

## Topic 19: Community Detection

by Sai Zhang

**Key points:** .

**Disclaimer:** The note is built on Prof. *Jinchi Lv*'s lectures of the course at USC, DSO 607, High-Dimensional Statistics and Big Data Problems.

### 19.1 Stochastic Block Model

Consider an undirected graph  $G$ , with nodes  $V$  and edges  $E$ . Let

- $n$  be a positive integer: the number of **vertices**
- $k$  be a positive integer: the number of **communities**
- $p = (p_1, \dots, p_k)$  be a probability vector on  $\{1, \dots, k\} := [k]$ : the **prior** on the  $k$  communities
- $\mathbf{W}$  be a  $k \times k$  symmetric matrix with entries  $W_{ij} \in [0, 1]$ : the matrix of **connectivity probabilities**

then we have

#### Definition 19.1.1: Stochastic Block Model

The pair  $(\mathbf{X}, G)$  is drawn under  $SBM(n, p, \mathbf{W})$  if  $\mathbf{X}$  is an  $n$  dimensional random vector with i.i.d. components distributed under  $p$ , and  $G$  is an  $n$ -vertex simple graph where vertices  $i$  and  $j$  are connected with probability  $W_{X_i, X_j}$ , **independently** of other pairs of vertices. And the **community** sets can be defined by

$$\Omega_i = \Omega_i(\mathbf{X}) := \{v \in [n] : X_v = i\}, i \in [k]$$

Immediately, we can define the symmetry of SBM as:

#### Definition 19.1.2: Symmetric SBM

An SBM is called symmetric if

- $p$  is **uniform**
- $\mathbf{W}$  takes the same value **on the diagonal** and the same value **off the diagonal**

$(\mathbf{X}, G)$  is drawn under  $SSBM(n, k, A, B)$  if  $p = \{1/k\}^k$  and  $\mathbf{W}$  takes value  $A$  on the diagonal and  $B$  off the diagonal.

#### 19.1.1 Recovery

The goal of community detection is to recover the labels  $\mathbf{X}$  by observing  $G$ , up to some level of accuracy. First, define **agreement** as

**Definition 19.1.3: Agreement of Communities**

The agreement between two community vectors  $\mathbf{x}, \mathbf{y} \in [k]^n$  is obtained by maximizing the common components between  $\mathbf{x}$  and any relabelling of  $\mathbf{y}$ , that is

$$A(\mathbf{x}, \mathbf{y}) = \max_{\pi \in S_k} \frac{1}{n} \sum_{i=1}^n \mathbf{1}[x_i = \pi(y_i)]$$

where  $S_k$  is the group of permutations on  $[k]$ .

The **relabelling** permutation is used to handle symmetric communities such as in SSBM, as it is impossible to recover the actual labels in this case. But it's possible to recover the **partition**. There are 2 types of partition recovery we consider

**Exact Recovery** First, consider the case of **exact recovery**:

**Definition 19.1.4: Exact Recovery**

Let  $(\mathbf{X}, G) \sim \text{SBM}(n, p, W)$ , the exact recovery is solved if there exists an algorithm that takes  $G$  as an input and outputs  $\hat{\mathbf{X}} = \hat{\mathbf{X}}(G)$  such that  $\mathbb{P}\{A(\mathbf{X}, \hat{\mathbf{X}}) = 1\} = 1 - o_p(1)$

In the SSBM case, algorithms that guarantee

$$A(\mathbf{X}, \hat{\mathbf{X}}) \rightarrow \frac{1}{k}$$

would be trivial.

**Weak Recovery** On the other hand, we the case of **weak recovery** defined as

**Definition 19.1.5: Weak Recovery**

Weak recovery or detection is solved  $\text{SSBM}(n, k, A, B)$  if for  $(\mathbf{X}, G) \sim \text{SSBM}(n, k, A, B)$ , then  $\exists \epsilon > 0$  and an algorithm that takes  $G$  as an input and outputs  $\hat{\mathbf{X}}$  such that

$$\mathbb{P}\left\{A(\mathbf{X}, \hat{\mathbf{X}}) \geq \frac{1}{k} + \epsilon\right\} = 1 - o(1)$$

**19.1.2 Example: SSBM(n,2)**

Let's look at the example of  $\text{SSBM}(n, 2, \alpha \frac{\log n}{n}, \beta \frac{\log n}{n})$ , where

- $n$ : number of vertices (assumed to be even for simplicity)
- for each  $v \in [n]$ , a binary label  $X_v$  is attached s.t.

$$|\{v \in [n] : X_v = 1\}| = n/2$$

- for each pair of distinct nodes  $u, v \in [n]$ , an edge is placed with probability

- $\alpha \frac{\log n}{n}$  if  $X_u = X_v$
- $\beta \frac{\log n}{n}$  if  $X_u \neq X_v$

where edges are placed independently conditionally on the vertex labels

- WLOG,  $\alpha > \beta$

then we have the following theorem

**Theorem 19.1.6: Exact Recovery in  $SSBM(n, 2, \alpha \log(n)/n, \beta \log(n)/n)$**