Inexact higher-order proximal algorithms for tensor factorization

Valentin Leplat[1], Anh-Huy Phan[1], Andersen Ang[2]
[1] Center of Artificial Inteligence Technology, Skoltech, Moscow, Russia
[2] School of Electronics and Computer Science, University of Southampton, Southampton, UK

***Abstract*—In the last decades, Matrix Factorization (MF) models and their multilinear extension – Tensor Factorization (TF) models have been shown to be powerful tools for high dimensional data analysis and features extraction. Computing MF's or TF's are commonly achieved by solving a constrained optimization subproblem on each block of variables, where the subproblems usually have a huge problem size that one has to rely on First-order Methods (FoM), i.e., gradient-based optimization methods.**

In this work, we consider Higher-order Methods (HoM), which are based on higher-order derivatives of the objective function. Compared to FoM, HoM are faster both in theory and practice. However, HoM has a higher per-iteration cost than FoM. Based on the recent development of efficient and implementable HoM, we consider higher-order proximal point methods within the BLUM framework which is potentially tractable for large-scale problems. For the newly proposed HoM, we introduce the appropriate objective functions, derive the algorithm, and show experimentally that the drop in the number of iterations with respect to their per-iteration cost make these HoM-based algorithms attractive for computing MF's and TF's.

***Index Terms*—Matrix Factorization, Tensor decomposition, higher-order proximal point methods, constrained optimization**

I. Introduction

Identifying the underlying structure of a data set and extracting meaningful information is a key problem in data analysis. Simple and powerful methods to achieve this goal are linear dimensionality reduction (LDR) techniques, which are equivalent to low-rank matrix and tensor approximations. Examples of LDR techniques are principal component analysis (PCA), their sparse and/or robust variants, independent component analysis (ICA), low-rank matrix completion, sparse component analysis, Canonical polyadic decomposition (CPD), and Tucker decomposition. This type of methods, besides being simple, are applicable in many applications. Among low-rank approximation techniques, nonnegative matrix factorization (NMF) and nonnegative tensor factorization (NTF) require the factors of the low-rank approximation to be component-wise nonnegative. This allows us to interpret them meaningfully, e.g., when they correspond to nonnegative physical quantities. Applications of NMF include extracting parts of faces (e.g., eyes, noses, and lips) in facial images, identifying topics in text documents, learning hidden Markov models, extracting materials and their abundances in hyperspectral images, separating audio sources from mixtures, detecting communities in networks, and decomposing gene expression micro-arrays [1]–[6].

The recent progress in technologies and telecommunications allows collecting huge volumes of information. Big data has led to many issues that are still challenging for the state-of-the-art MF and TF methods. Computing such decomposition commonly involves solving large scale optimization problems with special structure. The data community is urgently looking for new theoretical and algorithmic solutions. In this work, we focus on High-order Methods (HoM), which utilizes the higher-order derivatives of the objective function. Based on the recent development of efficient and implementable HoM [7]–[12], we propose a higher-order proximal point methods within the *Bi-level unconstrained minimization* (BLUM) framework [11] which is potentially tractable for large-scale problems. We propose new 3rd-order proximal point methods using only 2nd-order derivatives with convergence rate $O(k^{-4})$ and improves the lower bound $O(k^{-7/2})$, where k is the iteration counter[1]. Roughly speaking, the idea of these approaches consists of 1) an assumption on boundedness of the 4th-derivative and 2) the implementation of 3rd-order scheme from [8] using only the 2nd-order oracle.

Contributions: (1) we propose the first tractable 3rd-order algorithm relying on the BLUM framework to tackle low-rank tensor approximations, and (2) we extend BLUM to constrained problems with nonnegativity constraints.

II. Inexact higher-order proximal point methods

We now briefly discuss the notions of HoM and the key aspects of BLUM for the proposed algorithms.

