

# Gaussian Matching

Jiaxin Hu

April 2, 2022

## 1 Model Formulation

### 1.1 Notations

- Let the lowercase letters (e.g.,  $a, c$ ) denote the scalar; the bold lowercase letters (e.g.,  $\boldsymbol{\omega}, \boldsymbol{v}$ ) denote the vector; the uppercase letters (e.g.,  $S, T$ ) denote the index set with elements in  $\mathbb{Z}_+$ ; the bold uppercase letters (e.g.,  $\boldsymbol{P}, \boldsymbol{Q}$ ) denote the dimension-2 index set with elements in the form  $(i, j)$  where  $i, j \in \mathbb{Z}_+$ ; and the calligraphy letters (e.g.,  $\mathcal{A}, \mathcal{B}$ ) denote the tensor of order three or greater.
- For a positive integer  $n$ , let  $[n]$  denote the index set  $\{1, \dots, n\}$ .
- For an index set  $S$  and a positive integer  $m$ , let  $S^m$  denote the dimension- $m$  vector space of  $S$ , where  $S^m = \{(i_1, \dots, i_m) : i_k \in S, \text{ for all } k \in [m]\}$ .
- For two index sets  $S$  and  $T$ , we call the function  $\pi: S \mapsto T$  *the perfect matching between  $S$  and  $T$*  when  $\pi(i_1) = \pi(i_2)$  if and only if  $i_1 = i_2$  for any  $i_1, i_2 \in S$ . When  $T = S$ , we call the  $\pi$  *the permutation on  $S$* .
- For a perfect matching  $\pi: S \mapsto T$ , we call the dimension-2 index set  $\boldsymbol{P} = \{(i, \pi(i)) : i \in S\}$  *the set corresponding to  $\pi$* , and we call  $\pi$  *the perfect matching corresponding to  $\boldsymbol{P}$* .
- For a perfect matching  $\pi: S \mapsto T$  and an index set  $S_0 \subset S$ , let  $\pi|_{S_0}: S_0 \mapsto T_0$  denote the sub-matching of  $\pi$  for the nodes in  $S_0$ , where  $T_0 = \{\pi(i) : i \in S_0\} \subset T$ .
- For a perfect matching  $\pi: S \mapsto T$  and a dimension- $m$  vector  $\boldsymbol{v} = (v_1, \dots, v_m) \in S^m$ , let  $\pi \circ \boldsymbol{v} = (\pi(v_1), \dots, \pi(v_m)) \in T^m$  denote the permutation of the vector  $\boldsymbol{v}$ .
- Let  $\mathcal{A} \in \mathbb{R}^{n^{\otimes m}}$  denote an order- $m$  real tensor of dimension  $n$  on each mode. For the vector  $\boldsymbol{\omega} = (\omega_1, \dots, \omega_m) \in [n]^m$ , we use  $\mathcal{A}_{\boldsymbol{\omega}}$  to denote the  $(\omega_1, \dots, \omega_m)$ -th entry of  $\mathcal{A}$ .
- We call a tensor  $\mathcal{A} \in \mathbb{R}^{n^{\otimes m}}$  *super-symmetric* if for all  $\boldsymbol{\omega} = (\omega_1, \dots, \omega_m) \in [n]^m$  we have  $\mathcal{A}_{\boldsymbol{\omega}} = \mathcal{A}_{\pi \circ \boldsymbol{\omega}}$  for all permutations  $\pi$  on the set  $\{\omega_1, \dots, \omega_m\}$ .
- Let  $\times_k$  denote the tensor-by-matrix multiplication on the  $k$ -th mode.
- Let  $\|\cdot\|_F$  denote the Frobenius norm for tensors. Let  $\langle \cdot, \cdot \rangle$  denote the inner product for tensors.

