Received: 28 April 2019 | Accepted: 27 November 2020

DOI: 10.1111/rssb.12410

ROYAL STATISTICAL SOCIETY
Series B Statistical Methodology

ORIGINAL ARTICLE

Estimation and clustering in popularity adjusted block model

Majid Noroozi[1] | **Ramchandra Rimal**[2] | **Marianna Pensky**[3]

[1]Department of Mathematics and Statistics, Washington University in St. Louis, St. Louis, MO 63130, USA

[2]Department of Mathematical Sciences, Middle Tennessee State University, Murfreesboro, TN 37132, USA

[3]Department of Mathematics, University of Central Florida, Orlando, FL 32816, USA

Correspondence
Marianna Pensky, Department of Mathematics, University of Central Florida, Orlando 32816, FL, USA.
Email: Marianna.Pensky@ucf.edu

Funding information
All three authors were partially supported by National Science Foundation (NSF) grants DMS-1712977 and DMS-2014928.

Abstract

The paper considers the Popularity Adjusted Block model (PABM) introduced by Sengupta and Chen (Journal of the Royal Statistical Society Series B, 2018, **80**, 365–386). We argue that the main appeal of the PABM is the flexibility of the spectral properties of the graph which makes the PABM an attractive choice for modelling networks that appear in biological sciences. We expand the theory of PABM to the case of an arbitrary number of communities which possibly grows with a number of nodes in the network and is not assumed to be known. We produce estimators of the probability matrix and of the community structure and, in addition, provide non-asymptotic upper bounds for the estimation and the clustering errors. We use the Sparse Subspace Clustering (SSC) approach for partitioning the network into communities, the approach that, to the best of our knowledge, has not been used for the clustering network data. The theory is supplemented by a simulation study. In addition, we show advantages of the PABM for modelling a butterfly similarity network and a human brain functional network.

KEYWORDS

popularity adjusted block model, sparse subspace clustering, spectral clustering, stochastic block model

Equally contributing authors.

1 | INTRODUCTION

Statistical network analysis has become a major field of research, with applications as diverse as sociology, biology, genetics, ecology and information technology to name a few. An overview of statistical modelling of random graphs can be found in, for example, Goldenberg et al. (2010), Kolaczyk (2009).

Consider an undirected network with n nodes and no self-loops and multiple edges. Let $A \in \{0,1\}^{n\times n}$ be the symmetric adjacency matrix of the network with $A_{i,j} = 1$ if there is a connection between nodes i and j, and $A_{i,j} = 0$ otherwise. We assume that

$$A_{i,j} \sim \text{Bernoulli}(P_{i,j}), \quad 1 \le i \le j \le n, \tag{1}$$

where $A_{i,j}$ are conditionally independent given $P_{i,j}$ and $A_{i,j} = A_{j,i}$, $P_{i,j} = P_{j,i}$ for $i>j$.

The block models assume that each node in the network belongs to one of K distinct blocks or communities $\mathcal{N}_k$, $k = 1,\cdots,K$. Let c denote the vector of community assignment, with $c_i = k$ if the node i belongs to the community k. Then, the probability of connection between node $i \in \mathcal{N}_k$ and node $j \in \mathcal{N}_l$ depends on the pair of blocks (k, l) to which nodes (i, j) belong. One can also consider a corresponding *membership* (or *clustering*) matrix $Z \in \{0,1\}^{n\times K}$ such that $Z_{i,k} = 1$ if $i \in \mathcal{N}_k$, $i = 1,\ldots,n$.

A classical random graph model for networks with community structure is the Stochastic Block Model (SBM) that has been studied by a number of authors (see, e.g. Abbe, 2018; Gao et al., 2017 among others). Under this model, all nodes belonging to a community are considered to be stochastically equivalent, in the sense that the probability of connection between nodes is completely defined by the communities to which they belong. Specifically, under the K-block SBM, this probability is completely determined by the community assignment for nodes (i, j), so that $P_{i,j} = B_{c_i,c_j}$ where $B_{k,l}$ is the probability of connection between communities k and l. In particular, any nodes from the same community have the same degree distribution and the same expected degree.

Since the real-life networks usually contain a very small number of high-degree nodes while the rest of the nodes have very few connections (low degree), the SBM model fails to explain the structure of many networks that occur in practice. The Degree-Corrected Block Model (DCBM) addresses this deficiency by allowing these probabilities to be multiplied by the node-dependent weights (see, e.g. Chen et al., 2018; Karrer & Newman, 2011; Zhao et al., 2012 among others). Under the DCBM, the elements of matrix P are modelled as $P_{i,j} = \theta_i B_{c_i,c_j} \theta_j$, where θ_i, $i = 1,\ldots,n$, are the degree parameters of the nodes and B is the $(K\times K)$ matrix of baseline interaction between communities. Identifiability of the parameters is usually ensured by a constraint of the form $\sum_{i\in\mathcal{N}_k} \theta_i = 1$ for all $k = 1,\ldots,K$ (see, e.g. Karrer & Newman, 2011).

A network feature that is closely associated with community structure is the popularity of nodes across communities defined as the number of edges between a specific node and a specific community. While the DCBM allows to correctly detect the communities and accurately fits the total degree by enforcing the node-specific degree parameters, it enforces the node popularity to be uniformly proportional to the node degree. Hence, the DCBM fails to model node popularities in a flexible and realistic way. For this reason, recently, Sengupta and Chen (2018) introduced the Popularity Adjusted Stochastic Block Model (PABM) which models the probability of a connection between nodes as a product of popularity parameters that depend on the communities to which the nodes belong as well as on the pair of nodes themselves. In particular, in PABM

$$P_{i,j} = V_{i,c_j} V_{j,c_i}, \tag{2}$$

where $V_{i,k}$, $1 \le i \le n$, $1 \le k \le K$, is the scaling parameter that identifies popularity of node i in class k, and $0 \le P_{i,j} \le 1$ for any i and j. Specifically, Sengupta and Chen (2018) define the popularity of node i in

community k as $\mu_{i,k} = \sum_{j \in \mathcal{N}_k} P_{i,j}$. They noted that the ratio of popularities of the nodes $(i,j) \in \mathcal{N}_k$ in the same community k is equal to one for the SBM, is independent of community k (a function of i and j only) in DCBM but can vary between nodes and communities for the PABM, thus, allowing a more flexible modelling of connection probabilities. The authors showed that PABM generalises both the SBM and the DCBM, suggested the quasi-maximum likelihood type procedure for estimation and clustering and demonstrated the improvement achieved through this new methodology.

The flexibility of PABM, however, is not limited to modelling the popularity parameters of the nodes. In order to better understand the model, consider a rearranged version $P(Z, K)$ of matrix P where its first n_1 rows correspond to nodes from class 1, the next n_2 rows correspond to nodes from class 2 and the last n_K rows correspond to nodes from class K. Denote the (k, l)-th block of matrix $P(Z, K)$ by $P^{(k,l)}(Z,K)$. Since sub-matrix $P^{(k,l)}(Z,K) \in [0,1]^{n_k \times n_l}$ corresponds to pairs of nodes in communities (k, l), respectively, one obtains from (2) that $P^{(k,l)}_{i,j} = V_{i_k,l} V_{j_l,k}$ where i_k is the i-th element in $\mathcal{N}_k$ and j_l is the j-th element in $\mathcal{N}_l$. Thus, matrices $P^{(k,l)}(Z,K)$ are rank one matrices with the unique singular vectors generating them. Indeed, consider vectors $\Lambda^{(k,l)}$ with elements $\Lambda^{(k,l)}_i = V_{i_k,l}$, where $i = 1, \ldots, n_k$ and $i_k \in \mathcal{N}_k$. Then, Equation (2) implies that

$$P^{(k,l)}(Z,K) = \Lambda^{(k,l)} [\Lambda^{(l,k)}]^T. \tag{3}$$

Moreover, it follows from (2) and (3) that $P^{(k,l)}(Z,K) = [P^{(l,k)}(Z,K)]^T$ and that each pair of blocks (k, l) involves a unique combination of vectors $\Lambda^{(l,k)}$:

$$P(Z,K) = \begin{bmatrix} \Lambda^{(1,1)}(\Lambda^{(1,1)})^T & \Lambda^{(1,2)}(\Lambda^{(2,1)})^T & \cdots & \Lambda^{(1,K)}(\Lambda^{(K,1)})^T \\ \Lambda^{(2,1)}(\Lambda^{(1,2)})^T & \Lambda^{(2,2)}(\Lambda^{(2,2)})^T & \cdots & \Lambda^{(2,K)}(\Lambda^{(K,2)})^T \\ \vdots & \vdots & \cdots & \vdots \\ \Lambda^{(K,1)}(\Lambda^{(1,K)})^T & \Lambda^{(K,2)}(\Lambda^{(2,K)})^T & \cdots & \Lambda^{(K,K)}(\Lambda^{(K,K)})^T \end{bmatrix}$$

where

$$\Lambda = \begin{bmatrix} \Lambda^{(1,1)} & \Lambda^{(1,2)} & \cdots & \Lambda^{(1,K)} \\ \Lambda^{(2,1)} & \Lambda^{(2,2)} & \cdots & \Lambda^{(2,K)} \\ \vdots & \vdots & \cdots & \vdots \\ \Lambda^{(K,1)} & \Lambda^{(K,2)} & \cdots & \Lambda^{(K,K)} \end{bmatrix} \tag{4}$$

The latter implies that matrix $P(Z, K)$ is formed by arbitrary rank one blocks and hence $\text{rank}(P(Z, K)) = \text{rank}(P)$ can take any value between K and K^2. In comparison, all other block models restrict the rank of P to be exactly K. This is true not only for the SBM and DCBM discussed above but also for their generalisations such as the Mixed Membership models (MMM) (see, e.g. Airoldi et al., 2008; Cheng et al., 2017) and the Degree Corrected Mixed Membership (DCMM) (see, e.g. Jin et al., 2017). While the MMM and the DCMM allows more diverse structures of rank K matrices (those matrices have to be just a product of two rank K matrices with nonnegative components while the PABM requires a combination of K^2 rank one matrices), meaningful fitting of the MMM or DCMM relies on a variety of conditions (one needs to have pure nodes in the network and some identifiability conditions need to be satisfied). In addition, while the MMM and DCMM are extremely useful for the analysis of social and society-related networks such as publications networks, they may not be appropriate in some other applications where each node can

belong to one and only one class. The butterfly similarity network studied in this paper provides an example of such application.

In general, the flexibility makes the PABM an attractive choice for modelling networks that appear in biological sciences, especially in the situations where memberships in multiple communities are not allowed. Indeed, while social networks exhibit assortative behaviour due to the human tendency of forming strong associations, the biological networks tend to be more diverse.

However, while the PABM model is extremely valuable, the statistical inference in Sengupta and Chen (2018) has been incomplete. In particular, the authors considered only the case of a small finite number of communities K; they provided only asymptotic consistency results as $n\to\infty$ without any error bounds for finite values of n; their NP-hard clustering procedure was tailored to the case of a small K. In addition, the relaxation of this NP-hard procedure seems to be operational only in the case of $K = 2$ since all simulations and real data examples in Sengupta and Chen (2018) only tackled the case of $K = 2$.

The purpose of the present paper is to address some of those deficiencies and to advance the theory of the PABM. Specifically, the main merit of our paper lies in the fact that we recognise that the probability matrix of the PABM is formed by a unique collection of rank one matrices. This useful property has not been detected by Sengupta and Chen (2018) who worked in terms of the Poisson likelihood and the Poisson likelihood modularity maximisations. This observation on the structure of the probability matrix leads to a variety of breakthroughs.

