ESTIMATION OF LOW-RANK TENSORS VIA CONVEX OPTIMIZATION[*]

RYOTA TOMIOKA[†], KOHEI HAYASHI[‡], AND HISASHI KASHIMA[†]

Abstract. In this paper, we propose three approaches for the estimation of the Tucker decomposition of multi-way arrays (tensors) from partial observations. All approaches are formulated as convex minimization problems. Therefore, the minimum is guaranteed to be unique. The proposed approaches can automatically estimate the number of factors (rank) through the optimization. Thus, there is no need to specify the rank beforehand. The key technique we employ is the trace norm regularization, which is a popular approach for the estimation of low-rank matrices. In addition, we propose a simple heuristic to improve the interpretability of the obtained factorization. The advantages and disadvantages of three proposed approaches are demonstrated through numerical experiments on both synthetic and real world datasets. We show that the proposed convex optimization based approaches are more accurate in predictive performance, faster, and more reliable in recovering a known multilinear structure than conventional approaches.

1. Introduction.

Multi-way data analysis have recently become increasingly popular supported by modern computational power [23, 37]. Originally developed in the field of psychometrics and chemometrics, its applications can now also be found in signal processing (for example, for independent component analysis) [14], neuroscience [29], and data mining [28]. Decomposition of multi-way arrays (or tensors) into small number of factors have been one of the main concerns in multi-way data analysis, because interpreting the original multi-way data is often impossible. There are two popular models for tensor decomposition, namely the Tucker decomposition [44, 13] and the CANDECOMP/PARAFAC (CP) decomposition [11, 19]. In both cases, conventionally the estimation procedures have been formulated as non-convex optimization problems, which are in general only guaranteed to converge locally and could potentially suffer from poor local minima. Moreover, a popular approach for Tucker decomposition known as the higher order orthogonal iteration (HOOI) may converge to a stationary point that is not even a local minimizer [15].

Recently, convex formulations for the estimation of low-rank *matrix*, which is a special case of tensor, have been intensively studied. After the pioneering work of Fazel et al. [16], convex optimization has been used for collaborative filtering [38], multi-task learning [3], and classification over matrices [40]. In addition, there are theoretical developments that (under some conditions) guarantee *perfect reconstruction* of a low-rank matrix from partial measurements via convex estimation [10, 32]. The key idea here is to replace the rank of a matrix (a non-convex function) by the so-called trace norm (also known as the nuclear norm) of the matrix. One goal of this paper is to extend the trace-norm regularization for more than two dimensions. There have recently been related work by Liu et al. [27] and Signoretto et al. [36], which correspond to one of the proposed approaches in the current paper.

In this paper, we propose three formulations for the estimation of low rank tensors. The first approach is called "as a matrix" and estimates the low-rank matrix that is obtained by *unfolding* (or matricizing) the tensor to be estimated; thus this approach

[*] This work was supported by MEXT KAKENHI 22700138, 80545583, JST PRESTO, and NTT Communication Science Laboratories.

[†] Department of Mathematical Informatics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan ({tomioka,kashima}@mist.i.u-tokyo.ac.jp)

[‡] Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0192, Japan (kohei-h@is.naist.jp) ,

basically treats the unknown tensor as a matrix and only works if the tensor is low-rank in the mode used for the estimation. The second approach called "constraint" extends the first approach by incorporating the trace norm penalties with respect to all modes simultaneously. Therefore, there is no arbitrariness in choosing a single mode to work with. However, all modes being simultaneously low-rank might be a strong assumption. The third approach called "mixture" relaxes the assumption by using a mixture of K tensors, where K is the number of modes of the tensor. Each tensor is regularized to be low-rank in each mode.

We apply the above three approaches to the reconstruction of partially observed tensors. In both synthetic and real-world datasets, we show the superior predictive performance of the proposed approaches against conventional expectation maximization (EM) based estimation of Tucker decomposition model. We also demonstrate the effectiveness of a heuristic to improve the interpretability of the core tensor obtained by the proposed approaches on the amino acid fluorescence dataset.

This paper is structured as follows. In the next section, we first review the matrix rank and its relation to the trace norm. Then we review the definition of tensor mode-k rank, which suggests that a low rank tensor *is* a low rank matrix when appropriately unfolded. In Section 3, we propose three approaches to extend the trace-norm regularization for the estimation of low-rank tensors. In Section 4, we show that the optimization problems associated to the proposed extensions can be solved efficiently by the alternating direction method of multipliers [17]. In Section 5, we show through numerical experiments that one of the proposed approaches can recover a partly observed low-rank tensor almost perfectly from smaller fraction of observations compared to the conventional EM-based Tucker decomposition algorithm. The proposed algorithm shows a sharp threshold behaviour from a poor fit to a nearly perfect fit; we numerically show that the fraction of samples at the threshold is roughly proportional to the sum of the k-ranks of the underlying tensor when the tensor dimension is fixed. Finally we summarize the paper in Section 6. Earlier version of this manuscript appeared in NIPS2010 workshop "Tensors, Kernels, and Machine Learning".

2. Low rank matrix and tensor.

In this section, we first discuss the connection between the rank of a matrix and the trace-norm regularization. Then we review the CP and the Tucker decomposition and the notions of tensor rank connected to them.

2.1. Rank of a matrix and the trace norm.

The rank r of an $R \times C$ matrix $\boldsymbol{X}$ can be defined as the number of nonzero singular values of $\boldsymbol{X}$. Here, the singular-value decomposition (SVD) of $\boldsymbol{X}$ is written as follows:

$$\boldsymbol{X} = \boldsymbol{U}\mathrm{diag}(\sigma_1(\boldsymbol{X}), \sigma_2(\boldsymbol{X}), \ldots, \sigma_r(\boldsymbol{X}))\boldsymbol{V}^\top, \tag{2.1}$$

where $\boldsymbol{U} \in \mathbb{R}^{R\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{C\times r}$ are orthogonal matrices, and $\sigma_j(\boldsymbol{X})$ is the jth largest singular-value of $\boldsymbol{X}$. The matrix $\boldsymbol{X}$ is called *low-rank* if the rank r is less than $\min(R, C)$. Unfortunately, the rank of a matrix is a nonconvex function, and the direct minimization of rank or solving a rank-constrained problem is an NP-hard problem [32].

The trace norm is known to be the tightest convex lower bound of matrix rank [32] (see Fig. 2.1) and is defined as the linear sum of singular values as follows:

$$\|\boldsymbol{X}\|_* = \sum_{j=1}^{r} \sigma_j(\boldsymbol{X}).$$

Intuitively, the trace norm plays the role of the ℓ_1-norm in the subset selection problem [39], for the estimation of low-rank matrix[1]. The convexity of the above function follows from the fact that it is the dual norm of the spectral norm $\|\cdot\|$ (see [6, Section A.1.6]). Since it is a norm, the trace norm $\|\cdot\|_*$ is a convex function. The non-differentiability of the trace norm at the origin promotes many singular values of $\boldsymbol{X}$ to be zero when used as a regularization term. In fact, the following minimization problem has an analytic solution known as the spectral soft-thresholding operator (see [8]):

$$\underset{\boldsymbol{X}}{\operatorname{argmin}} \quad \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Y}\|_{\mathrm{Fro}}^2 + \lambda\|\boldsymbol{X}\|_*, \tag{2.2}$$

where $\|\cdot\|_{\mathrm{Fro}}$ is the Frobenius norm, and $\lambda > 0$ is a regularization constant. The spectral soft-thresholding operation can be considered as a shrinkage operation on the singular values and is defined as follows:

$$\mathrm{prox}_\lambda^{\mathrm{tr}}(\boldsymbol{Y}) = \boldsymbol{U} \max(\boldsymbol{S} - \lambda, 0)\boldsymbol{V}^\top, \tag{2.3}$$

where $\boldsymbol{Y} = \boldsymbol{USV}^\top$ is the SVD of the input matrix $\boldsymbol{Y}$, and the max operation is taken element-wise. We can see that the spectral soft-thresholding operation truncates the singular-values of the input matrix $\boldsymbol{Y}$ smaller than λ to zero, thus the resulting matrix $\boldsymbol{X}$ is usually low-rank. See also [42] for the derivation.

For the recovery of partially observed low-rank matrix, some theoretical guarantees have recently been developed. Candès and Recht [10] showed that in the noiseless case, $O(n^{6/5} r \log(n))$ samples are enough to perfectly recover the matrix under uniform sampling if the rank r is not too large, where $n = \max(R, C)$.

2.2. Rank of a tensor. For higher order tensors, there are several definitions of rank. Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_K}$ be a K-way tensor. The rank of a tensor (see [24]) is defined as the minimum number r of components required for rank-one decomposition of a given tensor $\boldsymbol{\mathcal{X}}$ in analogy to SVD as follows:

$$\begin{aligned}\boldsymbol{\mathcal{X}} &= \sum_{j=1}^{r} \lambda_j \boldsymbol{a}_j^{(1)} \circ \boldsymbol{a}_j^{(2)} \circ \cdots \circ \boldsymbol{a}_j^{(K)}, \\ &= \boldsymbol{\Lambda} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \cdots \times_K \boldsymbol{A}^{(K)} \end{aligned} \tag{2.4}$$

where $\circ$ denotes the outer product, $\boldsymbol{\Lambda} \in \mathbb{R}^{r \times \cdots \times r}$ denotes a K-way diagonal matrix whose (j, j, j)th element is λ_j, and $\times_k$ denotes the k-mode matrix product (see Kolda & Bader [23]); in addition, we define $\boldsymbol{A}^{(k)} = [\boldsymbol{a}_1^{(k)}, \ldots, \boldsymbol{a}_r^{(k)}]$. The above decomposition model is called CANDECOMP [11] or PARAFAC [19]. It is worth noticing that finding the above decomposition with the minimum r is a hard problem; thus there is no straightforward algorithm for computing the rank for higher-order tensors [24].

