Q & A about novelty

Jiaxin Hu

May 22, 2022

1. **Could we design *iterative* clean up (as local iterations in clustering problem) to show tensor benefits?**

   Yes. But finite sample accuracy theorems may be added.

   A possible iterative clean-up strategy is in following Algorithm 1, which is the higher-order analogy of Algorithm 5 in Ding et al. (2021). Notice that in Line 2, we only update the permutation on the first mode. The update for the other $m-1$ modes may not be necessary because we assume the permutation on every mode is the same, even though $\mathcal{A}$ and $\mathcal{B}$ are not super-symmetric.

   Ding et al. (2021) does not provide the accuracy for iterative clean-up. I believe the accuracy for clean-up should relate to the order m due to the larger sample size in tensor case. But to obtain the accuracy for clean-up, we need to first obtain the finite sample initialization accuracy (similar as Theorem 4 in dTBM, not the consistency/exact recovery when $n \to \infty$). Both Ding et al. (2021) and us do not have theoretical results on the finite sample accuracy, which may be a novel point if we provide such result.

Algorithm 1 Iterative clean-up

Input: Gaussian tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n^{\otimes m}}$, initialization $\pi^{(0)}$, maximal iteration number T

1: **for** $t = 0$ to $T-1$ **do**

2: Update the permutation via

$$\Pi^{(t+1)} = \mathop{\arg\max}_{\Pi \in \mathcal{P}_n} \left\langle \mathcal{A} \times_1 \Pi \times_2 \Pi^{(t)} \times_3 \cdots \times_m \Pi^{(t)}, \mathcal{B} \right\rangle,$$

where $\mathcal{P}_n$ is the collection of all possible permutation matrices on $[n]$.

3: **end for**

Output: Estimated permutation $\hat{\pi} = \pi^{(T)}$.

2. **Can we look into spectral method to get specificity of tensors in degree profile method?**

   Maybe no...

   Consider the scaled version of correlated Gaussian model

$$\boldsymbol{B} = \pi \circ \boldsymbol{A} + \sigma \boldsymbol{Z},$$

   where $\boldsymbol{Z}$ is a random standard Gaussian matrix independent with $\boldsymbol{A}, \boldsymbol{B}$ and σ is the noise level inversely related to the correlation parameter ρ. In both spectral and degree profile

method, we need to find the regime of noise level σ in which signal from permutation is dominant.

In spectral method, we match the nodes i, k by considering the corresponding eigen-values λ_i and ν_k of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. In degree profile method, we consider the empirical distribution F_n^i and G_n^k of the i-th row of $\boldsymbol{A}$ and k-th row of $\boldsymbol{B}$.

Consider the noiseless case, $\sigma = 0$. If $k = \pi^*(i)$, we have $\lambda_i = \nu_k$ and $F_n^i = G_n^k$. If $k \neq \pi^*(i)$, there are gaps between λ_i, ν_k and F_n^i, G_n^k. The noise condition is determined by the magnitude of gaps.

Now we extend both methods to tensor cases. Then, we need following changes

(a) for spectral method,
find gap between random **matrix** eigen-values $\lambda_i, \mu_k \quad \rightarrow$
find gap between random **tensor** eigen-values λ_i, μ_k;

(b) for degree profile method,
find gap between empirical distributions F_n^i, G_n^k calculated by **rows** $\boldsymbol{A}_{i:}, \boldsymbol{B}_{k:} \in \mathbb{R}^n \quad \rightarrow$
find gap between empirical distributions F_n^i, G_n^k calculated by **slices** $\boldsymbol{A}_{i:}, \boldsymbol{B}_{k:} \in \mathbb{R}^{n^{\otimes(m-1)}}$.

In spectral method, there is a technically difference when we handle the matrix or tensor eigenvalues. But in degree profile method, the empirical distribution does not use the spatial structure of the data, and we will vectorize the slices to vectors to calculate the empirical distribution.

Therefore, I think the matrix/tensor distinction in spectral method may not be applicable to the degree profile method.

In addition, the unseeded spectral method in Fan et al. (2019) has the same accuracy as the unseeded degree profile algorithm (not the seeded degree profile algorithm). The accuracy for spectral method may also be improved by adding seed procedures.

3. **What is the 1-d analogy of matching problem. Where does matrix-specific technique come in?**

The most significant difference between 1-d and 2-d matching is the multi-way permutation issue, which does not differ 2-d and m-d ($m \geq 3$) cases.

We discuss the non-symmetric observation in this question and leave the symmetry/dependency discussion to the next question.

