Data-Integrated Stochastic Block Models

Carter Allen

October 28, 2019

Motivation

Currently, we wish to better understand the gene networks related to **systemic sclerosis** (a.k.a *scleroderma* or *SSc*).

SSc is a chronic autoimmune disease involving fibrosis across multiple body systems. Currently, the etiology remains unknown and there are *no effective treatments available*.

Since SSc is a complex disease, genetic signals from experiments are often **weak and widespread**, posing challenges for classical network models.

Goal: Extend statistical models for network data to the data integration setting to address the issue of weak & widespread signal.

The Stochastic Block Model

The **stochastic block model** (SBM) is a generative model for network data. A simple, symmetric, and non-reflexive graph $\mathbf{G}$ is said to follow an SBM($n, \mathbf{P}, \mathbf{b}$) if

1. $\mathbf{G}$ has n nodes (vertices) denoted by $\mathcal{N} = \{\eta_1, \eta_2, \ldots \eta_n\}$.

2. Each node has exactly one label, denoted b_i for $i = 1,\ldots,n$.

3. An edge exists between nodes η_i and η_j randomly with probability $\mathbf{P}_{b_i,b_j}$ for $i = 1,\ldots,n$ and $j = i + 1,\ldots,n$.

Note: Often an additional parameter $\boldsymbol{\pi}$ is introduced to control the size of each community.

Bayesian Inference

Bayesian inference procedures for the SBM fall into two camps: Parametric and Non-parametric inference.

<u>Parametric Bayesian Inference</u> assumes the number of blocks B, and hence the dimension of the entire model, is known. These models generally have the following hierarchy.

$$\boldsymbol{\pi} \sim \text{Dirichlet}(B, \boldsymbol{\alpha})$$

$$P_{ab} \sim \text{Beta}(\beta_1, \beta_2)$$

$$G_{ij} \,|\, B, \mathbf{P}, \boldsymbol{\pi} \sim \text{Bernoulli}(P_{b_i, b_j})$$

Bayesian Inference

Non-parametric Bayesian Inference allows the model dimension B to be estimated from the data.

In `graph-tool`, a Python library with extensive features for working with network data and fitting SBMs, the non-parametric approach is used.

In principle, allowing B to vary greatly complicates model estimation. However, Piexoto (the author of `graph-tool`) outlines an efficient framework in several papers.

Bayesian Inference

Suppose interest lies primarily in estimating the community labeling vector $\mathbf{b}$.

$$\mathrm{P}(\mathbf{b} \mid \mathbf{G}) = \frac{\mathrm{P}(\mathbf{G} \mid \mathbf{b})\mathrm{P}(\mathbf{b})}{\mathrm{P}(\mathbf{G})},$$

where $\mathrm{P}(\mathbf{G} \mid \mathbf{b}) = \int_{\boldsymbol{\theta}} \mathrm{P}(\mathbf{G} \mid \boldsymbol{\theta}, \mathbf{b})\mathrm{P}(\boldsymbol{\theta} \mid \mathbf{b}) d\boldsymbol{\theta}$, and $\mathrm{P}(\mathbf{G})$ is a normalizing constant with respect to $\mathbf{b}$.

First, what is a good choice for $\mathrm{P}(\mathbf{b})$?

Bayesian Inference

Let $\mathrm{P}(\mathbf{b}) = \mathrm{P}(\mathbf{b}\,|\,\mathbf{n})\mathrm{P}(\mathbf{n}\,|\,B)\mathrm{P}(B)$, where $\mathbf{n} = (n_1, n_2, \ldots n_B)$ encodes the number of nodes in each community. We let

$$\mathrm{P}(\mathbf{b}\,|\,\mathbf{n}) = \left(\frac{N!}{\prod_{r=1}^{B} n_r!}\right)^{-1} \qquad \mathrm{P}(\mathbf{n}\,|\,B) = \binom{N-1}{B-1}^{-1} \qquad P(B) = \frac{1}{N}\,.$$

$\mathrm{P}(B)$ assigns equal probability to all $B \in \{1,\ldots,N\}$.

$\mathrm{P}(\mathbf{n}\,|\,B)$ gives equal probability to each of the possible ways to divide N total counts into B non-empty bins.

$\mathrm{P}(\mathbf{b}\,|\,\mathbf{n})$ gives equal probability to all possible arrangements of $\mathbf{n}$.

Bayesian Inference

Having dealt with $\mathrm{P}(\mathbf{b})$, we must specify the form of $\mathrm{P}(\mathbf{G}\,|\,\mathbf{b})$. Recall,

$$\mathrm{P}(\mathbf{G}\,|\,\mathbf{b}) = \int_{\boldsymbol{\theta}} \mathrm{P}(\mathbf{G}\,|\,\boldsymbol{\theta}, \mathbf{b})\mathrm{P}(\boldsymbol{\theta}\,|\,\mathbf{b})d\boldsymbol{\theta},$$

where $\boldsymbol{\theta}$ are the parameters that control the placement of edges in the graph $\mathbf{G}$.

The simplest SBM places edges according to independent Bernoulli trials, leading to the likelihood

$$\mathrm{P}(\mathbf{G}\,|\,\mathbf{b}, \mathbf{P}) = \prod_{i<j} P_{b_i,b_j}^{G_{ij}}(1 - P_{b_i,b_j})^{1-G_{ij}}.$$

Bayesian Inference

Piexoto chooses to generalize the simple SBM to accommodate *multigraphs,* allowing for multiple edges between any two nodes.

A natural model for edge placement in this case is **Poisson**. Thus,

$$\mathrm{P}(\mathbf{G} \mid \boldsymbol{\lambda}, \mathbf{b}) = \prod_{i<j} \frac{e^{-\lambda_{b_i,b_j}} \lambda_{b_i,b_j}^{G_{ij}}}{G_{ij}!} \times \prod_{i} \frac{e^{-\lambda_{b_i,b_i}/2} (\lambda_{b_i,b_i}/2)^{G_{ii}/2}}{G_{ii}/2!},$$

where $\lambda_{r,s}$ is the average number of edges between any two nodes in communities r and s, and we also allow for *reflexive,* but *still undirected* edges.

Bayesian Inference

To complete the specification of $P(\mathbf{G}\,|\,\mathbf{b})$, we must choose the form of $P(\boldsymbol{\theta}\,|\,\mathbf{b})$, or in case of the Poisson model, $P(\boldsymbol{\lambda}\,|\,\mathbf{b})$.

A natural choice is the **exponential** distribution. We let

$$\lambda_{rs}\,|\,\mathbf{b} \sim \text{exponential}(\bar{\lambda}_{rs}),$$

for $r \leq s$ and where $\bar{\lambda}_{rs}$ is the average number of edges between any two nodes in blocks r and s, which we assume is

$$\bar{\lambda}_{rs} = \frac{\bar{\lambda}(1+\delta_{rs})}{n_r n_s}, \text{ where } \bar{\lambda} = \frac{2E}{B(B+1)}.$$

Bayesian Inference

The Bayesian model is now fully specified and can be estimated using standard techniques that sample from

$$\mathrm{P}(\mathbf{b}\,|\,\mathbf{G}) \propto \mathrm{P}(\mathbf{G}\,|\,\mathbf{b})\mathrm{P}(\mathbf{b})\,.$$

However, through integration of $\mathrm{P}(\mathbf{G}\,|\,\boldsymbol{\lambda}, \mathbf{b})\mathrm{P}(\boldsymbol{\lambda}\,|\,\mathbf{b})$ with respect to $\boldsymbol{\lambda}$ and various combinatorial derivations, Piexoto is able to claim

$$\mathrm{P}(\mathbf{G}\,|\,\mathbf{b})\mathrm{P}(\mathbf{b}) = \mathrm{P}(\mathbf{G}\,|\,\mathbf{e}, \mathbf{b})\mathrm{P}(\mathbf{e}, \mathbf{b}) = 2^{-\Sigma},$$

where Σ is known as the **description length**, a concept borrowed from information theory. So, **minimizing Σ maximizes $\mathrm{P}(\mathbf{b}\,|\,\mathbf{G})$**.

MCMC Sampling

The `graph-tool` package includes functions for MCMC sampling of model parameters. In general, we sample the posterior distribution of **b** as follows.

1) For node η_i ($i = 1,\ldots,n$): randomly select η_j from the neighbors of η_i and record its community membership b_j.

2) Randomly draw a proposal community for η_i, namely b_i, from the B possibilities with uniform probability.

3) Accept the proposal b_i with probability R_j (more details later).

4) If b_i is rejected, choose edge e_k randomly from all edges incident upon η_j and assign $b_i = b_k$, where b_k is the community membership of η_k, the node connected to b_j by e_k.

MCMC Sampling

$$\mathrm{P}(b_i \to b_{i*} \,|\, b_j) = \frac{(1 - R_j)E_{i*j}}{E_j} + \frac{R_j}{B},$$

$$R_j = \frac{\epsilon B}{E_j + \epsilon B},$$

E_{i*j} is the number of edges between communities b_{i*} and b_j.

$$E_j = \sum_{i*} E_{i*j}.$$

$\epsilon > 0$ is a tuning parameter used to control the acceptance probabilities.

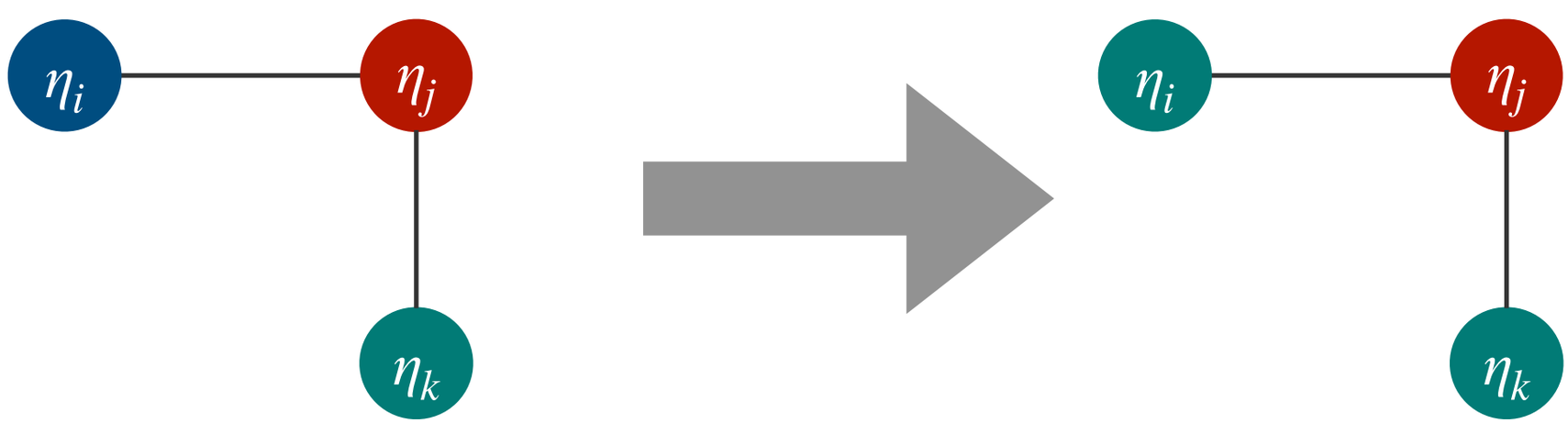
MCMC Sampling

1) Select η_i and a random neighbor η_j :

2) Randomly propose a new b_i :

3) Accept or reject b_i :

4) If reject b_i, choose $b_i = b_k$:



Simulation Studies

Sampling SBMs

In general, we sample a graph $\mathbf{G}$ from an SBM($n, \mathbf{P}, \mathbf{b}$) as follows.

1. Define the set of nodes of $\mathbf{G}$, $\mathcal{N} = \{\eta_1, \eta_2, \ldots, \eta_n\}$, and label them according to $\mathbf{b} = \{b_1, b_2, \ldots, b_n\}$, where $b_i \in \{1,\ldots,B\}$.

2. For all pairs of nodes, (η_i, η_j) subject to $i < j$, randomly place an edge between η_i and η_j with probability $\mathbf{P}_{b_i,b_j}$.

Note: The resultant graph $\mathbf{G}$ is simple, symmetric, and non-reflexive.

