Algorithm for hDCBM

Jiaxin Hu

July 19, 2021

1 Uniform framework

To analyze the models in (Gao et al., 2018), (Ke et al., 2019), (Han et al., 2020), we state an uniform higher-order degree-corrected block model(hDCBM). Consider the observation $\mathcal{Y} \in \mathbb{R}^{p_1 \times \cdot \times p_d}$. Let $r_k, k \in [d]$ denote the number clusters in the k-th mode, $z_k \in [r_k]^{p_k}$ denote the corresponding clustering assignment, $\theta_k = (\theta_{k,1}, ..., \theta_{k,p_k}) \in \mathbb{R}^{p_k}$ denote the degree-corrected parameters on k-th mode, where $\theta_{kj} > 0, j \in [p_k], k \in [d]$. We have following hDCBM model

$$\mathcal{Y} = \mathcal{S} \times_1 \Theta_1 \boldsymbol{M}_1 \times_2 \cdots \times_d \Theta_d \boldsymbol{M}_d + \mathcal{E},$$

where $\mathcal{S} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ is the core tensor whose entries refer to the interactive connections among the group, $\Theta_k = \operatorname{diag}(\theta_k) \in \mathbb{R}^{p_k \times p_k}$ is a diagonal matrix generated by θ_k, $\boldsymbol{M} \in \mathbb{R}^{p_k \times r_k}$ is the membership matrix generated by z_k, and $\mathcal{E}$ has independent sub-Gaussian noise entries. Let $\mathcal{X} = \mathcal{S} \times_1 \Theta_1 \boldsymbol{M}_1 \times_2 \cdots \times_d \Theta_d \boldsymbol{M}_d$ denote the true signal.

1. Biclustering model in (Gao et al., 2018): let $\mathcal{Y}$ is symmetric and follows Bernoulli distribution, $d = 2$, and $\Theta_1 = \Theta_2, \boldsymbol{M}_1 = \boldsymbol{M}_2$.
2. Tensor-SCORE in (Ke et al., 2019): let $\mathcal{Y}$ is symmetric and follows Bernoulli distribution, and $\Theta_1 = \cdots = \Theta_d = \Theta, \boldsymbol{M}_1 = \cdots = \boldsymbol{M}_d = \boldsymbol{M}$.
3. TBM in (Han et al., 2020): let $\mathcal{Y}$ follows Gaussian distribution, and $\Theta_k = \mathbf{1}_{p_k}$.

2 Algorithm comparison

Table 1 summarizes the algorithm comparison of these three methods. Here are few points need to be noticed:

1. In degree-corrected models, Biclustering and Tensor-SCORE, the algorithms use SCORE normalization to avoid estimating the degree-corrected parameters θ. Similar techniques may be useful for other hDCBM algorithms.
2. For block models (hard membership models), spectral methods do not lead to optimal classification rate. It is consistent to the theoretical results: 1) In Biclustering, without the last

refinement step, the loss of $\tilde{z}$ decays polynomially (Lemma 1); 2) tensor-SCORE, a spectral-based method, indicates a polynomial error (Theorem 1); 3) in TBM, the HSC initialization $\hat{z}^{(0)}$ indicates a polynomial error (Theorem 2).

3. The refinement step in Biclustering and the HLloyd in TBM improve the accuracy from polynomial to exponential. Both of these procedures take the advantages of the hard membership structure. I believe that may be the key to achieve the optimal rate.

Hence, a possible optimal algorithm for hDCBM should: 1) use SCORE normalization techniques to avoid the estimation of θ; 2) use hard membership structure in iteration or refinement.

Therefore, I would like to extend the TBM algorithm to a degree-corrected version. We consider the Gaussian model at this point. The detailed procedures are in Algorithm 1 and Algorithm 2. The SCORE normalization function $h(x)$ satisfies $h(ax) = ah(x)$. We have multiple choices of h, such as $\|\cdot\|_{\max}, \|\cdot\|_1, \|\cdot\|_F$.

In Algorithm 1, we use SCORE normalization to allow the marginal k-means on each mode, which is natural.

