Supplementary material for "Simultaneous Clustering and Model Selection for Tensor Affinities"

Zhuwen Li[1], Shuoguang Yang[2]*, Loong-Fah Cheong[1] and Kim-Chuan Toh[1]
[1]National University of Singapore [2]Columbia University

A. Proofs of the Theorems

A.1. Proof of Theorem 1

Theorem 1. *For any supersymmetric tensor* $\mathcal{X} \in \mathbb{R}^{N\times N\times\ldots\times N}$ *of order at least 3, if it has the form*

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{z}_r \circ \mathbf{z}_r \circ \cdots \circ \mathbf{z}_r, \quad \sum_{r=1}^{R} \mathbf{z}_r = \mathbf{e}, \quad \mathbf{z}_r \in \{0,1\}^N, \tag{1}$$

any slice of $\mathcal{X}$ *can only be either a rank-1 matrix* $\mathbf{z}_r \circ \mathbf{z}_r$ *or a* $\mathbf{0}$ *matrix.*

Proof. Let's first consider $\text{mode} = 3$. For a fixed node (index) i, its slice matrix $\mathcal{X}(:,:,i) = \sum_{r=1}^{R} \mathbf{y}_i(r) \cdot \mathbf{z}_r \circ \mathbf{z}_r$[1], where $\mathbf{y}_i \in \mathbb{R}^R$ indicates to which cluster the node i belong. More specifically, if node i belongs to cluster r, $\mathbf{y}_i(r) = 1$, otherwise $\mathbf{y}_i(r) = 0$. Since a node cannot belong to multiple clusters, let's say it is in cluster r. Then, $\mathcal{X}(:,:,i) = \mathbf{z}_r \circ \mathbf{z}_r$, which is rank one. When $\text{mode} = 3$, there is no $\mathbf{0}$ slice matrix, because a node i must belong to one cluster. We can also get an intuitive interpretation of the preceding by considering that the cube tensor must be in a block-diagonal form (obtained w.l.o.g. by permuting the nodes *w.r.t* their cluster labels). Now consider $\text{mode} = 4$. For fixed nodes (indices) i and j, the corresponding slice matrix $\mathcal{X}(:,:,i,j) = \sum_{r=1}^{R} \mathbf{y}_i(r) \cdot \mathbf{y}_j(r) \cdot \mathbf{z}_r \otimes \mathbf{z}_r$. There are two cases: (1) if i and j belong to the same cluster r, $\mathcal{X}(:,:,i,j) = \mathbf{z}_r \circ \mathbf{z}_r$; (2) if i and j belong to different clusters, $\mathcal{X}(:,:,i,j) = \mathbf{0}$. If $\text{mode} > 4$, the cases are very much the same as $\text{mode} = 4$.

A.2. Proof of Theorem 2

Theorem 2. *Let* $\mathbf{a} \in \mathbb{R}^m$ *be a given vector such that its elements* $a_1 \geq a_2 \geq \cdots \geq a_m$. *Consider the following problem:*

$$\mathbf{x}^* = \arg\min \|\mathbf{x} - \mathbf{a}\| + \rho(\mathbf{e}^T\mathbf{x})^2, \text{ s.t. } \mathbf{x} \in \{0,1\}^m, \tag{2}$$

where $\mathbf{e}$ *is an all-one vector. The components of* $\mathbf{x}^*$ *is also in a descending order and there exists an unique integer* $0 \leq s \leq m$ *such that* $x_i^* = 1$ *for* $i \leq s$ *and* $x_i^* = 0$ *for* $i > s$.

