Estimation and Clustering in Popularity Adjusted Stochastic Block Model

Marianna Pensky
University of Central Florida, USA

Joint work with
Majid Noroozi and Ramchandra Rimal

Meeting in Mathematical Statistics
Lumini, December 2020

Stochastic Networks

- **A network** is a collection of nodes or vertices and edges that describe the interactions between them
- **A degree of a node** represents the number of its edges it has to other nodes
- A network is called **directed** if all of its edges are directed; otherwise, it is called **undirected**
- The description of a network is summarized by its **adjacency matrix** A with $A_{i,j} = 1$ if there is an edge between nodes i and j, and $A_{i,j} = 0$ otherwise, $i, j = 1, \ldots, n$
- Assumption: **nodes can be partitioned into communities** with the specific patterns of behavior

Community structure in networks

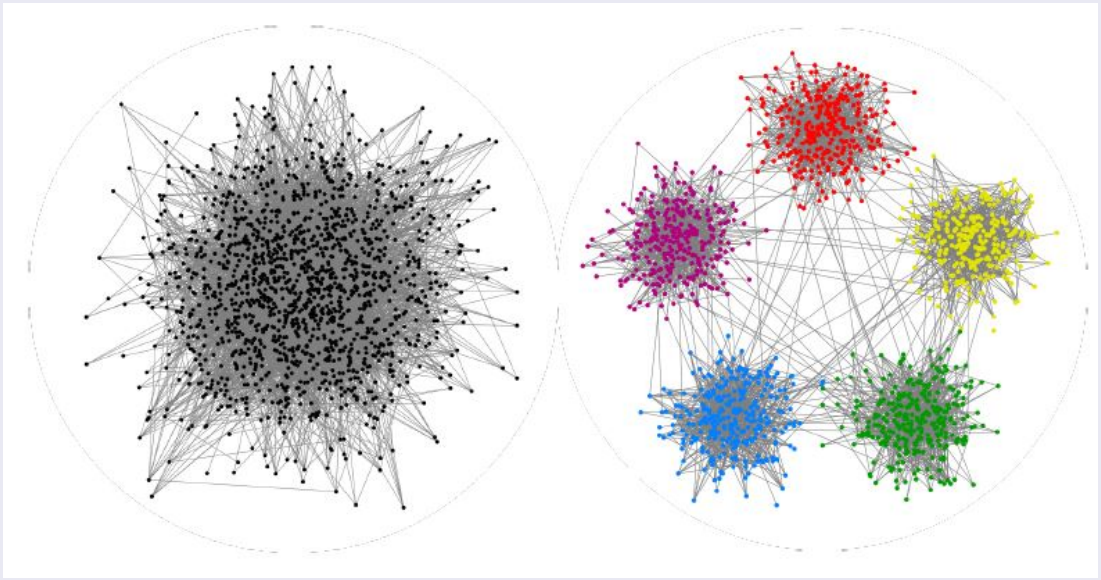

Figure 1 : Abbe (2018) *Journ. Mach. Learn. Research*

Data structure

- Consider an **undirected network** consisting of n **nodes** with K **communities**
- Let $P_{i,j}$ **be a probability of connection** between nodes i and j, $i, j = 1, \ldots, n$
- Since network is undirected, $P_{i,j} = P_{j,i}$ and $A_{i,j} = A_{j,i}$ and

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

- The total of $n(n-1)/2$ **unknown parameters**
- The value of $P_{i,j}$ **depends on the communities** to which nodes i and j belong

Community structure in networks

- **Communities** often correspond to important nodes' features.
- They allow a **low-dimensional representation** of a network.
- Community structures are often described by the **block models**.