Data Integration

Under each setting of n and $\mathbf{P}$, graphs $\mathbf{G}_1$ and $\mathbf{G}_2$ are integrated into the graph $\mathbf{G}$ using the **edge union** approach.

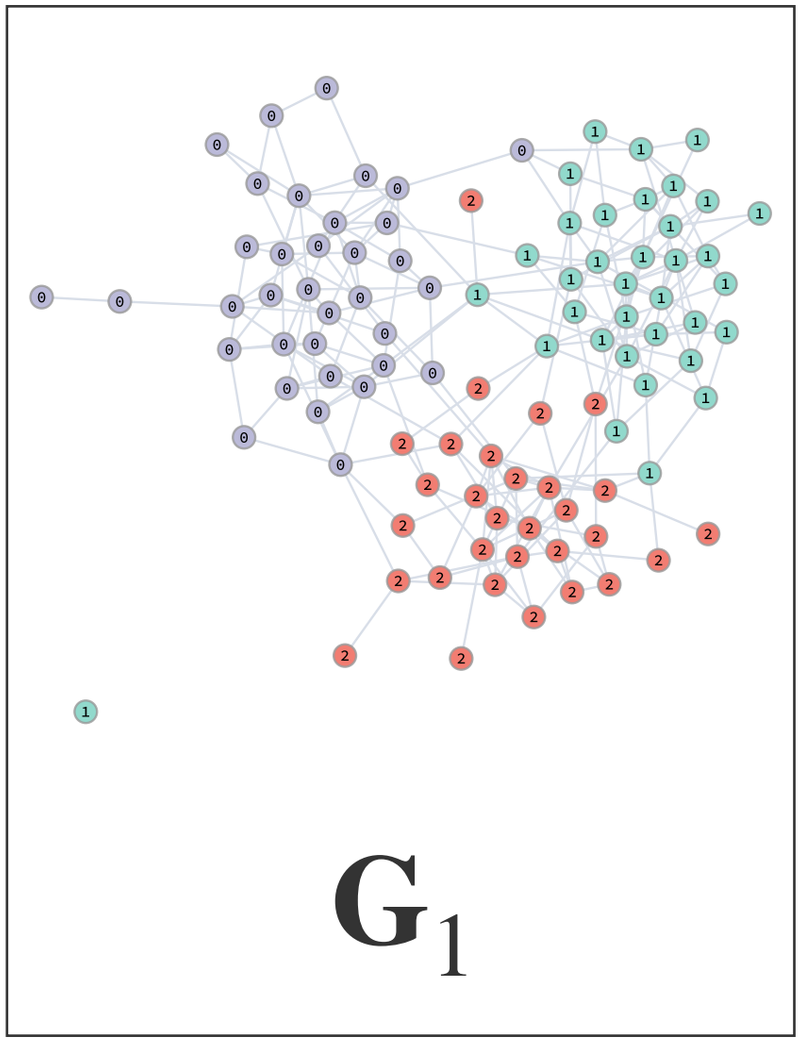
That is, the set of edges in $\mathbf{G}$, denoted by $\mathscr{E}$, is equal to $\mathscr{E}_1 \cup \mathscr{E}_2$, where $\mathscr{E}_1$ and $\mathscr{E}_2$ are the sets of edges in $\mathbf{G}_1$ and $\mathbf{G}_2$, respectively.

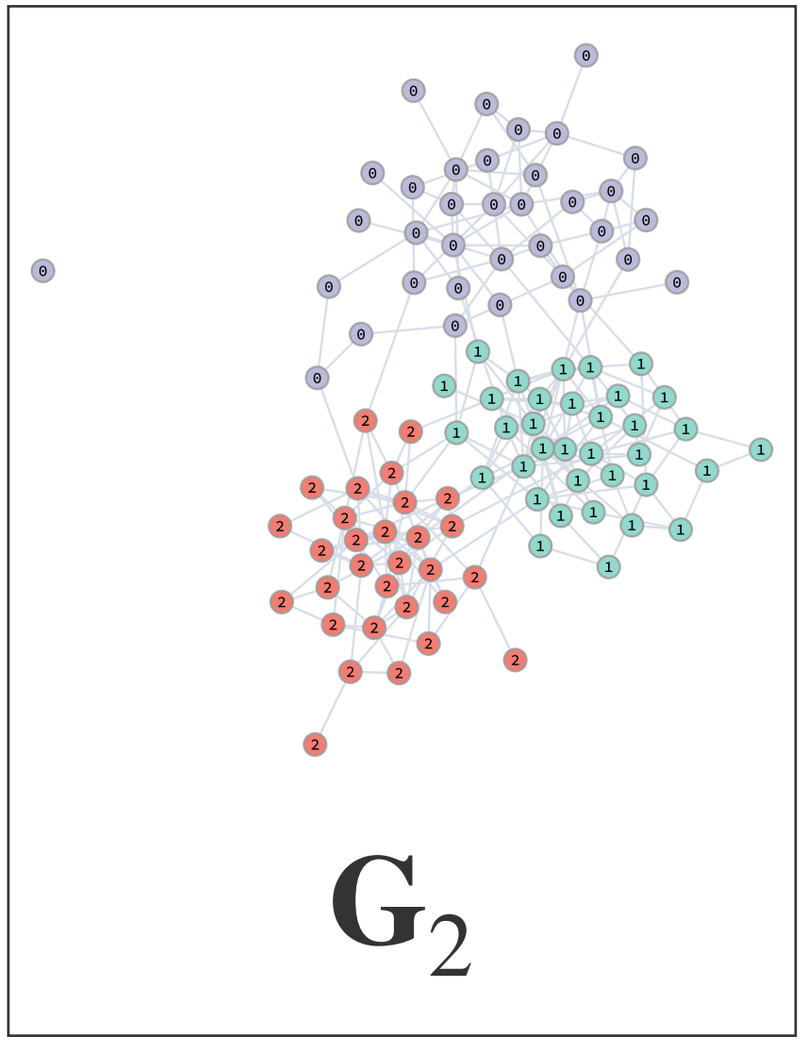
Edge Union:

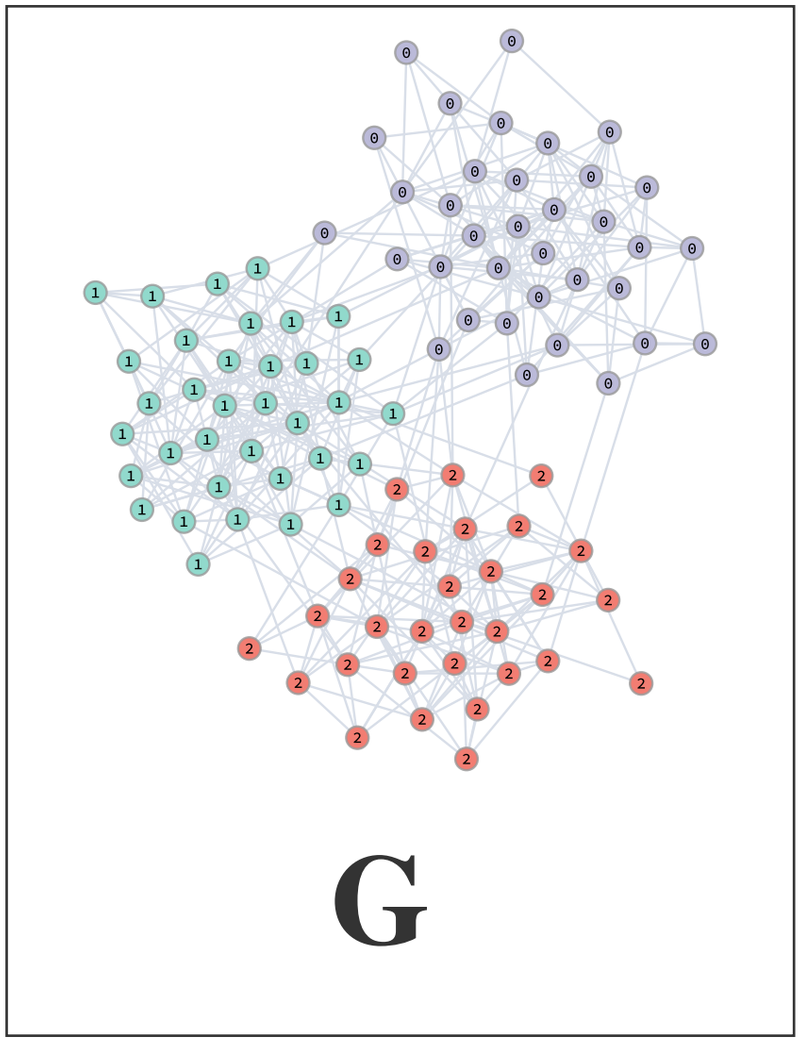
1. Since $\mathbf{G}_1$ and $\mathbf{G}_2$ share the same node set, so will $\mathbf{G}$. So, initialize $\mathbf{G}$ by setting $\mathbf{G} = \mathbf{G}_1$. Thus, $\mathscr{E} = \mathscr{E}_1$.

2. For $e \in \mathscr{E}_2$, add e to $\mathscr{E}$ if $e \notin \mathscr{E}_1$.

Edge Union Example







$$n = 100$$

$$\mathbf{P}_{ii} = 0.12; \ \mathbf{P}_{ij} = 0.01$$

Simulation Settings

Sample graphs $\mathbf{G}_1$ and $\mathbf{G}_2$ from SBM($n, \mathbf{P}, \mathbf{b}$) under varying settings of n and $\mathbf{P}$.

Specifically, $n_1 = 50$, $n_2 = 100$, and

$$\mathbf{P}_1 = \begin{bmatrix} 0.5 & 0.1 & 0.1 \\ 0.1 & 0.5 & 0.1 \\ 0.1 & 0.1 & 0.5 \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} 0.3 & 0.1 & 0.1 \\ 0.1 & 0.3 & 0.1 \\ 0.1 & 0.1 & 0.3 \end{bmatrix}.$$

Thus we have 8 total graphs; one *iid* pair for each combination of n and $\mathbf{P}$.

Estimating Model Dimension

To evaluate the ability of the data-integrated SBM to estimate B, the number of communities, we compare the estimates $\hat{B}, \hat{B}_1$, and $\hat{B}_2$ obtained by fitting a SBM to $\mathbf{G}$, $\mathbf{G}_1$, and $\mathbf{G}_2$, respectively.

For $i = 1,\ldots,I$:

1. Generate $\mathbf{G}_i$, $\mathbf{G}_{1i}$, and $\mathbf{G}_{2i}$ as described previously.

2. Obtain $\hat{B}_i, \hat{B}_{1i}$, and $\hat{B}_{2i}$ from the minimum the description length criterion.

3. Store $\hat{B}_i, \hat{B}_{1i}$, and $\hat{B}_{2i}$.

Table 1: *Simulation results for SBMs fit to G, G_1, and G_2* **under $\mathbf{P}_1$ and $\mathbf{P}_2$**, *with $n = 50$. The proportion of correctly specified models and the average number of clusters estimated are shown.*

	$\mathbf{P}_1$		$\mathbf{P}_2$	
$I = 100$	$\frac{1}{I}\sum_{i=1}^{I} 1_{\hat{B}=B}$	$\frac{1}{I}\sum_{i=1}^{I} \hat{B}$	$\frac{1}{I}\sum_{i=1}^{I} 1_{\hat{B}=B}$	$\frac{1}{I}\sum_{i=1}^{I} \hat{B}$
G	***0.85***	***2.85***	***0.00***	***1.01***
G_1	0.47	2.50	0.00	1.00
G_2	0.42	2.38	0.00	1.00

Table 2: *Simulation results for SBMs fit to G, G_1, and G_2* **under $\mathbf{P}_1$ and $\mathbf{P}_2$**, *with $n = 100$. The proportion of correctly specified models and the average number of clusters estimated are shown.*

	$\mathbf{P}_1$		$\mathbf{P}_2$	
$I = 100$	$\frac{1}{I}\sum_{i=1}^{I} 1_{\hat{B}=B}$	$\frac{1}{I}\sum_{i=1}^{I} \hat{B}$	$\frac{1}{I}\sum_{i=1}^{I} 1_{\hat{B}=B}$	$\frac{1}{I}\sum_{i=1}^{I} \hat{B}$
G	***1.00***	***3.00***	***0.97***	***2.99***
G_1	1.00	3.00	0.23	1.70
G_2	0.99	3.01	0.19	1.76