## 1.2 Higher-order Correlated Winger Model

Consider two random super-symmetric tensors  $\mathcal{A}, \mathcal{B}' \in \mathbb{R}^{n^{\otimes m}}$ . Assume that all the pairs  $\{(\mathcal{A}_\omega, \mathcal{B}'_\omega) : \omega \in [n]^m \cap \{\omega : \omega_1 \leq \dots \leq \omega_m\}\}$  follow the i.i.d. correlated multivariate zero-mean Gaussian distribution with variance 1 and correlation  $\rho \in (0, 1)$ ; i.e.,

$$\begin{pmatrix} \mathcal{A}_\omega \\ \mathcal{B}'_\omega \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right), \quad \text{and} \quad \begin{pmatrix} \mathcal{A}_\omega \\ \mathcal{B}'_\omega \end{pmatrix} \text{ is independent with } \begin{pmatrix} \mathcal{A}_{\omega'} \\ \mathcal{B}'_{\omega'} \end{pmatrix},$$

for all  $\omega' \neq \omega$  and  $\omega' \in [n]^m \cap \{\omega : \omega_1 \leq \dots \leq \omega_m\}$ . The tensors  $\mathcal{A}, \mathcal{B}'$  are two correlated Winger tensors. Let  $\pi^*$  be a permutation on  $[n]$ , and  $\Pi^* \in \{0, 1\}^{n \times n}$  denote the corresponding permutation matrix with entries  $\Pi^*_{ij} = 1$  if  $j = \pi^*(i)$  and  $\Pi^*_{ij} = 0$ , otherwise. Consider the permuted tensor  $\mathcal{B}$  such that for all  $\omega \in [n]^m$

$$\mathcal{B}_\omega = \mathcal{B}'_{\pi^* \circ \omega}, \quad \text{or equivalently} \quad \mathcal{B} = \mathcal{B}' \times_1 \Pi^* \times_2 \dots \times_m \Pi^*.$$

We call the pair  $(i, k) \in [n]^2$  as a *true pair* if  $k = \pi^*(i)$ , and  $(i, k)$  is a *fake pair*, otherwise.

Our goal is to recover  $\pi^*$  (or equivalently  $\Pi^*$ ) observing  $\mathcal{A}, \mathcal{B}$ .

## 1.3 Maximum Likelihood Estimate

Let  $\Sigma$  denote the covariance matrix  $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$  with inverse  $\Sigma^{-1} = \frac{1}{1-\rho^2} \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix}$ . The permutation  $\pi^*$  (or equivalently permutation matrix  $\Pi$ ) and the correlation  $\rho$  are unknown parameters, and  $\pi^*$  is our of main interest. We have the likelihood for the higher-order correlated Winger model

$$\begin{aligned} \mathcal{L}(\Pi, \rho | \mathcal{A}, \mathcal{B}) &= \frac{1}{[2\pi \det(\Sigma)]^{n^m/2}} \exp \left( -\frac{1}{2} \sum_{\omega \in [n]^m} (\mathcal{A}_{\pi \circ \omega}, \mathcal{B}_\omega) \Sigma^{-1} (\mathcal{A}_{\pi \circ \omega}, \mathcal{B}_\omega)^T \right) \\ &= \frac{1}{[2\pi \det(\Sigma)]^{n^m/2}} \exp \left( -\frac{1}{2(1-\rho^2)} [\|\mathcal{A}\|_F^2 + \|\mathcal{B}\|_F^2 - 2\rho \langle \mathcal{A} \times_1 \Pi \times_2 \dots \times_m \Pi, \mathcal{B} \rangle] \right). \end{aligned}$$

For any  $\rho \in (0, 1)$ , consider the estimator

$$\hat{\Pi}(\rho) = \arg \max_{\Pi} \mathcal{L}(\Pi | \rho, \mathcal{A}, \mathcal{B}) = \arg \max_{\Pi} \langle \mathcal{A} \times_1 \Pi \times_2 \dots \times_m \Pi, \mathcal{B} \rangle.$$

Note that  $\hat{\Pi}(\rho)$  is independent with  $\rho$ . Therefore, we let  $\hat{\Pi}$  denote the estimator  $\hat{\Pi}(\rho)$ , and  $\hat{\Pi}$  is the MLE of  $\Pi^*$ . The permutation  $\hat{\pi}$  corresponding to  $\hat{\Pi}$  is the MLE of  $\pi^*$ .