- The block models assume that **each node i belongs to one of K distinct blocks** or communities $\mathcal{N}_k, k = 1, \cdots, K$
- Community assignment is described by a function $c : \{1, \ldots, n\} \to \{1, \ldots, K\}$ where $c(i) = k$ if $i \in \mathcal{N}_k$
- Alternatively, one considers a corresponding **membership** (or **clustering**) matrix $Z \in \{0, 1\}^{n \times K}$ such that

$$Z_{i,k} = 1 \quad \text{iff} \quad i \in \mathcal{N}_k, \;\; i = 1, \ldots, n, \; k = 1, \cdots, K$$

Community structure in networks

Objectives:

- Identify **communities in a network:** estimate the clustering matrix $Z \in \{0,1\}^{n \times K}$
- Uncover the **low-dimensional structure** in a network
- Estimate the **matrix of connection probabilities** $P \in [0,1]^{n \times n}$, $P = P^T$

- **Stochastic Block Model (SBM)**
- **Degree Corrected Block Model (DCBM)**
- **Popularity Adjusted Block Model (PABM)**

Stochastic Block Model (SBM)

- **The SBM** is a classical random graph model for networks with community structure (Lorrain and White (1971))
- Under the **SBM**, the connection probability is completely determined by the community assignment

$$P_{i,j} = B_{c(i),c(j)}$$

where B is the $(K \times K)$ matrix of baseline interaction between communities, so that
$P = ZBZ^T, \quad Z \in \{0,1\}^{n \times K}, \; B \in [0,1]^{K \times K}, \; B = B^T$

- **All nodes from the same community have the same degree distribution and the same expected degree**

Degree Corrected Block Model (DCBM)

- **The real-life networks** usually contain a **very small number of high-degree nodes** (many connections) while the rest of the nodes are low degree (have very few connections)
- This is **impossible under the SBM** where the degree distribution is determined by the community
- In **DCBM** (introduced by Karrer and Newman, 2011), the probabilities of connections are multiplied by the node-dependent weights

$$P_{i,j} = \theta_i B_{c(i),c(j)} \theta_j$$

where θ_i, $i = 1, \ldots, n$, are the **degree parameters of the nodes**, $B \in [0,1]^{K \times K}$ is the **matrix of baseline interactions** between communities

- **Identifiability of the parameters** is usually ensured by a constraint $\sum_{i \in \mathcal{N}_k} \theta_i = 1$ for all $k = 1, ..., K$

Popularity of nodes across communities

Popularity of node i in community k is the number of edges between i and $\mathcal{N}_k$

$$M_{i,k} = \sum_{j \in \mathcal{N}_k} A_{i,j}, \quad \mu_{i,k} = \mathbb{E}M_{i,k} = \sum_{j \in \mathcal{N}_k} P_{i,j}$$

- Under the **SBM**, $\mu_{i,k} = n_k B_{c(i),k}$ where $n_k = |\mathcal{N}_k|$, the size of community $\mathcal{N}_k$. If nodes i and j are in the same community, then

$$\mu_{i,k} = \mu_{j,k}$$

- Under the **DCBM**, for nodes i and j in same community

$$\frac{\mu_{i,k}}{\theta_i} = \frac{\mu_{j,k}}{\theta_j}$$

  Popularity is proportional to the degree of the node

- **Both scenarios are often unrealistic**

Popularity Adjusted Block Model (PABM)

- In order to **model node popularities** in a flexible and realistic way, Sengupta and Chen (2018) introduced the **Popularity Adjusted Stochastic Block Model (PABM)**
- In PABM, the probabilities of a connections between nodes are products

$$P_{i,j} = V_{i,c(j)} V_{j,c(i)},$$

where $V_{i,k}$, $1 \leq i \leq n$, $1 \leq k \leq K$, are the **popularity scaling parameters**

- **The SBM and DCBM are the special cases of the PABM**

Understanding the PABM

- 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.
- The (k,l)-th block $P^{(k,l)}(Z,K)$ of matrix $P(Z,K)$ has elements

$$P_{i,j}^{(k,l)} = V_{i_k,l} V_{j_l,k}$$

where i_k is the i-th element in community k and j_l is the j-th element in community l.

