Supplementary Material

A flexible optimization framework for regularized matrix-tensor factorizations with linear couplings

Carla Schenker, Jeremy E. Cohen, Evrim Acar

October 30, 2020

1 Pairwise couplings

In many works related to CMTF, the coupling constraints are not expressed using a latent "consensus" variable $\mathbf{\Delta}$ but rather using direct, factor to factor "pairwise" coupling. A few natural questions, therefore, emerge:

- How are the consensus coupling and the pairwise coupling related? Is one more general than the other?
- Is one formulation better on the algorithmic side?

To avoid introducing heavy equations, we will tackle these questions for the particular cases of (1) two coupled tensors and (2) three coupled tensors, however generalization to more tensors is straightforward.

We answer the first question positively: consensus coupling is always more general than pairwise coupling, as the latter can always be recast into the former. We discuss various algorithmic advantages and disadvantages of the two approaches to answer the second question, and leave a thorough comparison for future works.

1.1 Two coupled tensors

Let us start with the simple case of two coupled tensors. A pairwise coupling may be written (with simplified notations) as

$$\mathbf{H}_1\mathbf{c}_1 = \mathbf{H}_2\mathbf{c}_2 \tag{17}$$

where $\mathbf{c}_{1,2}$ are vectorized matrices $\mathbf{C}_{1,2}$. We may recast (17) into the following consensus coupling

$$\mathbf{H}_1\mathbf{c}_1 = \delta = \mathbf{H}_2\mathbf{c}_2 \tag{18}$$

However, the converse is not always true. The following consensus coupling

$$\mathbf{c}_1 = \mathbf{H}_1\delta,\ \mathbf{c}_2 = \mathbf{H}_2\delta \tag{19}$$

can be cast as a pairwise coupling only if δ is identifiable in (19), *i.e.* if $\mathbf{M} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix}$ is full column rank. Then we have

$$\mathbf{c}_1 = \mathbf{H}_1\mathbf{M}^\dagger \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix},\ \mathbf{c}_2 = \mathbf{H}_2\mathbf{M}^\dagger \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}, \tag{20}$$

which is a quite unusual pairwise coupling.

Nevertheless, this condition is in fact exactly the one we impose in the next section of the Supplementary Material on the coupling matrices $\mathbf{H}_i$ (a similar reasoning holds for all special coupling cases studied in this paper), so that in practice we will only consider consensus couplings that are also pairwise couplings.

1.2 Three coupled tensors

We now suppose that three tensors are coupled. We start with a pairwise coupling

$$\mathbf{H}_{12}\mathbf{c}_1 = \mathbf{H}_{21}\mathbf{c}_2,\ \mathbf{H}_{13}\mathbf{c}_1 = \mathbf{H}_{31}\mathbf{c}_3,\ \mathbf{H}_{23}\mathbf{c}_2 = \mathbf{H}_{32}\mathbf{c}_3. \tag{21}$$

Introducing a consensus variable δ which has as many rows as there are coupling equations (here three blocks of such rows), we can equivalently rewrite (21)

$$\begin{bmatrix} \mathbf{H}_{12} \\ \mathbf{H}_{13} \\ 0 \end{bmatrix} \mathbf{c}_1 = \delta,\ \begin{bmatrix} \mathbf{H}_{21} \\ 0 \\ \mathbf{H}_{23} \end{bmatrix} \mathbf{c}_2 = \delta,\ \begin{bmatrix} 0 \\ \mathbf{H}_{31} \\ \mathbf{H}_{32} \end{bmatrix} \mathbf{c}_3 = \delta, \tag{22}$$

which is a consensus coupling formulation.

Again, it is not always possible to transform a consensus coupling into a pairwise coupling. However differently from (19), it is not sufficient than the stacked transformation matrices are full column rank, as pairwise coupling requires a relationship between pairs of factors rather than several linear relationships involving all the factors. To link all $\mathbf{c}_i$ and $\mathbf{c}_j$ in pairs, it is required (intuitively, but there may be weaker conditions) that for at least a single partition of pairs of indices, *e.g.* $[(1,2),(3,4),\ldots,(N-1,N)]$, every stacked matrix $M_{s_1 s_2}$ with s in that chain is full column rank. This way, we can pair factors according to that partition similarly to (20). Such conditions are harder than what we impose in this work for consensus couplings, and therefore *a priori* consensus couplings are strictly more general than pairwise couplings even given the restrictive conditions imposed in the next section.

