Generalized SBM Algorithm

Jiaxin Hu

05/20/2020

1 GENERALIZED SBM ALGORITHM

Now we generalize the refinement algorithm in Gao's paper to the asymmetric and high-order cases. For simplicity, let f denotes the function to calculate the penalty parameter, where

$$f(\hat{a}_u, \hat{b}_u) = -\frac{1}{2t_u}\log\left(\frac{\frac{\widehat{a}_u}{n}e^{-t_u}+1-\frac{\widehat{a}_u}{n}}{\frac{\hat{b}_u}{n}e^{t_u}+1-\frac{\widehat{b}_u}{n}}\right), \quad t_u(\hat{a}_u, \hat{b}_u) = \frac{1}{2}\log\frac{\widehat{a}_u\left(1-\widehat{b}_u/n\right)}{\hat{b}_u\left(1-\widehat{a}_u/n\right)}.$$

Let A_{-u} denotes the submatrix after removing the u-th row and column in A and A_o denotes the matrix after removing the diagonal elements in A. Algorithm 1 shows the original refinement algorithm.

Algorithm 1 Refinement for symmetric connection

```
Input: Adjacency matrix $A \in \{0,1\}^{n\times n}$, # of communities $K$, initialization algorithm $\sigma^0$.
Output: Community assignment $\hat{\sigma}$.
for $u = 1$ to $n$ do
    Apply $\sigma^0$ to $A_{-u}$ and obtain $\hat{\sigma}_u(v) = \sigma_u^0(v), \forall v \neq u$ and let $\sigma_u^0(u) = 0$.
    Define $\tilde{\mathcal{C}}_i^u = \{v : \hat{\sigma}_u(v) = i\}$. Estimate the connection:
        $$\widehat{B}_{ii}^u = \frac{\sum A_o[\tilde{\mathcal{C}}_i^u, \tilde{\mathcal{C}}_i^u]}{|\tilde{\mathcal{C}}_i^u|(|\tilde{\mathcal{C}}_i^u|-1)}, \quad \widehat{B}_{ii}^u = \frac{\sum A[\tilde{\mathcal{C}}_i^u, \tilde{\mathcal{C}}_j^u]}{|\tilde{\mathcal{C}}_i^u||\tilde{\mathcal{C}}_j^u|}, \quad \forall i \neq j \in [K].$$
    Define $\hat{a}_u = \min_i B_{ii}$ and $\hat{b}_u = \max_{i\neq j} B_{ij}$. Calculate the penalty parameter by $\rho_u = f(\hat{a}_u, \hat{b}_u)$.
    Obtain the partition estimate of $u$:
        $$\hat{\sigma}_u(u) = \arg\max_{k_0\in[K]} \sum_{\hat{\sigma}_u(v)=k_0} A_{uv} - \rho_u|\tilde{\mathcal{C}}_{k_0}^u|.$$
end for
Consensus: Define $\hat{\sigma}(1) = \hat{\sigma}_1(1)$, for $u = 2, ..., n$, define
    $$\widehat{\sigma}(u) = \operatorname*{argmax}_{l\in[K]} \left|\{v : \widehat{\sigma}_1(v) = l\} \cap \{v : \widehat{\sigma}_u(v) = \widehat{\sigma}_u(u)\}\right|.$$
```

1.1 ***Asymmetric case***

Next, I consider the asymmetric case. Suppose we have the network matrix $A \in \{0,1\}^{n\times m}$ and the number of communities on first and second mode are K_1 and K_2 respectively. We use TBM for order-2 tensor in Wang's paper as the initialization algorithm σ^0.