- Consider vectors $\Lambda_{(k,l)}$ with elements $(\Lambda_{(k,l)})_i = V_{i_k,l}$
  Then, $P^{(k,l)}(Z,K)$ are **rank-one matrices**

$$P^{(k,l)}(Z,K) = \Lambda_{(k,l)} \left[\Lambda_{(l,k)}\right]^T$$

Understanding the PABM

For a $K-$block network, let $\Lambda_{n\times K}$ be the matrix of popularity scaling parameters. Then,

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

where

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

Results of Sengupta and Chen (2018)

- Sengupta and Chen (2018) **did not understand the rank-one block structure** of the probability matrix
- They imposed **the identifiability constraint** $1_{n_k}^T \Lambda^{(k,l)} = 1_{n_l}^T \Lambda^{(l,k)}$ and found $\widehat{\Lambda}$ and $\hat{Z}$ by **maximizing the proxy Poisson likelihood** over $\Lambda^{(k,l)}$ $k, l = 1, \ldots, K$, and Z
- They used the **likelihood modularity maximization** to obtain community assignments
- Since the likelihood modularity maximization is exponentially hard, they applied the **Extreme Point (EP) algorithm** of Le, Levina and Vershynin (2016)
- Sengupta and Chen (2018) **assumed the number of communities K to be a known, small and fixed.** Since the EP algorithm is essentially designed for $K = 2$, **all simulations and real data examples were designed for $K = 2$**
- Sparsity was modeled as the **"uniform" sparsity:** $\max_{i,j} P_{i,j} \leq \tau_n$

Optimization procedure for estimation and clustering

- Instead of maximizing the proxy Poisson likelihood, we **minimize the Frobenius norm of the block differences**, which, for a given K, leads to

$$(\widehat{\Lambda}, \hat{Z}) \in \underset{\Lambda, Z}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \| A^{(k,l)}(Z,K) - \Lambda^{(k,l)} [\Lambda^{(l,k)}]^T \|_F^2 \right\}$$

- Since K is unknown and identifiability conditions can be imposed in a variety of ways, solve **the rank one optimization problem**

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

Optimization procedure for estimation and clustering

If $\hat{Z}$ and $\hat{K}$ were known, **the best solution would be given by the rank one approximations** $\hat{\Theta}^{(k,l)} = \Pi_{(1)}\left(A^{(k,l)}(\hat{Z},\hat{K})\right)$ of matrices $A^{(k,l)}(\hat{Z},\hat{K})$

Then, $(\hat{Z},\hat{K})$ are found by solving

$$(\hat{Z},\hat{K}) \in \underset{Z,K}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \|A^{(k,l)}(Z,K) - \Pi_{(1)}\left(A^{(k,l)}(Z,K)\right)\|_F^2 + \text{Pen}(n,K) \right\}$$

Solve the problem above for every K
After that, find the value $\hat{K}$ that delivers the global minimum

The penalty

Choose the penalty in order to offset the error of estimation and clustering

$$\text{Pen}(n, K) = C_1 nK + C_2 K^2 \ln n + C_3 n \ln K$$

C_1, C_2, C_3 are explicit absolute constants

- $C_1 nK$ is the **estimation error component:** the price of recovering nK unknown parameters (dominant)
- $C_2 K^2 \ln n$ is the **estimation error component:** the price of recovering K^2- block matrix
- $C_3 n \ln K$ is the **clustering error component:** the price of choosing one of K^n possible clustering assignments

Estimation and clustering errors: definitions

- **The average estimation error** $R_n(\hat{P}, P)$ is

$$R_n(\hat{P}, P) = n^{-2} \|\hat{P} - P\|_F^2$$

- **The average clustering error** $\mathrm{ERR}(\hat{Z}, Z)$ is evaluated as

$$\mathrm{ERR}_n(\hat{Z}, Z) = \min \left\{ n^{-1} \sum_{i=1}^{n} I(\mathscr{P}(\hat{Z}) \neq Z) \right\}$$

where the minimum is taken with respect to all permutations $\mathscr{P}$ of K clusters

Estimation errors

Let $(\hat{P}, \hat{Z}, \hat{K})$ be a solution of optimization problem above
Let P_* be the true probability matrix
Let K_* be the true number of communities
$\text{Pen}(n, K_*) = C_1 n K_* + C_2 K_*^2 \ln n + C_3 n \ln K_*$