Proof. It is clear that if $0 \geq a_i \geq \cdots \geq a_n$ for some i, then for the optimal solution $\mathbf{x}^*$, $x_n^* = \cdots = x_i^* = 0$. Thus without loss of generality, we assume that $a_n \geq 0$. We first prove $\mathbf{x}^*$ must also be arranged in a descending order by contradiction. Suppose for some $i < j$, we have that $x_i^* < x_j^*$. Consider $\hat{\mathbf{x}}$ such that $\hat{x}_k = x_k^*$ if $k \neq i, j$ and $\hat{x}_i = x_j^*, \hat{x}_j = x_i^*$. Denote the objective function as $f(\mathbf{x})$. Now

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) = 2(a_i - a_j)(x_i^* - x_j^*) \leq 0. \tag{3}$$

Case (i): $a_i > a_j$. We get $f(\hat{\mathbf{x}}) < f(\mathbf{x}^*)$ and this contradicts the fact that $\mathbf{x}^*$ is a minimizer; Case (ii): $a_i = a_j$. We get $f(\hat{\mathbf{x}}) = f(\mathbf{x}^*)$, and $\hat{\mathbf{x}}$ is also an optimal solution with $\hat{x}_i > \hat{x}_j$. Note that $\hat{\mathbf{x}}$ is also in a descending order in this case. The above result implied that there exists an integer $0 \leq s \leq m$ such that $x_i^* = 1$ for $i \leq s$ and $x_i^* = 0$ for $i > s$. Hence the minimum objective value is given by

$$f(\mathbf{x}^*) = (\|\mathbf{a}\|^2 + s + \rho s) - 2\sum_{i=1}^{s} a_i =: g(s). \tag{4}$$

Note that since $a_1 \geq a_2 \geq \cdots \geq a_m > 0$, the function $\sum_{i=1}^{s} a_i$ looks like an increasing concave function as a function of $s = 0, \ldots, m$. On the other hand, $(\|\mathbf{a}\|^2 + s + \rho s)$ looks like an increasing convex function as a function of $s = 0, \ldots, m$. Thus, we find the value s such that

$$g(s) - g(s-1) \leq 0, g(s) - g(s+1) \leq 0. \tag{5}$$

Hence the solution $\mathbf{x}^*$ can be found analytically. Note that $g(0), ..., g(m)$ can be computed in $O(m)$ operations, and the minimum value $\min\{g(0), ..., g(m)\}$ can be found in $O(m)$ operations.

*S. Yang worked on this project as a research engineer at NUS.

[1]Since $\mathcal{X}$ is supersymmetric, i can be at any position and $\mathcal{X}(:,:,i) = \mathcal{X}(:,i,:) = \mathcal{X}(i,:,:)$. Similar results hold when $\text{mode} > 3$.

B. The algorithmic details

B.1. ADMM

The problem we want to solve is:

$$\begin{aligned} \min \ & -\langle \mathcal{A}, \mathcal{G} \rangle + \lambda \text{rank}(\widehat{\mathbf{G}}) + \gamma \|\widehat{\mathbf{G}}\|_0, \\ \text{s.t.} \ & \mathcal{G} \in [0,1]^{N \times N \times \ldots \times N}, \text{diag}(\mathcal{G}) = 1, \\ & \mathbf{G}_i \in \mathbf{S}_+, \text{rank}(\mathbf{G}_i) \leq 1, i = 1, 2, \ldots n. \end{aligned} \tag{12}$$

For ease of representation, we let $\mathcal{W} = -\mathcal{A}$ and unfold it in the same form of $\widehat{\mathbf{G}}$ as $\widehat{\mathbf{W}}$. We solve (12) by Alternating Direction Method of Multipliers(ADMM) [1] with three block variables $\widehat{\mathbf{G}}$, $\widehat{\mathbf{H}}$ and $\{\mathbf{J}_i\}_{i=1}^n$:

$$\begin{aligned} \min \ & \langle \widehat{\mathbf{W}}, \widehat{\mathbf{G}} \rangle + \lambda \text{rank}(\widehat{\mathbf{G}}) + \gamma \|\widehat{\mathbf{H}}\|_0 + g(\widehat{\mathbf{H}}), \\ \text{s.t.} \ & \text{unfolding\_diag}(\widehat{\mathbf{H}}) = 1, \widehat{\mathbf{G}} = \widehat{\mathbf{H}}, \widehat{\mathbf{G}} = \widehat{\mathbf{J}}, \\ & \mathbf{J}_i \in \mathbf{S}_+, \text{rank}(\mathbf{J}_i) \leq 1, i = 1, 2, \ldots n, \end{aligned} \tag{13}$$