[1] Note however that the absolute value is not taken here because singular value is defined to be positive.

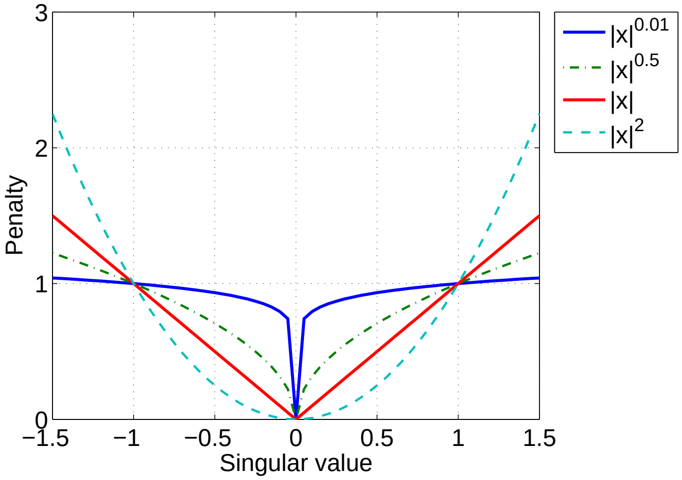

FIG. 2.1. *Penalty functions $|x|^p$ over one singular value x are schematically illustrated for various p. The absolute penalty function $|x|$ is the tightest convex lower bound of the rank ($p \to 0$) in the interval* $[-1, 1]$.

We consider instead the mode-k rank of tensors, which is the foundation of the Tucker decomposition [44, 13]. The mode-k rank of $\boldsymbol{\mathcal{X}}$, denoted $\mathrm{rank}_k(\boldsymbol{\mathcal{X}})$, is the dimensionality of the space spanned by the mode-k fibers of $\boldsymbol{\mathcal{X}}$. In other words, the mode-k rank of $\boldsymbol{\mathcal{X}}$ is the rank of the mode-k unfolding $\boldsymbol{X}_{(k)}$ of $\boldsymbol{\mathcal{X}}$. The mode-k unfolding $\boldsymbol{X}_{(k)}$ is the $n_k \times \bar{n}_{\backslash k}$ ($\bar{n}_{\backslash k} := \prod_{k' \neq k} n_{k'}$) matrix obtained by concatenating the mode-k fibers of $\boldsymbol{\mathcal{X}}$ as column vectors. In MATLAB this can be obtained as follows:

```
X=permute(X,[k:K,1:k-1]); X=X(:,:);
```

where the order of dimensions other than the first dimension k is not important as long as we use a consistent definition. We say that a K-way tensor $\boldsymbol{\mathcal{X}}$ is rank-$(r_1, \ldots, r_K)$ if the mode-k rank of $\boldsymbol{\mathcal{X}}$ is r_k ($k = 1, \ldots, K$). Unlike the rank of the tensor, mode-k rank is clearly computable; the computation of the mode-k ranks of a tensor boils down to the computation the rank of K matrices.

A rank-$(r_1, \ldots, r_K)$ tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times \cdots \times n_K}$ can be written as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \cdots \times_K \boldsymbol{U}_K, \tag{2.5}$$

where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1 \times \cdots \times r_K}$ is called a *core tensor*, and $\boldsymbol{U}_k \in \mathbb{R}^{n_k \times r_k}$ ($n = 1, \ldots, K$) are left singular-vectors from the SVD of the mode-k unfolding of $\boldsymbol{\mathcal{X}}$. The above decomposition is called the Tucker decomposition [44, 23].

The definition of a low-rank tensor (in the sense of Tucker decomposition) implies that a low-rank tensor *is* a low-rank matrix when unfolded appropriately. In order to see this, we recall that for the Tucker model (2.5), the mode-k unfolding of $\boldsymbol{\mathcal{X}}$ can be written as follows (see e.g., [23]):

$$\boldsymbol{X}_{(k)} = \boldsymbol{U}_k \boldsymbol{G}_{(k)} \left(\boldsymbol{U}_{k-1} \otimes \cdots \otimes \boldsymbol{U}_1 \otimes \boldsymbol{U}_K \otimes \cdots \otimes \boldsymbol{U}_{k+1}\right)^{\top}.$$

Therefore, if the tensor $\boldsymbol{\mathcal{X}}$ is low-rank in the kth mode (i.e., $r_k < \min(n_k, \bar{n}_{\backslash k})$), its unfolding is a low-rank matrix. Conversely, if $\boldsymbol{X}_{(k)}$ is a low-rank matrix (i.e., $\boldsymbol{X}_{(k)} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\top}$), we can set $\boldsymbol{U}_k = \boldsymbol{U}$, $\boldsymbol{G}_{(k)} = \boldsymbol{S}\boldsymbol{V}^{\top}$, and other Tucker factors $\boldsymbol{U}_{k'}$ ($k' \neq k$) as identity matrices, and we obtain a Tucker decomposition (2.5).

Note that if a given tensor $\boldsymbol{\mathcal{X}}$ can be written in the form of CP decomposition (2.4) with rank r, we can always find a rank-$(r, r, \ldots, r)$ Tucker decomposition (2.5) by ortho-normalizing each factor $\boldsymbol{A}^{(k)}$ in Equation (2.4). Therefore, the Tucker decomposition is more general than the CP decomposition.

However, since the core tensor $\boldsymbol{\mathcal{G}}$ that corresponds to singular-values in the matrix case (see Equation (2.1)) is not diagonal in general, it is not straightforward to generalize the trace norm from matrices to tensors.

3. Three strategies to extend the trace-norm regularization to tensors.

In this section, we first consider a given tensor as a matrix by unfolding it at a given mode k and propose to minimize the trace norm of the unfolding $\boldsymbol{X}_{(k)}$. Next, we extend this to the minimization of the weighted sum of the trace norms of the unfoldings. Finally, relaxing the condition that the tensor is *jointly* low-rank in every mode in the second approach, we propose a mixture approach. For solving the optimization problems, we use the alternating direction method of multipliers (ADMM) [17] (also known as the split Bregman iteration [18]). The optimization algorithms are discussed in Section 4.

3.1. Tensor as a matrix.

If we assume that the tensor we wish to estimate is (at least) low-rank in the kth mode, we can convert the tensor estimation problem into a matrix estimation problem. Extending the minimization problem (2.2) to accommodate missing entries we have the following optimization problem for the reconstruction of partially observed tensor:

$$\underset{\boldsymbol{\mathcal{X}}\in\mathbb{R}^{n_1\times\cdots\times n_K}}{\text{minimize}} \qquad \frac{1}{2\lambda}\|\Omega(\boldsymbol{\mathcal{X}})-\boldsymbol{y}\|^2+\|\boldsymbol{X}_{(k)}\|_*, \tag{3.1}$$

where $\boldsymbol{X}_{(k)}$ is the mode-k unfolding of $\boldsymbol{\mathcal{X}}$, $\boldsymbol{y}\in\mathbb{R}^M$ is the vector of observations, and $\Omega:\mathbb{R}^{n_1\times\cdots\times n_K}\rightarrow\mathbb{R}^M$ is a linear operator that reshapes the prespecified elements of the input tensor into an M dimensional vector; M is the number of observations. In Equation (3.1), the regularization constant $\lambda>0$ is moved to the denominator of the loss term from the numerator of the regularization term in Equation (2.2); this equivalent reformulation allows us to consider the noiseless case ($\lambda\rightarrow 0$) in the same framework. Note that λ can also be interpreted as the variance of the Gaussian observation noise model.

Since the estimation procedure (3.1) is essentially an estimation of a low-rank matrix $\boldsymbol{X}_{(k)}$, we know that in the noiseless case $O(\tilde{n}_k^{6/5}r_k\log(\tilde{n}_k))$ samples are enough to perfectly recover the unknown true tensor $\boldsymbol{\mathcal{X}}^*$, where $r_k=\mathrm{rank}_k(\boldsymbol{\mathcal{X}}^*)$ and $\tilde{n}_k=\max(n_k,\bar{n}_{\setminus k})$, if the rank r_k is not too high [10]. This holds regardless of whether the unknown tensor $\boldsymbol{\mathcal{X}}$ is low-rank in other modes $k'\neq k$. Therefore, when we can estimate the mode-k unfolding of $\boldsymbol{\mathcal{X}}^*$ perfectly, we can also recover the whole $\boldsymbol{\mathcal{X}}^*$ perfectly, including the ranks of the modes we did not use during the estimation.

However, the success of the above procedure is conditioned on the choice of the mode to unfold the tensor. If we choose a mode with a large rank, even if there are other modes with smaller ranks, we cannot hope to recover the tensor from a small number of samples.

Various advanced methods [42, 43, 25, 22] for the estimation of low-rank matrices can be used for solving the minimization problem (3.1). Here we use ADMM to keep the presentation concise; see Section 4 for the details.