Classification Performance

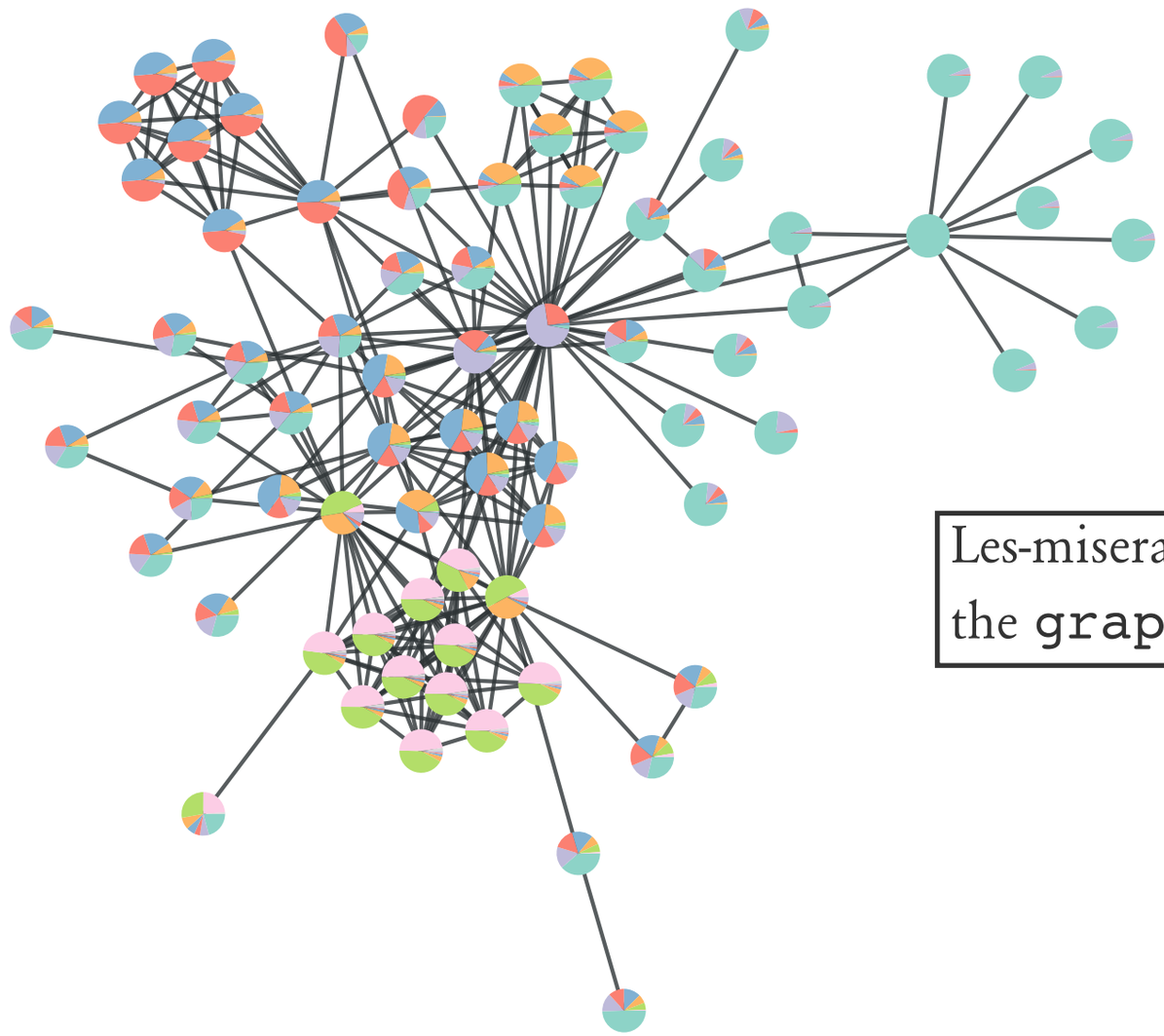
Goal: assess the classification performance of the data integrated SBM (i.e., how often are nodes assigned to the correct communities?).

The problem of **label switching** complicates the calculation of sensitivity and specificity.

Since B is estimated by the model, B may vary from one MCMC iteration to the next, making it difficult to apply relabeling algorithms to MCMC samples.

Solution: assess the posterior distribution of cluster assignments for each node in the network.

Classification Performance



Les-miserables example data from the `graph-tool` package.

For each node η_i, we plot the posterior probabilities $\mathrm{P}(b_i = 1), \mathrm{P}(b_i = 2), \ldots, \mathrm{P}(b_i = B)$ as small pie charts.

Classification Performance

To assess the classification performance of the data integrated SBM using edge union integration, we conduct the following simulation.

Setting 1: *Sparse networks with high signal to noise.*

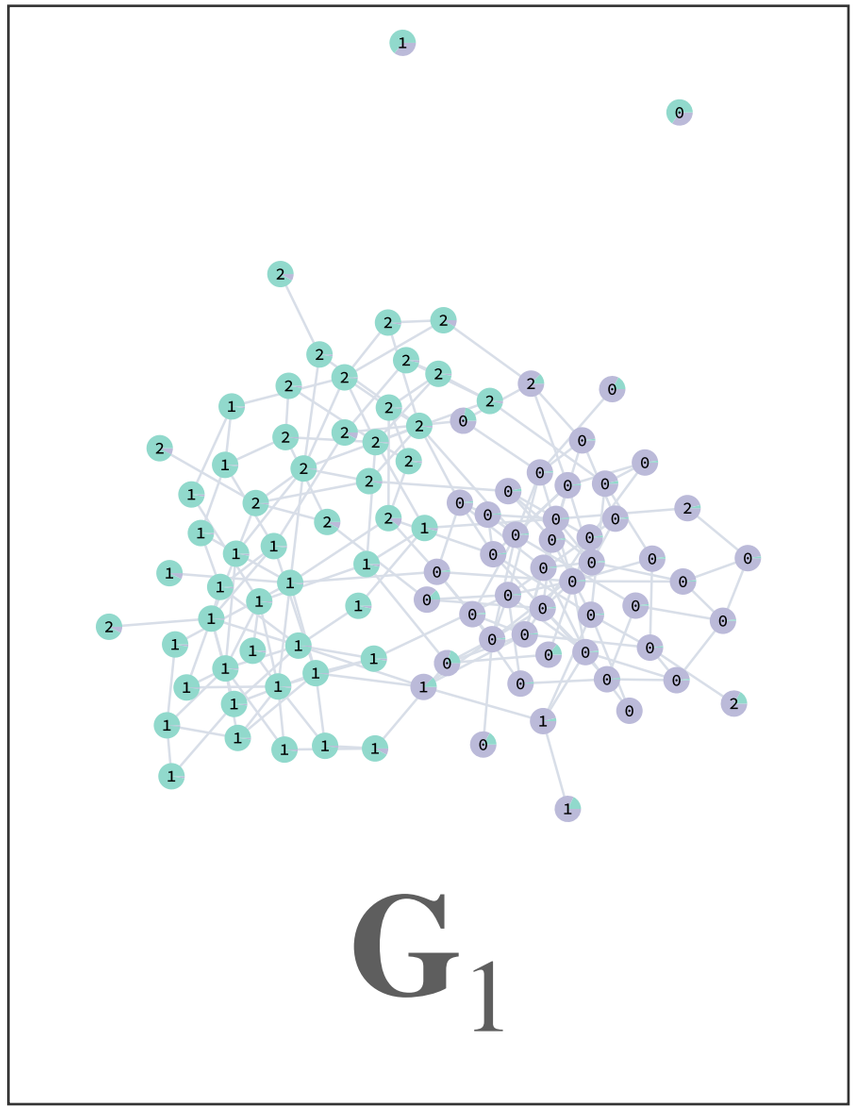
Setting 2: *Sparse networks with low signal to noise.*

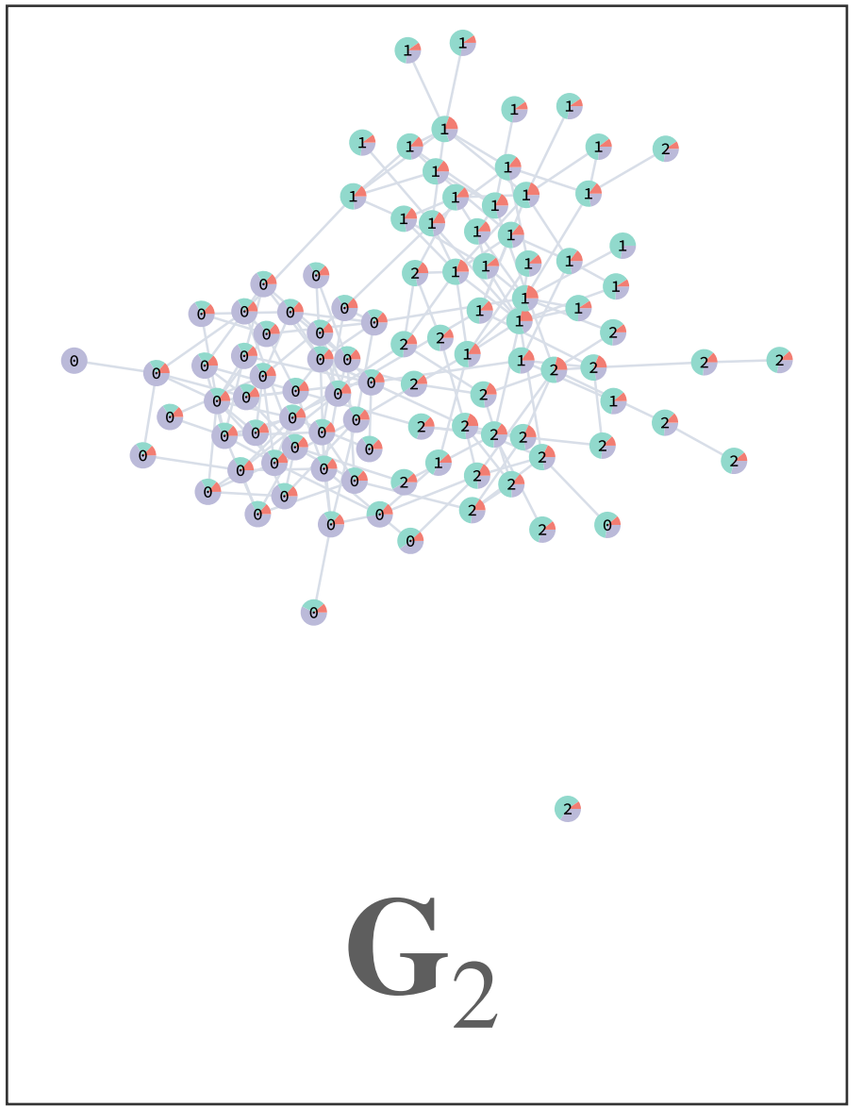
Setting 3: *Dense networks with high signal to noise.*