First, it enables us to carry out estimation and clustering for the PABM, without imposing any identifiability conditions, similarly to SBM and unlike the DCBM and MMM. Second, our understanding of the probability matrix structure leads to the Frobenius norm minimisation as the basis of optimisation procedure and to estimation of probability matrices by rank one approximations of the community matrices. The latter allows us to derive non-asymptotic upper bounds for the estimation error, even in the case when the number of communities is unknown and is possibly growing with n. In addition, we use the accuracy of approximation of the adjacency matrix for various number of communities, to identify the number of communities in the network. Moreover, we formulate detectability conditions that guarantee that communities are identifiable, that is, for the true probability matrix, the solution of the optimisation problem is given by the true community assignment. Under those conditions, we provide a non-asymptotic upper bound on the proportion of the misclassified nodes when the clustering is based on the solution of the optimisation problem above.

Furthermore, we note that, under the detectability condition, the columns of the probability matrix that correspond to any of the communities lie in a K-dimensional subspace, which is different from subspaces corresponding to all other communities. The latter conclusion results in the introduction of the Sparse Subspace Clustering (SSC) approach for partitioning the network into communities. While the SSC is widely used in computer vision, to the best of our knowledge, it has never been used for clustering network data. The advantage of the SSC procedure is that it is known to work very well in practice and has several well studied versions (see Section 3.1 for the discussion of the SSC algorithms). Moreover, unlike the Extreme Point algorithm which Sengupta and Chen (2018) managed to implement only in the case of $K = 2$, the SSC works well for an arbitrary number of communities. Our simulation study, as well as the real data examples, handle various number of communities between 2 and 6. In particular, we demonstrate the advantages of the PABM for modelling networks that appear in biological sciences.

We show that, under the detectability condition, the SSC delivers the correct community assignment at population level. We discuss the state-of-the-art results for the accuracy of the SSC approach and point out why they cannot be applied directly in the case of the independent Bernoulli errors. Investigation of the precision of the SSC for such errors is the matter of future work.

The rest of the paper is organised as follows. Section 2 considers estimation and clustering in PABM as a solution of a penalised optimisation procedure and investigates its accuracy. Specifically, Section 2.1 introduces notations used throughout the paper. Section 2.2 formulates estimation and clustering as solutions of an optimisation procedure. Section 2.3 derives an upper bound for the estimation error in the case when the number of communities in the PABM is unknown. Section 2.4 delivers an upper bound for estimation errors in the case when all probabilities of connections are uniformly small and also discusses advantages of the PABM for modelling sparsity when this assumption is not true. Section 2.5 provides detectability conditions at the population level. Section 2.6 offers sufficient conditions for the proportion of misclassified nodes to be bounded above by a pre-specified quantity ρ_n with a high probability. Since the optimisation problem in Section 2.2 is NP-hard, Section 3 presents a computationally tractable way of finding communities by the Subspace Clustering. In particular, Section 3.1 reviews the SSC methodologies and elaborates on what kind of SSC procedure we employ in this paper. Section 3.2 shows that the SSC delivers correct community assignment at the population level while Section 3.3 investigates this question in the case when the SSC is applied to the adjacency matrix. Section 4 deliberates about practical implementation of clustering and provides a simulation study and real data examples. Proofs for all the statements of the paper can be found in the Supplementary Material. Finally, the codes for the simulations are available from the third author's website at https://sciences.ucf.edu/math/mpensky/recent-publications/

2 | ESTIMATION AND CLUSTERING

2.1 | Notation

For any two positive sequences $\{a_n\}$ and $\{b_n\}$, $a_n \lesssim b_n$ and $a_n \asymp b_n$ mean that there exists a constant $C > 0$ independent of n such that, respectively, $a_n \le Cb_n$ and $C^{-1}a_n \le b_n \le Ca_n$ for any n. For any set Ω, denote cardinality of Ω by $|\Omega|$. For any numbers a and b, $a \wedge b = \min(a, b)$. For any vector $t \in \mathbb{R}^p$, denote its ℓ_2, ℓ_1, ℓ_0 and ℓ_∞ norms by, respectively, $\|t\|$, $\|t\|_1$, $\|t\|_0$ and $\|t\|_\infty$. Denote by 1_m the m-dimensional column vector with all components equal to one. For any matrix A, denote its spectral and Frobenius norms by, respectively, $\|A\|_{op}$ and $\|A\|_F$. Let vec(A) be the vector obtained from matrix A by sequentially stacking its columns.

Denote by $\mathcal{M}_{n,K}$ a collection of clustering matrices $Z \in \{0,1\}^{n\times K}$ such that $Z_{i,k} = 1$ if $i \in \mathcal{N}_k$, $i = 1,\ldots,n$ and $Z^TZ = \mathrm{diag}(n_1,\ldots,n_K)$ where $n_k = |\mathcal{N}_k|$ is the size of community k, where $k = 1,\ldots,K$. Denote $n_{\min} = \min_k n_k$. Denote by $\mathcal{P}_{Z,K} \in \{0,1\}^{n\times n}$ the permutation matrix corresponding to $Z \in \mathcal{M}_{n,K}$ that rearranges any matrix $B \in R^{n,n}$, so that its first n_1 rows correspond to nodes from class 1, the next n_2 rows correspond to nodes from class 2 and the last n_K rows correspond to nodes from class K. Recall that $\mathcal{P}_{Z,K}$ is an orthogonal matrix with $\mathcal{P}_{Z,K}^{-1} = \mathcal{P}_{Z,K}^T$. For any $\mathcal{P}_{Z,K}$ and any matrix $B \in \mathbb{R}^{n\times n}$ denote the permuted matrix and its blocks by, respectively, $B(Z,K)$ and $B^{(k,l)}(Z,K)$, where $B^{(k,l)}(Z,K) \in \mathbb{R}^{n_k\times n_l}$, $k, l = 1,\ldots,K$, and

$$B(Z,K) = \mathcal{P}_{Z,K}^T B \mathcal{P}_{Z,K}, \quad B = \mathcal{P}_{Z,K} B(Z,K) \mathcal{P}_{Z,K}^T. \tag{5}$$

Also, throughout the paper, we use the star symbol to identify the true quantities. In particular, we denote the true matrix of connection probabilities by P_*, the true number of classes by K_* and the true clustering matrix that partitions n nodes into K_* communities by Z_*.

2.2 | Optimisation procedure for estimation and clustering

In this section we consider estimation of the true probability matrix P_*. Consider block $P_*^{(k,l)}(Z_*, K_*)$ of the rearranged version $P_*(Z_*, K_*)$ of P_*. Let $\Lambda \equiv \Lambda(Z_*, K_*) \in [0,1]^{n \times K_*}$ be a block matrix with each column l partitioned into K_* blocks $\Lambda^{(k,l)} \equiv \Lambda^{(k,l)}(Z_*, K_*) \in [0,1]^{n_k}$. Then, due to (3), $P_*^{(k,l)}(Z_*, K_*)$ are rank one matrices such that $P_*^{(k,l)}(Z_*, K_*) = [P_*^{(l,k)}(Z_*, K_*)]^T$ and that each pair of blocks (k, l) involves a unique combination of vectors $\Lambda^{(k,l)}$. The structures of matrices $P_*(Z_*, K_*)$, Λ and P_* are illustrated in Figure 1.

Observe that although matrices $P_*^{(k,l)}(Z_*, K_*)$ in Equation (3) are well defined, vectors $\Lambda^{(k,l)}$ and $\Lambda^{(l,k)}$ can be determined only up to a multiplicative constant. In particular, under the constraint

$$1_{n_k}^T \Lambda^{(k,l)} = 1_{n_l}^T \Lambda^{(l,k)}, \tag{6}$$

Sengupta and Chen (2018) obtained explicit expressions for vectors $\Lambda^{(k,l)}$ and $\Lambda^{(l,k)}$ in Equation (3). In reality, K_* and matrices Z_* and P_* are unknown and need to be recovered. If K_* were known, in order to estimate Z_* and P_*, one could permute the rows and the columns of the adjacency matrix A using permutation matrix $\mathcal{P}_{Z,K_*}$ obtaining matrix $A(Z, K_*) = \mathcal{P}_{Z,K_*}^T A \mathcal{P}_{Z,K_*}$ and then, following assumption (3), minimise some divergence measure between blocks of $A(Z, K_*)$ and the products $\Lambda^{(k,l)}[\Lambda^{(l,k)}]^T$. One of such measures is the Bregman divergence between $A(Z, K_*)$ and $\Lambda^{(k,l)}[\Lambda^{(l,k)}]^T$.

The Bregman divergence between vectors x and y associated with a continuously differentiable, strictly convex function F is defined as

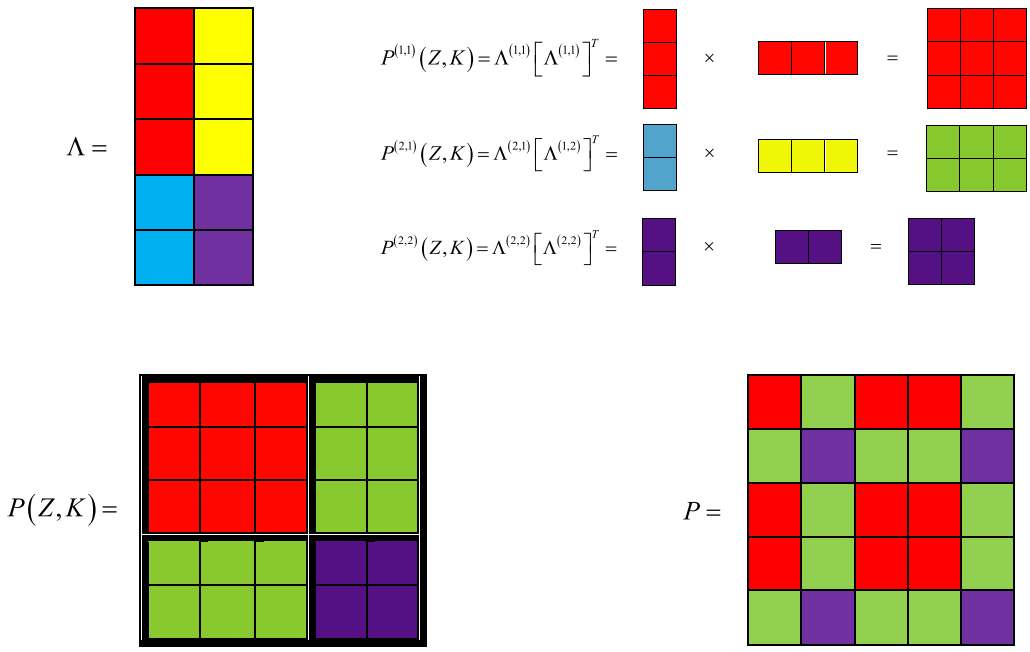

FIGURE 1 Matrices Λ, $P(Z, K)$ and P in the case of $n = 5$ and $K = 2$. Matrix Λ (top left): $\Lambda^{(1,1)}$ (red), $\Lambda^{(2,1)}$ (blue), $\Lambda^{(1,2)}$ (yellow), $\Lambda^{(2,2)}$ (violet). Assembling re-organised probability matrix $P(Z, K)$ (top right): $P^{(1,1)}(Z, K)$ (red), $P^{(2,1)}(Z, K)$ (green), $P^{(2,2)}(Z, K)$ (violet). Re-organised probability matrix $P(Z, K)$ (bottom left): $P^{(1,1)}(Z, K)$ (red), $P^{(2,1)}(Z, K)$ and $P^{(1,2)}(Z, K)$ (green), $P^{(2,2)}(Z, K)$ (violet). Probability matrix P (bottom right): nodes 1,3,4 are in community 1; nodes 2 and 5 are in community 2

$$D_F(x,y) = F(x) - F(y) - \langle \nabla F(y), x - y \rangle$$

where $\nabla F(y)$ is the gradient of F with respect to y. The Bregman divergence between any matrices X and Y of the same dimension can be defined as the Bregman divergence between their vectorised versions: $D_F(X,Y) = D_F(\mathrm{vec}(X), \mathrm{vec}(Y))$. It is well known that $D_F(X,Y) \geq 0$ for any X and Y and $D_F(X,Y) = 0$ if $X = Y$. In particular, the Poisson log-likelihood maximisation used in Sengupta and Chen (2018) corresponds to minimising the Bregman divergence with

$$F(x) = \sum_i (x_i \ln x_i - x_i).$$

Under the assumption (3) and the constraint (6) of Sengupta and Chen (2018), the latter leads to maximisation over $\Lambda^{(k,l)}$ and $Z \in \mathcal{M}_{n,K_*}$ of the following quantity

$$l(\Lambda | A) = -D_F(A, \Lambda) = \sum_{k,l=1}^{K_*} \sum_{i=1}^{n_k} \sum_{j=1}^{n_l} \left[A_{i,j}^{(k,l)} \ln \left(\Lambda_i^{(k,l)} \Lambda_j^{(l,k)} \right) - \left(\Lambda_i^{(k,l)} \Lambda_j^{(l,k)} \right) \right]. \tag{7}$$

where $A^{(k,l)}$ stands for $A^{(k,l)}(Z, K_*)$, the (k, l)-th block of matrix $A(Z, K_*)$. It is easy to see that the expression (7) coincides with the Poisson log-likelihood up to a term which depends on matrix A only, and is independent of P, Z and K_*. Maximisation of (7) over Λ, under condition (6), for given Z and K_*, leads to the estimators of Λ obtained in Sengupta and Chen (2018)

$$\widehat{\Lambda}^{(k,l)} = \frac{A^{(k,l)}(Z, K_*) 1_{n_l}}{\sqrt{1_{n_k}^T A^{(k,l)}(Z, K_*) 1_{n_l}}}; \quad \widehat{\Lambda}^{(l,k)} = \frac{(A^{(k,l)}(Z, K_*))^T 1_{n_k}}{\sqrt{1_{n_k}^T A^{(k,l)}(Z, K_*) 1_{n_l}}}. \tag{8}$$