Then, for any $t > 0$ and $n > 3$

$$\mathbb{P}\left\{ n^{-2} \left\| \hat{P} - P_* \right\|_F^2 \leq C_0\, n^{-2}\, \text{Pen}(n, K_*) + \tilde{C}_0\, n^{-2}\, t \right\} \geq 1 - 3e^{-t}$$

$$n^{-2}\, \mathbb{E} \left\| \hat{P} - P_* \right\|_F^2 \leq C_0\, n^{-2}\, \text{Pen}(n, K_*) + 3\tilde{C}_0\, n^{-2}$$

Here, C_0 and $\tilde{C}_0$ are explicit absolute constants.

Clustering errors

- Assume that **the true number of classes $K = K^*$ is known**
- The success of clustering relies upon the fact that **matrix P_* is a collection of K^2 rank-one blocks**
- Hence, **the operator and the Frobenius norms of each block are the same**
- If clustering were incorrect, **the ranks of the blocks would increase**
- This would lead to the **discrepancy between the operator and Frobenius norms of the blocks**

Clustering errors

- Let $K = K^*$ be the true number of clusters and $Z_* \in \mathcal{M}_{n,K_*}$ be the true clustering matrix
- Let $\hat{Z} \equiv \hat{Z}_K$ be a solution of the optimization problem
- Denote the set of clustering matrices with the **proportion of misclassified nodes being at least** ρ by $\Upsilon(Z_*, \rho)$, $\rho < 1$

If for some $\alpha_n \in (0, 1/2)$ and $\rho_n \in (0, 1)$, one has

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

then, with probability at least $1 - 2e^{-n}$, the proportion of the nodes, misclassified by $\hat{Z}$, **is at most** ρ_n.

Sparsity in Block Models

- **The real life networks are usually sparse** in a sense that a large number of nodes have small degrees
- **In the SBM**, one does not assume that the average block probabilities $B_{k,l} = 0$ for some k and l, since this **makes communities k and l disconnected**
- **In the DCBM**, setting any node specific weight to zero will force the respective **node to be totally disconnected from the network**
- As a result, **sparsity in block models is defined as a low maximum probability of connections** between the nodes: $\max_{i,j} P_{i,j} \leq \tau_n$ where $\tau_n \to 0$ as $n \to \infty$
- Hence, **sparsity describes only the behavior of network as a whole**, without distinguishing between the block-dependent sparsity patterns

Sparsity in Block Models

To the best of our knowledge, **the PABM** is the only block model that allows to model **structural sparsity** where

- **some connection probabilities are equal to zero**
- **the average connection probabilities between classes are above certain level**
- **the network is connected**

Sparsity in Block Models

- In the PABM, setting $\Lambda_i^{(k,l)} = 0$ simply **means that that node i in class k is not active ("popular") in class l.** This does not prevent this node from having high probability of connection with nodes in another class.
- Setting some elements of vectors $\Lambda^{(k,l)}$ to zero will merely lead to **some of the rows (columns) of sub-matrices $P^{(k,l)}(Z,K)$ being zero.**
- Since $A_{i,j}$ are Bernoulli variables with the means $P_{i,j}$, **those zeros are fairly easy to identify**, as $P_{i,j} = 0$ leads to $A_{i,j} = 0$.
- Setting connection probabilities to zero rather than to a very small positive number **leads to better understanding of network topology and more precise estimation of the probability matrix P**

Practical Implementation of Clustering

- Similarly to the likelihood modularity maximization of Sengupta and Chen (2018), **our optimization procedure is NP-hard**
- We apply **subspace clustering (well studied in computer vision)**

- **Subspace clustering** is designed for separation of points that lie in the union of subspaces
- **The matrix P is constructed by K clusters of columns (rows) of rank K**
- The Subspace Clustering allows one to take advantage of the knowledge that columns of matrix P lie in **the union of K distinct subspaces, each of the dimension K**

The Structure of the PABM

For a $K-$block network, let $\Lambda_{n\times K}$ be the matrix of popularity scaling parameters. Then,

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

where

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

Subspace Clustering versus Spectral Clustering

- Typically, **clustering algorithms rely on construction of an affinity matrix** whose entries are based on some distance measures between the points.
- In the subspace clustering, 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.**
- Expectation: a point is **more likely to be presented as a linear combination of points in its own subspace** rather than points from a different one.