In Algorithm 2, note that the estimation of $\mathcal{S}$ is related to the clustering on all modes. Thus, we need to remove the heterogeneity on all modes, which is highlighted in blue. To illustrate why the blue part works, we consider the noiseless tensor $\mathcal{X} = \mathcal{S} \times_1 \Theta_1 \boldsymbol{M}_1 \times_2 \cdots \times_d \Theta_d \boldsymbol{M}_d$. Suppose $k = 1$ and $h(x) = \|x\|_F$. We have

$$\boldsymbol{X}_1 = \Theta_1 \boldsymbol{M}_1 \mathcal{M}_1(\mathcal{S}) \left(\Theta_2 \boldsymbol{M}_2 \otimes \cdots \otimes \Theta_d \boldsymbol{M}_d\right)^T,$$

which implies for $j \in [p_1]$ such that $(z_1)_j = u, u \in [r_1]$

$$\boldsymbol{X}_{1j} = \theta_j v_u = \tilde{\theta}_j b_u,$$

where $v_u = \left[\mathcal{M}_1(\mathcal{S}) \left(\Theta_2 \boldsymbol{M}_2 \otimes \cdots \otimes \Theta_d \boldsymbol{M}_d\right)^T\right]_u$ and $b_u = v_u / \|v_u\|$. Let $\boldsymbol{X}^s_{1j} = \boldsymbol{X}^s_{1j}/h(\boldsymbol{X}^s_{1j})$, $\boldsymbol{V}_1 = \operatorname{diag}(\|v_1\|, ..., \|v_{r_1}\|)$, and $\mathcal{M}_1(\tilde{\mathcal{S}}) = \boldsymbol{V}_1 \mathcal{M}_1(\mathcal{S})$. We have

$$\boldsymbol{X}^s_1 = \boldsymbol{M}_1 \mathcal{M}_1(\tilde{S}) \left(\Theta_2 \boldsymbol{M}_2 \otimes \cdots \otimes \Theta_d \boldsymbol{M}_d\right)^T.$$

Then, we remove the effect of Θ_1. Fold back $\boldsymbol{X}^s_1$ to tensor $\mathcal{X}^s$ and repeat the steps with $\mathcal{X}_s$ for the other modes. We finally obtain

$$\mathcal{X}^s = \tilde{\mathcal{S}} \times_1 \boldsymbol{M}_1 \times \cdots \times_d \boldsymbol{M}_d.$$

If we apply the above de-heterogeneity procedures to the noisy observation $\mathcal{Y}$, the only thing we need to worry about is the noise. Obviously, the independence of the noise tensor still holds, but the variance for each entry may change. We need to check it more.

Biclustering	Tensor-SCORE	TBM
Input: Adjacency matrix $\mathcal{Y}$, # of clusters r, tuning para τ **Output:** Estimated assignment $\hat{z}$	**Input**: Adjacency tensor $\mathcal{Y}$, # of clusters r, tuning paras δ, T **Output:** Estimated assignment $\hat{z}$	**Input**: Observation $\mathcal{Y}$, # of clusters $(r_1, ..., r_d)$, tuning para M **Output:** Estimated assignment $\hat{z}$
Initialization: 1. Obtain truncated $\mathcal{Y}$, $T_\tau(\mathcal{Y})$ 2. Obtain estimated rank-r signal $\hat{\mathcal{X}} = \arg\min_{\text{rank}(\mathcal{X}) \leq r} \lVert T_\tau(\mathcal{Y}) - \mathcal{X} \rVert_F^2$	**Initialization:** 1. Apply HOSVD or randomized graph projection to $\mathcal{Y}$ 2. Obtain initial factor matrix $\hat{\boldsymbol{U}}^{(0)}$	**Initialization:** 1. Obtain singular space $\hat{\boldsymbol{U}}_k$ via matrix SVD 2. Obtain initial $\hat{z}^{(0)}$ by applying relaxed (with para M) k-means to $\hat{\boldsymbol{U}}_k$
Iteration:	**Iteration:** 1. Apply regularized (with para δ) HOOI with $\hat{\boldsymbol{U}}^{(0)}$ 2. Obtain $\hat{\boldsymbol{U}}^{(t)}$	**Iteration:** 1. Apply HLloyd with $\hat{z}^{(0)}$ 2. Obtain estimated assignment $\hat{z} = z_k^{(t)}$
Post-iteration: 1. Obtain SCORE normalized $\hat{\mathcal{X}}$, $\tilde{\mathcal{X}}$ 2. Apply k-median clustering to $\tilde{\mathcal{X}}$ and obtain $\tilde{z}$ 3. Apply refinement procedures to $\tilde{z}$ and obtain $\hat{z}$: $\hat{z}(i) = \arg\max_{u \in [r]} \frac{1}{\lvert\{j : \tilde{z}_j = u\}\rvert} \sum_{\{j:\tilde{z}_j = u\}} \mathcal{Y}_{ij}$	**Post-iteration:** 1. Obtain truncated (with para T) SCORE normalized $\hat{\boldsymbol{U}}^{(t)}$, $\tilde{\boldsymbol{U}}$ 2. Apply k-means to $\tilde{\boldsymbol{U}}$ and obtain $\hat{z}$	**Post-iteration:**
Guarantee: $\ell(\hat{z}, z^*) = \mathcal{O}(\exp\{-n\})$	**Guarantee:** $\ell(\hat{z}, z^*) = \mathcal{O}(n^{-d+2})$	**Guarantee:** $\ell(\hat{z}, z^*) = \mathcal{O}(\exp\{-n^{d-1}\})$

Table 1: Algorithm comparison of three methods.