We can consider the subtraction $a_u - b_u$ as the minimal gap in this model. If $a_u = b_u$ and $B_{ii} = B_{jl}$, the connection within cluster i is equal to the connection between cluster j and cluster l, which implies cluster j and cluster l should be merged into one cluster. Therefore, $a_u - b_u$ is the minimal gap with k different clusters. In the asymmetric case, the minimal gaps for n and m nodes are defined as

$$\delta^{(1)} = \min_{k_1 \neq k_1' \in [K_1]} \max_{k_2 \in [K_2]} \delta_1(k_1, k_1', k_2) = \min_{k_1 \neq k_1' \in [K_1]} \max_{k_2 \in [K_2]} (B_{k_1,k_2} - B_{k_1',k_2})^2$$

$$\delta^{(2)} = \min_{k_2 \neq k_2' \in [K_2]} \max_{k_1 \in [K_1]} \delta_2(k_2, k_2', k_1) = \min_{k_2 \neq k_2' \in [K_2]} \max_{k_1 \in [K_1]} (B_{k_1,k_2} - B_{k_1,k_2'})^2.$$

Let $(\hat{k}_1, \hat{k}_1', \hat{k}_2) = \arg\min\max \delta_1(k_1, k_1', k_2)$ and suppose $B_{\hat{k}_1,\hat{k}_2} > B_{\hat{k}_1',\hat{k}_2}$. Define $a_u^{(1)} = B_{\hat{k}_1,\hat{k}_2}, b_u^{(1)} = B_{\hat{k}_1',\hat{k}_2}$. If $a_u^{(1)} = b_u^{(1)}$, the connection matrix B is not irreducible and cluster $\hat{k}_1, \hat{k}_1'$ should be merged. Similarly, let $(\tilde{k}_2, \tilde{k}_2', \tilde{k}_1) = \arg\min\max \delta_2(k_2, k_2', k_1)$ and $a_u^{(2)} = B_{\tilde{k}_1,\tilde{k}_2}, b_u^{(2)} = B_{\tilde{k}_1,\tilde{k}_2'}$. Then we can use the estimated $(a_u^{(1)}, b_u^{(1)})$, $(a_u^{(2)}, b_u^{(2)})$ to calculate the penalty parameter $(\rho_u^{(1)}, \rho_u^{(2)})$ for each mode.

The function f may be different in asymmetric case, however, f is too complicate to modify explicitly at this time. Although, f should still be an increasing function along with a_u when b_u is fixed. In the following discussion, we still use f as the penalty function.

Let $A_{-i,-j}$ denotes the submatrix after removing the i-th row and j-th column in A and $A[C_i, C_j] = \sum_{p \in C_i, q \in C_j} A_{pq}$. Algorithm 2 shows the generalized refinement algorithm for asymmetric case.

In step 5, we estimate the row cluster and column cluster for one node separately. Here we can also estimate the row and column cluster simultaneously with following step:

$$(\hat{\sigma}_u^{(1)}(i), \hat{\sigma}_u^{(2)}(j)) = \arg\max_{k_0 \in [K_1], k_0' \in [K_2]} \sum_{\hat{\sigma}_u^{(1)}(v_1) = k_0, \hat{\sigma}_u^{(2)}(v_2) = k_0'} \{A_{v_1 j} + A_{i v_2}\} - \rho_u |\tilde{\mathcal{C}}_{k_0}^{u,1}||\tilde{\mathcal{C}}_{k_0'}^{u,2}|,$$

where $\rho_u = \min\{\rho_u^{(1)}, \rho_u^{(2)}\}$. The result may be different with step 5 because the penalty terms are distinctive. The other steps should maintain the same.

Another generalization idea is to do clustering on each mode separately. Algorithm 2 assigns nm nodes to $K_1 K_2$ blocks while clustering separately aims to assign n nodes to K_1 clusters and m nodes to K_2 clusters respectively. We can use the symmetric refinement algorithm on each mode. Algorithm 3 shows the simplified procedures.

Algorithm 3 may be faster than Algorithm 2, however, it doesn't utilize the block structure. And the transformation from original network data to two adjacency matrices may lose some information. Therefore, I think Algorithm 2 is more efficient.