The 1-d analogy of Gaussian matching problem is stated as following. Suppose that we have n nodes, a true permutation π^* on $[n]$, and two sets of N-dimensional vector representations $\{\boldsymbol{a}_i\}_{i\in[n]}$ and $\{\boldsymbol{b}_i\}_{i\in[n]}$ for each node, where for all $i \in [n], j \in [N]$

$$(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{\pi^*(i),j}) \sim \mathcal{N}\left(\boldsymbol{0}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right), \quad (\boldsymbol{a}_{i,j}, \boldsymbol{b}_{\pi^*(i),j}) \perp (\boldsymbol{a}_{i',j'}, \boldsymbol{b}_{\pi^*(i'),j'}), \text{ for all } (i,j) \neq (i',j').$$

For the matrix Gaussian matching problem, with observed non-symmetric adjacency matrices $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times n}$, for all $i \in [n]$ and $j \in [n]$, we assume that

$$(\boldsymbol{A}_{i,j}, \boldsymbol{B}_{\pi^*(i),\pi^*(j)}) \sim \mathcal{N}\left(\boldsymbol{0}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right), (\boldsymbol{A}_{i,j}, \boldsymbol{B}_{\pi^*(i),\pi^*(j)}) \perp (\boldsymbol{A}_{i',j'}, \boldsymbol{B}_{\pi^*(i'),\pi^*(j')}), \text{ for all } (i,j) \neq (i',j').$$

To match the nodes $i, k \in [n]$ in two observations, we measure the similarity between $\boldsymbol{a}_i$ versus $\boldsymbol{b}_k$ in 1-d case while consider rows $\boldsymbol{A}_{i:}$ versus $\boldsymbol{B}_{k:}$ in 2-d case. The order of components in $\boldsymbol{a}_i$ and $\boldsymbol{b}_k$ is known because the permutation only happens on the first index of $\boldsymbol{a}_{i,j}$. Hence, the distance based on direct subtraction, e.g., $\|\boldsymbol{a}_i - \boldsymbol{b}_k\|$, works for 1-d case. In contrast, we do not know the component correspondence between $\boldsymbol{A}_{i:}$ and $\boldsymbol{B}_{k:}$ since both indices in $\boldsymbol{A}_{i,j}$ are permuted and thus $\|\boldsymbol{A}_{i:} - \boldsymbol{B}_{k:}\|$ does not work in 2-d case.

The multi-way permutation in matrix or tensor matching requires us to adapt a component-order-invariant distance, e.g., $\|F_n - G_n\|$ where F_n, G_n are empirical distributions of $\boldsymbol{A}_{i:}, \boldsymbol{B}_{k:}$, respectively. **This multi-way permutation issue make the 1-d and 2-d matching problem significantly different. However, multi-way permutation issue does not differ the 2-d and m-d with $m \geq 3$ cases.** Without the concerns for symmetry or dependency, the order-invariant distance for 2-d case can always be smoothly extended for higher-order cases.

4. **Symmetry.**

   We now consider the symmetric observations.

   Symmetry in matrix matching algorithm.

   Another difference between 1-d and 2-d matching is the dependence in adjacency matrices. In 1-d case, the vector representation $\boldsymbol{a}_i$ is independent with $\boldsymbol{b}_k$ if $\pi^*(i) \neq k$. But in 2-d case, the independence between $\boldsymbol{A}_{i:}$ and $\boldsymbol{B}_{k:}$ for $\pi^*(i) \neq k$ does not always hold. Note that $\boldsymbol{A}_{i,j}$ and $\boldsymbol{B}_{\pi^*(j),\pi^*(i)}$ are correlated because $\boldsymbol{B}_{\pi^*(j),\pi^*(i)} = \boldsymbol{B}_{\pi^*(i),\pi^*(j)}$ and $\boldsymbol{A}_{i,j}$ and $\boldsymbol{B}_{\pi^*(i),\pi^*(j)}$ are correlated. Thus, $\boldsymbol{A}_{i:}$ is not independent with $\boldsymbol{B}_{k:}$ when $k = \pi^*(j)$ for $j \neq i$.

   Ding et al. (2021) does not handle this symmetry issue in Gaussian case. They claim that the degree profiles $(\boldsymbol{A}_{i:}, \boldsymbol{B}_{k:})$ are either correlated in pairs or independent; see their Remark 2. As $n \to \infty$, the symmetry effect becomes minor because the number of correlated components between $\boldsymbol{A}_{i:}$ and $\boldsymbol{B}_{\pi^*(j):}$ for $j \neq i$ is negligible compared with n. That's may be the reason why Ding et al. (2021) does not address this issue.