Afterwards, Sengupta and Chen (2018) plug the estimators (8) into (7), thus, obtaining the likelihood modularity function which they further maximise in order to obtain community assignments.

In the present paper, we use the Bregman divergence associated with the Euclidean distance ($F(x) = \|x\|^2$) which, for a given K, leads to the following optimisation problem

$$(\widehat{\Lambda}, \widehat{Z}) \in \underset{\Lambda, Z}{\mathrm{argmin}} \left\{ \sum_{k,l=1}^{K} \| A^{(k,l)}(Z, K) - \Lambda^{(k,l)} [\Lambda^{(l,k)}]^T \|_F^2 \right\} \quad \text{s.t.} \quad A(Z,K) = \mathcal{P}_{Z,K}^T A \mathcal{P}_{Z,K}$$

Note that recovery of the components $\Lambda^{(k,l)}$ and $\Lambda^{(l,k)}$ of the products above relies on an identifiability condition of the type (6). Since these conditions can be imposed in a variety of ways, we denote $\Theta^{(k,l)} = \Lambda^{(k,l)} [\Lambda^{(l,k)}]^T$ and recover the uniquely defined rank one matrix $\Theta^{(k,l)}$. In addition, since the number of clusters K is unknown, we impose a penalty on K in order to safeguard against choosing too many clusters. Hence, we need to solve the following optimisation problem

$$\begin{aligned} (\widehat{\Theta}, \widehat{Z}, \widehat{K}) \in \underset{\Theta, Z, K}{\mathrm{argmin}} & \left\{ \sum_{k,l=1}^{K} \| A^{(k,l)}(Z,K) - \Theta^{(k,l)} \|_F^2 + \mathrm{Pen}(n,K) \right\} \\ \text{s.t.} \quad & A(Z,K) = \mathcal{P}_{Z,K}^T A \mathcal{P}_{Z,K}, \quad \mathrm{rank}(\Theta^{(k,l)}) = 1; \quad k,l = 1, 2, \cdots, K. \end{aligned} \tag{9}$$

Here, $\widehat{\Theta}$ is the block matrix with blocks $\widehat{\Theta}^{(k,l)}$, $k, l = 1, \ldots, \widehat{K}$ and Pen(n, K) will be defined later.

Observe that, if $\widehat{Z}$ and $\widehat{K}$ were known, the best solution of problem (9) would be given by the rank one approximations $\widehat{\Theta}^{(k,l)}$ of matrices $A^{(k,l)}(\widehat{Z}, \widehat{K})$

$$\widehat{\Theta}^{(k,l)}(\widehat{Z}, \widehat{K}) = \Pi_{\widehat{u},\widehat{v}}\left(A^{(k,l)}(\widehat{Z}, \widehat{K})\right) = \widehat{\sigma}_1^{(k,l)}\widehat{u}^{(k,l)}(\widehat{Z}, \widehat{K})(\widehat{v}^{(k,l)}(\widehat{Z}, \widehat{K}))^T, \tag{10}$$

where $\widehat{\sigma}_1^{(k,l)}$ are the largest singular values of matrices $A^{(k,l)}(\widehat{Z}, \widehat{K})$; $\widehat{u}^{(k,l)}(\widehat{Z}, \widehat{K})$, $\widehat{v}^{(k,l)}(\widehat{Z}, \widehat{K})$ are the corresponding singular vectors and $\Pi_{\widehat{u},\widehat{v}}\left(A^{(k,l)}(\widehat{Z}, \widehat{K})\right)$ is the rank one projection of matrix $A^{(k,l)}(\widehat{Z}, \widehat{K})$ (see Lemma 3 in Supplementary Material for the exact expression). Due to the Perron–Frobenius theorem (Rao & Rao, 1998, **P.15.1.14**), $\widehat{\sigma}_1^{(k,l)} > 0$ and elements of vectors $\widehat{u}^{(k,l)}(\widehat{Z}, \widehat{K})$ and $\widehat{v}^{(k,l)}(\widehat{Z}, \widehat{K})$ are non-negative. Plugging (10) into (9), we rewrite optimisation problem (9) as

$$\begin{aligned}(\widehat{Z}, \widehat{K}) \in \underset{Z,K}{\operatorname{argmin}} &\left\{\sum_{k,l=1}^{K} \|A^{(k,l)}(Z, K) - \Pi_{\widehat{u},\widehat{v}}\left(A^{(k,l)}(Z, K)\right)\|_F^2 + \text{Pen}(n, K)\right\} \\ \text{s.t.} \quad & A(Z, K) = \mathcal{P}_{Z,K}^T A \mathcal{P}_{Z,K}\end{aligned} \tag{11}$$

In order to obtain $(\widehat{Z}, \widehat{K})$, one needs to solve optimisation problem (11) for every K, obtaining

$$\widehat{Z}_K \in \underset{Z \in \mathcal{M}_{n,K}}{\operatorname{argmin}} \left\{\sum_{k,l=1}^{K} \|A^{(k,l)}(Z, K) - \Pi_{\widehat{u},\widehat{v}}\left(A^{(k,l)}(Z, K)\right)\|_F^2\right\} \tag{12}$$

and then, find $\widehat{K}$ as

$$\widehat{K} \in \underset{K}{\operatorname{argmin}} \left\{\sum_{k,l=1}^{K} \|A^{(k,l)}(\widehat{Z}_K, K) - \Pi_{\widehat{u},\widehat{v}}\left(A^{(k,l)}(\widehat{Z}_K, K)\right)\|_F^2 + \text{Pen}(n, K)\right\}. \tag{13}$$

Note that if the true number of clusters K_* were known, the penalty in Equations (9) and (11) would be unnecessary.

2.3 | The penalty and the estimation errors

In this section we evaluate the estimation and the clustering errors. We choose the penalty which, with high probability, exceeds the random errors. In particular, we denote

$$\text{Pen}(n, K) = H_1 nK + H_2 K^2 \ln n + H_3 n \ln K, \tag{14}$$

where H_1, H_2 and H_3 are positive absolute constants that can be evaluated. Then, the following statement holds.

Theorem 1 *Let $(\widehat{\Theta}, \widehat{Z}, \widehat{K})$ be a solution of optimisation problem (9). Construct the estimator $\widehat{P}$ of P_* of the form*

$$\widehat{P} = \mathcal{P}_{\widehat{Z},\widehat{K}} \widehat{\Theta}(\widehat{Z}, \widehat{K}) \mathcal{P}^T_{\widehat{Z},\widehat{K}} \tag{15}$$

where $\mathcal{P}_{\widehat{Z},\widehat{K}}$ is the permutation matrix corresponding to $(\widehat{Z}, \widehat{K})$. Then, for any $t>0$ and some absolute positive constants H and $\tilde{C}$, one has

$$\mathbb{P}\left\{ n^{-2} \|\widehat{P} - P_*\|_F^2 \leq n^{-2} H \mathrm{Pen}(n, K_*) + n^{-2} \tilde{C} t \right\} \geq 1 - 3e^{-t}, \tag{16}$$

$$n^{-2}\, \mathbb{E} \|\, \widehat{P} - P_* \,\|_F^2 \leq n^{-2}\, H\, \mathrm{Pen}(n, K_*) + n^{-2}\, \tilde{C}. \tag{17}$$

The exact values of H and $\tilde{C}$ can be found in the proof of Theorem 1. Observe that estimation is always consistent as long as $K/n \to 0$. Note also that the estimation errors in Equations (16) and (17) are proportional to the right-hand side of Equation (14). The first term in Equation (14) corresponds to the error of estimating nK unknown entries of matrix Λ, the second term is associated with estimation of rank K^2 matrix while the last term is due to the clustering of n nodes into K communities. If K grows with n, that is, $K=K(n)\to\infty$ as $n\to\infty$, then the first term in Equation (14) dominates the other two terms. However, in the case of a fixed K, the first and the third terms grow at the same rate as $n\to\infty$. The second term is always of a smaller order provided $K(n)/n\to 0$.

2.4 | The sparse PABM

The real-life networks are usually sparse in a sense that a large number of nodes have small degrees. One of the advantages of the PABM is that it allows flexible modelling of sparsity. Traditionally, in most statistical models, sparsity of a vector means that a large proportion of its components is equal to zero. One of the shortcomings of both the SBM and the DCBM is that they do not allow to impose the condition that some of the connection probabilities are equal to zero. Naturally, for the SBM, it is not realistic to assume that all nodes in a pair of communities have no connections. Neither can one set any of the node-specific weight to zero, since this will force the respective node to be totally disconnected from the network. For this reason, unlike in other numerous statistical settings, sparsity in block models is defined as a low maximum probability of connections between the nodes:

$$\max_{i,j} P_{i,j} \leq \tau_n, \tag{18}$$

where τ_n is small when n is large (see, e.g. Klopp et al., 2017; Lei & Rinaldo, 2015).

There are several shortcomings of this definition of sparsity. First, definition (18) characterises sparsity of the network as a whole, without describing community-specific sparsity patterns. Second, even in the context of the simplest model, the SBM, in order to take a full advantage of assumption (18), one needs to carry out the estimation under the restriction that all entries of the matrix $\widehat{P}$ are bounded above by τ_n (see Klopp et al., 2017), which is an unknown quantity.

Moreover, in the context of the PABM, one can take advantage of sparsity in a different, more natural, in our opinion, way. Indeed, unlike the SBM and the DCBM, the PABM setting allows some connection probabilities to be zero while keeping average connection probabilities between classes

above certain level and the network connected. This is certainly true since setting $\Lambda_i^{(k,l)} = 0$ in the PABM simply means that that node i in class k is not active ('popular') in class l. The latter does not prevent node i from having high probability of connection with nodes in another class. This approach enables a more nuanced representation of sparsity patterns where they are community-based and do not necessarily apply to the whole network.

Therefore, the PABM, similarly to other sparse statistical settings, allows structural sparsity where small parameters are set to zero rather than considered to be infinitesimally small. Setting some of the connection probabilities to zero, rather than bounding all of the connection probabilities by a very small number, as in Equation (18), not only leads to better understanding of network topology but also allows more precise estimation of the probability matrix P_*. Furthermore, this approach enables one to handle the unknown number of communities that is possibly growing with n. While we do not consider the structurally sparse PABM in this paper, we investigate the structurally sparse PABM in depth in our subsequent publication Noroozi et al. (2019).