1.2 ***Tensor case***

My generalization for tensor case is similar with Algorithm 2 for asymmetric case. We use M to denote the tensor order, since K has already represented the number of clusters. Let $d_1, ..., d_M$ denote the dimensions, $K_1, ..., K_M$ denote the numbers of clusters. We also use $TBM(order - M)$ for initialization. Algorithm 4 shows the procedures, in which we omit the consensus steps.

Algorithm 2 Refinement for asymmetric connection

```
Input: Network matrix $A \in \{0,1\}^{n\times m}$, # of communities on two modes $(K_1, K_2)$, initialization algorithm $\sigma^0 = TBM(order-2)$.
Output: Community assignment on two modes $(\hat\sigma^{(1)}, \hat\sigma^{(2)})$.
for $u = (i,j), i \in [n], j \in [m]$ do
    Apply $\sigma^0$ to $A_{-i,-j}$ and obtain $\hat\sigma^{(1)}_u(v_1) = \sigma^{0,(1)}_u(v_1), \hat\sigma^{(2)}_u(v_2) = \sigma^{0,(2)}_u(v_2), \forall v_1 \neq i, v_2 \neq j$ and let $\sigma^{(1)}_u(i) = 0,\ \sigma^{(2)}_u(j) = 0$.
    Define $\tilde{\mathcal{C}}^{u,1}_{k_1} = \{v : \hat\sigma^{(1)}_u(v) = k_1\}, \forall k_1 \in [K_1]$ and $\tilde{\mathcal{C}}^{u,2}_{k_2} = \{v : \hat\sigma^{(1)}_u(v) = k_2\}, \forall k_2 \in [K_2]$. Estimate the connection:
        $$\widehat{B}^u_{k_1,k_2} = \frac{\sum A[\tilde{\mathcal{C}}^{u,1}_{k_1}, \tilde{\mathcal{C}}^{u,2}_{k_2}]}{|\tilde{\mathcal{C}}^{u,1}_{k_1}||\tilde{\mathcal{C}}^{u,2}_{k_2}|}$$
    Let $(\hat k_1, \hat k_1', \hat k_2) = \arg\min\max \delta_1(k_1, k_1', k_2), (\tilde k_2, \tilde k_2', \tilde k_1) = \arg\min\max \delta_2(k_2, k_2', k_1)$. Define $\hat a^{(1)}_u = B_{\hat k_1,\hat k_2}, \hat b^{(1)}_u = B_{\hat k_1',\hat k_2}$ and $\hat a^{(2)}_u = B_{\tilde k_1,\tilde k_2}, \hat b^{(2)}_u = B_{\tilde k_1,\tilde k_2'}$. Calculate the penalty parameter by $\rho^{(1)}_u = f(\hat a^{(1)}_u, \hat b^{(1)}_u)$ and $\rho^{(2)}_u = f(\hat a^{(2)}_u, \hat b^{(2)}_u)$.
    Obtain the partition estimate of $u = (i,j)$:
        $$\hat\sigma^{(1)}_u(i) = \arg\max_{k_0\in[K_1]} \sum_{\hat\sigma^{(1)}_u(v)=k_0} A_{vj} - \rho^{(1)}_u|\tilde{\mathcal{C}}^{u,1}_{k_0}|;$$
        $$\hat\sigma^{(2)}_u(j) = \arg\max_{k_0'\in[K_2]} \sum_{\hat\sigma^{(2)}_u(v)=k_0'} A_{iv} - \rho^{(2)}_u|\tilde{\mathcal{C}}^{u,2}_{k_0'}|$$
end for
Consensus: Define
        $$\hat\sigma^{(1)}_j(i) = \arg\max_{k_1\in[K_1]} |\{v : \hat\sigma^{(1)}_{(i,1)}(v) = k_1\} \bigcap \{v : \hat\sigma^{(1)}_{(i,j)}(v) = \hat\sigma^{(1)}_{(i,j)}(i)\}|$$
        $$\hat\sigma^{(2)}_i(j) = \arg\max_{k_2\in[K_2]} |\{v : \hat\sigma^{(2)}_{(1,j)}(v) = k_2\} \bigcap \{v : \hat\sigma^{(2)}_{(i,j)}(v) = \hat\sigma^{(2)}_{(i,j)}(j)\}|$$
    and
        $$\tilde\sigma^{(1)}(i) = Mode(\hat\sigma^{(1)}_1(i), \cdots, \hat\sigma^{(1)}_m(i)); \quad \tilde\sigma^{(2)}(j) = Mode(\hat\sigma^{(2)}_1(j), \cdots, \hat\sigma^{(2)}_n(j)).$$
    Let $\hat\sigma^{(1)}(1) = \tilde\sigma^{(1)}(1)$ and $\hat\sigma^{(2)}(1) = \tilde\sigma^{(2)}(1)$, for $i = 2, ..., n$ and $j = 2, ..., m$:
        $$\hat\sigma^{(1)}(i) = \arg\max_{k_1\in[K_1]} |\{v : \hat\sigma^{(1)}_{(1,1)}(v) = k_1\} \bigcap \{v : \hat\sigma^{(1)}_{(i,1)}(v) = \tilde\sigma^{(1)}(i)\}|$$
        $$\hat\sigma^{(2)}(j) = \arg\max_{k_2\in[K_2]} |\{v : \hat\sigma^{(2)}_{(1,1)}(v) = k_2\} \bigcap \{v : \hat\sigma^{(2)}_{(1,j)}(v) = \tilde\sigma^{(2)}(j)\}|.$$
```