where $\widehat{\mathbf{J}} = [\text{vec}(\mathbf{J}_1)\ \text{vec}(\mathbf{J}_2) \ldots \text{vec}(\mathbf{J}_n)]$, g is the indicator function of the convex set $[0,1]^{NN \times n}$, which returns 0 if it is in the set, ∞ otherwise, and unfolding_diag$(\cdot)$ are those entries of the unfolded form $\widehat{\mathbf{H}}$ corresponding to the diagonal entries of the tensor. The overall framework is summarized in Algorithm 1.

Algorithm 1 Solving (12) by ADMM

Input: Negative affinity matrix $\mathcal{W}$, parameters λ and γ.

Initialize: $\widehat{\mathbf{G}} = \widehat{\mathbf{H}} = \widehat{\mathbf{J}} = \mathbf{Y}_1 = \mathbf{Y}_2 = \mathbf{0}_{NN \times n}$, $\mu = 10^6$, $\rho = 1.1$, $\mu_{\min} = 10^{-10}$ and $\epsilon = 10^{-8}$.

while not converged **do**

Step 1 Fix the others and update $\widehat{\mathbf{G}}$:

$$\min \ \|\widehat{\mathbf{G}} - \frac{1}{2}(\widehat{\mathbf{H}} + \widehat{\mathbf{J}} - \mu(\widehat{\mathbf{W}} + \mathbf{Y}_1 + \mathbf{Y}_2))\|_F^2 + \lambda\mu \text{rank}(\widehat{\mathbf{G}}). \tag{6}$$

Step 2 Fix the others and update $\widehat{\mathbf{H}}$:

$$\begin{aligned} \min \ & \|\widehat{\mathbf{H}} - (\widehat{\mathbf{G}} + \mu\mathbf{Y}_1)\|_F^2 + 2\mu\gamma\|\widehat{\mathbf{H}}\|_0 + g(\widehat{\mathbf{H}}), \\ \text{s.t.} \ & \text{unfolding\_diag}(\widehat{\mathbf{H}}) = 1. \end{aligned} \tag{7}$$

Step 3 Fix the others and update $\{\mathbf{J}_i\}_{i=1}^n$:

For $i = 1, 2, \ldots n$, solve

$$\begin{aligned} \min \ & \|\mathbf{J}_i - (\mathbf{G}_i + \mu\mathbf{Y}_{2i})\|_F^2, \\ \text{s.t.} \ & \mathbf{J}_i \in \mathbf{S}_+, \ \text{rank}(\mathbf{J}_i) \leq 1, \end{aligned} \tag{8}$$

where $\mathbf{G}_i$ is extracted from the i-th column of $\widehat{\mathbf{G}}$ and reorganized into a square matrix. Similarly $\mathbf{Y}_{2i}$ is extracted from the i-th column of $\mathbf{Y}_2$ and reorganized.

Step 4 Update the multipliers: $\mathbf{Y}_1 = \mathbf{Y}_1 + \frac{1}{\mu}(\widehat{\mathbf{G}} - \widehat{\mathbf{H}})$ and $\mathbf{Y}_2 = \mathbf{Y}_2 + \frac{1}{\mu}(\widehat{\mathbf{G}} - \widehat{\mathbf{J}})$.

Step 5 Update the parameter μ by $\mu = \max(\frac{\mu}{\rho}, \mu_{\min})$.