Sparse Subspace Clustering (SSC)

- If the probability matrix P were known, **the coefficient matrix W of the SSC** would be based on writing every column P_j of P as a sparse linear combination of all other columns

$$\min_{W_j} \|W_j\|_0 \quad \text{s.t.} \quad P_j = \sum_{k \neq j} W_{k,j} P_k$$

- Since we have adjacency matrix A instead of P, columns W_j of W are found as

$$\min_{W_j} \left\{ \|W_j\|_0 + \gamma \|A_j - AW_j\|_2^2 \quad \text{s.t.} \quad W_{j,j} = 0,\ j = 1, \ldots, n \right\}$$

Sparse Subspace Clustering (SSC)

- Problem above can be rewritten in an equivalent form as

$$\min_{W_j} \|A_j - AW_j\|_2^2 \quad \text{s.t.} \quad \|W_j\|_0 \leq L, \quad W_{jj} = 0, \quad j = 1, \ldots, n$$

  where L is the maximum number of nonzero elements in each column of W.
- We solve the problem using the **Orthogonal Matching Pursuit (OMP)** algorithm with $L = K$
- Given W, the **affinity matrix** is defined as $|W| + |W^T|$
- The cluster assignment is then obtained by applying the **spectral clustering** to $|W| + |W^T|$

Practical Estimation of the Number of Communities

Using $\hat{P} \equiv \hat{P}(K)$, the estimator $\hat{K}$ of K, can be found by solving the following optimization problem:

$$\hat{K} = \underset{K}{\operatorname{argmin}} \{ \| \hat{P}(K) - A \|_F^2 + \operatorname{Pen}(n, K) \}$$

with

$$\operatorname{Pen}(n, K) = \rho(A) n K \sqrt{\ln n \, (\ln K)^3}$$

where $\rho(A)$ is the density of matrix A, the proportion of nonzero entries of A

This is an empirical formula which was validated by simulations

Real Data Example: the Butterfly Similarity Network

- Leeds Butterfly dataset (Wang *et al.* (2018)) contains fine-grained images of **832 butterfly species that belong to 10 different classes**
- There are **between 55 and 100 images in each class**
- 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 $K_* = 4$ largest classes** as an unweighted graph with **373 nodes and 20566 edges**
- We **draw an edge between the nodes if the visual similarity between those nodes is greater than zero**