3.2. Constrained optimization of low rank tensors.

In order to exploit the rank deficiency of more than one mode, it is natural to consider the following extension of the estimation procedure (3.1)

$$\underset{\boldsymbol{\mathcal{X}}\in\mathbb{R}^{n_1\times\cdots\times n_K}}{\text{minimize}} \qquad \frac{1}{2\lambda}\|\Omega(\boldsymbol{\mathcal{X}})-\boldsymbol{y}\|^2+\sum_{k=1}^{K}\gamma_k\|\boldsymbol{X}_{(k)}\|_*. \tag{3.2}$$

This is a convex optimization problem, because it can be reformulated as follows:

$$\begin{aligned} &\underset{\boldsymbol{x},\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_K}{\text{minimize}} \qquad \frac{1}{2\lambda}\|\boldsymbol{\Omega}\boldsymbol{x}-\boldsymbol{y}\|^2+\sum_{k=1}^{K}\gamma_k\|\boldsymbol{Z}_k\|_*, && (3.3)\\ &\text{subject to} \qquad \boldsymbol{P}_k\boldsymbol{x}=\boldsymbol{z}_k \quad (k=1,\ldots,K), && (3.4)\end{aligned}$$

where $\boldsymbol{x}\in\mathbb{R}^N$ is the vectorization of $\boldsymbol{\mathcal{X}}$ ($N=\prod_{k=1}^{K}n_k$), $\boldsymbol{P}_k$ is the matrix representation of mode-k unfolding (note that $\boldsymbol{P}_k$ is a permutation matrix; thus $\boldsymbol{P}_k{}^{\top}\boldsymbol{P}_k=\boldsymbol{I}_N$), $\boldsymbol{Z}_k\in\mathbb{R}^{n_k\times\bar{n}_{\backslash k}}$ is an auxiliary matrix of the same size as the mode-k unfolding of $\boldsymbol{\mathcal{X}}$, and $\boldsymbol{z}_k$ is the vectorization of $\boldsymbol{Z}_k$. With a slight abuse of notation $\boldsymbol{\Omega}\in\mathbb{R}^{M\times N}$ denotes the observation operator as a matrix.

This approach was considered earlier by Liu et al. [27] and Signoretto et al. [36]. Liu et al. relaxed the constraints (3.4) into penalty terms, therefore the factors obtained as the left singular vectors of $\boldsymbol{Z}_k$ does not equal the factors of the Tucker decomposition of $\boldsymbol{\mathcal{X}}$. Signoretto et al. have discussed the general Shatten-$\{p,q\}$ norms for tensors and the relationship between the regularization term in Equation (3.2) with $\gamma_k=1/K$ (which corresponds to Shatten-$\{1,1\}$ norm) and the function $\frac{1}{K}\sum_{k=1}^{K}\text{rank}_k(\boldsymbol{\mathcal{X}})$.

3.3. Mixture of low-rank tensors. The optimization problem (3.3) penalizes every mode of the tensor $\boldsymbol{\mathcal{X}}$ to be *jointly* low-rank, which might be too strict to be satisfied in practice. Thus we propose to predict instead with a mixture of K tensors; each mixture component is regularized by the trace norm to be low-rank in each mode. More specifically, we solve the following minimization problem:

$$\underset{\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_K}{\text{minimize}} \qquad \frac{1}{2\lambda}\left\|\boldsymbol{\Omega}\left(\sum\nolimits_{k=1}^{K}\boldsymbol{P}_k{}^{\top}\boldsymbol{z}_k\right)-\boldsymbol{y}\right\|^2+\sum_{k=1}^{K}\gamma_k\|\boldsymbol{Z}_k\|_*. \tag{3.5}$$

Note that when $\boldsymbol{z}_k=\frac{1}{K}\boldsymbol{P}_k\boldsymbol{x}$ for all $k=1,\ldots,K$, the problem (3.5) reduces to the problem (3.3) with $\gamma_k'=\gamma_k/K$.

3.4. Interpretation. All three proposed approaches inherit the *lack of uniqueness* of the factors from the conventional Tucker decomposition [23]. Some heuristics to improve the interpretability of the core tensor $\boldsymbol{\mathcal{G}}$ are proposed and implemented in the N-way toolbox [2]. However, these approaches are all restricted to orthogonal transformations. Here we present another simple heuristic, which is to apply PARAFAC decomposition on the core tensor $\boldsymbol{\mathcal{G}}$. This approach has the following advantages over applying PARAFAC directly to the original data. First, the dimensionality of the core tensor $(r_1,\ldots,r_K)$ is automatically obtained from the proposed algorithms. Therefore, the range of the number of PARAFAC components that we need to look for is much narrower than applying PARAFAC directly to the original data. Second, the PARAFAC problem does not need to take care of missing entries. In other words, we can separate the prediction problem and the interpretation problem, which are separately tackled by the proposed algorithms and PARAFAC, respectively. Finally, empirically the proposed heuristic seems to be more robust in recovering the underlying factors compared to applying PARAFAC directory when the rank is misspecified (see Section 5.2).

More precisely, let us consider the second "Constraint" approach. Let $\boldsymbol{U}_1 \ldots, \boldsymbol{U}_K$ be the left singular vectors of the auxiliary variables $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_K$. From Equation (2.5) we can obtain the core tensor $\boldsymbol{\mathcal{G}}$ as follows:

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{X}} \times_1 {\boldsymbol{U}_1}^\top \times_2 {\boldsymbol{U}_2}^\top \cdots \times_K {\boldsymbol{U}_K}^\top.$$

Let $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(K)}$ be the factors obtained by the PARAFAC decomposition of $\boldsymbol{\mathcal{G}}$ as follows:

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\Lambda} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \cdots \times_K \boldsymbol{A}^{(K)}.$$

Therefore, we have the following decomposition

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\Lambda} \times_1 (\boldsymbol{U}_1\boldsymbol{A}^{(1)}) \times_2 (\boldsymbol{U}_2\boldsymbol{A}^{(2)}) \cdots \times_K (\boldsymbol{U}_K\boldsymbol{A}^{(K)}), \tag{3.6}$$

which gives the kth factor as $\boldsymbol{U}_k\boldsymbol{A}^{(k)}$.

4. Optimization.

In this section, we describe the optimization algorithms based on the alternating direction method of multipliers (ADMM) for the problems (3.1), (3.3), and (3.5).

4.1. ADMM.

The alternating direction method of multipliers [17] (see also [5]) can be considered as an approximation of the method of multipliers [31, 21] (see also [4, 30]). The method of multipliers generates a sequence of primal variables $(\boldsymbol{x}^t, \boldsymbol{z}^t)$ and multipliers $\boldsymbol{\alpha}^t$ by iteratively minimizing the so called augmented Lagrangian (AL) function with respect to the primal variables $(\boldsymbol{x}^t, \boldsymbol{z}^t)$ and updating the multiplier vector $\boldsymbol{\alpha}^t$. Let us consider the following linear equality constrained minimization problem:

$$\underset{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{z}\in\mathbb{R}^m}{\text{minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{z}), \tag{4.1}$$

$$\text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}, \tag{4.2}$$

where f and g are both convex functions. The AL function $L_\eta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\alpha})$ of the above minimization problem is written as follows:

$$L_\eta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \boldsymbol{\alpha}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}) + \frac{\eta}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|^2,$$

where $\boldsymbol{\alpha} \in \mathbb{R}^m$ is the Lagrangian multiplier vector. Note that when $\eta = 0$, the AL function reduces to the ordinary Lagrangian function. Intuitively, the additional penalty term enforces the equality constraint to be satisfied. However, different from the penalty method (which was used in [27]), there is no need to increase the penalty parameter η very large, which usually makes the problem poorly conditioned.

The original method of multipliers performs minimization of the AL function with respect to $\boldsymbol{x}$ and $\boldsymbol{z}$ *jointly* followed by a multiplier update as follows:

$$(\boldsymbol{x}^{t+1}, \boldsymbol{z}^{t+1}) = \underset{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{z}\in\mathbb{R}^m}{\operatorname{argmin}} \; L_\eta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\alpha}^t), \tag{4.3}$$

$$\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \eta(\boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{z}^{t+1}). \tag{4.4}$$

Intuitively speaking, the multiplier is updated proportionally to the violation of the equality constraint (4.2). In this sense, η can also be regarded as a step-size parameter.

Under fairly mild conditions, the above method converges super-linearly to a solution of the minimization problem (4.1); see [34, 41]. However, the joint minimization of the AL function (4.3) is often hard (see [41] for an exception).

The ADMM decouples the minimization with respect to $\boldsymbol{x}$ and $\boldsymbol{z}$ as follows:

$$\boldsymbol{x}^{t+1} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} L_\eta(\boldsymbol{x}, \boldsymbol{z}^t, \boldsymbol{\alpha}^t), \tag{4.5}$$

$$\boldsymbol{z}^{t+1} = \operatorname*{argmin}_{\boldsymbol{z}\in\mathbb{R}^m} L_\eta(\boldsymbol{x}^{t+1}, \boldsymbol{z}, \boldsymbol{\alpha}^t), \tag{4.6}$$

$$\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \eta(\boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{z}^{t+1}). \tag{4.7}$$

Note that the new value of $\boldsymbol{x}^{t+1}$ obtained in the first line is used in the update of $\boldsymbol{z}^{t+1}$ in the second line. The multiplier update step is identical to that of the ordinary method of multipliers (4.4). It can be shown that the above algorithm is an application of firmly nonexpansive mapping and that it converges to a solution of the original problem (4.1). Surprisingly, this is true for any positive penalty parameter η [26]. This is in contrast to the fact that a related approach called forward-backward splitting [26] (which was used in [36]) converges only when the step-size parameter η is chosen appropriately.