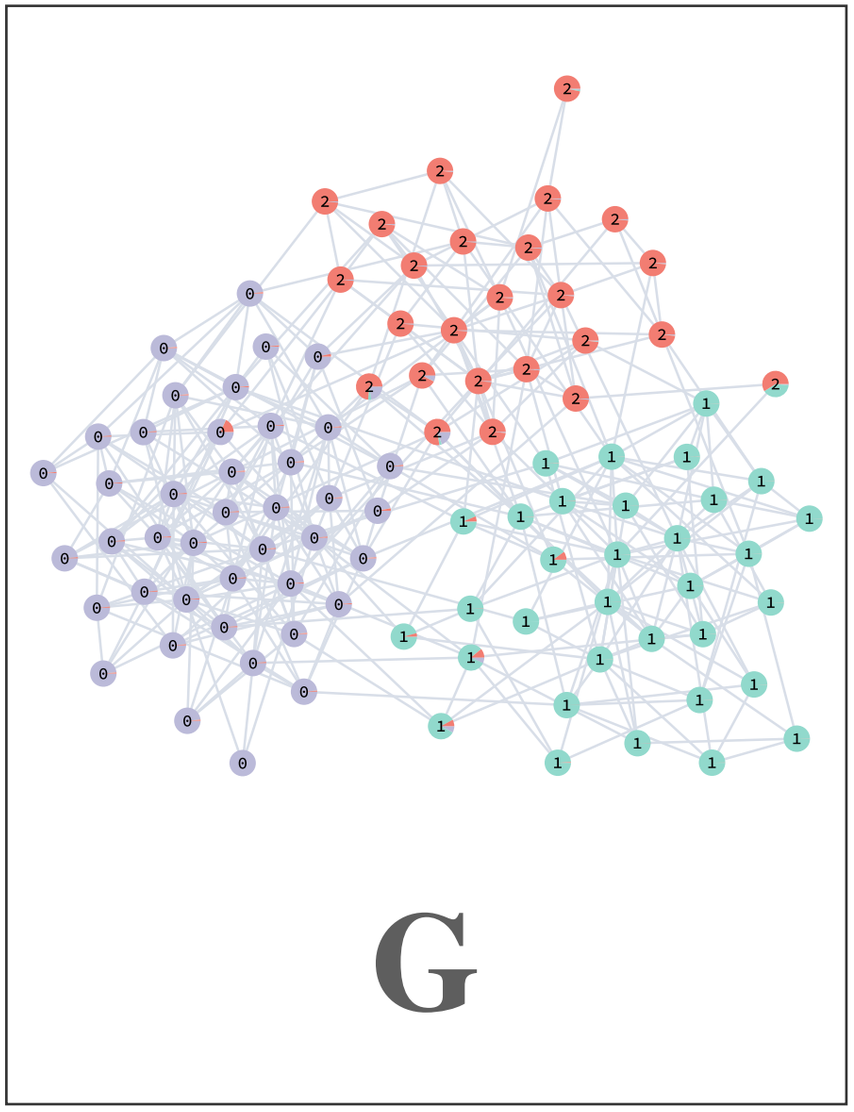
Setting 4: *Dense networks with low signal to noise.*

In each setting, we fix $n = 100$, draw $\mathbf{G}_1$ and $\mathbf{G}_2$, and use edge union to form $\mathbf{G}$. We fit SBMs to $\mathbf{G}$, $\mathbf{G}_1$, and $\mathbf{G}_2$ and examine classification performance qualitatively.