The butterfly similarity network

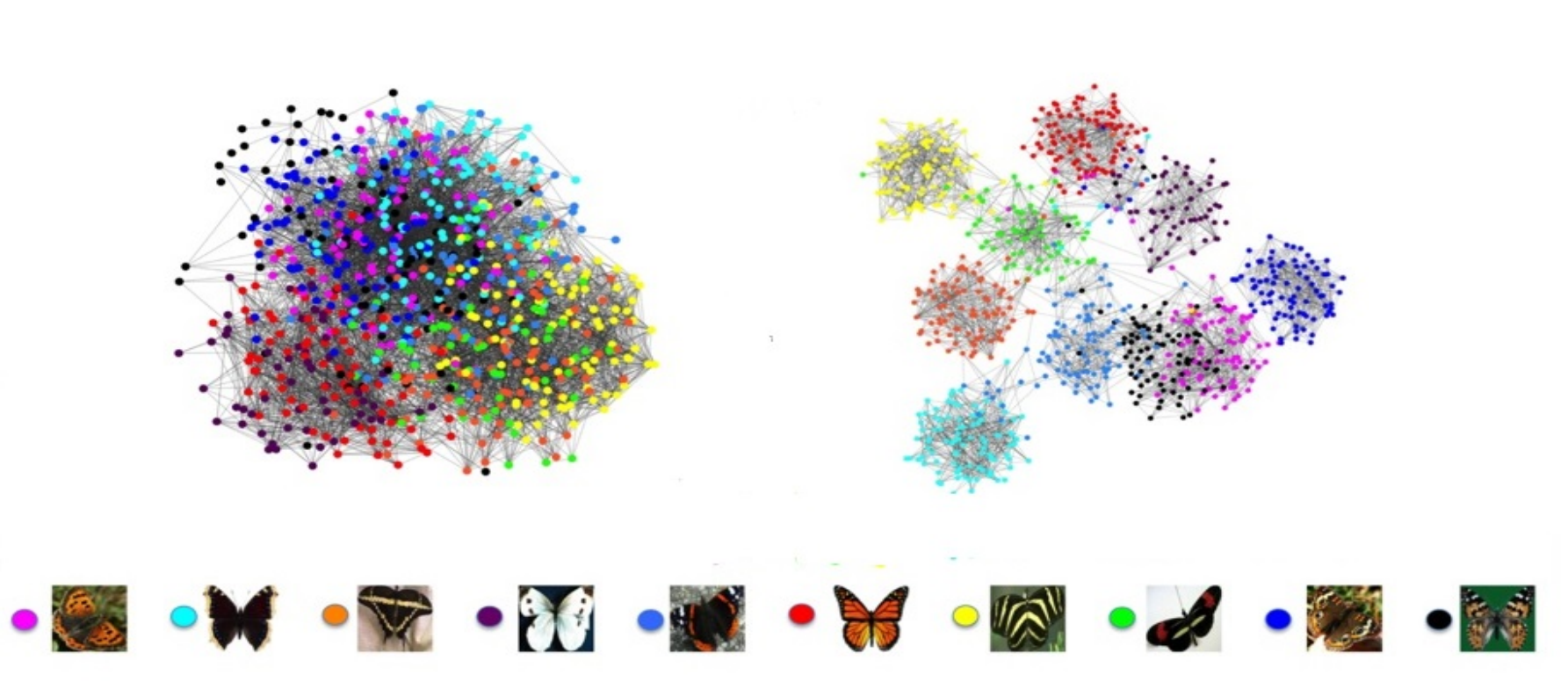

Figure 2 : A butterfly similarity network extracted from the Leeds Butterfly dataset described in Wang *et al.* (2018).

The butterfly similarity network

- We carry out clustering of the nodes using the **Sparse Subspace Clustering (SSC)** and the **Spectral Clustering (SC)** and compared the clustering assignments of both methods with the true class specifications of the species
- The **SSC** provides **89%** accuracy
- The **SC** provides **64%** accuracy
- In addition, we estimated the number of classes with $K = 2 \ldots 6$ obtaining the **true number of clusters**

Real Data Example: a Human Brain Functional Network

- We analyze a human brain functional network, measured using the **resting-state fMRI**
- In particular we use the **co-activation brain connectivity dataset** described in Crossley *et al.* (2013)
- The brain is partitioned into **638 distinct regions**
- The network topology is characterized by a weighted graph
- We set all nonzero weights to one, obtaining the **network with 18625 undirected edges**