Algorithm 3 Refinement for asymmetric connection separately

```
Input: Network matrix $A \in \{0,1\}^{n\times m}$, # of communities on two modes $(K_1, K_2)$, initialization algorithm $\sigma^0$.
Output: Community assignment on two modes $(\hat{\sigma}^{(1)}, \hat{\sigma}^{(2)})$.
Obtain the correlation matrix for $A$'s rows and columns, where $Z_1 = [\![cor(A_{i.}, A_{j.})]\!], i,j \in [n]$ and $Z_2 = [\![cor(A_{.p}, A_{.q})]\!], p,q \in [m]$.
Calculate the adjacency matrix $Z_1'$ for $A$'s rows:
    $$Z_1'[i,j] = \begin{cases} 0, & Z_1[i,j] < 0.5 \\ 1, & Z_1[i,j] > 0.5 \\ Ber(1/2), & Z_1[i,j] = 0.5 \end{cases}$$
Calculate the adjacency matrix $Z_2'$ for $A$'s columns as $Z_1'$ dose.
Apply Algorithm 1 to $Z_1'$ and $Z_2'$ and get the assignment $(\hat{\sigma}^{(1)}, \hat{\sigma}^{(2)})$.
```

Algorithm 4 Refinement for tensor connection

```
Input: Network matrix $A \in \{0,1\}^{d_1\times\cdots d_M}$, # of communities on two modes $(K_1, \cdots K_M)$, initialization algorithm $\sigma^0 = TBM(order-M)$.
Output: Community assignment on $M$ modes $(\hat{\sigma}^{(1)}, \cdots, \hat{\sigma}^{(M)})$.
for $u = (i_1, \cdots, i_M), i_m \in [d_m], \forall m \in [M]$ do
    Apply $\sigma^0$ to $A_{-i_1,\cdots,-i_M}$ and obtain $\hat{\sigma}_u^{(1)}(v_1), \cdots, \hat{\sigma}_u^{(M)}(v_M), \forall v_m \neq i_m$ and let $\hat{\sigma}_u^{(m)}(i_m) = 0, \forall m \in [M]$.
    Define $\tilde{\mathcal{C}}_{k_1}^{u,m} = \{v : \hat{\sigma}_u^{(m)}(v) = k_1\}, \forall m \in [M]$. Estimate the connection:
        $$\widehat{B}_{k_1,\cdots,k_M}^{u} = \frac{\sum A[\tilde{\mathcal{C}}_{k_1}^{u,1}, \cdots, \tilde{\mathcal{C}}_{k_M}^{u,M}]}{|\tilde{\mathcal{C}}_{k_1}^{u,1}| \cdots |\tilde{\mathcal{C}}_{k_M}^{u,M}|}$$
    For every mode $m \in [M]$, let $(\hat{k}_m, \hat{k}_m', \hat{k}_1, \cdots, \hat{k}_{m-1}, \hat{k}_{m+1}, \cdots, \hat{k}_M) = \arg\min\max \delta_m$. Define $\hat{a}_u^{(m)} = B_{\hat{k}_1,\cdots,\hat{k}_m,\cdots\hat{k}_M}$ and $b_u^{(m)} = B_{\hat{k}_1,\cdots,\hat{k}_m',\cdots,\hat{k}_2}$. Calculate $\rho_u^{(m)} = f(\hat{a}_u^{(m)}, \hat{b}_u^{(m)}), \forall m \in [M]$.
    Obtain the partition estimate of $u = (i_1, \cdots, i_M)$:
        $$\hat{\sigma}_u^{(1)}(i) = \arg\max_{k_1 \in [K_1]} \sum_{\hat{\sigma}_u^{(1)}(v) = k_0} A_{vi_2,\cdots,i_M} - \rho_u^{(1)} |\tilde{\mathcal{C}}_{k_1}^{u,1}|;$$
        $$\cdots$$
        $$\hat{\sigma}_u^{(M)}(i_M) = \arg\max_{k_M \in [K_M]} \sum_{\hat{\sigma}_u^{(M)}(v) = k_0} A_{i_1,\cdots,i_{M-1},v} - \rho_u^{(M)} |\tilde{\mathcal{C}}_{k_0}^{u,M}|$$
end for
Consensus
```

2 COMPARISON BETWEEN TBM & SBM ALGORITHM

Here we apply order-2 TBM algorithm to the symmetric connection network data and compare the MCR with SBM Algorithm. Figure 1 shows the MCR comparison.

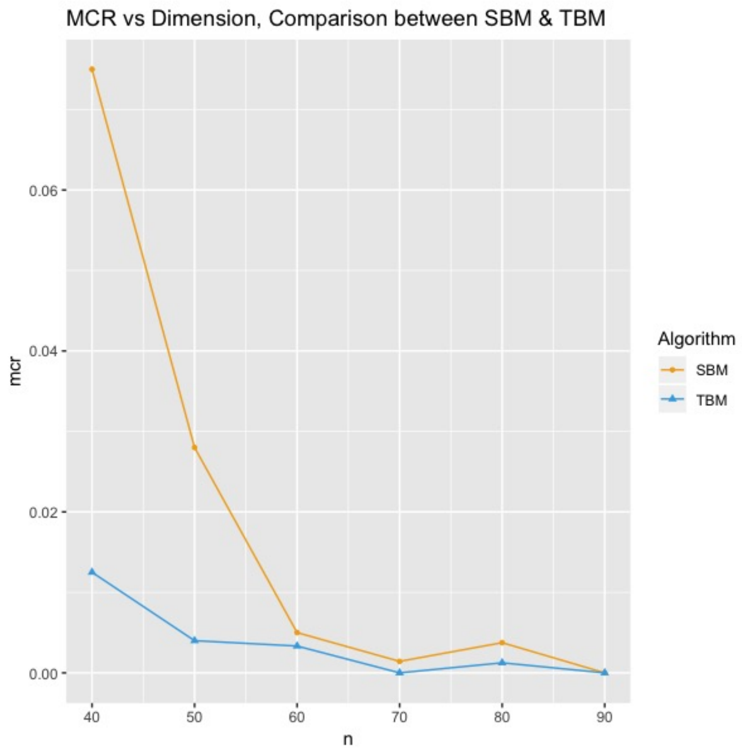

Figure 1. MCR change along with dimension growth. Each result is the mean value of 5 duplicate simulations.