1.3 Discussing algorithmic performance

The pairwise formulation involves fewer parameters. To be precise, if there are p coupling equations (counting for instance each row in (21)), then p additional parameters are required. If N tensors are involved, for the case of squared matrices $\mathbf{H}_{ij}$ of size $n \times n$ in a pairwise coupling involving all possible pairs such as (21), this means introducing exactly $nN(N-1)/2$ parameters, while in total the coupled matrices contain nN parameters. Therefore, it would tempting to conclude that the pairwise formulation, as N grows, is computationally more attractive.

However, the consensus formulation is what allows the AO-ADMM algorithm to compute the coupled factors $\mathbf{c}_i$ in parallel, therefore in principle speeding up the algorithm by a factor N. Again, introducing the consensus variable may drastically increase the number of parameters to estimate on the coupled mode, but also allows to use parallel computing to reduce the computational burden. At the end of the day we are facing a trade-off between memory and computing power, which we will study in future works. A very efficient implementation of the proposed framework may therefore possibly work with both consensus and pairwise couplings depending on the problem at hand.

It may be worth mentioning that in most applications we are aware of, the number of tensors N remains quite small. In that case, we observed no difference between pairwise couplings and consensus couplings, see for instance the comparisons in the Experiments section.

2 Restrictions on transformation matrices

In order for the coupled model to make sense, the transformation matrices $\mathbf{H}_{i,1}$ and $\mathbf{H}_{i,1}^{\Delta}$ should fulfil the following two properties, which will also ensure that the algorithm works as intended:

1. Given all coupled factor matrices $\{\mathbf{C}_{i,1}\}_{i \leq N}$ and transformation matrices, the consensus variable $\boldsymbol{\delta}_1$ should be uniquely defined via

$$\boldsymbol{\delta}_1 = \underset{\mathbf{z}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \mathbf{H}_{i,1} \operatorname{vec}(\mathbf{C}_{i,1}) - \mathbf{H}_{i,1}^{\Delta}\mathbf{z} + \boldsymbol{\mu}_{i,1_{(\delta)}} \right\|_2^2.$$

2. For any given $\boldsymbol{\Delta}_1$, there should exist at least one solution $\mathbf{C}_{i,1}$ to the equation

$$\mathbf{H}_{i,1} \operatorname{vec}(\mathbf{C}_{i,1}) = \mathbf{H}_{i,1}^{\Delta} \operatorname{vec}(\boldsymbol{\Delta}_1)$$

for each coupled tensor i.

This leads to some restrictions on the transformation matrices for the cases described in Section IV-A:

Case 2a: Here, the first property is always fulfilled. To ensure the second property, we require all transformation matrices $\tilde{\mathbf{H}}_{i,1}$ to be right-invertible, since we want the corresponding linear map to be surjective. It follows that $n_\Delta \leq \min_i n_{1_i}$, since all $\tilde{\mathbf{H}}_{i,1}$ are supposed to have linearly independent rows. Furthermore, we note that the solution $\mathbf{X}$ of the Sylvester equation $\mathbf{AX} + \mathbf{XB} = \mathbf{C}$ is unique if and only if $\mathbf{A}$ and $-\mathbf{B}$ have distinct spectra, which will almost always be the case for the matrices in this algorithm (equation (11)).

Case 2b: To ensure that $\boldsymbol{\Delta}_1$ is uniquely defined by the union of all coupled factor matrices $\{\mathbf{C}_{i,1}\}$, we require the matrix

$$\left[\tilde{\mathbf{H}}_{1,1}^{\Delta^T} | \tilde{\mathbf{H}}_{2,1}^{\Delta^T} | ... | \tilde{\mathbf{H}}_{N,1}^{\Delta^T}\right]$$

of size $n_{\Delta_1} \times \sum_i n_{1_i}$ to have rank n_{Δ_1}. This is sufficient for the invertibility of the matrix $\sum_i^N \left(\tilde{\mathbf{H}}_{i,1}^{\Delta^T} \tilde{\mathbf{H}}_{i,1}^{\Delta}\right)$, which is needed for the update of $\boldsymbol{\Delta}_1$.

Case 3a: Analogously to Case 2a, here we require all transformation matrices $\hat{\mathbf{H}}_{i,1}$ to be left-invertible. This implies the size restriction $R_{\Delta_1} \leq \min_i R_i$.