Algorithm 1 High-order degree-corrected spectral clustering (HDCSC)

Input: Observation $\mathcal{Y} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$, $(r_1, ..., r_d)$, relaxation factor in k-means $M > 1$, SCORE normalization function h

Compute $\tilde{\boldsymbol{U}}_k = SVD_{r_k}(\mathcal{M}_k(\mathcal{Y})$ for $k \in [d]$

for $k \in [d]$ **do**

Estimate the singular space $\hat{\boldsymbol{U}}_k$ via

$$\hat{U}_k = SVD_{r_k}(\mathcal{M}_k(\mathcal{Y} \times_1 \tilde{\boldsymbol{U}}_1^T \times \cdots \times_{k-1} \tilde{\boldsymbol{U}}_{k-1}^T \times_{k+1} \tilde{\boldsymbol{U}}_{k+1}^T \times \cdots \times_d \tilde{\boldsymbol{U}}_d^T))$$

end for

for $k \in [d]$ **do**

Calculate $\hat{\mathbf{Y}}_k$ via $\hat{\mathbf{Y}}_k = \hat{\boldsymbol{U}}_k \hat{\boldsymbol{U}}_k^T \mathcal{M}_k(\mathcal{Y} \times_1 \hat{\boldsymbol{U}}_1^T \times \cdots \times_{k-1} \hat{\boldsymbol{U}}_{k-1}^T \times_{k+1} \hat{\boldsymbol{U}}_{k+1}^T \times \cdots \times_d \hat{\boldsymbol{U}}_d^T)$

Let $\hat{\mathbf{Y}}_{kj}$ denote the rows of $\hat{\mathbf{Y}}_k$ for $j \in [p_k]$. Obtain the SCORE normalized $\hat{\mathbf{Y}}_k^s$ via $\hat{\mathbf{Y}}_{kj}^s = \frac{\hat{\mathbf{Y}}_{kj}}{h(\hat{\mathbf{Y}}_{kj})}$

Find the initial assignment $z_k^{(0)} \in [r_k]^{p_k}$ and centroids $\hat{x}_1, ..., \hat{x}_{rk} \in \mathbb{R}^{r-k}$ such that

$$\sum_{j=1}^{p_k} \left\| (\hat{\mathbf{Y}}_{kj}^s)^T - \hat{x}_{(z_k^{(0)})_j} \right\|^2 \leq M \min_{x_1, ..., x_{r_k}, z_k} \sum_{j=1}^{p_k} \left\| (\hat{\mathbf{Y}}_{kj}^s)^T - x_{(z_k^{(0)})_j} \right\|^2$$

end for

Output: $\left\{ z_k^{(0)} \in [r_k]^{p_k}, k \in [d] \right\}$

Algorithm 2 High-order degree-corrected Lloyd Algorithm (HDCLloyd)

Input: Observation $\mathcal{Y} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$, initialization $\{z_k^{(0)} \in [r_k]^{p_k}\}$, iteration number T, SCORE normalization function h

Let $\mathcal{Y}^s = \mathcal{Y}$.

for $k \in [d]$ **do**

Let $\boldsymbol{Y}_k^s$ denote the k-th model matricization of $\mathcal{Y}^s$. Apply SCORE normalization to $\boldsymbol{Y}_k^s$ and obtain $\tilde{\boldsymbol{Y}}_k^s$ via

$$\tilde{\mathbf{Y}}_{kj}^s = \mathbf{Y}_{kj}^s / h(\mathbf{Y}_{kj}^s)$$

Fold $\tilde{\mathbf{Y}}_k^s$ back to tensor $\tilde{\mathcal{Y}}^s$. Update $\mathcal{Y}^s = \tilde{\mathcal{Y}}^s$.

end for

Apply HLloyd with $\mathcal{Y}^s$, initialization $\{z_k^{(0)} \in [r_k]^{p_k}\}$, and iteration number T.

Output: $\left\{ z_k^{(T)} \in [r_k]^{p_k}, k \in [d] \right\}$

References

Gao, C., Ma, Z., Zhang, A. Y., and Zhou, H. H. (2018). Community detection in degree-corrected block models. The Annals of Statistics, 46(5):2153–2185.

Han, R., Luo, Y., Wang, M., and Zhang, A. R. (2020). Exact clustering in tensor block model: Statistical optimality and computational limit. arXiv preprint arXiv:2012.09996.

Ke, Z. T., Shi, F., and Xia, D. (2019). Community detection for hypergraph networks via regularized tensor power iteration. arXiv preprint arXiv:1909.06503.