Step 6 Check the convergence conditions: $\|\widehat{\mathbf{G}} - \widehat{\mathbf{H}}\|_\infty \leq \epsilon$ and $\|\widehat{\mathbf{G}} - \widehat{\mathbf{J}}\|_\infty \leq \epsilon$.

end while

Algorithm 2 Solving (14) by Stochastic ADMM

Input: Negative affinity tensor $\mathcal{W}$, an initialization $\widehat{\mathbf{G}}^0$ and parameters γ.

Initialize: $\widehat{\mathbf{G}} = \widehat{\mathbf{H}} = \widehat{\mathbf{J}} = \widehat{\mathbf{G}}^0$, $\mathbf{Y}_1 = \mathbf{Y}_2 = \mathbf{0}_{NN \times R_0}$, $\mathbf{Y}_3 = \mathbf{0}_N$, $\mu = 1$, $\rho = 1.1$, $\mu_{\min} = 10^{-10}$, $k = 0$ and $\epsilon = 10^{-8}$.

while not converged **do**

$\eta_{k+1} = N\sqrt{\frac{R_0}{2(k+1)}}$.

Randomly extract/construct one slice from $\mathcal{W}$.

Step 1 Fix the others and update $\widehat{\mathbf{G}}$:

$$\begin{aligned} \min_{\widehat{\mathbf{G}}} \ & \|\widehat{\mathbf{G}} - \widehat{\mathbf{P}}\|_F^2, \\ \text{s.t.} \ & \mathbf{G}_j \in \mathbf{S}_+, \text{rank}(\mathbf{G}_j) \leq 1, \quad j = 1 \ldots R_0, \end{aligned} \tag{9}$$

where $\widehat{\mathbf{P}} = (\frac{1}{\mu}\widehat{\mathbf{H}} + \frac{1}{\mu}\widehat{\mathbf{J}} + \frac{1}{\eta_{k+1}}\widehat{\mathbf{G}}^k - \mathbf{Y}_1 - \mathbf{Y}_2 - \ell'_{k+1}(\widehat{\mathbf{G}}^k))/(\frac{2}{\mu} + \frac{1}{\eta_{k+1}})$.

Step 2 Fix the others and update $\widehat{\mathbf{H}}$:

$$\begin{aligned} \min_{\widehat{\mathbf{H}}} \ & \|\widehat{\mathbf{H}} - \widehat{\mathbf{Q}}\|_F^2 + 2\mu\gamma \sum_{j=1}^{R_0} ||\mathbf{H}_j||_0^2, \\ \text{s.t.} \ & \widehat{\mathbf{H}} \in \{0,1\}^{NN \times R_0}, \end{aligned} \tag{10}$$

where $\widehat{\mathbf{Q}} = \widehat{\mathbf{G}} + \mu\mathbf{Y}_1$.

Step 3 Fix the others and update $\widehat{\mathbf{J}}$:

$$\begin{aligned} \min_{\widehat{\mathbf{J}}} \ & \left\langle \mathbf{Y}_2, \widehat{\mathbf{G}} - \widehat{\mathbf{J}} \right\rangle + \frac{1}{2\mu}\|\widehat{\mathbf{G}} - \widehat{\mathbf{J}}\|_F^2 + \\ & \left\langle \mathbf{Y}_3, \mathbf{S}\widehat{\mathbf{J}}\mathbf{e}_{R_0} - \mathbf{e}_N \right\rangle + \frac{1}{2\mu}\|\mathbf{S}\widehat{\mathbf{J}}\mathbf{e}_{R_0} - \mathbf{e}_N\|_F^2 \end{aligned} \tag{11}$$

Step 4 Update the multipliers:

$\mathbf{Y}_1 = \mathbf{Y}_1 + \frac{1}{\mu}(\widehat{\mathbf{G}} - \widehat{\mathbf{H}})$,

$\mathbf{Y}_2 = \mathbf{Y}_2 + \frac{1}{\mu}(\widehat{\mathbf{G}} - \widehat{\mathbf{J}})$,

$\mathbf{Y}_3 = \mathbf{Y}_3 + \frac{1}{\mu}(\mathbf{S}\widehat{\mathbf{J}}\mathbf{e}_{R_0} - \mathbf{e}_N)$.