Case 3b: Here, similar to Case 2b, in order to ensure the uniqueness of $\boldsymbol{\Delta}_1$, we require the matrix

$$\left[\hat{\mathbf{H}}_{1,1}^{\Delta} | \hat{\mathbf{H}}_{2,1}^{\Delta} | ... | \hat{\mathbf{H}}_{N,1}^{\Delta}\right]$$

of size $R_{\Delta_1} \times \sum_i^N R_i$ to have rank R_{Δ_1}.

3 Loss function gradient

The derivative of $\mathcal{L}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{X}})$, where $\boldsymbol{\mathcal{X}} = [\![\mathbf{C}_{i,1}, \mathbf{C}_{i,2}, ..., \mathbf{C}_{i,D_i}]\!]$, with respect to the factor matrix $\mathbf{C}_{i,d}$ is given by

$$\frac{\partial \mathcal{L}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{X}})}{\partial \mathbf{C}_{i,d}} = \boldsymbol{\mathcal{Y}}_{[d]} \mathbf{M}_{i,d},$$

where $\boldsymbol{\mathcal{Y}}_{[d]}$ is the mode d unfolding of the tensor $\boldsymbol{\mathcal{Y}}$ defined by

$$y_j = \frac{\partial \ell(t_j, x_j)}{\partial x_j},$$

where ℓ is the element-wise loss function , see [1] for a detailed derivation of this. Thus, one can take advantage of efficient implementations of the matricized tensor times Khatri-Rao product for the computation of the gradient. To obtain the complete gradient of the objective function in line 3 of Algorithm 2, the gradient of the terms related to coupling and constraints have to be added, which is straightforward, but changes slightly with the different linear coupling options and is omitted here.

4 Additional plots and experiments

Experiment 1a. This experiment is described in Section VI-B 1a. Figure 13 shows the distribution of computing times after reaching a relative tolerance of 10^{-6} difference in function value. After reaching this tolerance, FMS has typically converged for this problem.

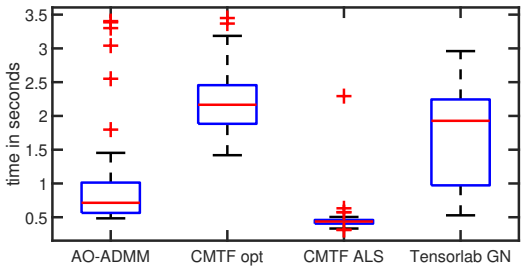


Figure 13: Experiment 1a with coupling Case 1 and congruence 0.5 in all factor matrices: Boxplots showing distribution of computing times of different algorithms after reaching a relative tolerance of 10^{-6}.

Experiment 1b. This experiment is described in Section VI-B 1b. Figure 14 shows the distribution of computing times after reaching a relative tolerance of 10^{-8} difference in function value. Since this problem is more difficult than the previous one, it typically takes longer to find the correct factors, therefore the tolerance is chosen as 10^{-8}.

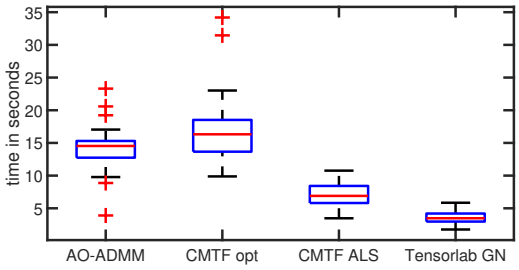
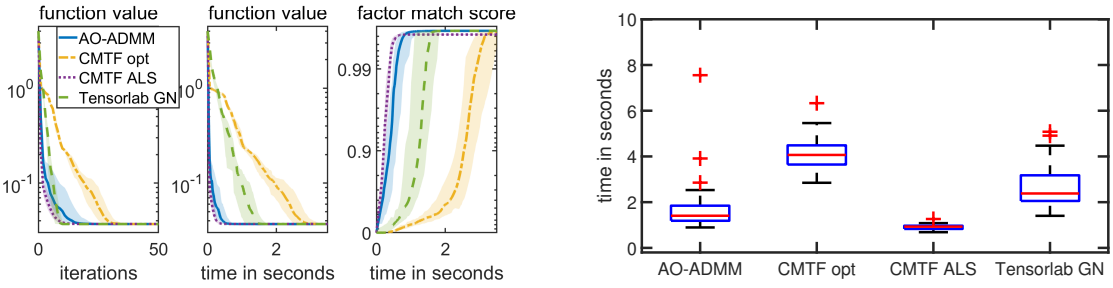


Figure 14: Experiment 1b with coupling Case 1 and congruence 0.9 in all factor matrices: Boxplots showing distribution of computing times of different algorithms after reaching a relative tolerance of 10^{-8}.