4.2. Stopping criterion. As a stopping criterion for terminating the above ADMM algorithm, we employ the relative duality gap criterion; that is, we stop the algorithm when the current primal objective value $p(\boldsymbol{x}, \boldsymbol{z}) := f(\boldsymbol{x}) + g(\boldsymbol{z})$ and the largest dual objective value $\max_{t'=1,\dots,t} d(\tilde{\boldsymbol{\alpha}}^{t'})$ obtained in the past satisfies the following equality

$$(p(\boldsymbol{x}^t, \boldsymbol{z}^t) - \max_{t'=1,\dots,t} d(\tilde{\boldsymbol{\alpha}}^{t'}))/p(\boldsymbol{x}^t, \boldsymbol{z}^t) < \epsilon. \tag{4.8}$$

Note that the multiplier vector $\boldsymbol{\alpha}^t$ computed in Equation (4.7) cannot be directly used in the computation of the dual objective value, because typically $\boldsymbol{\alpha}^t$ violates the dual constraints. See Appendix A for the details.

The reason we use the duality gap is that the criterion is invariant to the scale of the observed entries $\boldsymbol{y}$ and the size of the problem N.

4.3. ADMM for the “As a Matrix” approach. We consider the following constrained reformulation of problem (3.1)

$$\operatorname*{minimize}_{\boldsymbol{x}\in\mathbb{R}^N, \boldsymbol{Z}\in\mathbb{R}^{n_k\times\bar{n}_{\backslash k}}} \quad \frac{1}{2\lambda}\|\boldsymbol{\Omega}\boldsymbol{x} - \boldsymbol{y}\|^2 + \|\boldsymbol{Z}\|_*, \quad \text{subject to} \quad \boldsymbol{P}_k\boldsymbol{x} = \boldsymbol{z}, \tag{4.9}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is a vectorization of $\boldsymbol{\mathcal{X}}$, $\boldsymbol{Z} \in \mathbb{R}^{n_k\times\bar{n}_{\backslash k}}$ is an auxiliary variable that corresponds to the mode-k unfolding of $\boldsymbol{\mathcal{X}}$, and $\boldsymbol{z} \in \mathbb{R}^N$ is the vectorization of $\boldsymbol{Z}$. The AL function of the above constrained minimization problem can be written as follows:

$$L_\eta(\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{\alpha}) = \frac{1}{2\lambda}\|\boldsymbol{\Omega}\boldsymbol{x} - \boldsymbol{y}\|^2 + \|\boldsymbol{Z}\|_* + \eta\boldsymbol{\alpha}^\top(\boldsymbol{P}_k\boldsymbol{x} - \boldsymbol{z}) + \frac{\eta}{2}\|\boldsymbol{P}_k\boldsymbol{x} - \boldsymbol{z}\|^2, \tag{4.10}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^N$ is the Lagrangian multiplier vector that corresponds to the constraint $\boldsymbol{P}_k\boldsymbol{x} = \boldsymbol{z}$. Note that we rescaled the Lagrangian multiplier vector $\boldsymbol{\alpha}$ by the factor η for the sake of notational simplicity.

Starting from an initial point $(\boldsymbol{x}^0, \boldsymbol{Z}^0, \boldsymbol{\alpha}^0)$, we apply the ADMM explained in the previous section to the AL function (4.10). All the steps (4.5)–(4.7) can be implemented in closed forms. First, minimization with respect to $\boldsymbol{x}$ yields,

$$\boldsymbol{x}^{t+1} = \left(\boldsymbol{\Omega}^\top \boldsymbol{y} + \lambda\eta {\boldsymbol{P}_k}^\top (\boldsymbol{z}^t - \boldsymbol{\alpha}^t)\right) ./ (\mathbf{1}_\Omega + \lambda\eta \mathbf{1}_N), \tag{4.11}$$

where $\mathbf{1}_\Omega$ is an N-dimensional vector that has one for observed elements and zero otherwise; $\mathbf{1}_N$ is an N-dimensional vector filled with ones; ./ denotes element-wise division. Note that when $\lambda \to 0$ (no observational noise), the above expression can be simplified as follows:

$$x_i^{t+1} = \begin{cases} (\boldsymbol{\Omega}^\top \boldsymbol{y})_i, & i \in \Omega, \\ ({\boldsymbol{P}_k}^\top (\boldsymbol{z}^t - \boldsymbol{\alpha}^t))_i, & i \notin \Omega \end{cases} \quad (i = 1, \ldots, N). \tag{4.12}$$

Here the observed entries of $\boldsymbol{x}$ are overwritten by the observed values $\boldsymbol{y}$ and the unobserved entries are filled with the mode-k tensorization of the current prediction $\boldsymbol{z}^t - \boldsymbol{\alpha}^t$. In the general case (4.11), the predicted values also affect the observed entries. The primal variable $\boldsymbol{x}^t$ and the auxiliary variable $\boldsymbol{z}^t$ becomes closer and closer as the optimization proceeds. This means that eventually the multiplier vector $\boldsymbol{\alpha}^t$ takes non-zero values only on the observed entries when $\lambda \to 0$.

Next, the minimization with respect to $\boldsymbol{Z}$ yields,

$$\boldsymbol{Z}^{t+1} = \mathrm{prox}^{\mathrm{tr}}_{1/\eta}\left(\boldsymbol{P}_k \boldsymbol{x}^{t+1} + \boldsymbol{\alpha}^t\right),$$

where $\mathrm{prox}^{\mathrm{tr}}_{1/\eta}$ is the spectral soft-threshold operation (2.3) in which the argument $\boldsymbol{P}_k \boldsymbol{x}^{t+1} + \boldsymbol{\alpha}^t$ is considered as a $n_k \times \bar{n}_{\backslash k}$ matrix.

The last step is the multiplier update (4.7), which can be written as follows:

$$\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \left(\boldsymbol{P}_k \boldsymbol{x}^{t+1} - \boldsymbol{z}^{t+1}\right). \tag{4.13}$$

Note that the step-size parameter η does not appear in (4.13) due to the rescaling of $\boldsymbol{\alpha}$ in (4.10).

The speed of convergence of the algorithm mildly depends on the choice of the step-size η. Here as a guideline to choose η, we require that the algorithm is invariant to scalar multiplication of the objective (4.9). More precisely, when the input $\boldsymbol{y}$ and the regularization constant λ are both multiplied by a constant c, the solution of the minimization (4.9) (or (3.1)) should remain essentially the same as the original problem, except that the solution $\boldsymbol{x}$ is also multiplied by the constant c. In order to make the algorithm (see (4.11)-(4.13)) follow the same path (except that $\boldsymbol{x}^t$, $\boldsymbol{z}^t$, and $\boldsymbol{\alpha}^t$ are all multiplied by c), we need to scale η inversely proportional to c. We can also see this in the AL function (4.10); in fact, the first two terms scale linearly to c, and also the last two terms scale linearly if η scales inversely to c. Therefore we choose η as $\eta = \eta_0/\mathrm{std}(\boldsymbol{y})$, where η_0 is a constant and $\mathrm{std}(\boldsymbol{y})$ is the standard deviation of the observed values $\boldsymbol{y}$.

4.4. ADMM for the "Constraint" approach. The AL function of the constrained minimization problem (3.3)-(3.4) can be written as follows:

$$\begin{aligned} L_\eta(\boldsymbol{x}, \{\boldsymbol{Z}_k\}_{k=1}^K, \{\boldsymbol{\alpha}_k\}_{k=1}^K) &= \frac{1}{2\lambda}\|\boldsymbol{\Omega}\boldsymbol{x} - \boldsymbol{y}\|^2 + \sum_{k=1}^K \gamma_k \|\boldsymbol{Z}_k\|_* \\ &\quad + \sum_{k=1}^K \left(\eta {\boldsymbol{\alpha}_k}^\top (\boldsymbol{P}_k \boldsymbol{x} - \boldsymbol{z}_k) + \frac{\eta}{2}\|\boldsymbol{P}_k \boldsymbol{x} - \boldsymbol{z}_k\|^2\right). \end{aligned}$$

Note that we rescaled the multiplier vector $\boldsymbol{\alpha}$ by the factor η as in the previous subsection.

Starting from an initial point $(\boldsymbol{x}^0, \{\boldsymbol{Z}_k^0\}_{k=1}^K, \{\boldsymbol{\alpha}_k^0\}_{k=1}^K)$, we take similar steps as in (4.11)-(4.13) except that the last two steps are performed for all $k = 1, \ldots, K$. That is,

$$\boldsymbol{x}^{t+1} = \left(\boldsymbol{\Omega}^\top \boldsymbol{y} + \lambda\eta \sum\nolimits_{k=1}^K {\boldsymbol{P}_k}^\top (\boldsymbol{z}_k^t - \boldsymbol{\alpha}_k^t)\right) ./(\mathbf{1}_\Omega + \lambda\eta K \mathbf{1}_N), \tag{4.14}$$

$$\boldsymbol{Z}_k^{t+1} = \mathrm{prox}_{\gamma_k/\eta}^{\mathrm{tr}} \left(\boldsymbol{P}_k \boldsymbol{x}^{t+1} + \boldsymbol{\alpha}_k^t\right) \qquad (k = 1, \ldots, K), \tag{4.15}$$

$$\boldsymbol{\alpha}_k^{t+1} = \boldsymbol{\alpha}_k^t + (\boldsymbol{P}_k \boldsymbol{x}^{t+1} - \boldsymbol{z}_k^{t+1}) \qquad (k = 1, \ldots, K). \tag{4.16}$$

By considering the scale invariance of the algorithm, we choose the step-size η as $\eta = \eta_0/\mathrm{std}(\boldsymbol{y})$ as in the previous subsection.