Below, we briefly consider the case of the uniformly sparse PABM satisfying condition (18). In this case, the main error term nK in Equations (16) and (17) is replaced by $\tau_n nK$, which can significantly reduce the error if $K=K(n)\to\infty$. However, the drawback of this approach is that one needs to know either the sparsity level τ_n or the number of communities K. The reason for this is that the penalty term, which offsets the random error, should contain a component $C\tau_n nK$ where C is an absolute constant. The latter quantity may not be monotone since $K=K(n)$ is growing with n, while τ_n is decreasing with n.

For this reason, we derive the estimation error under a more common scenario that the number of communities is known: $K = K_*$. In this case, penalty is unnecessary and one can just solve optimisation problem (12) for the known number of communities.

Theorem 2 *Let $K = K_*$ be known, $\widehat{Z}$ be a solution of optimisation problem (12) with $K = K_*$. Let $\widehat{\Theta} = \widehat{\Theta}(\widehat{Z})$ be the matrix with blocks $\widehat{\Theta}^{(k,l)}$ given by Equation (10). Construct the estimator $\widehat{P}$ of P_* of the form Equation (15) where $\mathcal{P}_{\widehat{Z}}$ is the permutation matrix corresponding to $\widehat{Z}$. Assume that $n_{\min} = \min_k(n_k)$ is large enough, so that*

$$\log(2n_{\min}) \le (2\,n_{\min})^{2/13}, \quad \tau_n \ge C_\tau \log(2\,n_{\min})/n_{\min} \tag{19}$$

for some absolute constant $C_\tau > 0$ and τ_n in Equation (18). Then, for any $t>0$ and some absolute positive constants $\tilde{H}$, H_1, H_2 and H_3, one has

$$\mathbb{P}\left\{ n^{-2}\|\widehat{P}-P_*\|_F^2 \le H_1\tau_n n^{-1}K + H_2 n^{-2}K^2 + H_3 n^{-1}\ln K + n^{-2}\tilde{H}t \right\} \ge 1-3e^{-t}, \tag{20}$$

$$n^{-2}\,\mathbb{E}\,\|\widehat{P}-P_*\|_F^2 \le H_1\tau_n n^{-1}K + H_2 n^{-2}K^2 + H_3 n^{-1}\ln K + n^{-2}\tilde{H}. \tag{21}$$

The advantage of Theorem 2, in comparison with Theorem 1, is that it takes sparsity into account and replaces the main error term $O(nK)$ in Theorem 1 by the smaller quantity $O(\tau_n nK)$, and this is done without any knowledge of τ_n. If $\tau_n \to 0$ as $n\to\infty$, the latter may be significantly smaller than the former. This reduction, however, comes at a price. First, application of Theorem 2 requires the knowledge of the number of communities K. Second, while results in Theorem 1 are non-asymptotic and are valid for any combination of n and K, Theorem 2 requires not only n but also $n_{\min}$ to be large via conditions (19).

In addition, Theorem 2 puts natural constraints on the sparsity level τ_n. Note that under the assumption (18), one has $\|P_*\|_F^2 \le n^2\tau_n^2$. Therefore, in order $\mathbb{E}\|\widehat{P}-P_*\|_F^2/\|P_*\|_F^2 \to 0$ according to (21), one needs

$$H_1K(n\tau_n)^{-1} + H_2K^2(n\tau_n)^{-2} + H_3\ln K(n\tau_n^2)^{-1} + \tilde{H}(n\tau_n)^{-2} \to 0 \quad (n\to\infty).$$

The latter requires $n\tau_n^2/\ln K \to \infty$. Hence, to achieve a meaningful consistency, one needs to ensure that the expected network degree is $n\tau_n \gg \sqrt{n}$, which implies that Theorem 2 works for a moderately dense graph only.

2.5 | Detectability of clusters

In order to evaluate the clustering error, we assume that the true number of communities $K = K_*$ is known. Let $Z_* \in \mathcal{M}_{n,K_*}$ be the true clustering matrix. In order for clustering to be successful, one needs a detectability condition that guarantees that communities are identifiable.

Assumption A1 There exists k, $1 \le k \le K_*$, such that vectors $\Lambda^{(k,1)}, \ldots, \Lambda^{(k,K_*)}$ are linearly independent and have all positive components.

Assumption A1 is an alternative formulation of the Detectability Assumption 4.4 of Sengupta and Chen (2018) which states that, for any two nodes j_1 and j_2 that belong to different communities, the set $\{P_{i,j_1}/P_{i,j_2}\}_{i=1}^n$ assumes at least (K+1) distinct values. Similarly, to Sengupta and Chen (2018), Assumption A1 guarantees that, for the true $K = K_*$, expression (12) is minimised at $Z = Z_*$.

Note that the assumption, that all elements of matrix Λ are positive, is necessary. Indeed, consider a PABM with $K = 2$ and matrix Λ such that $\Lambda^{(1,1)} = \Lambda^{(1,2)} = u$ and $\Lambda^{(2,1)} = v$ and $\Lambda^{(2,2)} = w$, where v and w are linearly independent. If $u_i = 0$ and $w_j = 0$ for some i and j, then matrix P has two proportional columns, i and j, of the form $(cu, 0)^T$, and nodes i and j can be placed in any of the two communities. In order to avoid the condition that all elements of matrix Λ are positive, one can use an alternative assumption.

Assumption A1* For any $k = 1, \ldots, K_*$, vectors $\Lambda^{(k,1)}, \ldots, \Lambda^{(k,K_*)}$ are linearly independent.

Lemma 1 *Let Assumption A1 or Assumption A1* holds. Let $Z_* \in \mathcal{M}_{n,K_*}$ be the true clustering matrix and $Z \in \mathcal{M}_{n,K_*}$ be an arbitrary clustering matrix. Then,*

$$\sum_{k,l=1}^{K} \|P_*^{(k,l)}(Z_*) - \Pi_{(1)}(P_*^{(k,l)}(Z_*)\|_F^2 \le \sum_{k,l=1}^{K} \|P_*^{(k,l)}(Z) - \Pi_{(1)}(P_*^{(k,l)}(Z)\|_F^2 \tag{22}$$

where, for any matrix B, $\Pi_{(1)}(B)$ is its rank one approximation. Moreover, equality in Equation (22) occurs if and only if matrices Z and Z_* coincide up to a permutation of columns.

Lemma 1 implies that if $K = K_*$ is known, then optimisation problem (9) leads to the true clustering assignment at a population level. The next section explores the clustering errors in the case when optimisation procedure (9) is applied to the adjacency matrix.

2.6 | The clustering errors

Note that if Z_* is the true clustering matrix and Z is any other clustering matrix, then the proportion of misclassified nodes can be evaluated as

$$\mathrm{Err}(Z, Z_*) = (2n)^{-1} \min_{\mathcal{P}_K \in \mathcal{P}_K} \|Z\mathcal{P}_K - Z_*\|_1 = (2n)^{-1} \min_{\mathcal{P}_K \in \mathcal{P}_K} \|Z\mathcal{P}_K - Z_*\|_F^2 \tag{23}$$

where $\mathcal{P}_K$ is the set of permutation matrices $\mathcal{P}_K$: $\{1, 2, \cdots, K\} \longrightarrow \{1, 2, \cdots, K\}$. Let

$$\Upsilon(Z_*, \rho) = \left\{ Z \in \mathcal{M}_{n,K} \colon (2n)^{-1} \min_{\mathcal{P}_K \in \mathcal{P}_K} \|Z\mathcal{P}_K - Z_*\|_1 \geq \rho \right\} \tag{24}$$

be the set of clustering matrices with the proportion of misclassified nodes being at least ρ_n, $0 < \rho_n < 1$.

The success of clustering in Equation (12) relies upon the fact that matrix P_* is a collection of K^2 rank one blocks, so that the operator and the Frobenius norms of each block are the same. Moreover, if clustering were incorrect, the ranks of the blocks would increase which would lead to the discrepancy between their operator and Frobenius norms. In particular, the following statement is true.

Theorem 3 *Let $K = K_* \geq 2$ be the true number of clusters and $Z_* \in \mathcal{M}_{n,K_*}$ be the true clustering matrix. Let Assumption A1 or A1* hold. Let $\widehat{Z} \equiv \widehat{Z}_K$ be a solution of the optimisation problem (12). If for some $\alpha_n \in (0, 1/2)$ and $\rho_n \in (0, 1)$, one has*

$$\|P_*\|_F^2 - (1 + \alpha_n) \max_{Z \in \Upsilon(Z_*, \rho_n)} \sum_{k,l=1}^{K} \|P_*^{(k,l)}(Z)\|_{op}^2 \geq \frac{H}{\alpha_n} (nK + K^2 \ln n), \tag{25}$$

then, with probability at least $1 - 2e^{-c_0 n}$, the proportion of the nodes, misclassified by $\widehat{Z}$, is at most ρ_n. If, moreover, the entries of the matrix P_* are bounded above by τ_n as in Equation (18) and, in addition, condition (19) holds for the true clustering Z_*, then, with probability at least $1 - 2e^{-c_0 n}$, the proportion of the nodes, misclassified by $\widehat{Z}$, is at most ρ_n, provided

$$\|P_*\|_F^2 - (1 + \alpha_n) \max_{Z \in \Upsilon(Z_*, \rho_n)} \sum_{k,l=1}^{K} \|P_*^{(k,l)}(Z)\|_{op}^2 \geq \frac{\tilde{H}}{\alpha_n} (\tau_n nK + K^2 + n \ln K). \tag{26}$$

Here, H, $\tilde{H}$ and c_0 are absolute positive constant independent of K, n, ρ_n and α_n.

In order to see what condition (25) means, we consider a simple example of the SBM with $K = 2$ and $P_{i,j} = b$ when nodes i and j belong to the same community and $P_{i,j} = r$ when they belong to different ones. Then, condition (25) reduces to the following inequality.

Lemma 2 *Consider the SBM with $K = 2$, $P = ZBZ^T$ where $B_{1,1} = B_{2,2} = b$, $B_{1,2} = r$ and $Z_* \in \mathcal{M}_{n,2}$ with equal size communities. Then, for $0 < \rho \equiv \rho_n < \min(1, r^2/b^2)$ and $\alpha \equiv \alpha_n$, one has*

$$\|P_*\|_F^2 - (1 + \alpha) \max_{Z \in \Upsilon(Z_*, \rho)} \sum_{k,l=1}^{2} \|P_*^{(k,l)}(Z)\|_{op}^2 \geq \tilde{C} \left[n^2 \rho^2 (b^2 - r^2)^2 b^{-2} - 4\alpha n^2 b^2 \right], \tag{27}$$

where $\tilde{C}$ is an absolute constant.

Corollary 1 *For the SBM in Lemma 2, one has*

$$\rho_n^2 \lesssim n^{-1/2}\, b^{-1}\, (b^2 - r^2)^{-2}. \tag{28}$$

The example above shows that condition (25) is less sensitive than conditions that are based on the difference between mean vectors of probabilities of connections between the communities in the case of the SBM (or scaled mean vectors of the communities in the case of the DCBM). Indeed, it follows from Gao et al. (2017) that for the SBM in Lemma 2 one can attain the misclassification rate

$$\rho_n \asymp \exp\left(-\frac{n(b-r)^2}{2Kb}\right) \tag{29}$$

which is much smaller than ρ_n in Equation (28). Nevertheless, achieving the misclassification rate (29) depends upon not only the knowledge that the data are generated by the SBM, but also that this SBM is strongly assortative and balanced, and, in addition, requires handling the Bernoulli likelihood. In contrast, Theorem 3 is designed to work in the case where the communities are not characterised by their means and are not necessarily assortative or balanced. In addition, our procedure is based on minimising the Frobenius norm which is much more computationally efficient but is less sensitive than the Bernoulli likelihood maximisation.