   In Bernoulli case, instead of $\boldsymbol{A}_{i:}$ and $\boldsymbol{B}_{k:}$, they use the sets of columns $A_i = \{\boldsymbol{A}_{:j}\mathbb{1}\{\boldsymbol{A}_{ij} = 1\}\}$ and $B_k = \{\boldsymbol{B}_{:j}\mathbb{1}\{\boldsymbol{B}_{kj} = 1\}\}$ to describe the distributions of node i and k. The "outdegree" solves the dependence issue among the columns inside the sets A_i and B_k. But the "outdegree" does not solve the dependence between A_i and B_k. Note that $\mathbb{1}\{\boldsymbol{A}_{ij} = 1\}$ is correlated with $\mathbb{1}\{\boldsymbol{B}_{\pi^*(j)\pi^*(i)} = 1\}$ and thus A_i is not independent with $B_{\pi^*(j)}$ for $j \neq i$. Again, as n goes to ∞, the dependence issue among A_i and $B_{\pi^*(j)}$ is negligible due to the small number of correlated components.

   Symmetry in tensor matching algorithm.

   We directly extend the method in Ding et al. (2021) to higher-order cases, and thus the same symmetry issue occurs. Without loss of generality, let π^* be identity mapping. Consider two tensor observations, $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n^{\otimes m}}$ and we measure the similarity between the slices $\mathcal{A}_{i:}, \mathcal{B}_{k:} \in \mathbb{R}^{n^{\otimes(m-1)}}$. For $i \neq k$, the entries in $\mathcal{A}_{i:}$ with indices $\{\omega = \{i, \omega_2, \ldots, \omega_m\} : k \in \{\omega_2, \ldots, \omega_m\}\}$ are correlated with the entries in $\mathcal{B}_{k:}$ due to the super-symmetry. So, there are $(m-1)n^{m-2}$ entries are correlated between $\mathcal{A}_{i:}$ and $\mathcal{B}_{k:}$, which has a smaller order than the number of entries in $\mathcal{A}_{i:}$, n^{m-1}.

   We also do not address this issue yet since we consider the non-symmetric observations now. But I think this issue might be solved by kicking off the correlated entries.

   Symmetry in tensor MLE phase transition.

Without loss of generality, let π^* be identity mapping. In MLE phase transition, the symmetry issue occurs when we find the relationship between node errors and edge errors. For every index $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m) \in [n]^m$, define the equivalent set of $\boldsymbol{\omega}$

$$S_{\boldsymbol{\omega}} = \{\boldsymbol{v} : \boldsymbol{v} = (\omega_{\pi(1)}, \ldots, \omega_{\pi(m)}) \text{ for some permutation } \pi \text{ on } [m] \}.$$

Therefore, by super-symmetry, we have $\mathcal{A}_{\boldsymbol{\omega}} = \mathcal{A}_{\boldsymbol{\omega}'}$ for all $\boldsymbol{\omega}' \in S_{\boldsymbol{\omega}}$.

Suppose we have a permutation π on $[n]$ with the set of unfixed point $D = \{i \in [n] : \pi(i) \neq i\}$ and $d = |D|$. We want to find the set of unfixed edges $D^E = \{\boldsymbol{\omega} \in [n]^m : \pi \circ \boldsymbol{\omega} \notin S_{\boldsymbol{\omega}}\}$ and lower bound its cardinality $d^E = |D^E|$. We have

$$d^E = d^m - e(m), \quad \text{where } e(m) = |E(m)| \text{ and } E(m) = \{\boldsymbol{\omega} \in D^m : \pi \circ \boldsymbol{\omega} \in S_{\boldsymbol{\omega}}\}.$$

The induction may not work here because there is a weak connection between $E(m)$ and $E(m+1)$. For example, let $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ be an arbitrary vector in $E(3)$ and $\boldsymbol{v} = (v_1, v_2, v_3, v_4)$ be an arbitrary vector in $E(4)$. We must have

$$\{\omega_1, \omega_2, \omega_3\} \nsubseteq \{v_1, v_2, v_3, v_4\}.$$

If $\omega_1 = v_1, \omega_2 = v_2, \omega_3 = v_3$, then $\pi \circ (v_1, v_2, v_3) \in S_{(v_1, v_2, v_3)}$ and $\pi(v_4) = v_4$, which indicates v_4 is a fixed point, $v_4 \notin D$, and contradicts to the definition of $E(4)$. Therefore, we can not use the information we already know about $E(3)$ to infer the information about $E(4)$.

References

Ding, J., Ma, Z., Wu, Y., and Xu, J. (2021). Efficient random graph matching via degree profiles. *Probability Theory and Related Fields*, 179(1):29–115.

Fan, Z., Mao, C., Wu, Y., and Xu, J. (2019). Spectral graph matching and regularized quadratic relaxations i: The gaussian model. *arXiv preprint arXiv:1907.08880*.