## 2 Algorithm

### 2.1 Matching via Empirical Distributions

We describe each node by the slice empirical distribution and adapt the sup-norm distance for distributions as the similarity measure to construct the mapping. Specifically, we define the sup-norm distance for all pairs  $(i, k) \in [n]^2$  as

$$d_{ik} = \sup_{t \in \mathbb{R}} |F_n^i(t) - G_n^k(t)|, \tag{1}$$

where for all  $t \in \mathbb{R}$

$$F_n^i(t) = \frac{1}{n^{m-1}} \sum_{\omega \in [n]^{m-1}} \mathbb{1}\{\mathcal{A}_{i,\omega} \leq t\}, \quad G_n^k(t) = \frac{1}{n^{m-1}} \sum_{\omega \in [n]^{m-1}} \mathbb{1}\{\mathcal{B}_{k,\omega} \leq t\}$$

are slice empirical distributions for node  $i$  with observation  $\mathcal{A}$  and node  $k$  with observation  $\mathcal{B}$ , respectively. The sup-norm distance  $d_{ik}$  is smaller when  $F_n^i$  and  $G_n^k$  are more correlated, which motivates our Algorithm 1 for Gaussian tensor matching.

---

**Algorithm 1** Gaussian tensor matching via empirical distribution

---

**Input:** Gaussian tensors  $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n^{\otimes m}}$ .

- 1: Calculate the sup-norm distances  $d_{ik}$  as (1) for all pairs  $(i, k) \in [n]^2$ .
- 2: Obtain the set  $\mathbf{P} = \{(i, k) \in [n]^2 : d_{ik} \text{ is no larger than the } n\text{-th smallest } d_{ik}\}$ .
- 3: **if** there exists a permutation on  $[n]$ ,  $\hat{\pi}$ , corresponding to the set  $\mathbf{P}$  **then**
- 4:     Output  $\hat{\pi}$ .
- 5: **else**
- 6:     Output error.
- 7: **end if**

**Output:** Estimated permutation  $\hat{\pi}$  or error.

---

Following theorem shows the guarantee for Algorithm 1 to exactly recover true permutation  $\pi^*$ .

**Theorem 2.1** (Guarantee for Algorithm 1). *Let  $\sigma^2 = \sqrt{1 - \rho^2}$ . Suppose  $\sigma \leq c \log^{-1} n$  for some sufficiently small positive constant  $c$ . Algorithm 1 exactly recovers the true permutation  $\pi^*$  with probability tends to 1 as  $n \rightarrow \infty$ .*

## 2.2 Improved Matching with Seeded Algorithm

We improve the Algorithm 1 with seeded algorithm. Our seeded algorithm involves three steps: (1) generating seeds that reveals the true permutation  $\pi^*$  for a subset of nodes; (2) recovering the full permutation with the seed; (3) refining the estimated permutation in (2) to achieve exact recovery.

Specifically, we consider the seed that involves high-similarity and high-degree pairs in step (1). We measure the similarity by the sup-norm distance defined in (1). The “degree” of node  $i$  in  $\mathcal{A}$  and node  $k$  in  $\mathcal{B}$  for all  $i, k \in [n]$  are represented as

$$a_i = \frac{1}{n^{(m-1)/2}} \sum_{\omega \in [n]^{m-1}} \mathcal{A}_{i,\omega}, \quad \text{and} \quad b_k = \frac{1}{n^{(m-1)/2}} \sum_{\omega \in [n]^{m-1}} \mathcal{B}_{k,\omega}. \quad (2)$$

Hence, we consider the following seed set  $\mathbf{Q}$  with given thresholds  $\xi, \zeta$

$$\mathbf{Q} = \{(i, k) \in [n]^2 : a_i, b_k \geq \xi, d_{ik} \leq \zeta\}. \quad (3)$$

The  $\mathbf{Q}$  contains more seeds with a smaller  $\xi$  and larger  $\zeta$ . As our proof shows later, the set  $\mathbf{Q}$  with proper thresholds  $\xi$  and  $\zeta$  contains only true pairs with high probability. Suppose there exists a perfect matching  $\pi_0 : S \mapsto T$  corresponding to  $\mathbf{Q}$ , where  $S$  and  $T$  are two subsets of  $[n]$ . Then, we solve the rest permutations by transferring the multiway tensor matching to a bipartite matching problem with  $\pi_0$ . Detail procedures are in Algorithm 2.