A. Notation, assumptions and higher-order proximal methods

Let $\mathbb{E} \subset \mathbb{R}^n$ denotes a vector space endowed with an inner product $\langle x, y\rangle$ and its induced norm $\|x\| = \langle x, x\rangle^{\frac{1}{2}}$, let $f : \mathbb{E} \to \mathbb{R}$ be a closed convex function, we want to solve

$$\operatorname*{argmin}_{x\in\mathbb{E}} f(x). \tag{1}$$

We assume f is p-times continuously differentiable on $\mathbb{E}$ such that the pth derivative is uniformly upper-bounded as $\sup_{x\in\mathbb{E}} \left\|D^p f(x)\right\| =: M_p < +\infty$, where $\|D^p f(x)\| = \max_h \left\{\left|D^p f(x)[h]^p\right| : \|h\| \leq 1\right\}$ and $D^p f(x)[h_1, ..., h_p]$ is the pth-order directional derivative of f at x along the direction $h = [h_1, ..., h_p]$.

[1]Note that others 2nd-order methods with similar rates have been concomitantly introduced in [7].

The higher-order proximal point methods are based on the pth-order proximal operators defined as

$$\mathbf{prox}^p_{f,\lambda}(\bar{x}) \coloneqq \underset{x\in\mathbb{E}}{\operatorname{argmin}} \Big\{ f^p_{\bar{x},\lambda} \coloneqq f(x) + \lambda d_{p+1}(x-\bar{x}) \Big\}, \quad (2)$$

where $\lambda \in \mathbb{R}^+$ and $p \in \mathbb{N}$, the function $f^p_{\bar{x},\lambda}$ is the *pth-order model* and $d_{p+1}(a) = \frac{1}{p+1}\|a\|^{p+1}$. Note that (2) generalizes the proximal operator

$$\mathbf{prox}_{f,\lambda}(\bar{x}) \coloneqq \underset{x\in\mathbb{E}}{\operatorname{argmin}} \Big\{ f(x) + \frac{1}{2\lambda}\|x-\bar{x}\|^2 \Big\},$$

and (2) reduces to the *proximal-point* algorithms [13], [14] if $p = 1$. The method achieves faster convergence even when it does not solve (2) exactly, as shown in [11].

B. Inexact proximal method - BLUM

Often (2) cannot be computed efficiently, hence an approximate solution is computed using an optimization procedure. The work [11] defines a set of acceptable solutions to (2) as

$$\mathcal{A}^p_{\lambda,f}(\bar{x},\beta) = \Big\{ x \in \mathbb{E} : \|\nabla f^p_{\bar{x},\lambda}(x)\| \le \beta \|\nabla f(x)\| \Big\}, \quad (3)$$

where $\beta \in [0,1)$ is a tolerance parameter. In case we have 1st-order optimality $\|\nabla f(x)\| = 0$ or we set $\beta = 0$, we have the ideal cases of $\mathcal{A}^p_{\lambda,f}(\bar{x},\beta)$ that (2) is solved exactly.

Let $A_k \coloneqq \frac{2(1-\beta)}{\lambda}(\frac{k}{2p+2})^{p+1}$, $a_{k+1} \coloneqq A_{k+1} - A_k$ and define $\Phi_0(x) \coloneqq d_{p+1}(x - x_0)$, Algorithm 1 shows the accelerated inexact pth-order proximal algorithm, abbreviated as AIPPA.

Algorithm 1 AIPPA.

Input: $x_0 \in \mathbb{E}$, $\beta \in [0,1)$, $\lambda > 0$, $\Phi_0(x) \coloneqq d_{p+1}(x - x_0)$
Output: An approximate solution to Problem (1)

1: **for** $k = 0, 1, \ldots$ **do**
2: $v_k \coloneqq \underset{x\in\mathbb{E}}{\operatorname{argmin}}\, \Phi_k(x)$ and $y_k \coloneqq \frac{A_k}{A_{k+1}}x_k + \frac{a_{k+1}}{A_{k+1}}v_k$
3: Compute $T_k \in \mathcal{A}^p_{\lambda,f}(y_k,\beta)$ and update Φ as $\Phi_{k+1}(x) = \Phi_k(x) + a_{k+1}\big(f(T_k) + \langle \nabla f(T_k), x - T_k\rangle\big)$
4: Choose x_{k+1} such that $f(x_{k+1}) \le f(T_k)$.
5: **end for**

Convergence rate: the sequence $\{x_k\}_k$ generated by AIPPA satisfies $f(x_k) - f^\star \le O(\frac{1}{k^{p+1}})$ [11, Theorem 2].

Stopping criterion: Condition (3) can be checked in practice and integrated directly as a stopping criterion for any procedure used to compute T_k given y_k.