Moreover, we believe that the assessment of Theorem 3 is valuable since it allows one to upper bound the misclassification rate rather than just stating that it tends to zero when the number of nodes in the network grows, as it is routinely done in the papers that draw clustering assignments on the basis of modularity maximisations (see, e.g. Bickel & Chen, 2009; Sengupta & Chen, 2018; Zhao et al., 2012).

3 | SPARSE SUBSPACE CLUSTERING

In Section 2, we obtained an estimator $\widehat{Z}$ of the true clustering matrix Z_* as a solution of optimisation problem (11). Minimisation in Equation (11) is somewhat similar to modularity maximisation in Bickel and Chen (2009), Sengupta and Chen (2018), Zhao et al. (2012), in the sense that modularity maximisation as well as minimisation in Equation (11) are NP-hard, and, hence, require some relaxation in order to obtain an implementable clustering solution.

In the case of the SBM and the DCBM, possible relaxations include semi-definite programming (see, e.g. Amini & Levina, 2018 and references therein), variational methods (Celisse et al., 2012) and spectral clustering and its versions (see, e.g. Joseph & Yu, 2016; Lei & Rinaldo, 2015; Rohe et al., 2011 among others). Since in the case of PABM, columns of matrix P_* that correspond to nodes in the same class are neither identical, nor proportional, application of spectral clustering (and its versions such as spherical spectral clustering) to matrix P_* directly does not deliver the partition of the nodes.

However, it is easy to see that the columns of matrix P_* that correspond to nodes in the same class form a matrix with K rank one blocks, hence, those columns lie in the subspace of the dimension at most K. Therefore, matrix P_* is constructed of K clusters of columns (rows) that lie in the union of K subspaces, each of the dimension K. Under Assumption A1*, those subspaces are independent in the sense that the dimension of their union (the rank of P_*) is equal to the sum K^2 of the dimensions K of individual subspaces, and they can be recovered. For this reason, the subspace clustering presents a technique for obtaining a fast and reliable solution of optimisation problem (11) or (12).

3.1 | Review of the subspace clustering

Subspace clustering has been widely used in computer vision and, for this reason, it is a very well studied and developed technique in comparison with the Extreme Points algorithm used in Sengupta and Chen (2018). Subspace clustering is designed for separation of points that lie in the union of subspaces. Let $\{X_j \in \mathbb{R}^D\}_{j=1}^n$ be a given set of points drawn from an unknown union of $K \geqslant 1$ linear or affine subspaces $\{S_i\}_{i=1}^K$ of unknown dimensions $d_i = \dim(S_i)$, $0 < d_i < D$, $i = 1,\ldots,K$. In the case of linear subspaces, the subspaces can be described as

$$S_i = \{\boldsymbol{x} \in \mathbb{R}^D : \boldsymbol{x} = \boldsymbol{U}_i \boldsymbol{y}\}, \quad i = 1, \ldots, K$$

where $\boldsymbol{U}_i \in \mathbb{R}^{D \times d_i}$ is a basis for subspace S_i and $\boldsymbol{y} \in \mathbb{R}^{d_i}$ is a low-dimensional representation for point $\boldsymbol{x}$. The goal of subspace clustering is to find the number of subspaces K, their dimensions $\{d_i\}_{i=1}^K$, the subspace bases $\{\boldsymbol{U}_i\}_{i=1}^K$, and the segmentation of the points according to the subspaces.

Several methods have been developed to implement subspace clustering such as algebraic methods (Boult & Gottesfeld Brown, 1991; Ma et al., 2008; Vidal et al., 2005), iterative methods (Agarwal & Mustafa, 2004; Bradley & Mangasarian, 2000; Tseng, 2000) and spectral clustering-based methods (Elhamifar & Vidal, 2009, 2013; Favaro et al., 2011; Liu et al., 2013, 2010; Soltanolkotabi et al., 2014; Vidal 2011). In this paper, we use the latter group of techniques.

Spectral clustering algorithms rely on construction of an affinity matrix whose entries are based on some distance measures between the points. In particular, in the case of the SBM, adjacency matrix itself serves as the affinity matrix, while for the DCBM, the affinity matrix is obtained by normalising rows/columns of A. In the case of the subspace clustering problem, one cannot use the typical distance-based affinity because two points could be very close to each other, but lie in different subspaces, while they could be far from each other, but lie in the same subspace. One of the solutions is to construct the affinity matrix using self-representation of the points with the expectation that a point is more likely to be presented as a linear combination of points in its own subspace rather than from a different one. A number of approaches such as Low-Rank Representation (see, e.g. Liu et al., 2010; Liu et al., 2013) and SSC (see, e.g. Elhamifar & Vidal, 2009; Elhamifar & Vidal, 2013) have been proposed in the past decade for the solution of this problem.

In this paper, we use SSC since it allows one to take advantage of the knowledge that, for a given K, columns of matrix P_* lie in the union of K distinct subspaces, each of the dimension at most K. If matrix P_* were known, the weight matrix W would be based on writing every data point as a sparse linear combination of all other points by minimising the number of nonzero coefficients

$$\min_{W_j} \|W_j\|_0 \quad \text{s.t } (P_*)_j = \sum_{k \neq j} W_{kj} (P_*)_k \tag{30}$$

where, for any matrix B, B_j is its j-th column. The affinity matrix of the SSC is the symmetrised version of the weight matrix W. If the subspaces are linearly independent, then the solution to the optimisation problem (30) is such that $W_{k,j} \neq 0$ only if points k and j are in the same subspace. In the case of data contaminated by noise, the SSC algorithm does not attempt to write data as an exact linear combination of other points. Instead, SSC is based on the solution of the following optimisation problem

$$\widehat{W}_j \in \underset{W_j}{\operatorname{argmin}} \left\{ \|W_j\|_0 + \gamma \|A_j - A W_j\|_2^2 \quad \text{s.t.} \quad W_{jj} = 0 \right\}, \quad j = 1, \ldots, n, \tag{31}$$

where $\gamma > 0$ is a tuning parameter. Problem (31) can be rewritten in an equivalent form as

$$\widehat{W}_j \in \underset{W_j}{\operatorname{argmin}} \left\{ \|A_j - AW_j\|_2^2 \quad \text{s.t.} \quad \|W_j\|_0 \le L, \quad W_{jj} = 0 \right\}, \quad j = 1, \ldots, n, \tag{32}$$

where L is the maximum number of nonzero elements in each column of W; in our case $L=K$. We solve (32) using the Orthogonal Matching Pursuit (OMP) algorithm (Mallat & Zhang, 1993; Weisberg, 2005) implemented in SPAMS Matlab toolbox (see Mairal et al., 2014). Given $\widehat{W}$, the affinity (similarity) matrix is defined as

$$S = |\widehat{W}| + |\widehat{W}^T| \tag{33}$$

where, for any matrix B, matrix $|B|$ has absolute values of elements of B as its entries.

The similarity matrix allows to construct the *similarity graph* $G=(V;E)$ where $(i, j) \in E$ if and only if $S_{i,j} > 0$ (see, e.g. Wang et al., 2016).

Remark 1 (Computational complexity of the SSC).Implementation of the SSC consists of two parts. The first part, evaluation of the matrix $\widehat{W}$. The second part is spectral clustering of the similarity matrix (33). While the first component of the problem is more computationally expensive (since it requires solution of n sparse regression problems), it is also the portion that can be easily carried out via parallel computing. Indeed, evaluation of each of the vectors W_j, $j = 1,\ldots,n$, is completely independent from evaluation of all the others. Hence, if one has m CPUs available, evaluation of matrix W can be accomplished m times faster. As a result, with the adequate facilities available, the computational limits of the technique is similar to the ones in the more traditional block models such as SBM and DCBM.

3.2 | Correctness of the SSC at population level

In order to apply the SSC for clustering in the PABM, we need to show that the SSC can detect communities correctly, at least at the population level. Let P_* be the true probability matrix. Since P_* contains no errors, one can obtain the coefficients matrix W as a solution of the optimisation problem (30). It turns out that Assumption A1* in Section 2.5 guarantees the correct community assignment (up to permutations of class labels).

Recovery of the clustering matrix Z_* relies on the fact that each column of the matrix P_* is represented as a linear combination of points in its own subspace rather than from a different one. This is

Algorithm 1: Consistent noiseless SSC (Wang et al., 2016)

Input: The noiseless data matrix P_*

Output: Vector of community assignments $\widehat{c}$, clustering matrix $\widehat{Z}$ and recovered subspaces $\widehat{\mathcal{S}}_k$, $k = 1,\ldots,K$

Steps:

1. **Constructing the similarity graph:** Solve the optimisation problem (30) and construct the similarity matrix S defined in Equation (33). Construct the similarity graph $G=(V;E)$ where $(i, j) \in E$ if and only if $S_{i,j} > 0$.
2. **Subspace recovery:** For each connected component $G_r = (V_r; E_r)$ of G, compute $\widehat{\mathcal{S}}_{(r)} = \text{Range}(P_{V_r})$ using any convenient linear algebraic method. Let $\{\widehat{\mathcal{S}}^{(k)}\}_{k=1}^{K}$ be the K unique subspaces in $\widehat{\mathcal{S}}_{(r)}$.
3. **Final clustering:** For each connected component V_r with $\widehat{\mathcal{S}}_{(r)} = \widehat{\mathcal{S}}^{(k)}$, set $\widehat{c}_i = k$, $\widehat{Z}_{i,l} = \mathbb{I}(l = k)$ for $i \in V_r$ and $l = 1,\ldots,K$.

formalised as the *Self-Expressiveness Property* (**SEP**) of the similarity graph S: $S_{i,j} > 0$ implies that nodes i and j belong to the same cluster (see, e.g. Elhamifar & Vidal, 2013). Note that the reverse is not necessarily true: the fact that nodes i and j are in the same cluster does not necessarily imply that $S_{i,j} > 0$. Under Assumption A1*, Theorem 1 of Elhamifar and Vidal (2013) ensures that the similarity matrix S obtained as a solution of optimisation problem (30) satisfies the SEP.

Nevertheless, the SEP alone does not lead to the perfect clustering because the obtained similarity graph G could be poorly connected (see, e.g. Nasihatkon & Hartley, 2011). It appears however that a simple post-clustering procedure (Algorithm 1) suggested in Wang et al. (2016), guarantees the correct recovery.

Theorem 4 *Assume that the correct number of communities $K = K_*$ is known. Then, under Assumption A1*, Algorithm 1 recovers communities correctly up to a permutation, that is, there exists a permutation π on {1,…,K} such that $\widehat{Z}_{i,\pi(k)} = (Z_*)_{i,k}$ for every* i = 1,…,n *and* k = 1,…,K.

3.3 | Accuracy of the data-based SSC

While there are many papers that evaluate clustering errors in the case of the k-means algorithm and spectral clustering, as well as their relaxations, there are very few results available for the accuracy of the SSC, and those results are quite recent. As it is evident from Section 3.2, the successful clustering relies on the fact that the SEP condition is satisfied with the high probability and that the similarity graph is sufficiently connected.

The main effort of the scientific community was devoted to establishing the SEP condition. Initially, this effort was directed to its justification when the true matrix is measured without errors Elhamifar and Vidal (2009) or with outliers Soltanolkotabi and Candes (2012). The latter paper assumes that the columns of the data matrix are generated at random using the bases of the respective subspace (semi-random model). Furthermore, the bases may themselves be generated uniformly, at random, from the unit sphere (random model) and the outliers are also uniformly distributed on the unit sphere. In the subsequent paper, Soltanolkotabi et al. (2014) handle the case where the data matrix contains small Gaussian errors. Few later papers refine the results of the last two publications. Specifically, Wang and Xu (2016) extend results of Soltanolkotabi et al. (2014) to the case of the deterministic model with random noise, that is, the model where true matrix is fixed in advance and is not generated at random. The noise vectors are assumed to be i.i.d. spherically symmetric with the lengths bounded above by a small quantity. Results of the latter paper are also featured in a very recent monograph of Shi and Iyengar (2019) which refines results of Wang and Xu (2016) in the case of missing observations or Gaussian noise.