4.5. ADMM for the "Mixture" approach. We consider the following dual problem of the mixture formulation (3.5):

$$\begin{aligned} \underset{\boldsymbol{\alpha}\in\mathbb{R}^M, \boldsymbol{W}_k\in\mathbb{R}^{n_k\times\bar{n}_{\setminus k}}}{\text{minimize}} \quad & \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2 - \boldsymbol{\alpha}^\top \boldsymbol{y} + \sum_{k=1}^K \delta_{\gamma_k}(\boldsymbol{W}_k), \\ \text{subject to} \quad & \boldsymbol{w}_k = \boldsymbol{P}_k \boldsymbol{\Omega}^\top \boldsymbol{\alpha} \qquad (k = 1, \ldots, K), \end{aligned} \tag{4.17}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^M$ is a dual vector; $\boldsymbol{W}_k \in \mathbb{R}^{n_k\times\bar{n}_{\setminus k}}$ is an auxiliary variable that corresponds to the mode-k unfolding of $\boldsymbol{\Omega}^\top\boldsymbol{\alpha}$, and $\boldsymbol{w}_k \in \mathbb{R}^N$ is the vectorization of $\boldsymbol{W}_k$; the indicator function δ_λ is defined as $\delta_\lambda(\boldsymbol{W}) = 0$, if $\|\boldsymbol{W}\| \le \lambda$, and $\delta_\lambda(\boldsymbol{W}) = +\infty$, otherwise, where $\|\cdot\|$ is the spectral norm (maximum singular-value of a matrix).

The AL function for the problem (4.17) can be written as follows:

$$\begin{aligned} L_\eta(\boldsymbol{\alpha}, \{\boldsymbol{W}_k\}_{k=1}^K, \{\boldsymbol{z}_k\}_{k=1}^K) = & \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2 - \boldsymbol{\alpha}^\top \boldsymbol{y} + \sum_{k=1}^K \delta_{\gamma_k}(\boldsymbol{W}_k) \\ & + \sum_{k=1}^K \left({\boldsymbol{z}_k}^\top(\boldsymbol{P}_k\boldsymbol{\Omega}^\top\boldsymbol{\alpha} - \boldsymbol{w}_k) + \frac{\eta}{2}\|\boldsymbol{P}_k\boldsymbol{\Omega}^\top\boldsymbol{\alpha} - \boldsymbol{w}_k\|^2\right) \end{aligned}$$

Similar to the previous two algorithms, we start from an initial point $(\boldsymbol{\alpha}^0, \{\boldsymbol{W}_k^0\}_{k=1}^K, \{\boldsymbol{z}_k^0\}_{k=1}^K)$, and compute the following steps:

$$\begin{aligned} \boldsymbol{\alpha}^{t+1} &= \underset{\boldsymbol{\alpha}}{\mathrm{argmin}}\, L_\eta(\boldsymbol{\alpha}, \{\boldsymbol{W}_k^t\}_{k=1}^K, \{\boldsymbol{z}_k^t\}_{k=1}^K) \\ \boldsymbol{W}_k^{t+1} &= \underset{\boldsymbol{W}_k}{\mathrm{argmin}}\, L_\eta(\boldsymbol{\alpha}^{t+1}, \{\boldsymbol{W}_k\}_{k=1}^K, \{\boldsymbol{z}_k^t\}_{k=1}^K) \\ \boldsymbol{z}_k^{t+1} &= \boldsymbol{z}_k^t + \eta(\boldsymbol{P}_k\boldsymbol{\Omega}^\top\boldsymbol{\alpha}^{t+1} - \boldsymbol{w}_k^{t+1}). \end{aligned} \tag{4.18}$$

The above steps can be computed in closed forms. In fact,

$$\boldsymbol{\alpha}^{t+1} = \frac{1}{\lambda + \eta K}\left(\boldsymbol{y} - \boldsymbol{\Omega}\sum\nolimits_{k=1}^K {\boldsymbol{P}_k}^\top(\boldsymbol{z}_k^t - \eta\boldsymbol{w}_k^t)\right), \tag{4.19}$$

$$\boldsymbol{W}_k^{t+1} = \mathrm{proj}_{\gamma_k}(\boldsymbol{P}_k\boldsymbol{\Omega}^\top\boldsymbol{\alpha}^{t+1} + \boldsymbol{z}_k^t/\eta), \tag{4.20}$$

where the projection operator proj_λ is the projection onto a radius λ-spectral-norm ball, as follows:

$$\mathrm{proj}_\lambda(\boldsymbol{w}) := \boldsymbol{U}\min(\boldsymbol{S}, \lambda)\boldsymbol{V}^\top,$$

where $\boldsymbol{W} = \boldsymbol{USV}^\top$ is the SVD of the matricization of the input vector $\boldsymbol{w}$. Moreover, combining the two steps (4.20) and (4.18), we have (see [41])

$$\boldsymbol{z}_k^{t+1} = \mathrm{prox}^{\mathrm{tr}}_{\gamma_k \eta}\left(\boldsymbol{z}_k^t + \eta \boldsymbol{P}_k \boldsymbol{\Omega}^\top \boldsymbol{\alpha}^{t+1}\right). \tag{4.21}$$

Note that we recover the spectral soft-threshold operation $\mathrm{prox}^{\mathrm{tr}}_{\gamma_k \eta}$ by combining the two steps. Therefore, we can simply iterate steps (4.19) and (4.21) (note that the term $\eta \boldsymbol{w}_k^t$ in (4.19) can be computed from (4.18) as $\eta \boldsymbol{w}_k^t = \boldsymbol{z}_k^{t-1} + \eta \boldsymbol{P}_k \boldsymbol{\Omega}^\top \boldsymbol{\alpha}^t - \boldsymbol{z}_k^t$.)

In order to see that the multiplier vector $\boldsymbol{z}_k^t$ obtained in the above steps converges to the primal solution of the mixture formulation (3.5), we take the derivative of the ordinary Lagrangian function L_0 with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{W}_k$ ($k = 1, \ldots, K$) and obtain the following optimality conditions:

$$\begin{gathered} \boldsymbol{\alpha} = \frac{1}{\lambda}\left(\boldsymbol{y} - \boldsymbol{\Omega} \sum\nolimits_{k=1}^{K} \boldsymbol{P}_k{}^\top \boldsymbol{z}_k\right), \\ \boldsymbol{P}_k \boldsymbol{\Omega}^\top \boldsymbol{\alpha} \in \partial \gamma_k \|\boldsymbol{Z}_k\|_* \qquad (k = 1, \ldots, K), \end{gathered}$$

where we used the relationship $\boldsymbol{w}_k = \boldsymbol{P}_k \boldsymbol{\Omega}^\top \boldsymbol{\alpha}$, and the fact that $\partial \delta_{\gamma_k}(\boldsymbol{W}_k) \ni \boldsymbol{z}_k$ implies $\boldsymbol{w}_k \in \partial \gamma_k \|\boldsymbol{Z}_k\|_*$ because the two functions δ_{γ_k} and $\gamma_k \|\cdot\|_*$ are conjugate to each other; see [33, Cor. 23.5.1]. By combining the above two equations, we obtain the optimality condition for the mixture formulation (3.5) as follows:

$$-\frac{1}{\lambda} \boldsymbol{P}_k \boldsymbol{\Omega}^\top \left(\boldsymbol{y} - \boldsymbol{\Omega} \sum\nolimits_{k=1}^{K} \boldsymbol{P}_k{}^\top \boldsymbol{z}_k\right) + \partial \gamma_k \|\boldsymbol{Z}_k\|_* \ni 0 \qquad (k = 1, \ldots, K).$$

As in the previous two subsections, we require that the algorithm (4.18)-(4.21) is invariant to scalar multiplication of the input $\boldsymbol{y}$ and the regularization constant λ by the same constant c. Since $\boldsymbol{z}_k^t$ appears in the final solution, $\boldsymbol{z}_k^t$ must scale linearly with respect to c. Thus from (4.18), if $\boldsymbol{\alpha}_k^t$ and $\boldsymbol{w}_k^t$ are constants with respect to c, the step-size η must scale linearly. In fact, from (4.19) and (4.20), we can see that these two dual variables remain constant when $\boldsymbol{y}$, $\boldsymbol{z}_k^t$, and η are multiplied by c. Therefore, we choose $\eta = \mathrm{std}(\boldsymbol{y})/\eta_0$.

5. Numerical experiments. In this section, we first present results on two synthetic datasets. Finally we apply the proposed methods to the Amino acid fluorescence data published by Bro and Andersson [7].