Bi-level framework: BLUM has two levels:

- an upper-level corresponds to a chosen pth-order proximal algorithm
- a lower-level where an algorithm running on lower-order derivative (for a instance the $(p-1)$ derivatives) is used to *approximately* solve Step 3 in AIPPA.

We are now ready to discuss the set up of the BLUM framework in the newly proposed algorithm for solving CPD.

III. Proposed algorithm for CPD

Now we present a new inexact 3rd-order proximal algorithms for computing a nonnegative CPD of an input nonnegative n-way tensor, i.e., approximating a tensor with a sum of R rank-one nonnegative tensors. Let a tensor $\mathcal{T}$ has dimensions $I_1 \times I_2 \times \cdots \times I_N$, let matrices $U^{(n)}$ have size $I_n \times R$ for $1 \le n \le N$. Computing a rank-R nonnegative CPD of $\mathcal{T}$ is achieved by solving:

$$\underset{\substack{U^{(n)}\ge 0\\ 1\le n\le N}}{\operatorname{argmin}} D\Big(\mathcal{T}\big|\mathcal{I}\times_1 U^{(1)}\times_2\cdots\times_N U^{(N)}\Big) \coloneqq F\Big(\big\{U^{(n)}\big\}\Big), \quad (4)$$

where D is a measure of discrepancy between two tensors (we consider two cases later), $\mathcal{I}$ is the identity tensor of dimensions all equal to R, the symbol $\times_n$ is the n-mode product [2]. Note that the proposed method can be used to compute a low-rank matrix approximation of an matrix, which is a special case of nonnegative CPD where input $\mathcal{T}$ is a 2-way tensor and the feasible set is $\mathbb{E}$.

We solve (4) by Block-Coordinate Descent (BCD) that consists of optimizing alternatively over one factor of the factorization while the others are kept fixed at their most recent value, i.e., at each iteration we successively solve n-subproblems for CPD; say one in $U^{(n)}$ and the others in $U^{(1)}, ..., U^{(n-1)}, U^{(n+1)}, ..., U^{(N)}$ alternatively (after rewritting tensor decomposition model using tensor matricization along the different modes). Each subproblem is solved using a variant of AIPPA. In the following we present the update for one factor $U^{(n)}$ to tackle (4). The updates of the other factors are performed in a similar fashion under permutation.

Consider the update of $U^{(1)}$, after matricization along the 1st-mode we have:

$$\begin{aligned} \mathcal{T} \quad &= \mathcal{I} \times_1 U^{(1)} \times_2 ... \times_N U^{(N)} \\ \Leftrightarrow\ \mathcal{T}_{(1)} \quad &= U^{(1)} \Big(U^{(2)} \odot \cdots \odot U^{(N)}\Big)^T, \end{aligned} \quad (5)$$

where $\mathcal{T}_{(i)}$ is the unfolding of tensor $\mathcal{T}$ along mode i, $\odot$ is the Khatri-Rao product. Note that (5) is an NMF of $V = \mathcal{T}^T_{(1)}$ with factors $X = (U^{(1)})^T$ and $W = U^{(2)} \odot \cdots \odot U^{(N)}$.

We now consider two functions D, denoted as $f(X)$:

$$\min_X f(X) \coloneqq \frac{1}{2}\|V - WX\|^2_F - \gamma \sum_{i,j} \log(X_{ij}), \quad (6)$$

$$\min_X f(X) \coloneqq \frac{1}{24}\|V - WX\|^4_4 - \gamma \sum_{i,j} \log(X_{ij}), \quad (7)$$

where $\gamma \ge 0$ is a penalty weight and $\log(x)$ is the log-barrier function that promotes nonnegativity of x. Note that both f in (6) and (7) are separable w.r.t. each column of X, i.e., denoting $V_{:,j} = v$ and $X_{:,j} = x$ (the j-th column of V and X resp.), solving (6) and (7) boils down to solving m minimization problems in parallel (m = number of columns of X), where each minimization being performed over one particular column of X. For instance (6) becomes:

$$\min_{x\in\mathbb{E}} f(x) \coloneqq \frac{1}{2}\|v - Wx\|^2_F - \gamma \sum_i \log(x_i).$$

To solve each of these subproblems in the CPD, we use a variant of AIPPA with $p = 3$ and $\beta := \frac{1}{p}$ (in the "upper-level" part). For such choices, we assume that the 4th derivative of f is bounded on $Q \subset \mathbb{E}$ with constant $M_4 > 0$ and let $\lambda = 3M_4$ [11], here Q is the nonnegative orthant of appropriate size.