The graph connectivity in the SSC has been much less studied. Indeed, it is mentioned in Nasihatkon and Hartley (2011) that the similarity graph may satisfy the SEP condition but be poorly connected. Fortunately, this issue can be addressed by post-processing procedures suggested in Wang et al. (2016). One of the procedures is presented in Algorithm 1.

Since our optimisation problem guarantees that coefficient vector has K non-zero components, and, hence, leads to a sufficiently well connected similarity graph, it is the SEP condition that presents the hardest challenge. Indeed, as it follows from the review above, there are two types of derivations of the SEP conditions in the existing literature. Specifically, in the case where $X \in \mathbb{R}^{n\times N}$ is a true matrix with columns drawn from K different subspaces $S_i, i = 1, \ldots, K$, and $Y{=}X{+}\Xi$ is its noisy version, the papers differ on whether they treat elements of matrix Ξ as deterministic or random. In both cases, the procedures start with scaling columns of matrix Y to the unit length.

The case of the random errors handles either the normally distributed errors (Shi & Iyengar, 2019; Soltanolkotabi et al., 2014), or, more broadly, i.i.d errors having a spherically symmetric distribution (Wang & Xu, 2016). Moreover, this assumption constitutes the cornerstone of the proofs since the arguments there are based on the fact that the errors are invariant under an orthogonal transformation. In addition, for both random and deterministic errors (see, e.g. Wang et al., 2016), it is assumed that for any columns X_i and Ξ_i of matrices X and Ξ, respectively, one has

$$\|\Xi_i\| \leq \delta_n \|X_i\|, \quad i = 1, \ldots, n, \tag{34}$$

where $\delta_n \to 0$ as $n \to \infty$ with high probability. The latter implies that $\|X_i\| = 1 + o(1)$ as $n \to \infty$.

Note that neither of the above assumptions are true in the case of Bernoulli errors. It is easy to see that Bernoulli errors are not i.i.d and that the columns of matrix Ξ are not spherically symmetric. Moreover, assumption (34) is not true in the case of the Bernoulli data. To understand this, consider a vector $p \in [0, 1]^n$, a vector a of independent Bernoulli variables $a_i \sim \text{Bernoulli}(p_i)$, $i = 1,\ldots,n$, and $\xi = a - p$. Then, for $p_i \leq 1/2$ one has $p_i(1 - p_i) \geq p_i^2$ and

$$\mathbb{E}\|\xi\|^2 = \sum_{i=1}^{n} (p_i - p_i^2) \geq \|p\|^2,$$

so that inequality (34) does not hold.

To make matters worse, scaling of the columns of the matrix A in Equation (1) to unit length is itself problematic. Indeed, since components of a Bernoulli vector are either zeros or ones, one has $\|a\|^2 = \sum_{i=1}^{n} a_i = a^T 1$, so $\|a\|^2$ is a very poor approximation of $\|p\|^2$. As a matter of fact, scaling column A_i of matrix A to unit length amounts to dividing this column by $\sqrt{d(i)}$ where $d(i)$ is the degree of the node i.

In conclusion, the existing error analysis for the SSC cannot be used in the case of Bernoulli data and one needs to establish a brand new theory. Development of such theory is a matter of future investigation.

4 | SIMULATIONS AND REAL DATA EXAMPLES

4.1 | Simulations on synthetic networks

In this section we evaluate the performance of our method using synthetic networks. We assume that the number of communities (clusters) K is known and for simplicity consider a perfectly balanced model with n/K nodes in each cluster. We generate each network from a random graph model with a symmetric probability matrix P given by the PABM model with a clustering matrix Z and a block matrix Λ.

Sengupta and Chen (2018), in their simulations, considered networks with $K = 2$ communities of equal sizes and matrices Λ in Equation (2) with elements $\Lambda_{i,r} = \alpha_i \sqrt{\frac{h}{1+h}}$ when node i lies in class r, and $\Lambda_{i,r} = \beta_i \sqrt{\frac{1}{1+h}}$ otherwise, where h is the homophily factor. The factors α_i and β_i were set to 0.8 for half of the nodes in each class and to 0.2 for another half at random, and h ranges between 1.5 and 4.0. Note that, although the data generated by the procedure above follows PABM, the probability matrix has constant blocks, for which the spectral clustering is known to deliver accurate results. In particular, the setting above leads to the SBM with four blocks. However, the spectral clustering incurs some difficulties as the probabilities of connections in every community become more diverse. In this paper, we make sure to generate networks that follow PABM with diverse probabilities of connections.

To generate a more diverse synthetic network, we start by producing a block matrix Λ in Equation (4) with random entries on the interval $(0,a)$, $0<a<1$. We multiply the non-diagonal blocks of Λ by ω, $0<\omega<1$, to ensure that most nodes in the same community have larger probability of interactions. Then matrix $P(Z, K)$ with blocks $P_{Z,K}^{(k,l)} = \Lambda^{(k,l)} (\Lambda^{(l,k)})^T$, $k, l = 1,\ldots,K$, mostly has larger entries in the diagonal blocks than in the non-diagonal blocks. The parameter ω is the heterogeneity parameter. Indeed, if $\omega = 0$, the matrix P_* is strictly block-diagonal, while in the case of $\omega = 1$, there is no difference between diagonal and non-diagonal blocks. Next, we generate a random clustering matrix $Z \in \mathcal{M}_{n,K}$ corresponding to the case of equal community sizes and the permutation matrix $\kappa P(Z, K)$ corresponding to the clustering matrix Z. Subsequently, we scramble rows and columns of $P(Z, K)$ to create the probability matrix $P = \mathcal{P}_{Z,K} P(Z, K) \mathcal{P}_{Z,K}^T$. Finally, we generate the lower half of the adjacency matrix A as independent Bernoulli variables $A_{ij} \sim \mathrm{Ber}(P_{ij})$, $i = 1,\ldots,n$, $j = 1,\ldots,i-1$, and set $A_{ij} = A_{ji}$ when $j>i$. In practice, the diagonal $\mathrm{diag}(A)$ of matrix A is unavailable, so we estimate $\mathrm{diag}(P)$ without its knowledge.

Sengupta and Chen (2018) used the Extreme Points (EP) algorithm, introduced in Le et al. (2016), as a clustering procedure. For $K = 2$, the EP algorithm computes the two leading eigenvectors of the adjacency matrix A, and finds the candidate assignments associated with the extreme points of the projection of the cube $[-1, 1]^n$ onto the space spanned by the two leading eigenvectors of A. The technique is becoming problematic when K grows and the probabilities of connections are getting more diverse, hence, Sengupta and Chen (2018) have only studied performances of estimation and clustering in the case of $K = 2$ and the choices of probability matrix P described above. As we have mentioned before, these are the settings for which the spectral clustering procedure allows to identify the communities. Considering that we are interested in studying $K>2$ and the more diverse probabilities of connections, we use the spectral clustering directly (SC thereafter) and compare its precision with the SSC procedure.

Since the diagonal elements of matrix A are unavailable, we initially set $A_{ii} = 0$, $i = 1,\ldots,n$. We solve optimisation problem (32) using the OMP algorithm. After matrix $\widehat{W}$ of weights is evaluated, we obtain the clustering matrix $\widehat{Z}$ by applying spectral clustering to $|\widehat{W}| + |\widehat{W}^T|$, as it was described in Section 3.1. Given $\widehat{Z}$, we generate matrix $A(\widehat{Z}) = \mathcal{P}_{\widehat{Z}}^T A \mathcal{P}_{\widehat{Z}}$ with blocks $A^{(k,l)}(\widehat{Z})$, $k, l = 1,\ldots,K$, and obtain $\widehat{\Theta}^{(k,l)}(\widehat{Z}, \widehat{K})$ using the rank one approximation for each of the blocks. Finally, we estimate matrix P by $\widehat{P} = \widehat{P}(\widehat{Z}, \widehat{K})$ using formula (15) with $\widehat{K} = K$.

We compared the accuracy of SSC and SC methods in terms of the average estimation errors $n^{-2} \|\widehat{P} - P\|_F^2$ and the average clustering errors $\mathrm{Err}(\widehat{Z}, Z)$ defined in Equation (23). Figures 2 and 3 show the results of these comparisons for the number of nodes ranging from $n = 600$ to $n = 1080$ with the increments of 120. The left panels display the clustering errors $\mathrm{Err}(\widehat{Z}, Z)$ while the right ones exhibit the estimation errors $n^{-2} \|\widehat{P} - P\|_F^2$, as functions of the number of nodes. All errors are averaged over 50 simulation runs. Figure 2 explores the effects of heterogeneity on the precision of estimation and clustering by carrying out simulations for $K = 4$ and $K = 8$, and for three different values of the parameter ω: $\omega = 0.5$, 0.7 and 0.9. Block matrix Λ in Equation (4) has random entries in the interval (0,1) in this case. Figure 2 confirms that the SSC is becoming more and more accurate in comparison with SC as ω grows. The latter is due to the fact that the SSC is more suitable for handling heterogeneous connections probabilities.

Figure 3 examines the impact of sparsity on clustering precision. For this round of simulations we used $K = 3$ and $K = 5$, fixed $\omega = 0.9$ and generated entries of the block matrix Λ in Equation (4) on the intervals (0,1), (0,0.9) and (0,0.8). The results are sorted by the average connection probability ρ which, in the above simulation setting, takes values 0.21833, 0.17685 and 0.13973 for $K = 3$, and 0.21200, 0.17172 and 0.13568 for $K = 5$. Figure 3 shows that the clustering errors decrease as ρ and n increase, with the effect of growth of n on the accuracy of clustering being much more significant in the case of the SSC.

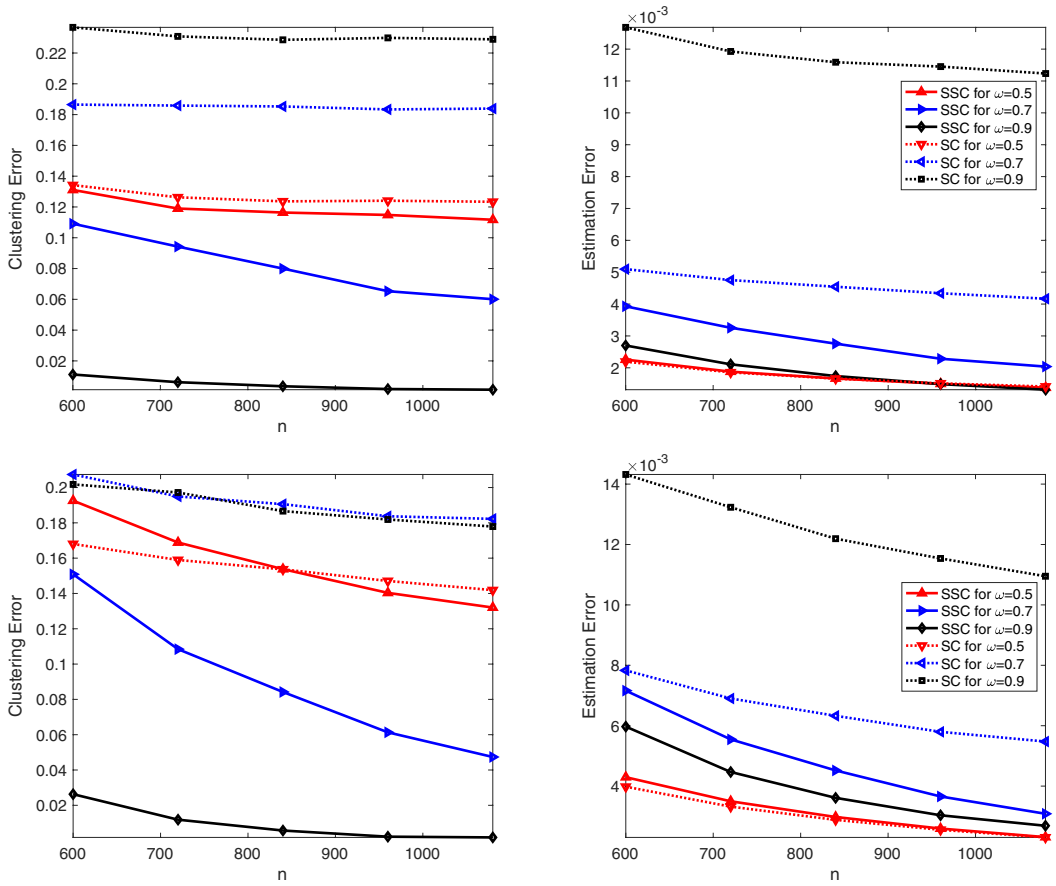

FIGURE 2 The clustering errors $\mathrm{Err}(\widehat{Z}, Z)$ defined in Equation (23) (left panels) and the estimation errors $n^{-2}\,\|\widehat{P} - P\|_F^2$ (right panels) for $K = 4$ (top) and $K = 8$ (bottom) clusters. The errors are evaluated over 50 simulation runs. The number of nodes ranges from $n = 600$ to $n = 1080$ with the increments of 120. SSC results are represented by the solid lines; SC results are represented by the dotted lines: $\omega = 0.5$ (red), $\omega = 0.7$ (blue) and $\omega = 0.9$ (black)