Experiment 1c. Unbalanced Data Size. In this additional experiment, we study the impact of unbalanced dimensions in the data sets. The setting corresponds to Experiment 1a as described in Section VI-B 1a. The tensor has now a size of $40 \times 5000 \times 50$ and the matrix is of size 40×6000. Factors have a congruence of 0.5. As shown in Figure 15, AO-ADMM performs as well as CMTF-ALS and all methods can find the true factors.

Table 5: Failed runs experiments 1c, 1d

	AO-ADMM		CMTF opt		CMTF ALS		TL GN	
Exp.	all init.	best runs	all init.	best runs	all init.	best runs	all init.	best runs
1c	23	0	1	0	10	0	0	0
1d	45	0	0	0	0	0	3	0



(a) Median and quartiles of function values and FMS for different algorithms.

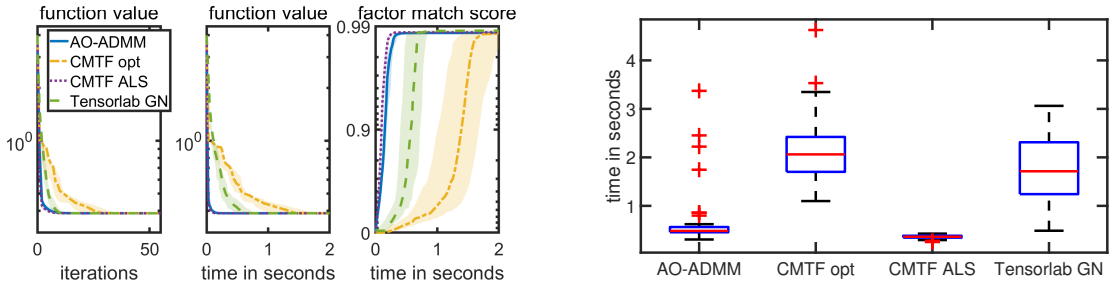
(b) Boxplots showing distribution of computing times of different algorithms after reaching a relative tolerance of 10^{-6}.

Figure 15: Experiment 1c with coupling Case 1 and congruence 0.5 and unbalanced sizes.

Experiment 1d. Higher Noise Level. In this additional experiment we study the effect of more noise in the data. The setting corresponds to Experiment 1a as described in Section VI-B 1a, except that the noise level is increased to 0.8. This means that a noise tensor $\boldsymbol{\mathcal{N}}_i$ is added to the ground truth tensor $\boldsymbol{\mathcal{T}}_i$ as follows,

$$\boldsymbol{\mathcal{T}}_i = \boldsymbol{\mathcal{X}}_i + 0.8(\| \boldsymbol{\mathcal{X}}_i \|_F \,/\, \| \boldsymbol{\mathcal{N}}_i \|_F)\boldsymbol{\mathcal{N}}_i.$$

As shown in Figure 16, the overall achieved factor match score has decreased due to the noise and AO-ADMM performs as well as CMTF-ALS.



(a) Median and quartiles of function values and FMS for different algorithms.

(b) Boxplots showing distribution of computing times of different algorithms after reaching a relative tolerance of 10^{-6}.

Figure 16: Experiment 1d with coupling Case 1 and congruence 0.5 and noise level 0.8.

Experiment 2. Here, we study the effect of different number of maximum inner iterations m ($m = 3, 5, 10, 20$) in the AO-ADMM algorithm 2 in the setting of experiment 2 (Section VI-B 2) with non-negativity constraints. As shown in Figure 17, a maximum of 3 inner iterations seems to be to small, but there is almost no difference in convergence speed for $m = 5, 10$ and 20.

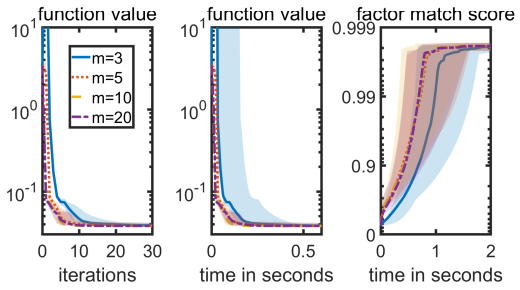


Figure 17: Experiment 2a: Median and minimum/maximum function values and FMS for AO-ADMM with different number of maximum inner iterations m.