Following theorem shows the guarantee for Algorithm 2 to exactly recover true permutation  $\pi^*$ .

---

**Algorithm 2 Improved Gaussian tensor matching with seeded algorithm**

---

**Step 1: Seeds generation**

---

**Input:** Gaussian tensors  $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n^{\otimes m}}$ , thresholds  $\xi$  and  $\zeta$ .

- 1: Calculate the sup-norm distances  $d_{ik}$  as (1) for all pairs  $(i, k) \in [n]^2$  and the degrees  $a_i$  and  $b_i$  as (2) for all  $i \in [n]$ .
- 2: Obtain the seed set  $\mathbf{Q}$  as (3) with  $\xi$  and  $\zeta$ .
- 3: **if** there exists a perfect matching  $\pi_0 : S \mapsto T$  corresponding to  $\mathbf{Q}$  **then**
- 4:     Output  $\pi_0$ .
- 5: **else**
- 6:     Stop the entire Algorithm 2 immediately and output error.
- 7: **end if**

**Output:** Perfect matching  $\pi_0$  or error.

---

**Step 2: Seeded matching**

---

**Input:** Gaussian tensors  $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n^{\otimes m}}$  and the perfect matching  $\pi_0 : S \mapsto T$  from **Step 1**.

- 8: Calculate the analogy of sample covariance  $H_{ik}$  for all pairs  $(i, k)$  such that  $i \in S^c$  and  $k \in T^c$ , where  $S^c = [n]/S, T^c = [n]/T$ , and

$$H_{ik} = \sum_{\omega \in S^{m-1}} \mathcal{A}_{i,\omega} \mathcal{B}_{k,\pi_0 \circ \omega}.$$

- 9: Find the optimal perfect matching  $\tilde{\pi}_1 : S^c \mapsto T^c$  such that

$$\tilde{\pi}_1 = \arg \max_{\pi : S^c \mapsto T^c} \sum_{i \in S^c} H_{i\pi(i)}.$$

- 10: Concatenate the matching  $\pi_0$  and  $\tilde{\pi}_1$  to a permutation  $\pi_1$  on  $[n]$  such that  $\pi_1|_S = \pi_0$  and  $\pi_1|_{S^c} = \tilde{\pi}_1$ .

**Output:** Estimated permutation  $\pi_1$ .

---

**Step 3: Non-iterative clean-up**

---

**Input:** Gaussian tensors  $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n^{\otimes m}}$  and the permutation  $\pi_1$  on  $[n]$  from **Step 2**.

- 11: Calculate  $W_{ik} = \sum_{\omega \in [n]^{m-1}} \mathcal{A}_{i,\omega} \mathcal{B}_{k,\pi_1 \circ \omega}$  for all pairs  $(i, k) \in [n]^2$ .
- 12: Obtain the set  $\mathbf{P} = \{(i, k) \in [n]^2 : W_{ik} \text{ is no smaller than the } n\text{-th largest } W_{ik}\}$ .
- 13: **if** there exists a permutation on  $[n]$ ,  $\hat{\pi}$ , corresponding to the set  $\mathbf{P}$  **then**
- 14:     Output  $\hat{\pi}$ .
- 15: **else**
- 16:     Output error.
- 17: **end if**

**Output:** Estimated permutation  $\hat{\pi}$  or error.

---

**Theorem 2.2** (Guarantee for Algorithm 2). *Let  $\sigma^2 = \sqrt{1 - \rho^2}$ . Suppose  $\sigma \leq c \log^{-1/3(m-1)} n$  for some sufficiently small positive constant  $c$ . Choose thresholds  $\xi \geq c_1 \sqrt{\log^{1/(m-1)} n}$  and  $\zeta \leq c_2 \sqrt{\sigma/n^{m-1}}$  for some positive constants  $c_1, c_2$ . Algorithm 2 exactly recovers the true permutation  $\pi^*$  with probability tends to 1 as  $n \rightarrow \infty$ .*

## References