Classification Performance



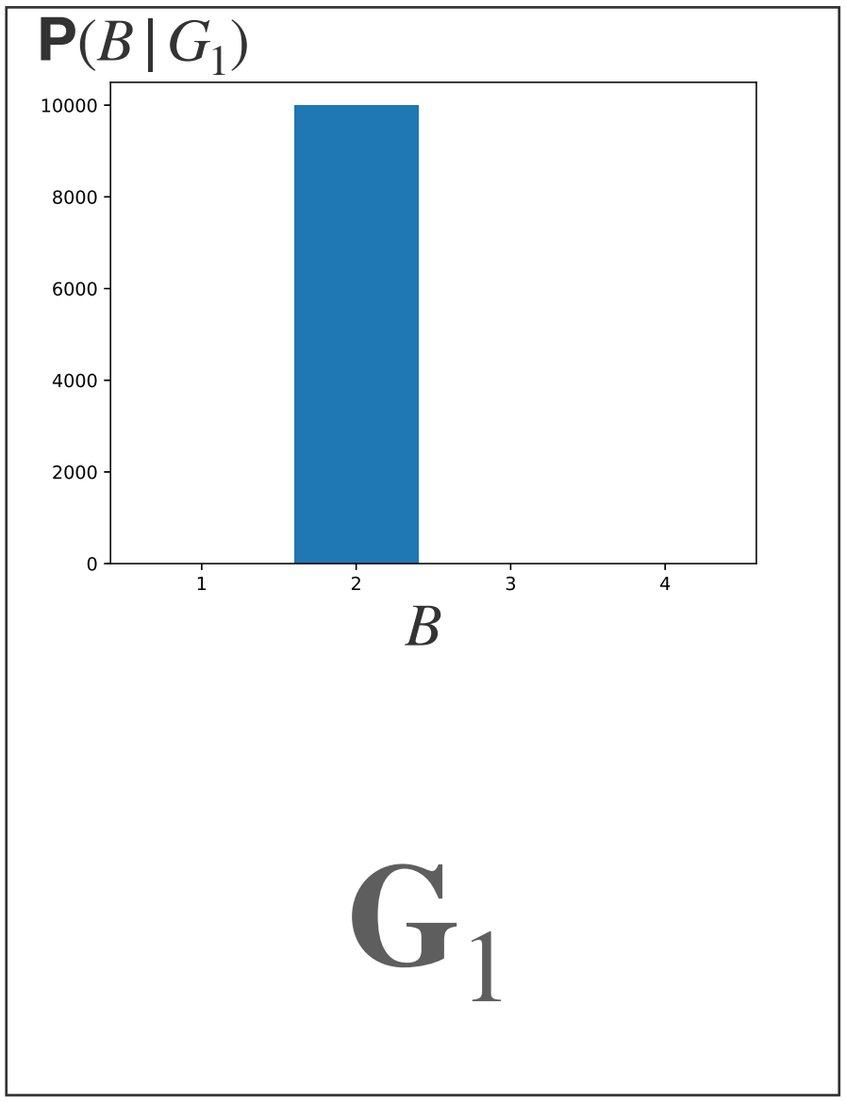


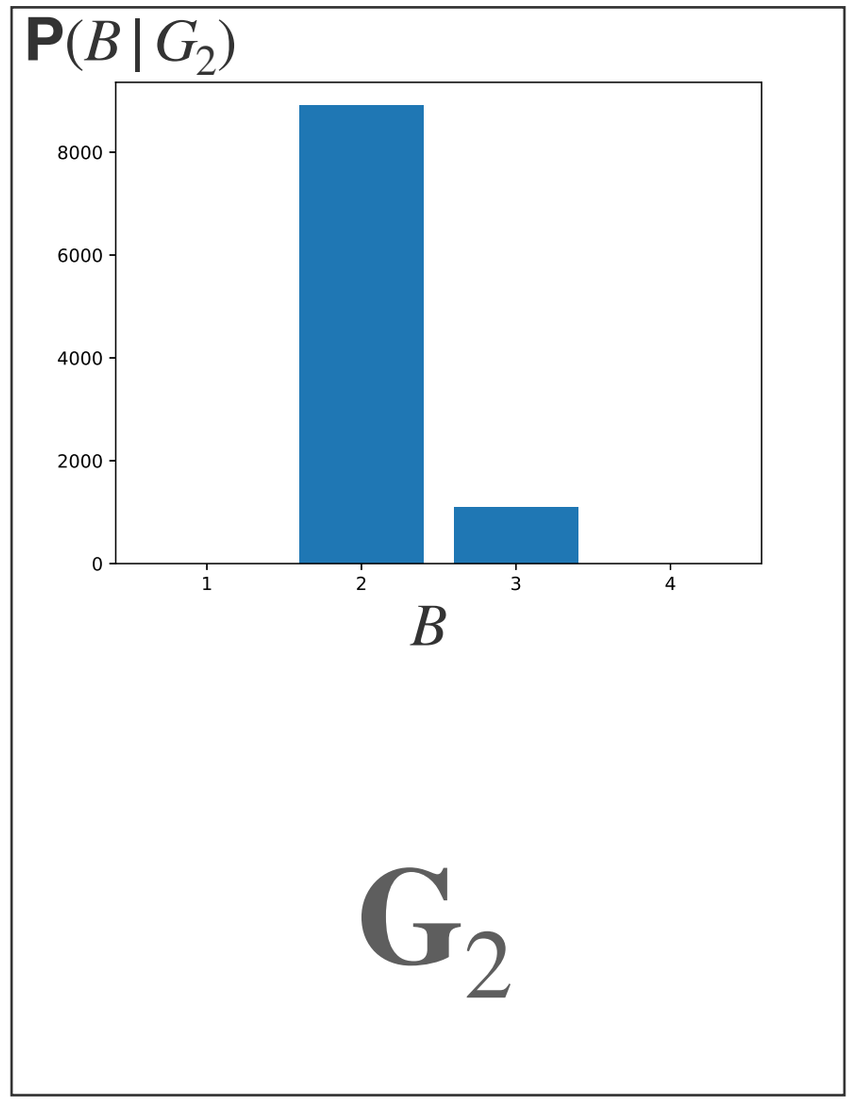


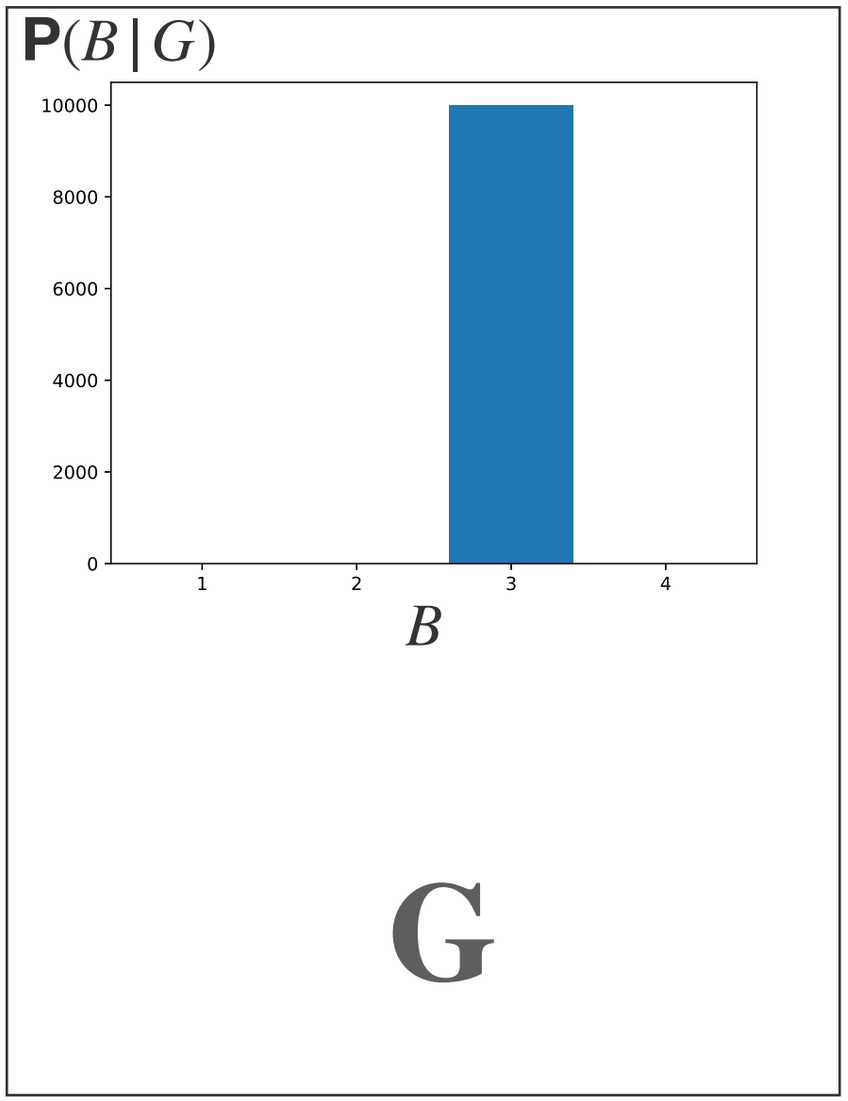
Setting 1

$$\mathbf{P}_{ii} = 0.10; \ \mathbf{P}_{ij} = 0.01$$

Estimating Model Dimension



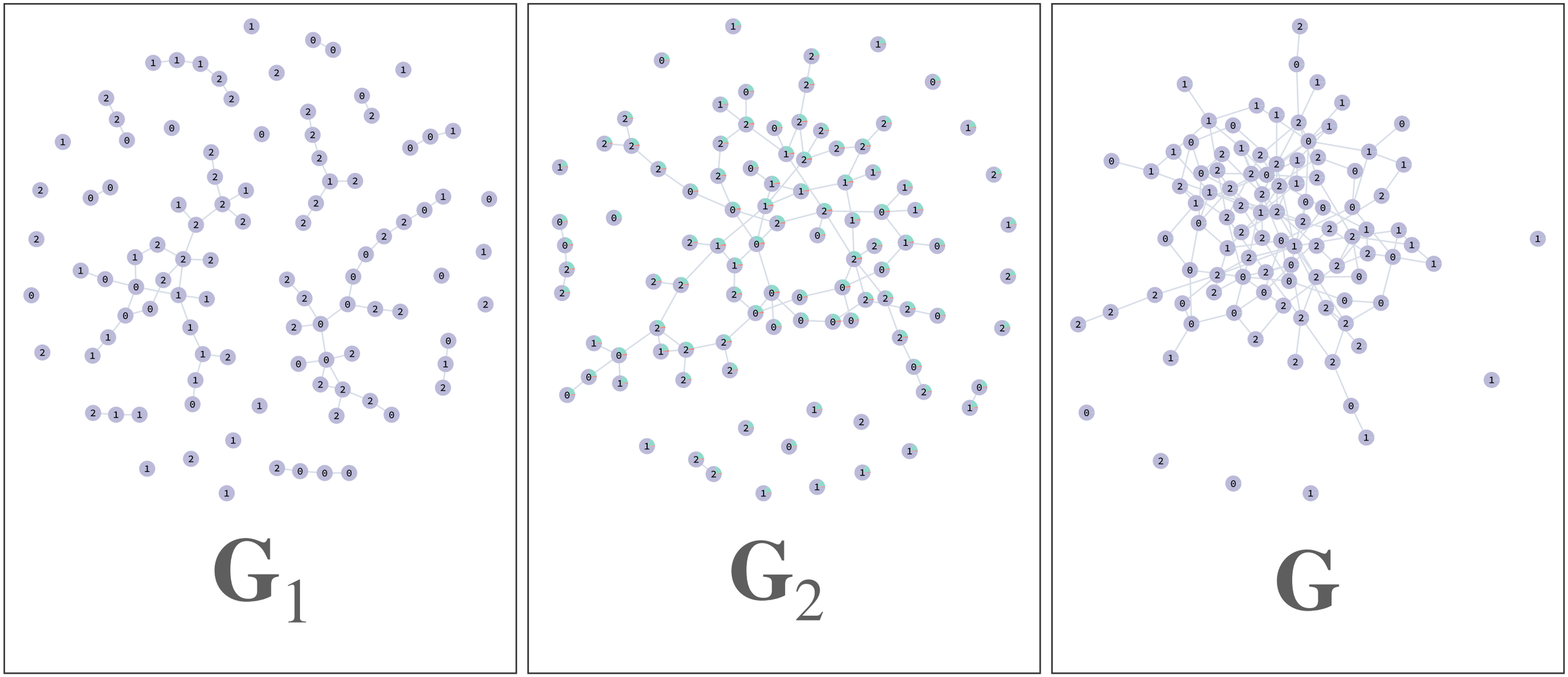




<u>Setting 1</u>

$$\mathbf{P}_{ii} = 0.10; \;\; \mathbf{P}_{ij} = 0.01$$

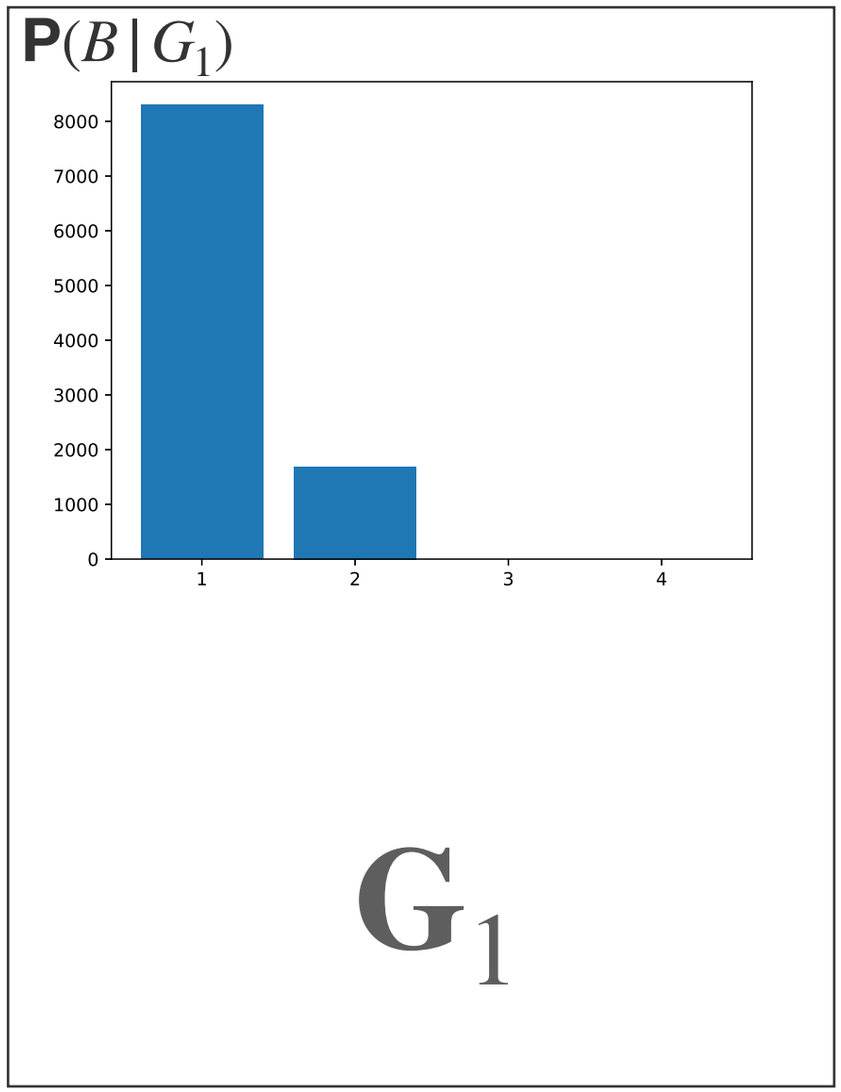
Classification Performance

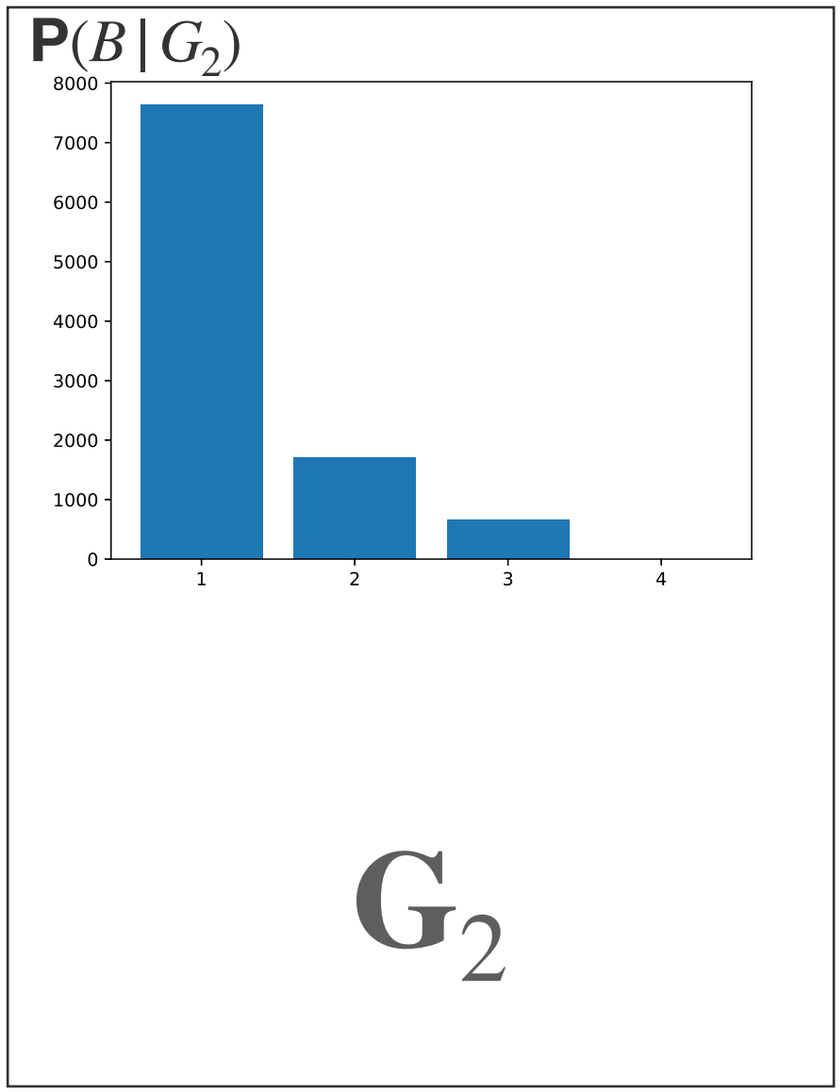


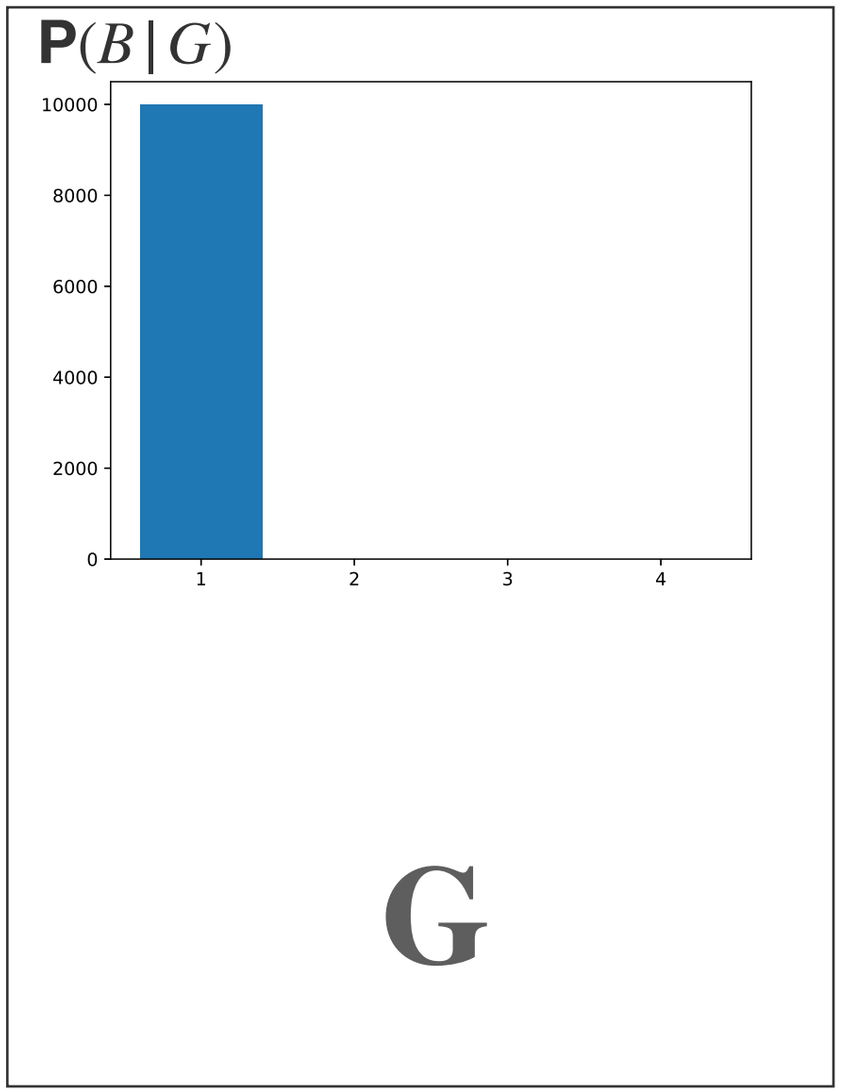
Setting 2

$$\mathbf{P}_{ii} = 0.03; \;\; \mathbf{P}_{ij} = 0.01$$

Estimating Model Dimension



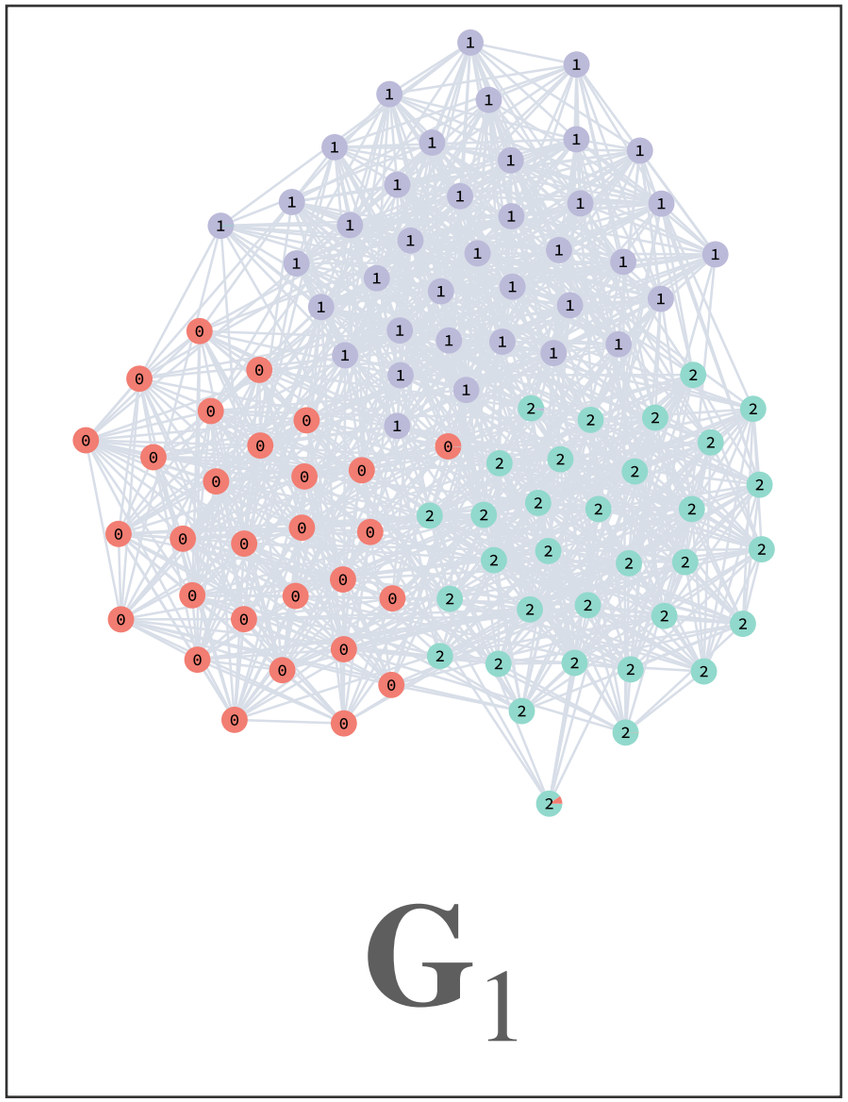


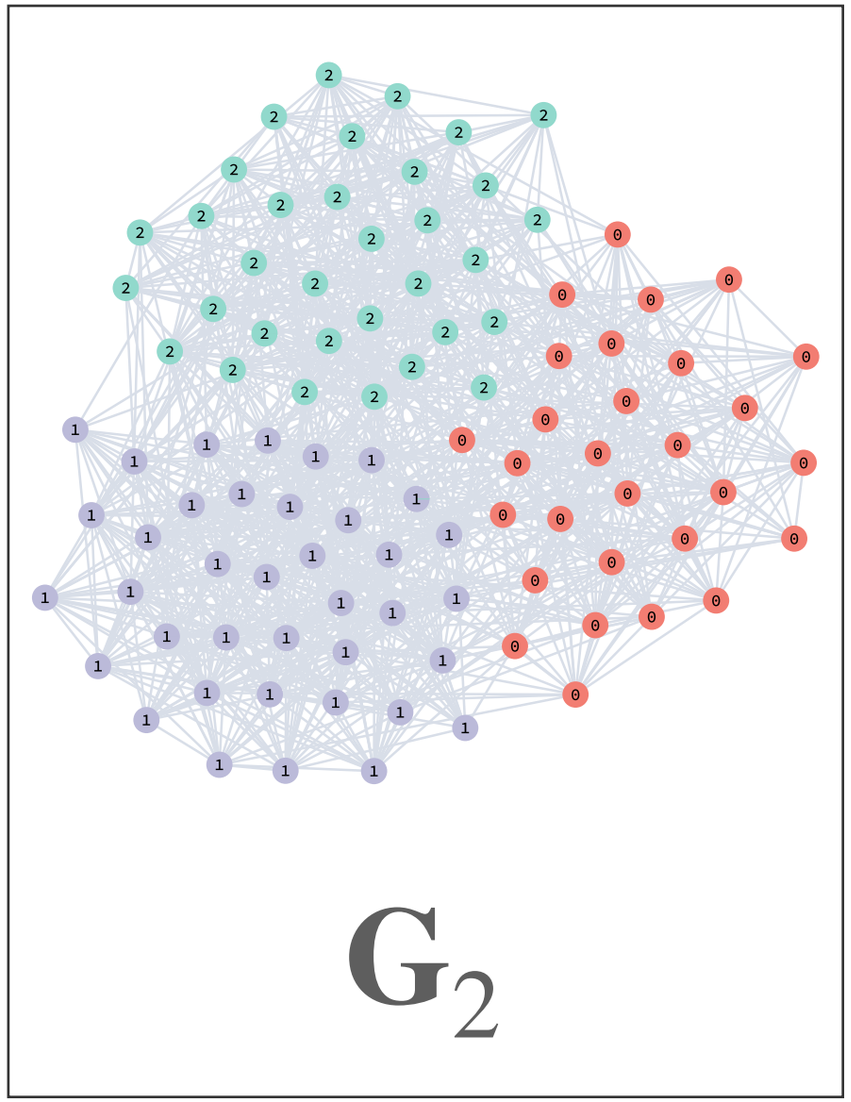