Experiment 2b. Unbalanced Data Size. We study here the impact of unbalanced dimensions in data sets together with non-negativity constraints, in the setting of experiment 2 (Section VI-B 2). The tensor has a size of $40 \times 5000 \times 50$ and the matrix is of size 40×6000. Figure 18 shows that AO-ADMM performs very well in this setting.

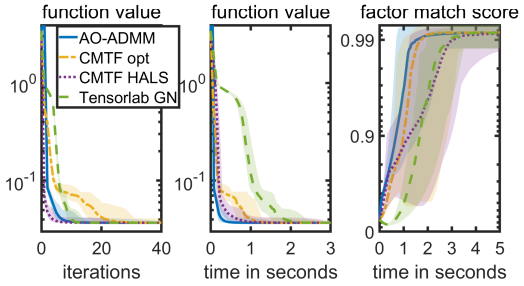


Figure 18: Experiment 2b with coupling Case 1, nonnegativity constraints and unbalanced sizes: Median and minimum/maximum function values and FMS for different algorithms.

Experiment 2c. Different number of components. Here, we compare the different algorithms on data sets with increasing ranks together with non-negativity constraints, in the setting of experiment 2 (Section VI-B 2). Figures 19, 20, and 21 show the comparisons for $R = 5$, $R = 10$ and $R = 15$, respectively. While the accuracy of HALS decreases with increasing rank, AO-ADMM continues to perform well even for rank $R = 15$.

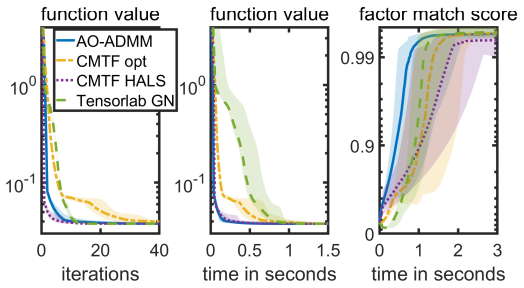


Figure 19: Experiment 2c with coupling Case 1, nonnegativity constraints and rank R=5: Median and minimum/maximum function values and FMS for different algorithms.

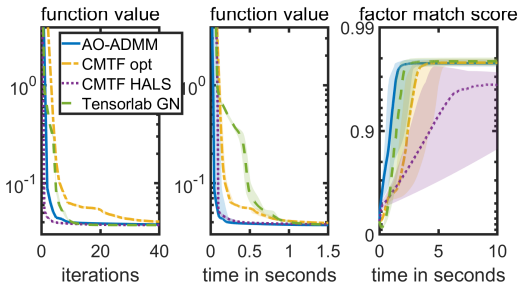


Figure 20: Experiment 2c with coupling Case 1, nonnegativity constraints and rank R=10: Median and minimum/maximum function values and FMS for different algorithms.

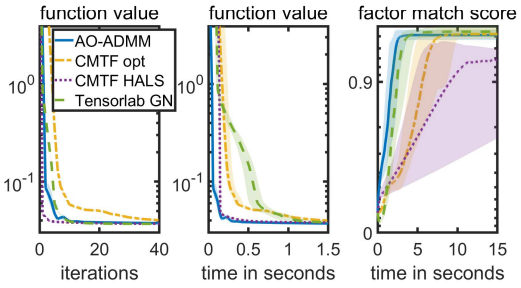


Figure 21: Experiment 2c with coupling Case 1, nonnegativity constraints and rank R=15: Median and minimum/maximum function values and FMS for different algorithms.

Experiment 3. This experiment is described in Section VI-B 3. Figure 22 shows the distribution of computing times after reaching a relative tolerance of 10^{-6} difference in function value.

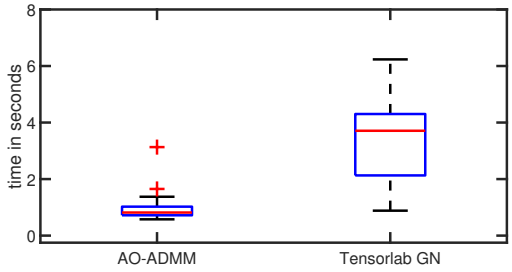


Figure 22: Experiment 3: Boxplots showing distribution of computing times of AO-ADMM and Tensorlab GN after reaching a rel. tol. of 10^{-6}.