A human brain co-activation functional network

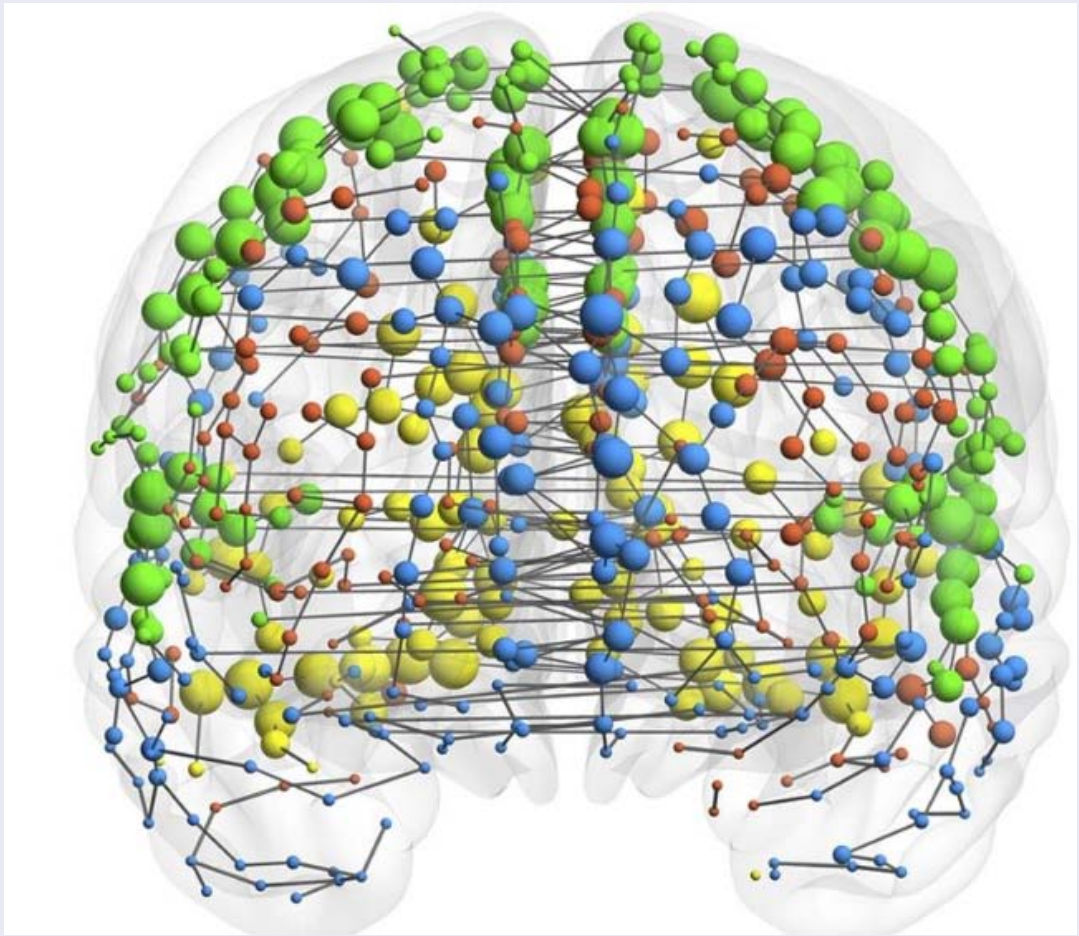

Figure 3 : A human brain co-activation functional network measured using the fMRI described in Crossley *et al.* (2013)

A human brain functional network

- Since the true number of clusters and the true clustering are unknown, we first tested the number of clusters with $K = 2 \ldots 10$ obtaining $\hat{K} = 6$
- Subsequently, we applied the SSC for partitioning the network into blocks and derived the estimator $\hat{P}$ of P_*
- The true probability matrix P_* is unknown, we can only report that