Step 5 **If** $k \geq 750$, update the parameter μ by $\mu = \max(\frac{\mu}{\rho}, \mu_{\min})$; **end if**

Step 6 Check the convergence conditions: $\|\widehat{\mathbf{G}} - \widehat{\mathbf{H}}\|_\infty \leq \epsilon$, $\|\widehat{\mathbf{G}} - \widehat{\mathbf{J}}\|_\infty \leq \epsilon$ and $\|\mathbf{S}\widehat{\mathbf{J}}\mathbf{e}_{R_0} - \mathbf{e}_N\|_\infty \leq \epsilon$ $k = k + 1$.

end while

B.2. Stochastic ADMM

The problem we want to solve is:

$$\begin{aligned} \min_{\widehat{\mathbf{G}}} \ & \frac{1}{n}\sum_{i=1}^{n} \min_{j=1}^{R_0} \langle \mathbf{W}_i, \mathbf{G}_j \rangle + \gamma \sum_{j=1}^{R_0} ||\mathbf{G}_j||_0^2, \\ \text{s.t.} \ & \widehat{\mathbf{G}} \in \{0,1\}^{NN \times R_0}, \mathbf{G}_j \in \mathbf{S}_+, \\ & \text{rank}(\mathbf{G}_j) \leq 1, \ j = 1, \ldots R_0, \mathbf{S}\widehat{\mathbf{G}}\mathbf{e}_{R_0} = \mathbf{e}_N, \end{aligned} \tag{14}$$

where $\mathbf{W}_i$ is a slice from $\mathcal{W}$, $\widehat{\mathbf{G}} = [vec(\mathbf{G}_1) \quad vec(\mathbf{G}_2) \dots vec(\mathbf{G}_{R_0})]$, $\mathbf{e}_{R_0}$ and $\mathbf{e}_N$ are all-one vectors of size R_0 and N respectively, and $\mathbf{S} \in \{0,1\}^{N \times NN}$ is a selection matrix with $\mathbf{S}(i,(N+1)i-N) = 1$, $i = 1, ..., N$ and other elements being 0.

For notational convenience, we denote the first term in the objective function as $f(\widehat{\mathbf{G}}) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\widehat{\mathbf{G}})$, and $\ell_i(\widehat{\mathbf{G}}) = \min_{j=1}^{R_0} \left\langle \widetilde{\mathbf{W}}_i^j, \widehat{\mathbf{G}} \right\rangle$, where $\widetilde{\mathbf{W}}_i^j \in \mathbb{R}^{NN \times R_0}$ with its j-th column given by $\mathrm{vec}(\mathbf{W}_i)$ and 0 elsewhere. Now the problem is in a form suited for optimization by the stochastic ADMM [3] in an online fashion. In particular, we randomly obtain one slice $\mathbf{W}_i$ from $\mathcal{W}$ for each iteration, solve the following constituent subproblems once, and then initiate the next iteration with a different slice from $\mathcal{W}$, repeating the above until convergence.

Again we introduce the intermediate variables $\widehat{\mathbf{H}}$ and $\widehat{\mathbf{J}}$ and solve

$$\begin{aligned} \min_{\widehat{\mathbf{G}}} \quad & \frac{1}{n}\sum_{i=1}^{n} \ell_i(\widehat{\mathbf{G}}) + \gamma \sum_{j=1}^{R_0} ||\mathbf{H}_j||_0^2, \\ \text{s.t.} \quad & \mathbf{G}_j \in \mathbf{S}_+, \widehat{\mathbf{G}} = \widehat{\mathbf{H}}, \widehat{\mathbf{G}} = \widehat{\mathbf{J}}, \widehat{\mathbf{H}} \in \{0,1\}^{NN \times R_0}, \\ & \mathrm{rank}(\mathbf{G}_j) \le 1, \quad j = 1, ...R_0, \mathbf{S}\widehat{\mathbf{J}}\mathbf{e}_{R_0} = \mathbf{e}_N, \end{aligned} \tag{15}$$

To obtain a good initialization for the algorithm, we can apply spectral clustering to the projected graph and oversegment the points into R_0 groups. The overall framework of the Stochastic ADMM is shown in Algorithm 2.