Experiment 4. This experiment is described in Section VI-B 4. Figure 23 shows the distribution of computing times after reaching a relative tolerance of 10^{-8} difference in function value.

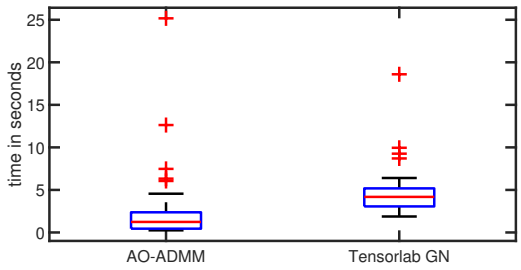


Figure 23: Experiment 4: Boxplots showing distribution of computing times of AO-ADMM and Tensorlab GN after reaching a rel. tol. of 10^{-8}.

Experiment 5. This experiment is described in Section VI-B 5. Figure 24 shows the distribution of computing times after reaching a relative tolerance of 10^{-6} difference in function value.

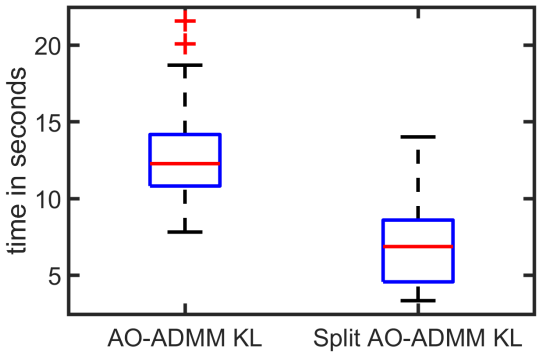


Figure 24: Experiment 5: Boxplots showing distribution of computing times of different algorithms after reaching a rel. tol. of 10^{-6}.

4.1 Hyperspectral super-resolution

In addition to the residual maps, we show here the various endmembers and abundances estimated by the baseline and the proposed AO-ADMM once a RMSE of 0.0144 has been reached for both methods. The results are qualitatively extremely similar.

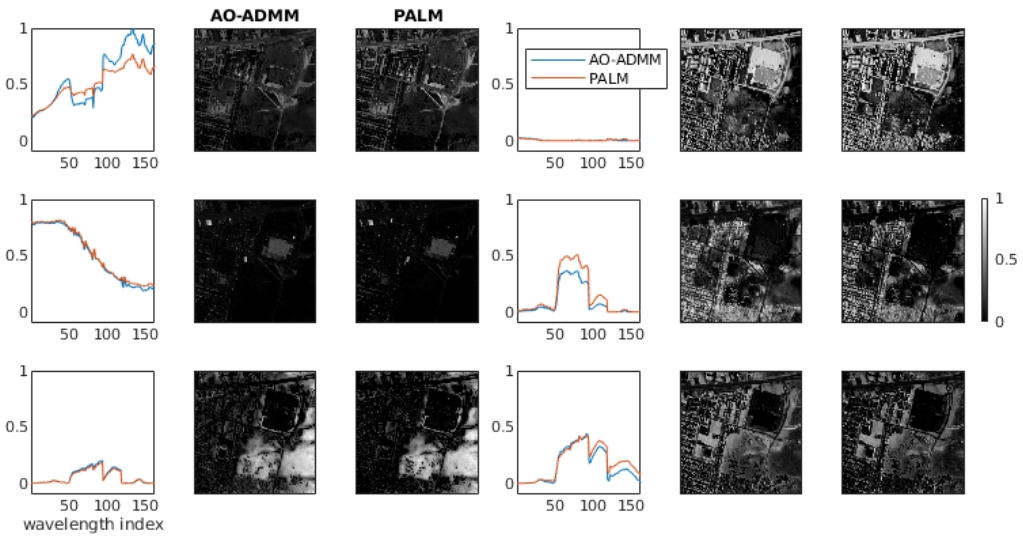


Figure 25: Endmembers and abundances estimated by PALM and AO-ADMM from the degraded images. The abundances are normalized by their maximal value, which is pulled in the endmembers. We can distinguish the following materials (from top left to bottom right): soil, rooftops, grass 1, asphalt + rooftops, trees, grass 2