Figure 4 presents the results of comparison of the clustering errors of SSC and SC in the simulations settings of Sengupta and Chen (2018). It is easy to see that, while for larger values of the homophily factor h both methods perform almost equally well, the accuracy of SC deteriorates as h is getting smaller, due to the fact that the differences between probabilities of connections within and between clusters become less significant. The latter shows that the SSC approach is beneficial for clustering in PABM model. Indeed, it delivers more accurate results than the SC when probabilities of connections are more diverse. Moreover, SSC is still applicable when the PABM reduces to the SBM, although SC is more accurate in the case of the SBM since it does not require an additional step of evaluating the affinity matrix.

Remark 2 (Spectral Clustering Versus SSC.)It is worth noting that when the matrix of probabilities P_* is close to being block diagonal, the spectral clustering can be still used for recovering community assignments, even if P_* does not follow the SBM. The latter is due to the fact that, in this situation, the graph can be well approximated by a union of distinct connected components, and, therefore, SC allows to identify the true clusters. Moreover, in such situation, SC has an advantage of not requiring an additional step of self-representation, which is computationally costly and produces

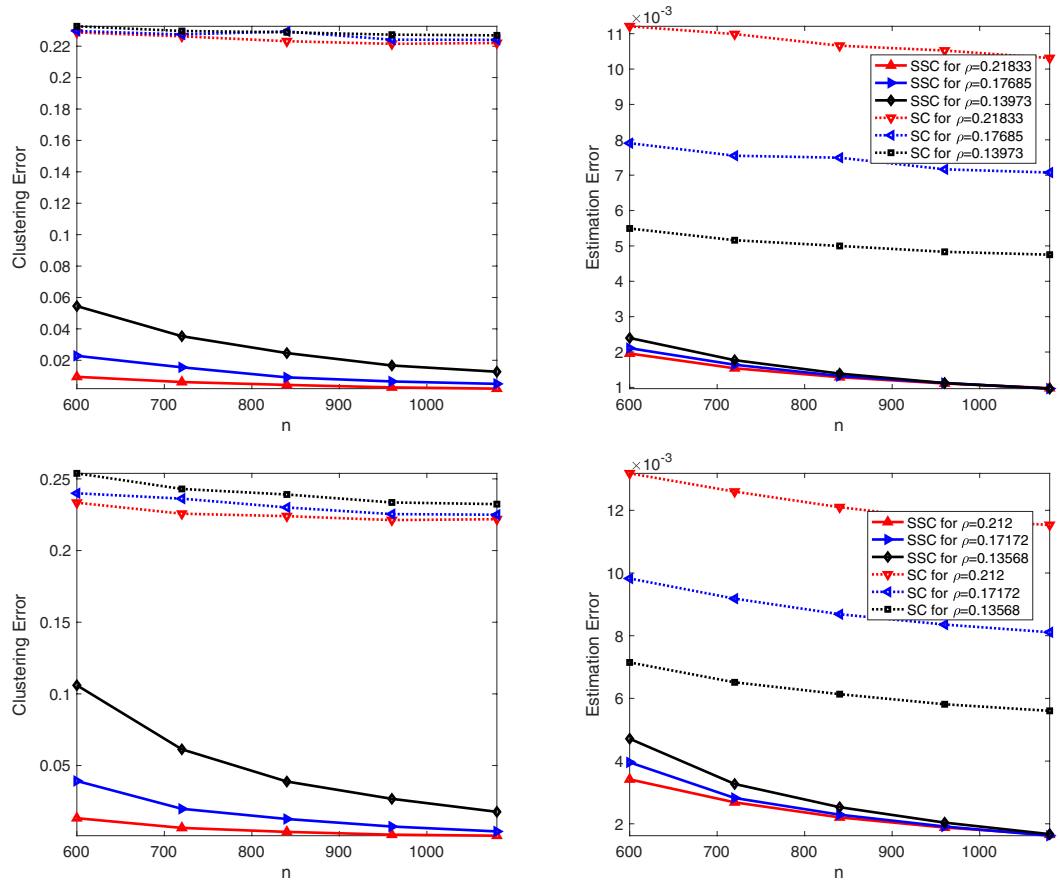

FIGURE 3 The clustering errors $\mathrm{Err}(\widehat{Z}, Z)$ defined in Equation (23) (left panels) and the estimation errors $n^{-2} \| \widehat{P} - P \|_F^2$ (right panels) for $K = 3$ (top) and $K = 5$ (bottom) clusters. The errors are evaluated over 50 simulation runs. The number of nodes ranges from $n = 600$ to $n = 1080$ with the increments of 120. SSC results are represented by the solid lines; SC results are represented by the dotted lines: for three different values of ρ and fixed $\omega = 0.9$

additional errors. Moreover, as we shall see from examples below, when probabilities of connections become more heterogeneous, SSC turns to be more precise than SC. In addition, since PABM has more unknown parameters than SBM, its correct fitting requires sufficient number of nodes per class (see, e.g. Soltanolkotabi et al., 2014); otherwise, its accuracy declines.

Remark 3 (Unknown number of clusters.)In our previous simulations we treated the true number of clusters as a known quantity. However, we can actually use $\widehat{P}$ to obtain an estimator $\widehat{K}$ of K by solving, for every suitable K, the optimisation problem (13), which can be equivalently rewritten as

$$\widehat{K} = \underset{K}{\operatorname{argmin}} \{ \| \widehat{P} - A \|_F^2 + \mathrm{Pen}(n, K) \}. \tag{35}$$

The penalty Pen(n, K) defined in Equation (14) is, however, motivated by the objective of setting it above the noise level with a very high probability. In our simulations, we also study the selection of an unknown K using somewhat smaller penalty

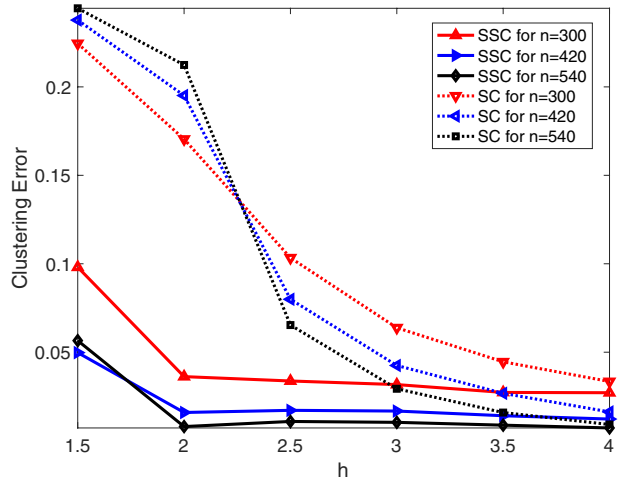

FIGURE 4 Clustering errors of SC and SSC for $K = 2$ clusters and $n = 300,420$ and 540 nodes in the simulations setting of Sengupta and Chen (2018). The homophily factor h ranges from 1.5 to 4 with increments of 0.5

TABLE 1 The relative of the estimators $\widehat{K}$ of K_* for K_* ranging from 3 to 6, $n = 420$ and $n = 840$ and $\omega = 0.5$, 0.7 and 0.9

		n = 420			**n = 840**		
K_*	$\widehat{K}$	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.9$	$\omega = 0.5$	ω=0.7	$\omega = 0.9$
3	2	0	0	0	0	0	0
	3	**0.76**	**0.80**	**0.90**	**0.52**	**0.60**	**0.80**
	4	0.24	0.16	0.10	0.36	0.26	0.16
	5	0	0.04	0	0.12	0.14	0.02
	6	0	0	0	0	0	0.02
4	2	0	0	0	0	0	0
	3	0.06	0.14	0	0.02	0.02	0
	4	**0.64**	**0.66**	**0.96**	**0.56**	**0.64**	**0.76**
	5	0.28	0.16	0.04	0.30	0.26	0.22
	6	0.02	0.04	0	0.12	0.08	0.02
5	2	0	0.02	0	0	0	0
	3	0.02	0	0.02	0	0	0
	4	0.14	0.16	0.04	0.04	0.04	0
	5	**0.64**	**0.66**	**0.82**	**0.78**	**0.68**	**0.90**
	6	0.20	0.16	0.12	0.18	0.28	0.10
6	2	0	0.04	0	0	0	0
	3	0.06	0.18	0.02	0	0	0
	4	0.18	0.22	0.02	0	0	0
	5	0.28	0.22	0.08	0.12	0.16	0.10
	6	**0.48**	**0.34**	**0.88**	**0.88**	**0.84**	**0.90**

The probabilities for the true values of K are given in bold.

$$\text{Pen}(n, K) = \rho(A) nK\sqrt{\ln n\,(\ln K)^3} \tag{36}$$

where $\rho(A)$ is the density of matrix A, the proportion of nonzero entries of A.
In order to assess the accuracy of $\widehat{K}$ as an estimator of K, we evaluated $\widehat{K}$ as a solution of optimisation problem (35) with the penalty (36) in each of the previous simulations settings over 50 simulation runs. Table 1 presents the relative frequencies of the estimators $\widehat{K}$ of K_* for K_* ranging from 3 to 6, $n = 420$ and $n = 840$ and $\omega = 0.5$, 0.7 and 0.9. Table 1 confirms that for majority of settings, $\widehat{K} = K_*$, the true number of clusters, with high probability. Moreover, the estimator $\widehat{K}$ of K is more reliable for higher values of ω and larger number of nodes per cluster.

4.2 | Real data examples

In this section, we report the performances of SSC and SC in studying real-life networks. The social networks usually exhibit strong assortative behaviour, the phenomenon which is possibly due to the tendency of humans to form strong associations. Perhaps, for this reason, the political blogs network, the British Twitter network and the Digital Bibliography & Library Project network which have been analysed by Sengupta and Chen (2018) have nearly block-diagonal adjacency matrices, so SC exhibits good performance in clustering of those networks (see Remark 2).