B.3. Complexity

In our algorithms, the most expensive part is usually the SVD decomposition due to the rank minimization. In Algorithm 1, the size of $\widehat{\mathbf{G}}$ is $N^2 \times N^{K-2}$. a) When $K = 3$, the complexity of (6) is $O(N^5)$; when $K > 3$, it is $O(N^{2K-2})$. b) It is $O(N^K)$ for (7), and c) $O(N^{K+1})$ for (8). Thus, solving (6) is the most expensive and sometimes it becomes intractable as the order increases to a large number. In this case, we have to resort to the stochastic version, in which $\widehat{\mathbf{G}}$ is of size $N^2 \times R_0$, where $R_0 \ll N^{K-2}$. In Algorithm 2, the complexities of solving the respective subproblems are as follows: a) $O(R_0 N^3)$ for (9), b) $O(N^2 log(N))$ for (10), and c) $O(R_0^3 N^6)$ for (11). As is evident, the complexity does not change as the tensor order increases. However, it does take more iterations (usually) for the stochastic version to converge. Based on our observation, Algorithm 1 usually needs about 300 iterations to converge, while Algorithm 2 needs about 900 iterations.

C. Multiple fundamental matrix fitting

In this subsection, we add an extra experiment and estimate multiple fundamental matrices on real images. Given the matched points in an image pair, usually 8 correspondences are needed to fit a fundamental matrix; a 9-th point is need to determine the goodness of the fitted model, meaning that the order in this problem is 9. Again, we only compare the stochastic ADMM version with the others and add LabelCost [2] into the evaluation.

Table 1. Multiple fundamental matrix fitting (F1-measure) on *AdelaideRMF dataset*

Method	Label-Cost	CExp-GH	CExp-SCAMS	SCAMSTA-SADMM
mean ± variance	0.7777± 0.1183	0.6421± 0.2031	0.6490± 0.2022	**0.8593 ± 0.1448**

In this experiment, the data also comes from the AdelaideRMF dataset [5]; it consists of 21 image pairs with matched points available for multiple fundamental matrix fitting. In each iteration of SCAMSTA-SADMM, we adopt the RCM sampling [4] to sample 7 correspondences, as $(9-2)$ samples are required to generate a slice. All the generated slices are also used to construct the projected 2D graph needed by other methods; using the same slices makes sure that the same amount of information are provided for each methods. We set the label cost to 10000 for all labels, parameters $\gamma = 8 \times 10^{-5}$ for SCAMSTA-SADMM, and $\lambda = 2$ and $\gamma = 0.16$ for SCAMS. The F1-measures averaged over 21 instances are reported in Table 1. Similarly, it is observed that our method outperforms the others significantly, being the only one with mean F1-measure greater than 0.8. Other thant that, it is noticeable that CExp-SCAMS performs very bad in this experiment, showing that the projected pairwise graph degrades significantly when the order increases.

References

[1] S. Boyd, N. Parikh, E. Chu, and B. Peleato. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 1(3):1–122, 2011. 2

[2] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *IJCV*, 96(1):1–27, 2012. 3

[3] H. Ouyang, N. He, L. Tran, and A. G. Gray. Stochastic alternating direction method of multipliers. In *ICML*, 2013. 3

[4] P. Purkait, T. Chin, H. Ackermann, and D. Suter. Clustering with hypergraphs: The case for large hyperedges. In *ECCV*, 2014. 3

[5] H. S. Wong, T. Chin, J. Yu, and D. Suter. Dynamic and hierarchical multi-structure geometric model fitting. In *ICCV*, 2011. 3