5 Extension to flexible couplings

So far, we have only assumed *hard* linear coupling structures, $\mathbf{H}_{i,1}\,\mathrm{vec}(\mathbf{C}_{i,1}) = \mathbf{H}_{i,1}^{\Delta}\,\mathrm{vec}(\mathbf{\Delta}_1)$, in the sense of exact equality. This is a very strict assumption and is oftentimes relaxed to allow for small variations between coupled factors from different datasets [2–4], referred to as *soft* or *flexible* coupling. Here, we derive a variant of our algorithmic framework which covers the case when the coupling relationship between the factors $\mathbf{C}_{i,1}$ is described by a joint density probability function, as discussed in [2]. For example, define the joint density of $\mathbf{C}_{i,1}$ as

$$p(\mathbf{C}_{1,1},\ldots,\mathbf{C}_{n,1}) = p(\mathbf{C}_{1,1},\ldots,\mathbf{C}_{N,1}|\mathbf{\Delta}_1)p(\mathbf{\Delta}_1) = p(\mathbf{\Delta}_1)\prod_{i=1}^{N} p(\mathbf{C}_{i,1}|\mathbf{\Delta}_1)$$

where $p(\mathbf{\Delta}_1)$ is a prior on the consensus variable $\mathbf{\Delta}_1$. Conditional independence of $\mathbf{C}_{i,1}$ is assumed. Then, using a MAP estimator, the optimization problem for mode 1 with flexible couplings takes the following form:

$$\underset{\{\mathbf{C}_{i,1}\}_{i\leq N},\mathbf{\Delta}_1}{\operatorname{argmin}} \sum_{i=1}^{N} \left[w_i\mathcal{L}_i\left(\boldsymbol{\mathcal{T}}_i, [\![\mathbf{C}_{i,1},\mathbf{C}_{i,2},...,\mathbf{C}_{i,D_i}]\!]\right) + g_{i,1}\left(\mathbf{C}_{i,1}\right) - \log(p(\mathbf{C}_{i,1}|\mathbf{\Delta}_1))\right] - \log(p(\mathbf{\Delta}_1)) \tag{23}$$

Note that problem (23) reduces to the following problem when $p(\mathbf{\Delta}_1)$ is flat, and $p(\mathbf{C}_{i,1}|\mathbf{\Delta}_1)$ is Gaussian with standard deviation ν_i:

$$\underset{\{\mathbf{C}_{i,1}\}_{i\leq N},\mathbf{\Delta}_1}{\operatorname{argmin}} \sum_{i=1}^{N} \left[w_i\mathcal{L}_i\left(\boldsymbol{\mathcal{T}}_i, [\![\mathbf{C}_{i,1},\mathbf{C}_{i,2},...,\mathbf{C}_{i,D_i}]\!]\right) + g_{i,1}\left(\mathbf{C}_{i,1}\right) + \frac{1}{\nu_i^2}\,\|\,\mathbf{C}_{i,1} - \mathbf{\Delta}_1\,\|_F^2\right]$$

We propose to use ADMM to solve problem (23) similar to before, but with a different splitting of variables. Variables $\mathbf{C}_{i,1}$ are split twice, once for handling the set constraint, and once for accounting for the coupling function. Additionally, hidden variable $\mathbf{\Delta}_1$ is split to account for

its prior. This yields the following optimization problem:

$$\begin{aligned} \underset{\{\mathbf{C}_{i,1},\mathbf{Z}_{i,1},\mathbf{B}_{i,1}\}_{i\leq N},\boldsymbol{\Delta}_1,\mathbf{Z}_{\boldsymbol{\Delta}_1}}{\operatorname{argmin}} \quad & \sum_{i=1}^{N} \left[w_i \mathcal{L}_i \left(\boldsymbol{\mathcal{T}}_i, [\![\mathbf{C}_{i,1}, \mathbf{C}_{i,2}, ..., \mathbf{C}_{i,D_i}]\!] \right) + g_{i,1} \left(\mathbf{Z}_{i,1} \right) + f_{i,1} \left(\mathbf{B}_{i,1}, \boldsymbol{\Delta}_1 \right) \right] + h \left(\mathbf{Z}_{\boldsymbol{\Delta}_1} \right) \\ \text{s.t.} \quad & \mathbf{C}_{i,1} = \mathbf{B}_{i,1} \\ & \mathbf{C}_{i,1} = \mathbf{Z}_{i,1} \\ & \boldsymbol{\Delta}_1 = \mathbf{Z}_{\boldsymbol{\Delta}_1} \end{aligned} \tag{24}$$

where $f_{i,1}(\cdot,\cdot) = -\log(p(\cdot|\cdot))$ and $h(\cdot) = -\log(p(\cdot))$ denote the negative log-likelihood of the coupling probability density function and the prior distribution on $\boldsymbol{\Delta}_1$, respectively. Applying ADMM to problem (24) leads to Algorithm 4 presented below. Also here, special forms of linear coupling transformations can be incorporated, for which we refer to [2].