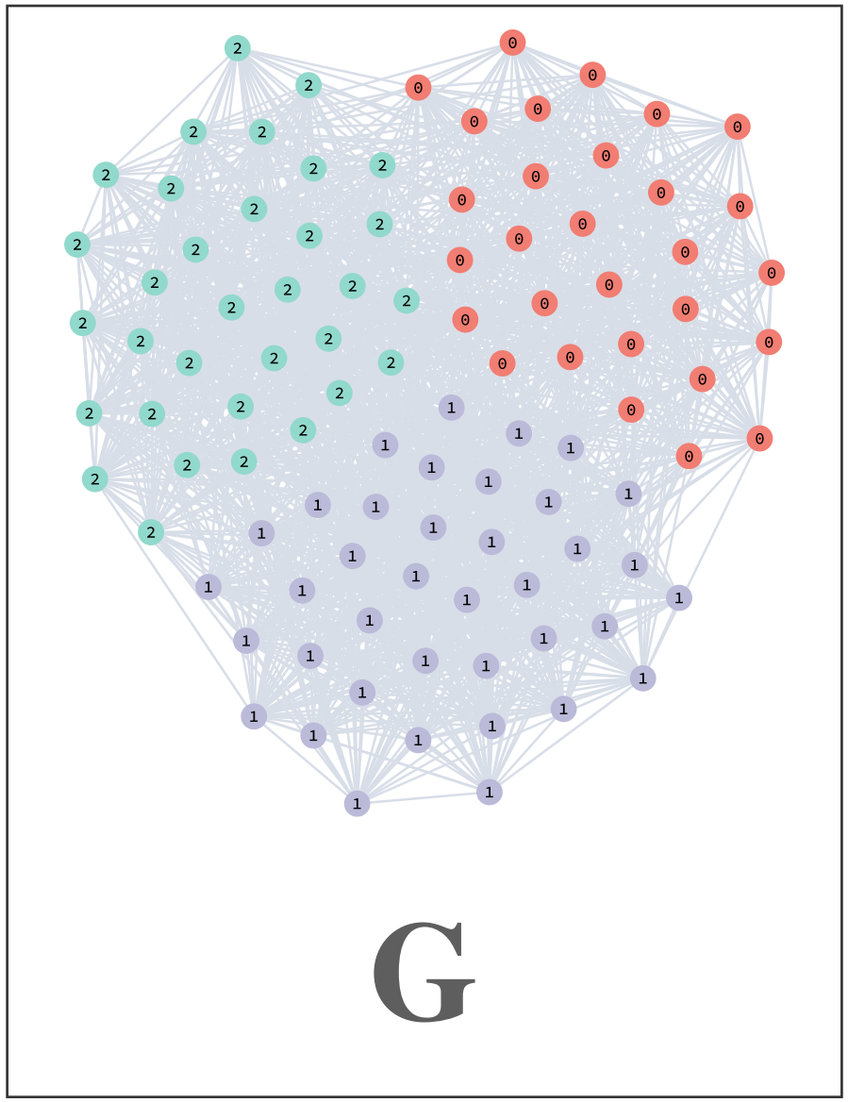
<u>Setting 2</u>

$$\mathbf{P}_{ii} = 0.03; \;\; \mathbf{P}_{ij} = 0.01$$

Classification Performance



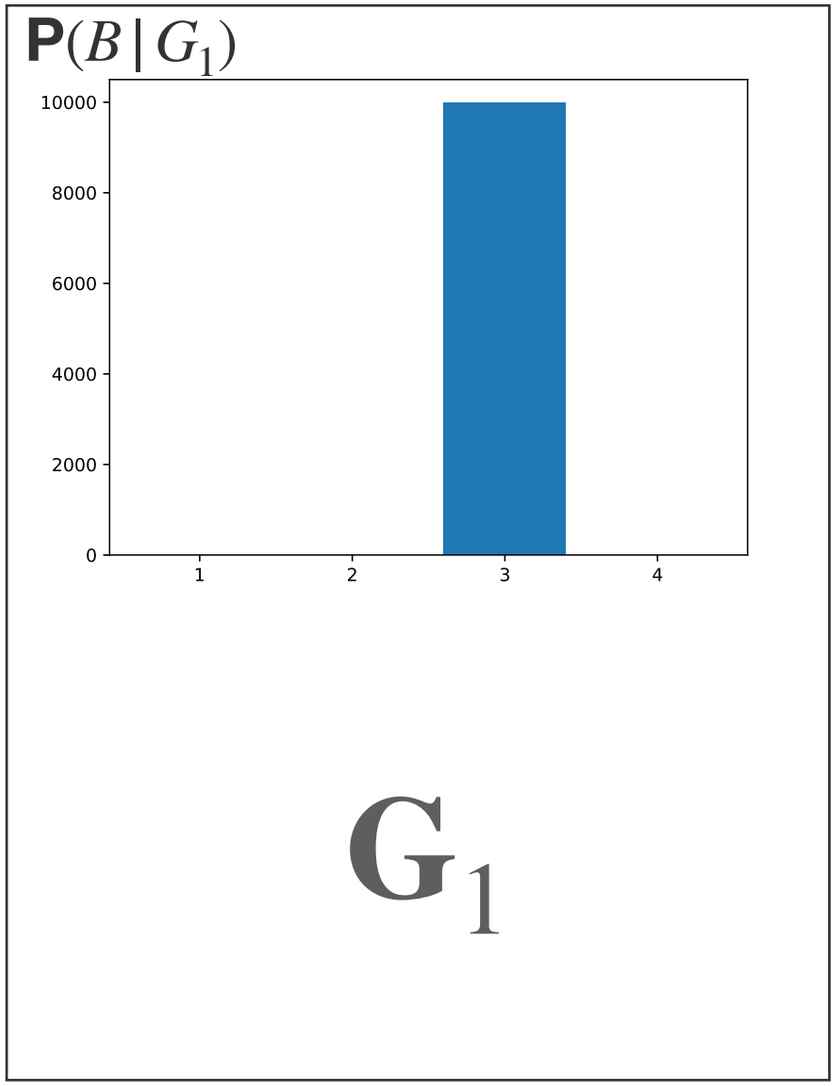


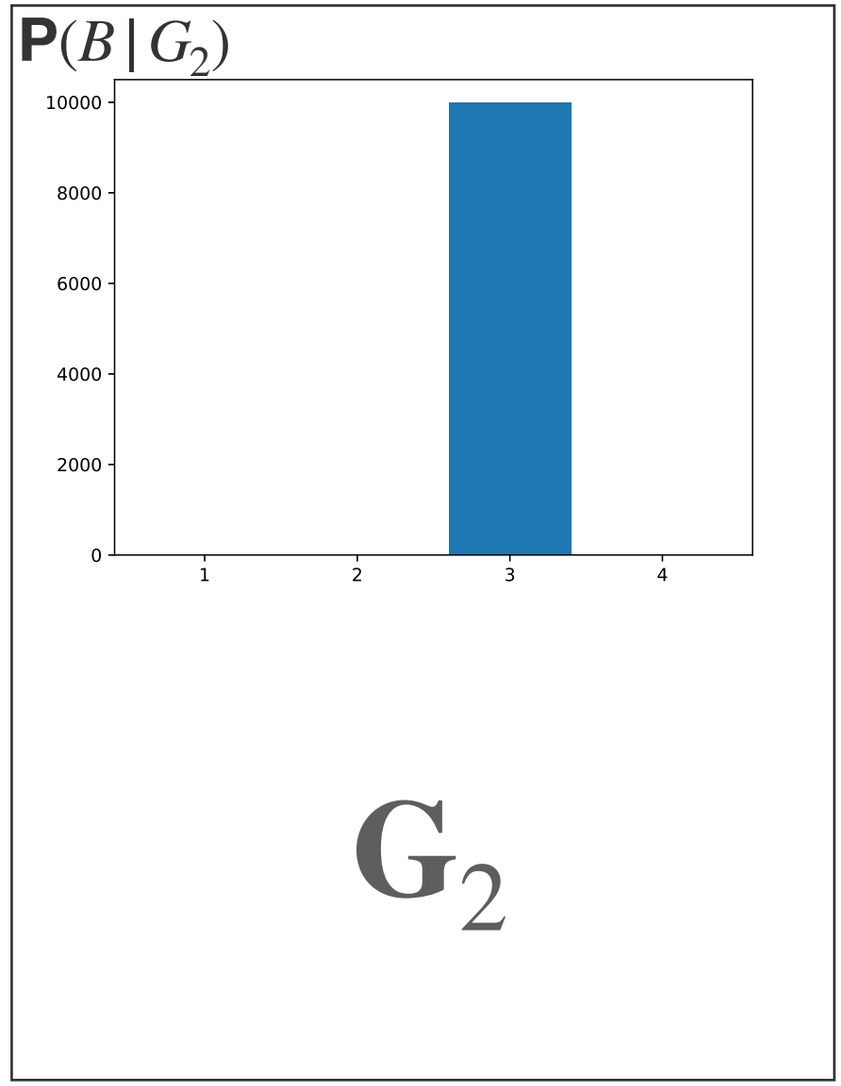


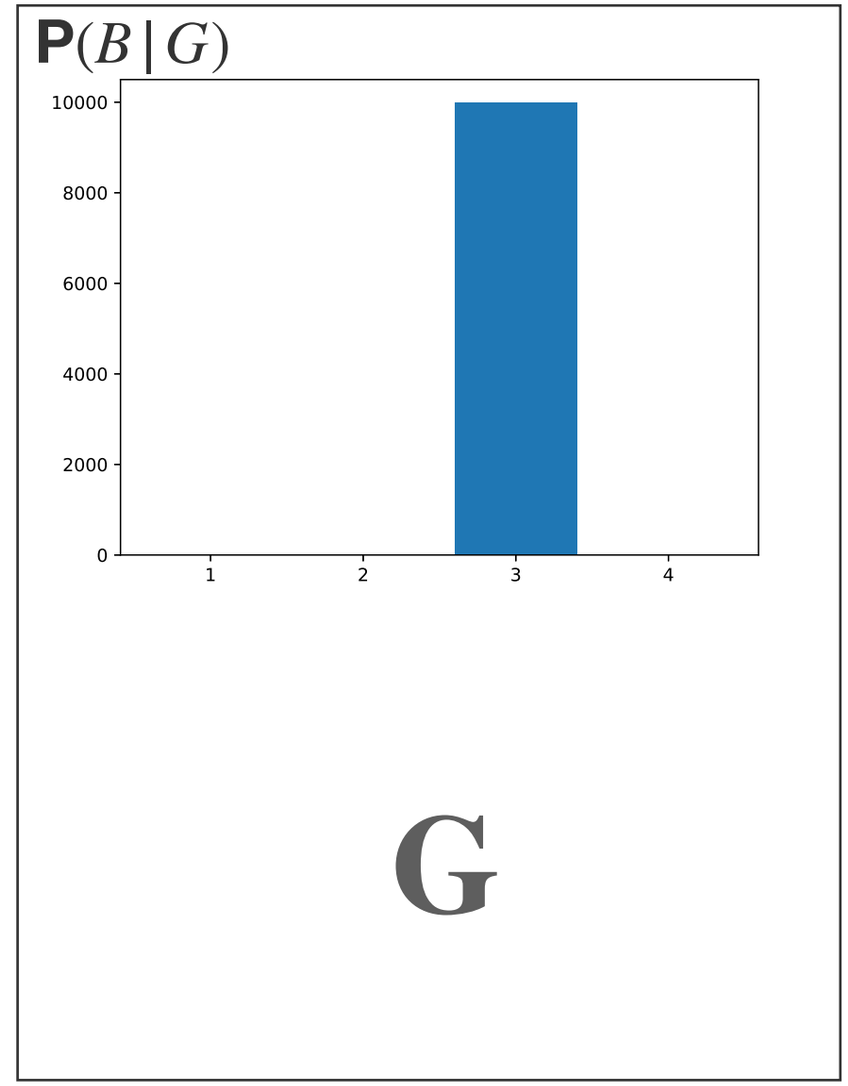
Setting 3

$$\mathbf{P}_{ii} = 0.50; \ \mathbf{P}_{ij} = 0.10$$

Estimating Model Dimension



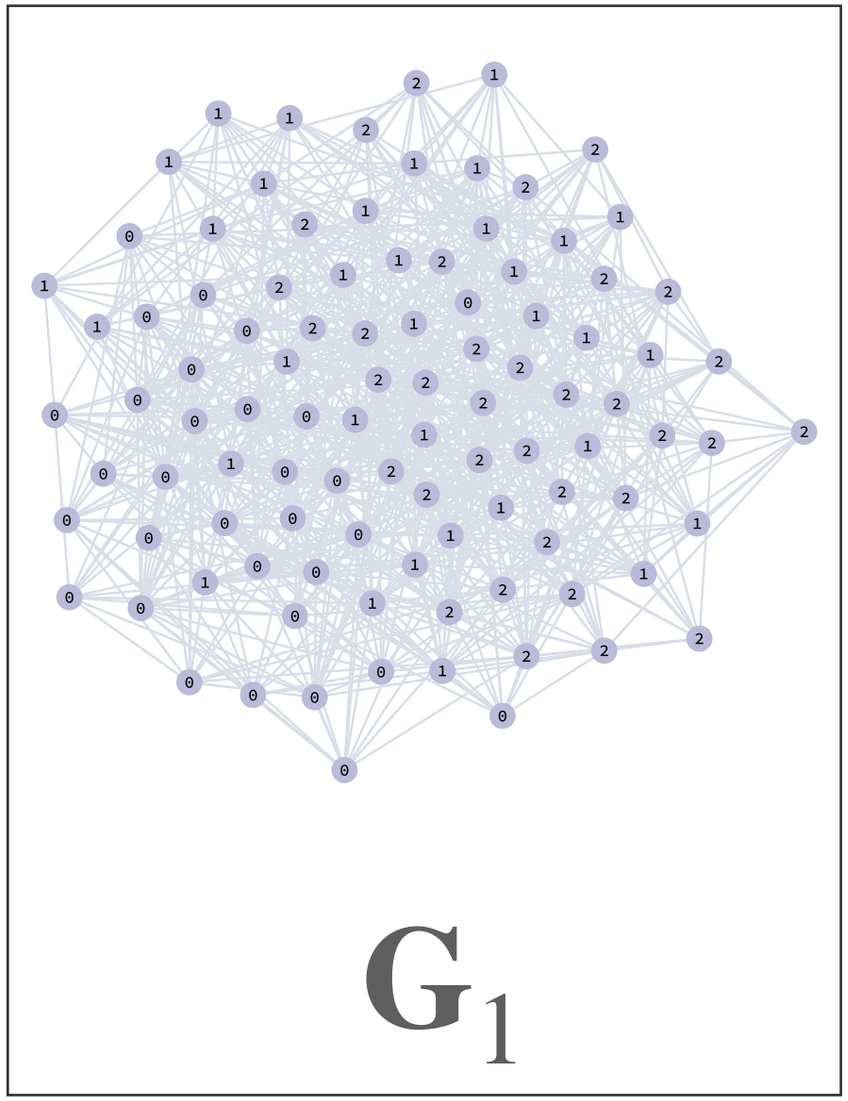


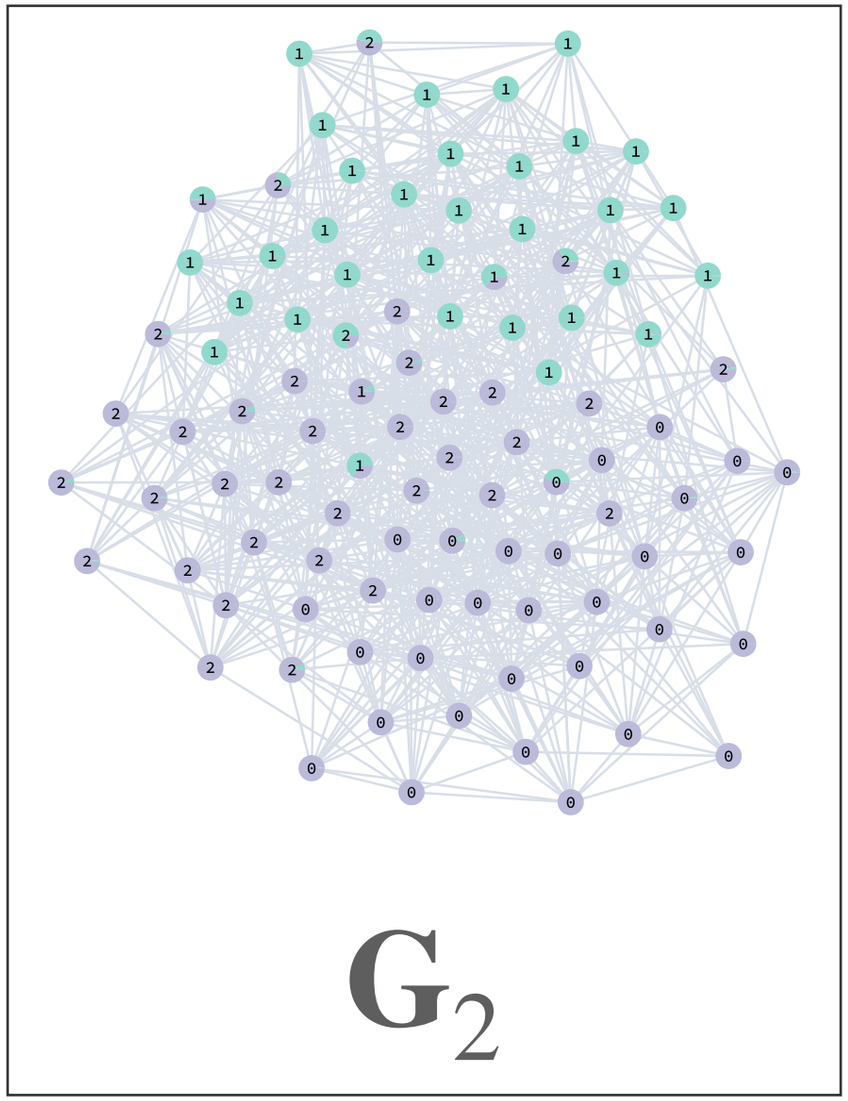


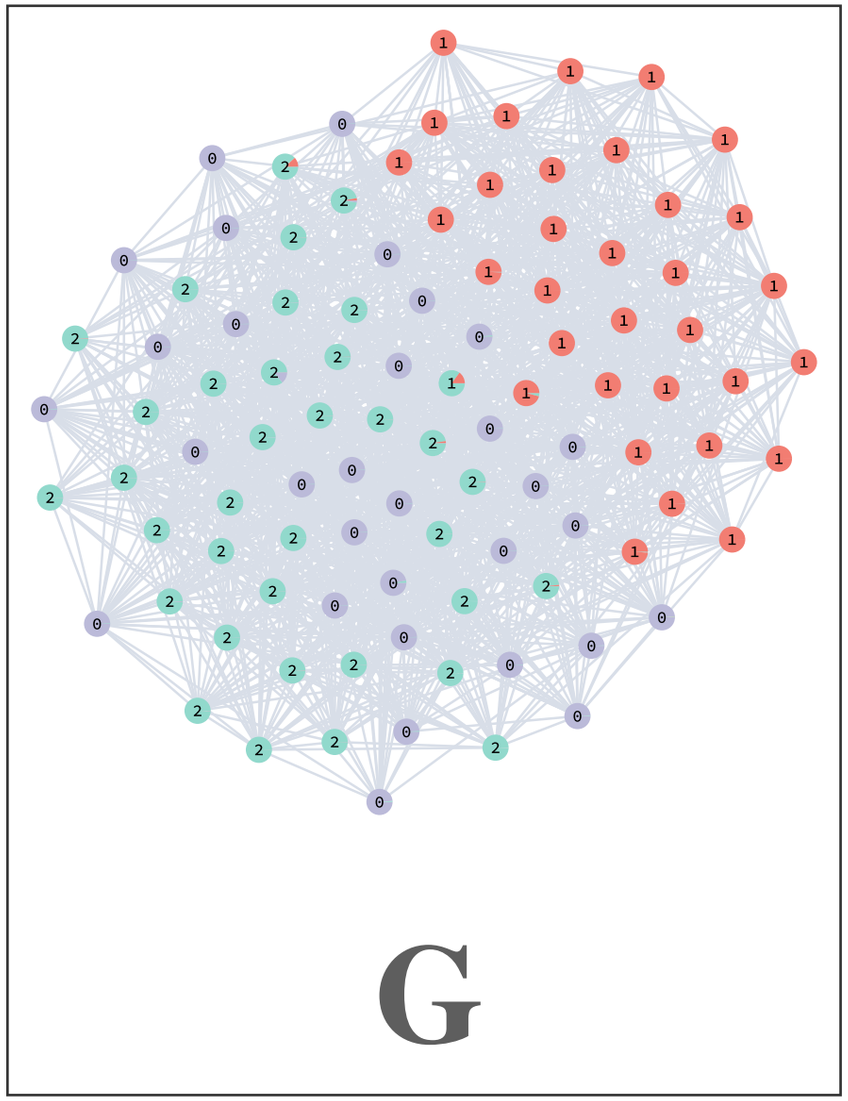
Setting 3

$$\mathbf{P}_{ii} = 0.50; \ \mathbf{P}_{ij} = 0.10$$

Classification Performance



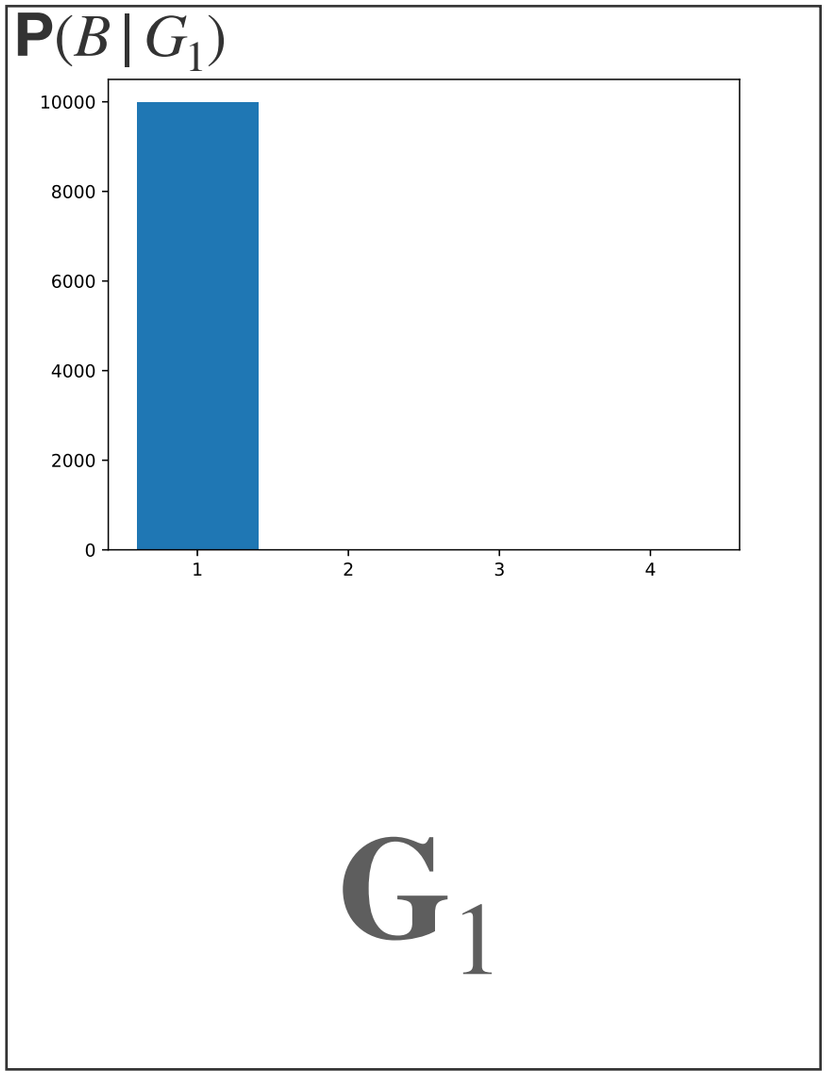


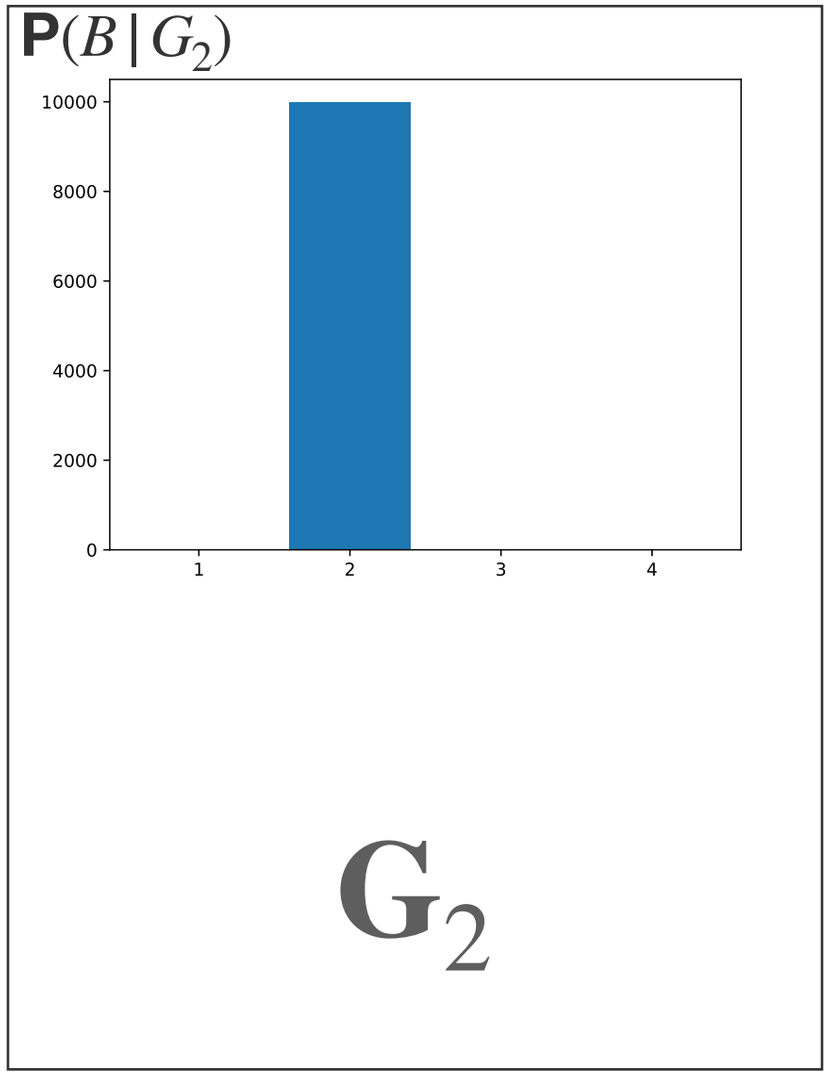