5.1. Synthetic experiments. We randomly generated a rank-(7,8,9) tensor of dimensions (50,50,20) by drawing the core from the standard normal distribution and multiplying its each mode by an orthonormal factor randomly drawn from the Haar measure. We randomly selected some elements of the true tensor for training and kept the remaining elements for testing. We used the algorithms described in the previous section with the tolerance $\epsilon = 10^{-3}$. We choose $\gamma_k = 1$ for simplicity in the later two approaches. The step-size η is chosen as $\eta = \eta_0/\mathrm{std}(\boldsymbol{y})$ for the first two approaches and $\eta = \mathrm{std}(\boldsymbol{y})/\eta_0$ for the third approach with $\eta_0 = 0.1$. For the first two approaches, $\lambda \to 0$ (zero observation error) was used; see (4.12). For the last approach, we used $\lambda = 0$. The Tucker decomposition algorithm `tucker` from the N-way toolbox [2] is also included as a baseline, for which we used the correct rank ("exact") and the 20% higher rank ("large"). Note that all proposed approaches can find the rank automatically. The generalization error is defined as follows:

$$\mathrm{error} = \frac{\|\boldsymbol{y}_{\mathrm{pred}} - \boldsymbol{y}_{\mathrm{test}}\|}{\|\boldsymbol{y}_{\mathrm{test}}\|},$$

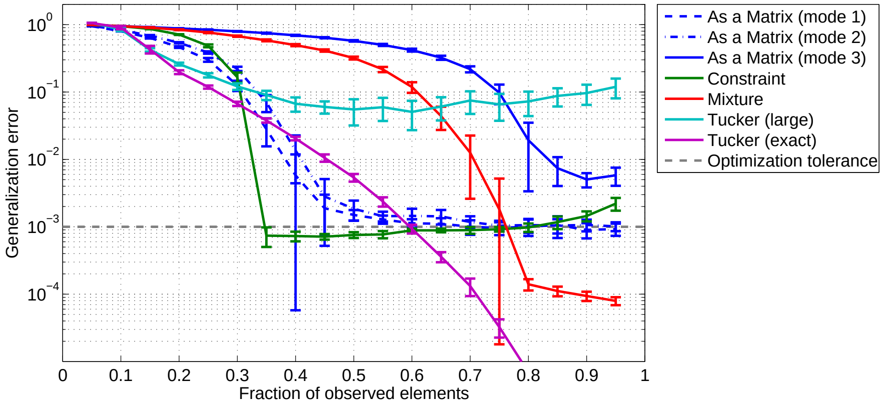

FIG. 5.1. *Comparison of three strategies, tensor as a matrix ("As a Matrix"), constrained optimization ("Constraint"), and mixture of low-rank tensors ("Mixture") on a synthetic rank-$(7, 8, 9)$ tensor (the dimensions are $50 \times 50 \times 20$). Also the Tucker decomposition with 20% higher rank ("large") and with the correct rank ("exact") implemented in the N-way toolbox [2] are included as baselines. The generalization error is plotted against the fraction of observed elements of the underlying low-rank tensor. Also the tolerance of optimization (10^{-3}) is shown.*

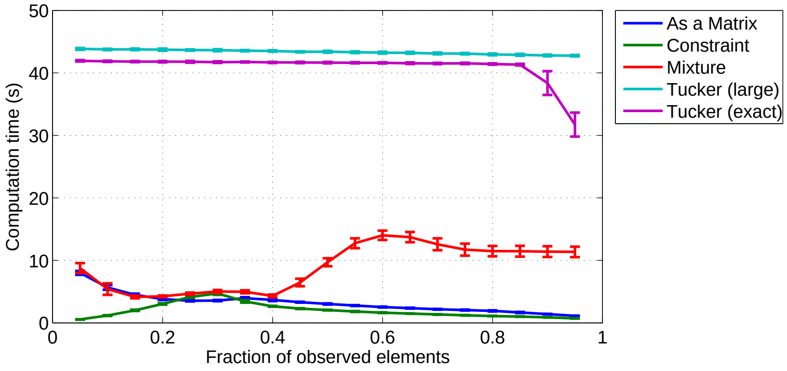

FIG. 5.2. *Comparison of computation times.*

where $\boldsymbol{y}_{\text{test}}$ is the vectorization of the unobserved entries and $\boldsymbol{y}_{\text{pred}}$ is the prediction computed by the algorithms. For the "As a Matrix" strategy, error for each mode is reported. All algorithms were implemented in MATLAB and ran on a computer with two 3.5GHz Xeon processors and 32GB of RAM. The experiment was repeated 20 times and averaged.

Figure 5.1 shows the result of tensor completion using three strategies we proposed above, as well as the Tucker decomposition. At 35% observation, the proposed "Constraint" obtains nearly perfect generalization. Interestingly there is a sharp transition from a poor fit (generalization error$>$ 1) to an almost perfect fit (generalization error$\simeq 10^{-3}$). The "As a Matrix" approach also show similar transition for mode 1 and mode 2 (around 40%), and mode 3 (around 80%), but even the first transition is slower than the "Constraint" approach. The "Mixture" approach shows a transition around 70% slightly faster than the mode 3 in the "As A Matrix" approach. Tucker shows early decrease in the generalization error, but when the rank is misspecified ("large"), the error remains almost constant; even when the correct rank is known

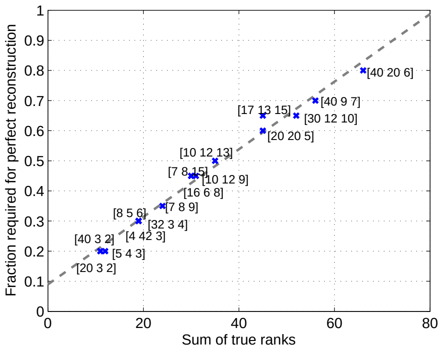

FIG. 5.3. *Fraction of observations at the threshold plotted against the sum of true ranks. Numbers in the brackets denote the k-rank of the underlying tensor. The dimension of the tensor is (50,50,20).*

("exact"), the convergence is slower than the proposed "Constraint" approach.

The proposed convex approaches are not only accurate but also fast. Fig. 5.2 shows the computation time of the proposed approaches and EM-based Tucker decomposition against the fraction of observed entries. For the "As a Matrix" approach the total time for all modes is plotted. We can see that the "As a Matrix" and "Constraint" approaches are roughly 4–10 times faster than the conventional EM-based tucker decomposition.

We have further investigated the condition for the threshold behaviour using the proposed "Constraint" approach. Here we generated different problems of different core dimensions (r_1, r_2, r_3). The sum of mode-k ranks is defined as $\min(r_1, r_2r_3) + \min(r_2, r_3r_1) + \min(r_3, r_1r_2)$. For each problem, we apply the "Constraint" approach for increasingly large fraction of observations and determine when the generalization error falls below 0.01. Fig. 5.3 shows the fraction of observations required to obtain generalization error below 0.01 (in other words, the fraction at the threshold) against the sum of mode-k ranks defined above. We can see that the fraction at the threshold is roughly proportional to the sum of the mode-k ranks of the underlying tensor. We do not have any theoretical argument to support this observation. Acar et al [1] also empirically discussed condition for successful recovery for the CP decomposition.

Figure 5.4 show another synthetic experiment. We randomly generated a rank-$(50, 50, 5)$ tensor of the same dimensions as above. We chose the same parameter values $\gamma_k = 1$, $\lambda \to 0$, $\epsilon = 10^{-3}$, and $\eta_0 = 0.1$. Here we can see that interestingly the "Constraint" approach perform poorly, whereas the "mode 3" and "Mixture" perform clearly better than other algorithms. It is natural that the "mode 3" approach works well because the true tensor is only low-rank in the third mode. In contrast, the "Mixture" approach can automatically detect the rank-deficient mode, because the regularization term in the formulation (3.5) is a linear sum of three $(K = 3)$ penalty terms. The linear sum structure enforces sparsity across $\boldsymbol{Z}_k$. Therefore, in this case $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ were switched off, and "Mixture" approach yielded almost identical results to the "mode 3" approach.

5.2. Amino acid fluorescence data. The amino acid fluorescence data is a semi-realistic data contributed by Bro and Andersson [7], in which they measured the fluorescence of five laboratory-made solutions that each contain different amounts of tyrosine, tryptophan and phenylalanine. Since the "factors" are known to be the

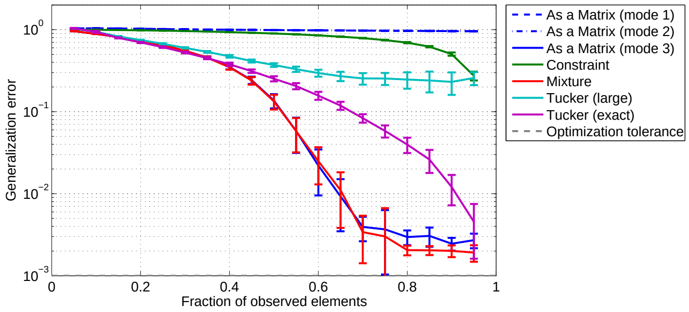

FIG. 5.4. *Synthetic experiment on a rank-*$(50, 50, 5)$ *tensor of dimensions* $50 \times 50 \times 20$*. See also Fig. 5.1.*

three amino acids, this is a perfect data for testing whether the proposed method can automatically find those factors.

For the experiments in this subsection, we chose the same parameter setting $\gamma_k = 1$, $\lambda \to 0$, $\epsilon = 10^{-3}$, and $\eta_0 = 0.1$. Setting $\lambda \to 0$ corresponds to assuming no observational noise. This can be justified by the fact that the original data is already approximately low-rank (rank-$(3, 3, 3)$) in the sense of Tucker decomposition. The dimensionality of the original tensor is $201 \times 61 \times 5$, which correspond to emission wavelength (250–450 nm), excitation wavelength (240–300 nm), and samples, respectively.

Fig. 5.5 show the generalization error obtained by the proposed approaches as well as EM-based Tucker and PARAFAC decompositions. Here PARAFAC is included because the dataset is originally designed for PARAFAC. We can see that the proposed “Constraint” approach show fast decrease in generalization error, which is comparable to the PARAFAC model knowing the correct dimension. Tucker decomposition of rank-$(3, 3, 3)$ performs as good as PARAFAC models when more than half the entries are observed. However, a slightly larger rank-$(4, 4, 4)$ Tucker decomposition could not decrease the error below 0.05.