Algorithm 4 ADMM for subproblem w.r.t. mode 1 of regularized CPD with flexible couplings

while convergence criterion is not met **do**
  for $i = 1, ..., N$ **do**

$$\mathbf{C}_{i,1}^{(k+1)} = \underset{\mathbf{X}}{\operatorname{argmin}}\ w_i \mathcal{L}_i \left(\boldsymbol{\mathcal{T}}_i, [\![\mathbf{X}, \mathbf{C}_{i,2}, ..., \mathbf{C}_{i,D_i}]\!] \right) + \frac{\rho}{2} \left(\left\| \mathbf{X} - \mathbf{Z}_{i,1}^{(k)} + \boldsymbol{\mu}_{i,1}{}_{(z)}^{(k)} \right\|_F^2 + \left\| \mathbf{X} - \mathbf{B}_{i,1}^{(k)} + \boldsymbol{\mu}_{i,1}{}_{(B)}^{(k)} \right\|_F^2 \right)$$

  end for
  $\boldsymbol{\Delta}_1^{(k+1)} = \operatorname{prox}_{\frac{1}{\rho}\sum_{i=1}^{N} f_{i,1}(\mathbf{B}_{i,1}^{(k)},\cdot)}(\mathbf{Z}_{\Delta_1}^{(k)} - \mu_{\Delta_1}^{(k)})$
  for $i = 1, ..., N$ **do**
    $\mathbf{Z}_{i,1}^{(k+1)} = \operatorname{prox}_{\frac{1}{\rho} g_{i,1}} \left(\mathbf{C}_{i,1}^{(k+1)} + \boldsymbol{\mu}_{i,1}{}_{(z)}^{(k)} \right)$
    $\mathbf{B}_{i,1}^{(k+1)} = \operatorname{prox}_{\frac{1}{\rho} f_{i,1}(\cdot,\boldsymbol{\Delta}_1^{(k+1)})} \left(\mathbf{C}_{i,1}^{(k+1)} + \boldsymbol{\mu}_{i,1}{}_{(B)}^{(k)} \right)$
    $\boldsymbol{\mu}_{i,1}{}_{(z)}^{(k+1)} = \boldsymbol{\mu}_{i,1}{}_{(z)}^{(k)} + \mathbf{C}_{i,1}^{(k+1)} - \mathbf{Z}_{i,1}^{(k+1)}$
    $\boldsymbol{\mu}_{i,1}{}_{(B)}^{(k+1)} = \boldsymbol{\mu}_{i,1}{}_{(B)}^{(k)} + \mathbf{C}_{i,1}^{(k+1)} - \mathbf{B}_{i,1}^{(k+1)}$
    $\boldsymbol{\mu}_{i,1}{}_{(\Delta)}^{(k+1)} = \boldsymbol{\mu}_{i,1}{}_{(\Delta)}^{(k)} + \boldsymbol{\Delta}_1^{(k+1)} - \mathbf{Z}_{\Delta_1}^{(k+1)}$
  end for
  $k = k + 1$
end while

References

[1] D. Hong, T. G. Kolda, and J. A. Duersch, "Generalized canonical polyadic tensor decomposition," *SIAM Review*, vol. 62, no. 1, pp. 133–163, 2020.

[2] R. Cabral Farias, J. E. Cohen, and P. Comon, "Exploring multimodal data fusion through joint decompositions with flexible couplings," *IEEE Trans. Sig. Proc.*, vol. 64, no. 18, pp. 4830–4844, Sep. 2016.

[3] B. Rivet, M. Duda, A. Guérin-Dugué, C. Jutten, and P. Comon, "Multimodal approach to estimate the ocular movements during EEG recordings: a coupled tensor factorization method," in *EMBC'15*, 2015, pp. 6983–6986.

[4] C. Chatzichristos, M. Davies, J. Escudero, E. Kofidis, and S. Theodoridis, "Fusion of EEG and fMRI via soft coupled tensor decompositions," *EUSIPCO'18*, pp. 56–60, 2018.