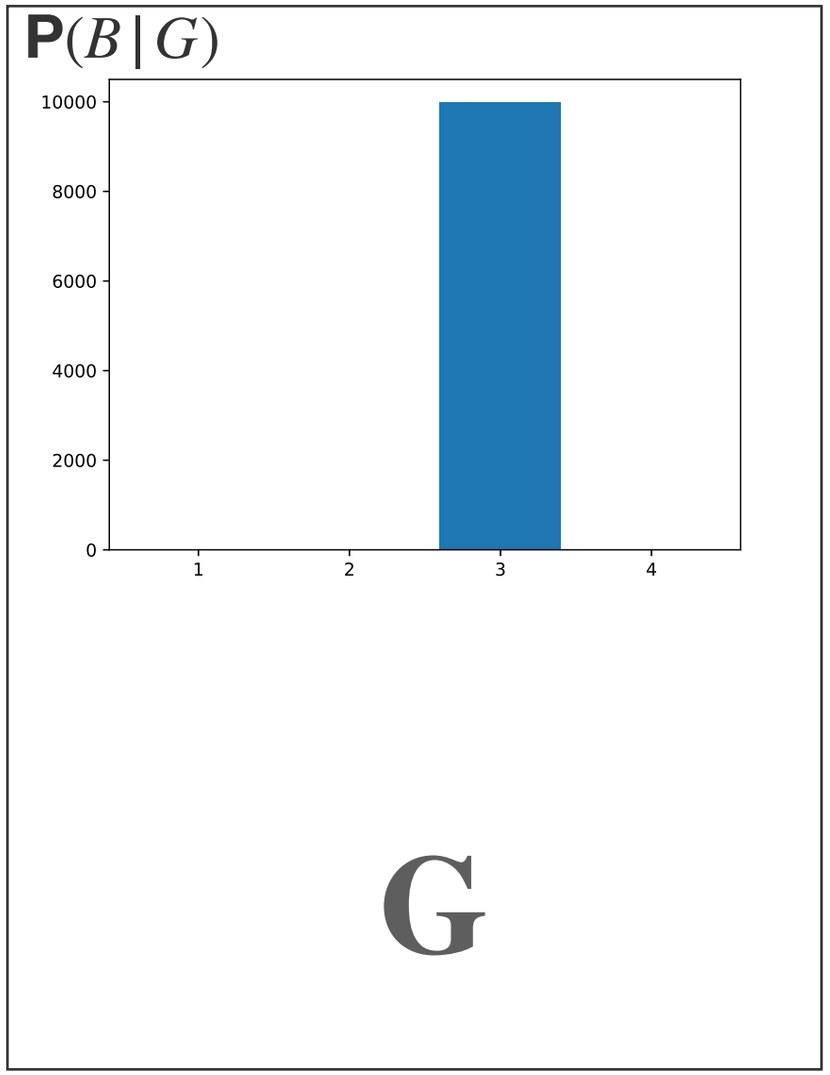
<u>Setting 4</u>

$$\mathbf{P}_{ii} = 0.30; \;\; \mathbf{P}_{ij} = 0.10$$

Classification Performance







<u>Setting 4</u>

$$\mathbf{P}_{ii} = 0.30; \;\; \mathbf{P}_{ij} = 0.10$$

Summary of Results

1. The edge union approach to data integration improves the ability of the SBM to perform community detection.

2. Improvements are most noticeable in dense networks with low signal to noise ratio or in sparse networks with high signal to noise ratio.

3. In dense networks with high signal to noise ratio, data integrated SBMs perform no worse than non-data integrated SBMs.

Other Approaches

We implemented a few other approaches to data integration, which performed surprisingly poorly.

Multigraph SBM: We form $\mathbf{G}$ by combining all edges in $\mathbf{G}_1$ and $\mathbf{G}_2$, allowing for multiple edges between any two nodes.

Weighted SBM: We construct $\mathbf{G}$ with Binomial edge weights corresponding to the number of times the edge appeared in $\mathbf{G}_1$ and $\mathbf{G}_2$.

Both approaches tend to overestimate B since they tend to cluster all nodes with multiple or weighted edges together.

Limitations and Future Work

1. Better quantify classification performance.

2. Relax the restrictions placed on the priors in the `graph-tool` package.

3. Assess performance on real data.

4. Apply to SSc data.

Thanks!