$$n^{-2} \|\hat{P} - A\|_F^2 = 0.05$$

This indicates high agreement between the two matrices

A human brain co-activation functional network

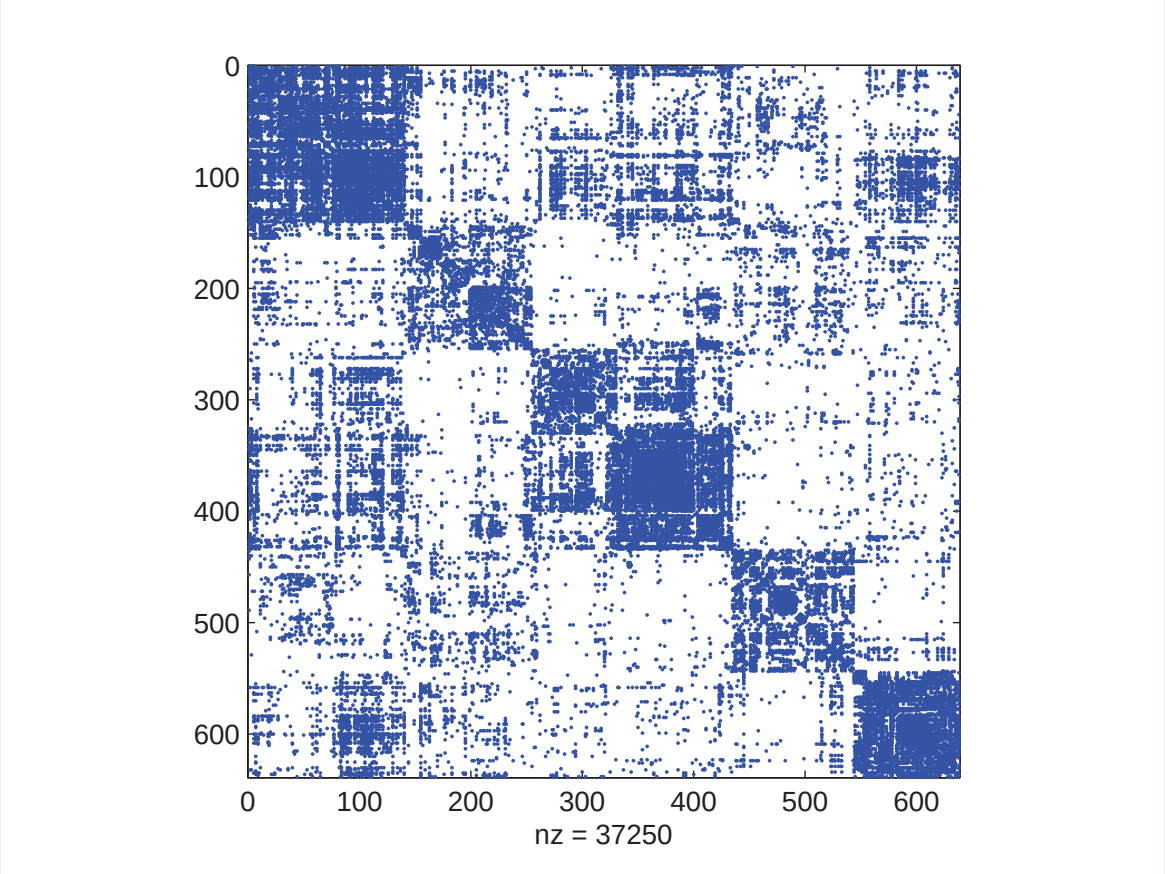

Figure 4 : The adjacency matrix of the graph after clustering

Advantages of the PABM

- Due to the flexibility of its spectral properties, **PABM provides a valuable tool for modeling** diverse networks, especially in the situations when the mixed memberships are not allowed
- PABM **does not require identifiability conditions** for its fitting. The number of communities can be arbitrary and unknown, and grow with the number of nodes.
- PABM **allows to model structural sparsity** by setting some of the small probabilities of connections to zero. This allows to better understand the block-dependent sparsity patterns.

Our Contributions

- We provide **probabilistic guarantees for non-asymptotic upper bounds on estimation and clustering errors** that are **valid for any combination of the parameters**
- We identified communities in the PABM by Sparse Subspace Clustering. This is **the first application of the Subspace Clustering to random network models**
- Our papers contains **simulations and real data examples**. Our simulation study as well as the real data examples handle **various number of communities** between 2 and 8
- We demonstrate the advantages of the PABM for **modeling networks that appear in biological sciences**

References

- **M. Noroozi, R. Rimal and M. Pensky (2020)** Estimation and Clustering in Popularity Adjusted Stochastic Block Model. *J. Royal Stat. Soc., Ser. B*, accepted. *ArXiv 1902.00431*.

- **M. Noroozi, M. Pensky and R. Rimal (2020)** Sparse Popularity Adjusted Stochastic Block Model. *ArXiv: 1910.01931*

- **S. Sengupta and Y. Chen (2018)** A block model for node popularity in networks with community structure. *J. Royal Stat. Soc., Ser. B*, **80**, 365–386.

NOBODY KNOWS

NOBODY KNOWS