However, PABM provides a more accurate description of more diverse networks, in particular, the networks that appear in biological sciences. Below, we consider a butterfly similarity network extracted from the Leeds Butterfly dataset described in Wang et al. (2018). Leeds Butterfly dataset contains fine-grained images of 832 butterfly species that belong to 10 different classes, with each class containing between 55 and 100 images. In this network, the nodes represent butterfly species and edges represent visual similarities between them. Visual similarities are evaluated on the basis of butterfly images and range from 0 to 1. We study a network by extracting the four largest classes as

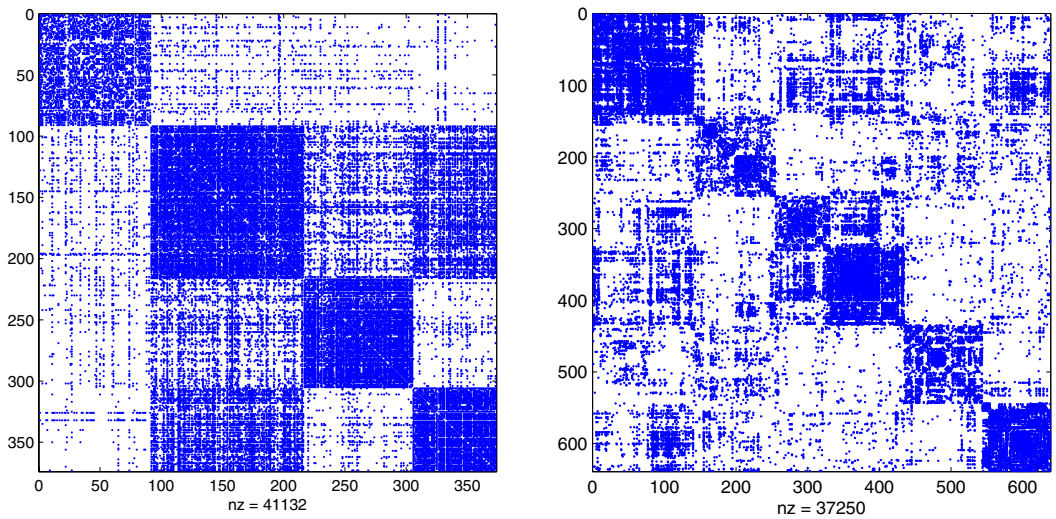

FIGURE 5 Adjacency matrices of the butterfly similarity network with 41,132 nonzero entries and 4 clusters (left) and the brain network with 37,250 nonzero entries and 6 clusters (right)

a simple graph with 373 nodes and 20,566 edges. We draw an edge between the nodes if the visual similarity between those nodes is greater than zero.

Classification of species on the basis of their visual similarities is a very important task. In many applications, the goal is to classify species automatically on the basis of their images, captured by a remote camera. This type of monitoring is essential for surveying bio-diversity and tracking abundance and habitats of species that may be affected by climate change and human activities. While the related species may look similar, classification of species does not allow mixed memberships: each of the actual species belong to one and only one class.

Figure 5 (left) shows the adjacency matrix of the graph (after clustering), which suggests that the PABM is a reasonable model to fit to the network. The latter is due to the fact that, since the phenotype of the species in the same class can vary, the SBM may not provide an adequate summary for the class similarities. Replacing the SBM by the DCBM does not solve the problem either, since it is unlikely that few butterflies are 'more similar' to the others than the rest. Moreover, the PABM allows some of the butterflies in one class to be 'more similar' to species of another specific class than the others, thus, justifying application of the PABM.

We carried out clustering of the nodes using the SSC, the SC and the weighted k-median algorithm, one of the popular clustering methods for the DCBM used in Lei and Rinaldo (2015) and Gao et al. (2018). We compared the clustering assignments of those methods with the true class specifications of the species using the adjusted Rand index that measures the agreement between two clustering assignments. The value of the adjusted Rand index between the true class specifications and the clustering assignments obtained by the SSC is 0.73; the weighted k-median algorithm is 0.67; and the SC is 0.61. In addition, we applied formula (35) with K ranging from 2 to 6 and obtained the true number of clusters.

As the second real network, we analyse a human brain functional network, measured using the resting-state functional MRI (fMRI). In particular, we use the co-activation matrix of the brain connectivity dataset, described in Crossley et al. (2013). In this dataset, the brain is partitioned into 638 distinct regions and a weighted graph is used to characterise the network topology. In our analysis, we set all nonzero weights to one, obtaining the network with 18625 undirected edges. Since, for this network, the true clustering as well as the true number of clusters are unknown, we first applied formula (35) with K ranging from 2 to 10 to find the number of clusters obtaining $\widehat{K} = 6$. This agrees with the assessment in Crossley et al. (2013) where the authors partitioned the network into six groups (if one considers the 'rich-club' communities as separate clusters). Subsequently, we applied the SSC for partitioning the network into blocks and derived the estimator $\widehat{P}$ of P_*. Figure 5 (right) shows the adjacency matrix of the graph after clustering. The true probability matrix P_* is unknown, we can only report that $n^{-2} \| \widehat{P} - A \|_F^2 = 0.05$, which indicates high agreement between the two matrices. We also carried out clustering of the nodes using the weighted k-median algorithm and the SC, that correspond, respectively, to modelling P via the DCBM and the SBM, and calculated the adjusted Rand index between the clustering assignments obtained by the three clustering methods. The adjusted Rand index between the clustering assignments obtained by the SSC and the weighted k-median algorithm is 0.47; the SSC and the SC is 0.64; and the weighted k-median algorithm and the SC is 0.51.

ORCID

Marianna Pensky http://orcid.org/0000-0002-0011-2075

REFERENCES

Abbe, E. (2018) Community detection and stochastic block models: Recent developments. *The Journal of Machine Learning Research*, 18, 1–86.

Agarwal, P.K. & Mustafa, N.H. (2004) K-means projective clustering. In: *Proceedings of the twenty-third ACM SIGMODSIGACT-SIGART symposium on Principles of database systems*, ACM, pp. 155–165..

Airoldi, E.M., Blei, D.M., Fienberg, S.E. & Xing, E.P. (2008) Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9, 1981–2014.

Amini, A.A. & Levina, E. (2018) On semidefinite relaxations for the block model. *The Annals of Statistics*, 46, 149–179.

Bickel, P.J. & Chen, A. (2009) A nonparametric view of network models and Newman–Girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106, 21068–21073.

Boult, T. & Gottesfeld Brown, L. (1991) Factorization-based segmentation of motions. 179–186.

Bradley, P.S. & Mangasarian, O.L. (2000) k-plane clustering. *Journal of Global Optimization*, 16, 23–32.

Celisse, A., Daudin, J.-J. & Pierre, L. (2012) Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics*, 6, 1847–1899.

Chen, Y., Li, X. & Xu, J. (2018) Convexified modularity maximization for degree-corrected stochastic block models. *The Annals of Statistics*, 46, 1573–1602.

Cheng, J., Li, T., Levina, E. & Zhu, J. (2017) High-dimensional mixed graphical models. *Journal of Computational and Graphical Statistics*, 26, 367–378.

Crossley, N.A., Mechelli, A., Vértes, P.E., Winton-Brown, T.T., Patel, A.X., Ginestet, C.E. et al. (2013) Cognitive relevance of the community structure of the human brain functional coactivation network. Vol. 110, 11583–11588. National Acad Sciences.

Elhamifar, E. & Vidal, R. (2009) Sparse subspace clustering. In 2009 *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2790–2797.

Elhamifar, E. & Vidal, R. (2013) Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 2765–2781.

Favaro, P., Vidal, R. & Ravichandran, A. (2011) A closed form solution to robust subspace estimation and clustering. In: CVPR'11, Washington, DC, USA: IEEE Computer Society, 1801–1807.

Gao, C., Ma, Z., Zhang, A.Y. & Zhou, H.H. (2017) Achieving optimal misclassification proportion in stochastic block models. *The Journal of Machine Learning Research*, 18, 1980–2024.

Gao, C., Ma, Z., Zhang, A.Y., Zhou, H.H. (2018) Community detection in degree-corrected block models. *The Annals of Statistics*, 46, 2153–2185.

Goldenberg, A., Zheng, A.X., Fienberg, S.E. & Airoldi, E.M. (2010) A survey of statistical network models. *Foundations and Trends\circledR in Machine Learning*, 2, 129–233.

Jin, J., Ke, Z.T. & Luo, S. (2017) Estimating network memberships by simplex vertex hunting. *arXiv e-prints*, arXiv:1708.07852.

Joseph, A. & Yu, B. (2016) Impact of regularization on spectral clustering. *The Annals of Statistics*, 44, 1765–1791.

Karrer, B. & Newman, M.E.J. (2011) Stochastic blockmodels and community structure in networks. *Physical Review, E, Statistical, Nonlinear, and Soft Matter Physics*, 83, 016107.

Klopp, O., Tsybakov, A.B. & Verzelen, N. (2017) Oracle inequalities for network models and sparse graphon estimation. *The Annals of Statistics*, 45, 316–354.

Kolaczyk, E.D. (2009) *Statistical analysis of network data: Methods and models*, 1st edn. New York: Springer Publishing Company.

Le, C.M., Levina, E. & Vershynin, R. (2016) Optimization via low-rank approximation for community detection in networks. *The Annals of Statistics*, 44, 373–400.

Lei, J. & Rinaldo, A. (2015) Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43, 215–237.

Liu, G., Lin, Z. & Yu, Y. (2010) Robust subspace segmentation by low-rank representation. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, 663–670. USA: Omnipress. https://doi.org/10.1007/s11063-018-9783-y

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y. & Ma, Y. (2013) Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 171–184.

Ma, Y., Yang, A.Y., Derksen, H. & Fossum, R. (2008) Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50, 413–458.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., Jenatton, R. & Obozinski, G. (2014) Spams: A sparse modeling software, v2.3. Available from: http://spams-devel. gforge.inria.fr/downloads.html.

Mallat, S. & Zhang, Z. (1993) Matching pursuits with time-frequency dictionaries. *Transactions on Signal Processing*, 41, 3397–3415.

Nasihatkon, B. & Hartley, R. (2011) Graph connectivity in sparse subspace clustering. In: *CVPR '11*. Washington, D.C: IEEE Computer Society, pp. 2137–2144. https://doi.org/10.1109/CVPR.2011.5995679

Noroozi, M., Rimal, R. & Pensky, M. (2019) Sparse popularity adjusted stochastic block model.

Rao, C. & Rao, M. (1998) *Matrix algebra and its applications to statistics and econometrics*, vol. 528. Singapore: World Scientific. https://doi.org/10.1142/3599

Rohe, K., Chatterjee, S., Yu, B. (2011) Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39, 1878–1915.

Sengupta, S. & Chen, Y. (2018) A block model for node popularity in networks with community structure. *Journal of the Royal Statistical Society Series B*, 80, 365–386.

Shi, B. & Iyengar, S. (2019) *Mathematical theories of machine learning—theory and applications*. Berlin: Springer. ISBN 978-3-030-17076-9.

Soltanolkotabi, M. & Candes, E.J. (2012) A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40, 2195–2238.

Soltanolkotabi, M., Elhamifar, E. & Candes, E.J. (2014) Robust subspace clustering. *The Annals of Statistics*, 42, 669–699.

Tseng, P. (2000) Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105, 249–252.

Vidal, R. (2011) Subspace clustering. *IEEE Signal Processing Magazine*, 28, 52–68.

Vidal, R., Ma, Y. & Sastry, S. (2005) Generalized principal component analysis (gpca). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 1945–1959.

Wang, Y.-X. & Xu, H. (2016) Noisy sparse subspace clustering. *Journal of Machine Learning Research*, 17, 1–41. Available from: http://jmlr.org/papers/v17/13-354.html.

Wang, Y., Wang, Y.-X. & Singh, A. (2016) Graph connectivity in noisy sparse subspace clustering. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (eds. A. Gretton and C. C. Robert),* vol. 51 of *Proceedings of Machine Learning Research*, pp. 538–546. Cadiz, Spain: PMLR. Available from: http://proceedings.mlr.press/v51/wang16b.html.

Wang, B., Pourshafeie, A., Zitnik, M., Zhu, J., Bustamante, C. D., Batzoglou, S., Leskovec, J. (2018) Network enhancement as a general method to denoise weighted biological networks. *Nature Communications*, 9, 3108.

Weisberg, S. (2005) *Applied linear regression*, vol. 528. Hoboken: John Wiley & Sons. ISBN:9780471663799.

Zhao, Y., Levina, E., Zhu, J. (2012) Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40, 2266–2292.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

How to cite this article: Noroozi M, Rimal R, . Estimation and clustering in popularity adjusted block model. *J R Stat Soc Series B*. 2021;83:293–317. https://doi.org/10.1111/rssb.12410