If we drop the constraints (set $\gamma = 0$), one need to only consider (7) for computing X since $M_4 = 0$ for (6) (and then loosing high convergence rates offered by the higher-order proximal algorithm).

A. The step 2 in AIPPA

We now present solution to the problem at step 2 in AIPPA, i.e., solving the minimization $v_k = \operatorname{argmin}_{x \in \mathbb{E}} \Phi_k(x)$, where

$$\Phi_k(x) = \Phi_{k-1}(x) + a_k\Big(f(T_k) + \big\langle \nabla f(T_k), x - T_k \big\rangle\Big).$$

Define $g_0 = 0$ and $g_k = g_{k-1} + a_k \nabla f(T_k)$, the above minimization problem is simplified to

$$v_k = \operatorname*{argmin}_{x \in \mathbb{E}} g_k^T x + d_{p+1}(x - x_k),$$

which has an optimal solution

$$v_k^\star = x_k - \frac{g_k}{\|g_k\|^{1-\frac{1}{p}}}.$$

Note that y_k can be written as

$$y_k = x_k - \left(1 - \Big(\frac{k}{k+1}\Big)^4\right) \frac{g_k}{\|g_k\|^{1-\frac{1}{p}}}.$$

B. The step 3 in AIPPA

Two issues arise when solving the higher-order proximal at step 3 in AIPPA.

a) Determine M (the upper bound of directional derivative): For (7), one can show that computing the uniform upper-bound for the 4th derivative boils down to solving an eigenvalue problem for the 4th order derivative tensor

$$\mathcal{I} \times_1 W^T \times_2 W^T \times_3 W^T \times_4 W^T.$$

Moreover, if $\gamma > 0$, one can easily see that $D^4 \log(x_i)$ is not bounded above for $x_i \to 0$. In this case, we tune numerically M_4 along iteration of our algorithm to ensure numerical stability and the objective function along iterations is monotonically decreasing.

b) Solving the high-order prox: Recall Section II, a crucial step for AIPPA is the computation of the "lower-level part in BLUM: solving $T_k \in \mathcal{A}_{H,f}(y_k, \frac{1}{3})$, where we dropped the symbol p to ease the notation. In our case, we want to compute an approximate solution for the following 3rd-order proximal operator:

$$\mathbf{prox}_{f,3M_4}(y_k) := \operatorname*{argmin}_{x \in \mathbb{E}} \Big\{ \bar{f}(x) := f(x) + 3M_4 d_4(x - y_k) \Big\}, \tag{8}$$

where f is the objective function of (6) or (7).

For an efficient computation of T_k, we use Bregman gradient descent (BGD, Algorithm 2), where the update of x_{i+1} involved the minimization of the linearized f, denoted as $\bar{f}$ in (8).

Algorithm 2 Bregman gradient descent (BGD)

Input: Given y_k, β, M_4, $\gamma \geq 0$, set $x_0 = y_k$
Output: An approximate solution to Problem (8)
1: **while** $\|\nabla f_{3M_4,y_k}(x_i)\| > \beta \|\nabla f(x_i)\|$ **do**
2: $\quad x_{i+1} := \operatorname*{argmin}_{x \in \mathbb{E}} \langle \nabla \bar{f}(x_i), x - x_i \rangle + L\beta_{\rho_{y_k}}(x_i, x)$
3: **end while**

In step 2 of BGD, we use $L = \frac{3}{2}$ (suggested in [11]), and the term $\beta_{\rho_{y_k}}(x_i, x)$ is the Bregman divergence between x_i and x with respect to ρ_{y_k} , defined as follows [11]:

$$\beta_{\rho_{y_k}}(x_i, x) = \rho_{y_k}(x) - \rho_{y_k}(x_i) - \langle \nabla \rho_{y_k}(x_i), x - x_i \rangle,$$

where

$$\rho_{y_k} := \frac{1}{2}\langle \nabla^2 f(y_k)(x - y_k), x - y_k \rangle + 3M_4 d_4(x - y_k).$$

After some algebra, step 2 of BGD can be simplified to a quartic minimization

$$x_{i+1} = \operatorname*{argmin}_x \frac{(x - y_k)^T Q (x - y_k)}{2} + \frac{2g_{ki}^T x}{3} + \frac{3M_4}{4}\|x - y_k\|^4,$$

where

- $Q = \nabla^2 f(y_k)$ is Hessian of $f(x)$,
- $g_{ki} = \nabla f(y_k) - \frac{3}{2} Q(x_i - y_k) - \frac{3M_4}{4}\|x_i - y_k\|^2 (x_i - y_k)$.