Fig. 5.6 show the factors obtained by fitting directly three-component PARAFAC model, four-component PARAFAC model, and applying a four component PARAFAC model to the core obtained by the proposed “Constraint” approach. The fraction of observed entries was 0.5. The two conventional approaches used EM iteration for the estimation of missing values. For the proposed model, the dimensionality of the core was $4 \times 4 \times 5$; this was obtained by keeping the singular-values of the auxiliary variable $\boldsymbol{Z}_k$ that are larger than 1% of its largest singular-value for each $k = 1, \ldots, K$. Then we applied a four-component (fully-observed) PARAFAC model to this core and obtained the factors as in Equation (3.6). Interestingly, although the four component-PARAFAC model is redundant for this problem [7], the proposed approach seem to be more robust than applying four-component PARAFAC directly to the data. We can see that the shape of the major three components (blue, green, red) obtained by the proposed approach (the right column) are more similar to the three-component PARAFAC model (the left column) than the four-component PARAFAC model (the center column).

6. Summary. In this paper we have proposed three strategies to extend the framework of trace norm regularization to the estimation of partially observed low-rank tensors. The proposed approaches are formulated in convex optimization prob-

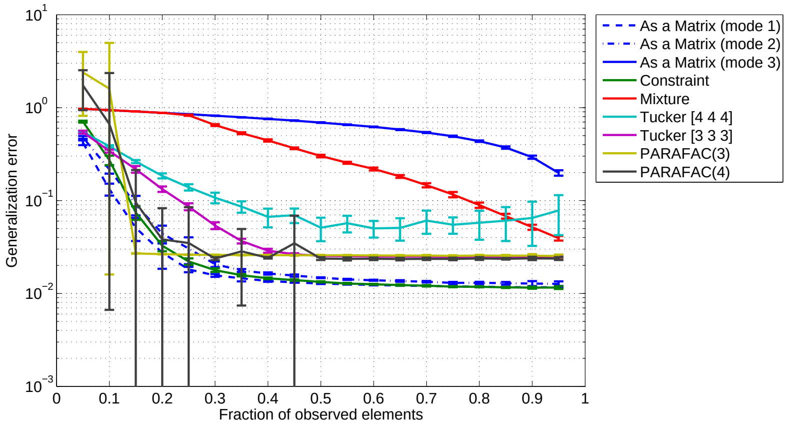

FIG. 5.5. *Generalization performance of proposed methods on the amino acid fluorescence data is compared to conventional EM-based Tucker decomposition and PARAFAC. See also Figures 5.1 and 5.4.*

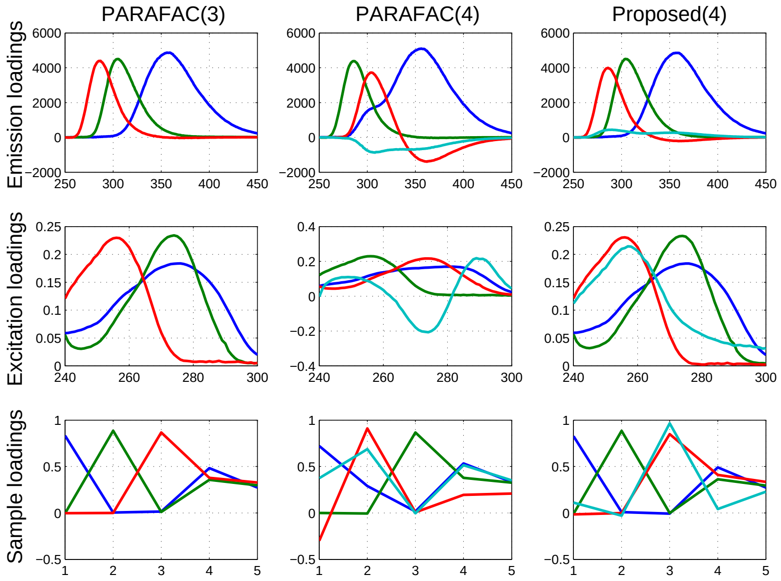

FIG. 5.6. *Factors obtained by three-component PARAFAC (left), four-component PARAFAC (center), and the heuristic proposed in Section 3.4 (right) at the fraction of observation* 0.5*. Even when a redundant four-component PARAFAC is used in the post-processing, the proposed heuristic estimates the factors more reliably than directly applying the PARAFAC model.*

lems and the rank of the tensor decomposition is automatically determined through the optimization.

In the simulated experiment, tensor completion using the “Constraint” approach showed nearly perfect reconstruction from only 35% observations. The proposed approach shows a sharp threshold behaviour and we have empirically found that the fraction of samples at the threshold is roughly proportional to the sum of mode-k ranks of the underlying tensor.

We have also shown the weakness of the “Constraint” approach. When the unknown tensor is only low-rank in certain mode, the assumption that the tensor is low-rank in every mode, which underlies the “Constraint” approach, is too strong. We have demonstrated that the “Mixture” approach is more effective in this case. The “Mixture” approach can automatically detect the rank-deficient mode and lead to better performance.

In the amino acid fluorescence dataset, we have shown that the proposed “Constraint” approach outperforms conventional EM-based Tucker decomposition and is comparable to PARAFAC model with the correct number of components. Moreover, we have demonstrated a simple heuristic to obtain a PARAFAC-style decomposition from the decomposition obtained by the proposed method. Moreover, we have shown that the proposed heuristic can reliably recover the true factors even when the number of PARAFAC factors is misspecified.

The proposed approaches can be extended in many ways. For example, it would be important to handle non-Gaussian noise model [12, 20]; for example a tensor version of robust PCA [9] would be highly desirable. For classification over tensors, extension of the approach in [40] would be meaningful in applications including brain-computer interface; see also [35] for another recent approach. It is also important to extend the proposed approach to handle large scales tensors that cannot be kept in the RAM. Combination of the first-order optimization proposed by Acar et al. [1] with our approach is a promising direction. Moreover, in order to understand the threshold behaviour, further theoretical analysis is necessary.

Appendix A. Computation of the dual objectives. In this Appendix, we show how we compute the dual objective values for the computation of the relative duality gap (4.8).

A.1. Computation of dual objective for the “As a Matrix” approach. The dual problem of the constrained minimization problem (4.9) can be written as follows:

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^N}{\text{maximize}} \quad -\frac{\lambda}{2}\left\|\boldsymbol{\Omega}\boldsymbol{P}_k{}^{\top}\boldsymbol{\alpha}\right\|^2 + \boldsymbol{y}^{\top}\boldsymbol{\Omega}\boldsymbol{P}_k{}^{\top}\boldsymbol{\alpha} \tag{A.1}$$

$$\text{subject to} \quad \bar{\boldsymbol{\Omega}}\boldsymbol{P}_k{}^{\top}\boldsymbol{\alpha} = \boldsymbol{0}, \quad \|\boldsymbol{A}\| \leq 1. \tag{A.2}$$

Here $\bar{\boldsymbol{\Omega}} : \mathbb{R}^N \to \mathbb{R}^{N-M}$ is the linear operator that reshapes the elements of a given N dimensional vector that correspond to the unobserved entries into an $N-M$ dimensional vector. In addition, $\boldsymbol{A} \in \mathbb{R}^{n_k \times \bar{n}_{\setminus k}}$ is the matricization of $\boldsymbol{\alpha}$ and $\|\cdot\|$ is the spectral norm (maximum singular value).

Note that the multiplier vector $\boldsymbol{\alpha}^t$ obtained through ADMM does not satisfy the above two constraints (A.2). Therefore, similar to the approach used in [45, 41], we apply the following transformations. First, we compute the projection $\hat{\boldsymbol{\alpha}}^t$ by projecting $\boldsymbol{\alpha}^t$ to the equality constraint. This can be done easily by setting the elements of $\boldsymbol{\alpha}^t$ that correspond to unobserved entries to zero. Second, we compute the maximum singular value σ_1 of the matricization of $\hat{\boldsymbol{\alpha}}^t$ and shrink $\hat{\boldsymbol{\alpha}}^t$ as follows:

$$\tilde{\boldsymbol{\alpha}}^t = \min(1, 1/\sigma_1)\hat{\boldsymbol{\alpha}}^t.$$

Clearly this operation does not violate with the equality constraint. Finally we substitute $\tilde{\boldsymbol{\alpha}}^t$ into the dual objective (A.1) to compute the relative duality gap as in Equation (4.8).

A.2. Computation of dual objective for the "Constraint" approach. The dual problem of the constrained minimization problem (3.3) can be written as follows:

$$\begin{aligned} \underset{\{\boldsymbol{\alpha}_k\}_{k=1}^K}{\text{maximize}} \quad & -\frac{\lambda}{2}\left\|\boldsymbol{\Omega}\sum_{k=1}^{K}\boldsymbol{P}_k{}^\top\boldsymbol{\alpha}_k\right\|^2 + \boldsymbol{y}^\top\boldsymbol{\Omega}\sum_{k=1}^{K}\boldsymbol{P}_k{}^\top\boldsymbol{\alpha}_k, \\ \text{subject to} \quad & \bar{\boldsymbol{\Omega}}\sum_{k=1}^{K}\boldsymbol{P}_k{}^\top\boldsymbol{\alpha}_k = \boldsymbol{0}, \quad \|\boldsymbol{A}_k\| \leq \gamma_k \, (k = 1, \dots, K). \end{aligned} \tag{A.3}$$

Here the anti-observation operator $\bar{\boldsymbol{\Omega}}$ is defined as in the last subsection, and $\boldsymbol{A}_k \in \mathbb{R}^{n_k \times \bar{n}_{\setminus k}}$ is the matricization of $\boldsymbol{\alpha}_k$ $(k = 1, \dots, K)$.