According to the graph, the MCR of SBM decreases more rapid than the MCR of TBM. That consists with the theoretical misclassification rates, where $MCR_{SBM} \leq Cexp(-n)$ and $MCR_{TBM} \leq C'n^{-1/2}$.

3 ASYMMETRIC ESTIMATE IN TBM

TBM gives asymmetric partitions even if the true partitions on second and third modes are the same. I change the input the observation to a symmetric tensor to test whether TBM would give symmetric results. Figure 2 and Figure 3 show the results for asymmetric input and symmetric input.

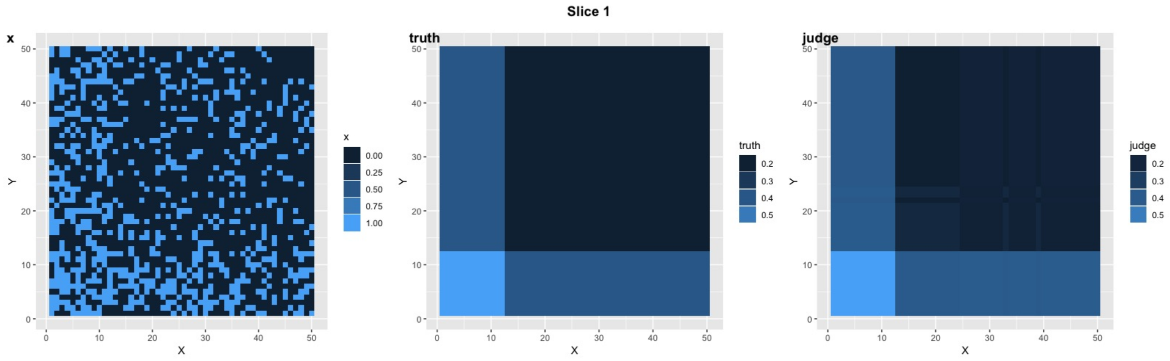

Figure 2. First slice from mode 1 of asymmetric observation $\mathcal{X}[3,,]$, true connectivity $B[3,,]$, estimated connectivity $\hat{B}[3,,]$.

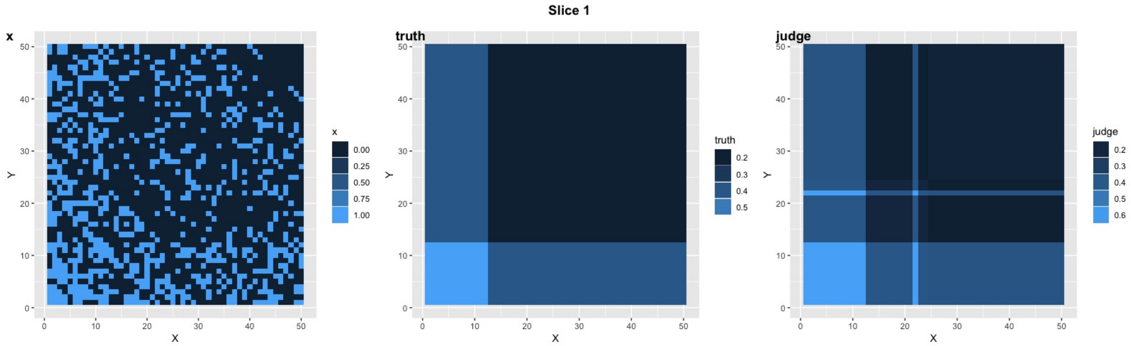

Figure 3. First slice from mode 1 of symmetric observation $\mathcal{X}[3,,]$, true connectivity $B[3,,]$, estimated connectivity $\hat{B}[3,,]$.

Therefore, TBM will give symmetric results if the input observation is strictly symmetric. The reason why TBM gives asymmetric estimates is the asymmetric input.