In order to obtain a dual feasible point from the current multiplier vectors $\boldsymbol{\alpha}_k^t$ $(k = 1, \dots, K)$, we apply similar transformations as in the last subsection. First, we compute the projection to the equality constraint. This can be done by computing the sum over $\boldsymbol{\alpha}_1^t, \dots, \boldsymbol{\alpha}_K^t$ for each unobserved entry. Then the sum divided by K is subtracted from each corresponding entry for $k = 1, \dots, K$. Let us denote by $\hat{\boldsymbol{\alpha}}_k^t$ the multiplier vectors after the projection. Next, we compute the largest singular-values $\sigma_{k,1} = \sigma_1(\hat{\boldsymbol{A}}_k^t)$ where $\hat{\boldsymbol{A}}_k^t$ is the matricization of the projected multiplier vector $\hat{\boldsymbol{\alpha}}_k^t$ for $k = 1, \dots, K$. Now in order to enforce the inequality constraints, we define the *shrinkage factor* c as follows:

$$c = \min(1, \gamma_1/\sigma_{1,1}, \gamma_2/\sigma_{2,1}, \dots, \gamma_K/\sigma_{K,1}). \tag{A.4}$$

Using the above shrinkage factor, we obtain a dual feasible point $\tilde{\boldsymbol{\alpha}}_k^t$ as follows:

$$\tilde{\boldsymbol{\alpha}}_k^t = c\hat{\boldsymbol{\alpha}}_k^t \qquad (k = 1, \dots, K).$$

Finally, we substitute $\tilde{\boldsymbol{\alpha}}_k^t$ into the dual objective (A.3) to compute the relative duality gap as in Equation (4.8).

A.3. Computation of dual objective for the "Mixture" approach. The dual problem of the mixture formulation is already given in Equation (4.17). Making the implicit inequality constraints explicit, we can rewrite this as follows:

$$\begin{aligned} \underset{\boldsymbol{\alpha} \in \mathbb{R}^M}{\text{maximize}} \quad & -\frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2 + \boldsymbol{\alpha}^\top\boldsymbol{y}, \\ \text{subject to} \quad & \|\boldsymbol{P}_k\boldsymbol{\Omega}^\top\boldsymbol{\alpha}\| \leq \gamma_k \quad (k = 1, \dots, K). \end{aligned}$$

Note that the norm in the second line should be interpreted as the spectral norm of the matricization of $\boldsymbol{P}_k\boldsymbol{\Omega}^\top\boldsymbol{\alpha}$. Although the ADMM presented in Section 4.5 was designed to solve this dual formulation, we did not discuss how to evaluate the dual objective. Again the dual vector $\boldsymbol{\alpha}^t$ obtained through the ADMM does not satisfy the inequality constraints.

In order to obtain a dual feasible point, we compute the largest singular-values $\sigma_{k,1} = \sigma_1(\boldsymbol{P}_k\boldsymbol{\Omega}^\top\boldsymbol{\alpha})$ for $k = 1, \dots, K$. From the singular-values $\sigma_{k,1}$, we can compute the shrinkage factor c as in Equation (A.4) in the previous subsection. Finally, a dual feasible point can be obtained as $\tilde{\boldsymbol{\alpha}} = c\boldsymbol{\alpha}$, which we use for the computation of the relative duality gap (4.8).

REFERENCES

[1] E. Acar, D.M. Dunlavy, T.G. Kolda, and M. Mørup, *Scalable tensor factorizations with missing data*, tech. report, arXiv:1005.2197v1 [math.NA], 2010.

[2] C. A. Andersson and R. Bro, *The N-way toolbox for MATLAB*, Chemometrics & Intelligent Laboratory Systems, 52 (2000), pp. 1–4. http://www.models.life.ku.dk/source/nwaytoolbox/.

[3] A. Argyriou, T. Evgeniou, and M. Pontil, *Multi-task feature learning*, in Advances in Neural Information Processing Systems 19, B. Schölkopf, J. Platt, and T. Hoffman, eds., MIT Press, Cambridge, MA, 2007, pp. 41–48.

[4] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 1982.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, 2011. Unfinished working draft.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[7] R. Bro, *PARAFAC. Tutorial and applications*, Chemometrics and Intelligent Laboratory Systems, 38 (1997), pp. 149–171.

[8] J.-F. Cai, E. J. Candes, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, tech. report, arXiv:0810.3286, 2008.

[9] E. J. Candes, X. Li, Y. Ma, and J. Wright, *Robust principal component analysis?*, tech. report, arXiv:0912.3599, 2009.

[10] E. J. Candes and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computational Mathematics, 9 (2009), pp. 717–772.

[11] J.D. Carroll and J.J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[12] E. C. Chi and T. G. Kolda, *Making tensor factorizations robust to non-gaussian noise*, tech. report, arXiv: 1010.3043v1, 2010.

[13] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1324–1342.

[14] L. De Lathauwer and J. Vandewalle, *Dimensionality reduction in higher-order signal processing and rank-$(r_1, r_2, \ldots, r_n)$ reduction in multilinear algebra*, Linear Algebra and its Applications, 391 (2004), pp. 31–55.

[15] L. Eldén and B. Savas, *A Newton–Grassmann method for computing the best multilinear rank-(r_1, r_2, r_3) approximation of a tensor,*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 248–271.

[16] M. Fazel, H. Hindi, and S. P. Boyd, *A Rank Minimization Heuristic with Application to Minimum Order System Approximation*, in Proc. of the American Control Conference, 2001.

[17] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Comput. Math. Appl., 2 (1976), pp. 17–40.

[18] T. Goldstein and S. Osher, *The split Bregman method for L1 regularized problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 323–343.

[19] R.A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis*, UCLA working papers in phonetics, 16 (1970), pp. 1–84.

[20] K. Hayashi, T. Takenouchi, T. Shibata, Y. Kamiya, D. Kato, K. Kunieda, K. Yamada, and K. Ikeda, *Exponential family tensor factorization for missing-values prediction and anomaly detection*, in 2010 IEEE International Conference on Data Mining, 2010, pp. 216–225.

[21] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.

[22] S. Ji and J. Ye, *An accelerated gradient method for trace norm minimization*, in Proceedings of the 26th International Conference on Machine Learning (ICML2009), New York, NY, 2009, ACM, pp. 457–464.

[23] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.

[24] J. B. Kruskal, *Rank, decomposition, and uniqueness for 3-way and n-way arrays*, in Multiway data analysis, R. Coppi and S. Bolasco, eds., Elsevier, North-Holland, Amsterdam, 1989, pp. 7–18.

[25] Z. Lin, M. Chen, L. Wu, and Y. Ma, *The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices*, Mathematical Programming, (2009). submitted.

[26] P. L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM Journal on Numerical Analysis, 16 (1979), pp. 964–979.

[27] J. Liu, P. Musialski, P. Wonka, and J. Ye, *Tensor completion for estimating missing values in visual data*, in Prof. ICCV, 2009.

[28] M. Mørup, *Applications of tensor (multiway array) factorizations and decompositions in data mining*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1 (2011), pp. 24–40.

[29] M. Mørup, L.K. Hansen, C.S. Herrmann, J. Parnas, and S.M. Arnfred, *Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG*, NeuroImage, 29 (2006), pp. 938–947.

[30] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 1999.

[31] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, London, New York, 1969, pp. 283–298.

[32] B. Recht, M. Fazel, and P.A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Review, 52 (2010), pp. 471–501.

[33] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.

[34] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. of Oper. Res., 1 (1976), pp. 97–116.

[35] M. Signoretto, L. De Lathauwer, and J.A.K. Suykens, *Convex multilinear estimation and operatorial representations*, in NIPS2010 Workshop: Tensors, Kernels and Machine Learning (TKML), 2010.

[36] ———, *Nuclear norms for tensors and their use for convex multilinear estimation*, Tech. Report 10-186, ESAT-SISTA, K.U.Leuven, 2010.

[37] A.K. Smilde, R. Bro, and P. Geladi, *Multi-way analysis with applications in the chemical sciences*, Wiley, 2004.

[38] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola, *Maximum-margin matrix factorization*, in Advances in Neural Information Processing Systems 17, Lawrence K. Saul, Yair Weiss, and Léon Bottou, eds., MIT Press, Cambridge, MA, 2005, pp. 1329–1336.

[39] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Stat. Soc. B, 58 (1996), pp. 267–288.

[40] R. Tomioka and K. Aihara, *Classifying matrices with a spectral regularization*, in Proceedings of the 24th International Conference on Machine Learning (ICML2007), ACM Press, 2007, pp. 895–902.

[41] R. Tomioka, T. Suzuki, and M. Sugiyama, *Super-linear convergence of dual augmented-Lagrangian algorithm for sparsity regularized estimation*, tech. report, arXiv:0911.4046v2, 2009.

[42] R. Tomioka, T. Suzuki, and M. Sugiyama, *Augmented Lagrangian methods for learning, selecting, and combining features*, in Optimization for Machine Learning, Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright, eds., MIT Press, 2011.

[43] R. Tomioka, T. Suzuki, M. Sugiyama, and H. Kashima, *A fast augmented lagrangian algorithm for learning low-rank matrices*, in Proceedings of the 27 th Annual International Conference on Machine Learning (ICML2010), Johannes Fürnkranz and Thorsten Joachims, eds., Omnipress, 2010.

[44] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[45] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Signal Process., 57 (2009), pp. 2479–2493.