We now discuss how to solve the quartic problem. Recall by assumption f is convex thus the Hessian admits an eigenvalue decomposition (EVD). Let $U \operatorname{diag}(\sigma) U^T$ be the EVD of the Hessian Q and let the vector $c = U^T g_{ki}$ where g_{ki} is defined above, the optimal solution $x_{i+1}^\star$ of the quartic minimization is given by

$$x_{i+1}^\star = y_k - \frac{2}{3} U \frac{c}{\sigma_i + \lambda^\star},$$

where $\lambda^\star$ is the unique nonnegative solution of the following non-linear scalar problem

$$\lambda^\star = \operatorname*{argmin}_\lambda \frac{M_4}{3}\left(\sum_n \frac{c_n^2}{(\sigma_n + \lambda)^2}\right)^2 - \sum_n \frac{c_n^2 (\lambda + 1/2 s_n)}{(\sigma_n + \lambda)^2}.$$

We obtain $\lambda^\star$ by fixed-point iteration: setting the gradient of the above function w.r.t. λ to zero gives the following update

$$\lambda \leftarrow \frac{4M_4}{3} \sum_n \frac{c_n^2}{(\sigma_n + \lambda)^2}.$$

Remark: on quartic function For the cases (6) and (7) with W full rank, the function f is strongly convex. However, the model $\bar{f}$ is not globally strongly convex due to d_4 as x^4 is not bounded below by x^2 when $|x| < 1$.

Remark: why BCD At first glance, it seems possible to optimize all the factor matrices all-at-once [15] by stacking all the block variables into one large variable. However this creates an explosion in dimension, and making the BGD iteration very expensive to compute as we now need to run a huge EVD in the order of $\prod_{i=1}^N I_i$ for every iteration.

Algorithm 3 IAHOM for nonnegative CPD.

Input: a nonnegative N-way tensor, $M_4 > 0$, $\gamma \geq 0$, rank R.
Output: Nonnegative factors $U^{(1)}, ..., U^{(N)}$
Initialization: $\{U_0^{(1)}, ..., U_0^{(N)}\}$
for $k = 0, ...$ **do**
  for $n = 1, ..., N$ **do**
    Update $U_k^{(n)}$ as an inexact solution of:
    $\min_{U^{(n)} \geq 0} F(U_k^{(1)}, ..., U_k^{(n-1)}, U^{(n)}, U_{k-1}^{(n+1)}, ...)$
    by Algorithms 1 and 2.
  end for
end for

C. IAHOM: Inexact Accelerated Higher-Order Method

Lastly, Algorithm 3 IAHOM summarizes the proposed general method for computing the rank-R CPD of a N-way input tensor $\mathcal{T}$ (with or without nonnegativity constraints), for both objective functions from Problems (6) and (7).

IV. Numerical experiments

We now compare Algorithm 3 for both Problems (6) and (7), respectively dubbed as *IAHOM-O2* and *IAHOM-O4*, with the well known methods *Hierarchical Alternating Least Squares* (HALS) [16] and *SDF-NLS* [17] (a L-BFGS method) implemented in TensorLab [18]. We consider low-rank synthetic datasets: we generate each entry of $\{U^{(n)}\}$ $(1 \leq n \leq N)$ using the uniform distribution in $[0, 1]$ and let $\mathcal{T} := \mathcal{I} \times_1 U^{(1)} \times_2 \cdots \times_N U^{(N)}$. We consider $N = 3$, $R \in \{5, 10\}$ and $I_n \in \{50, 100\}$. To compare the solutions generated by the algorithms, we report the evolution of the relative data fitting error defined as

$$E(k) := \frac{\|\mathcal{T} - \mathcal{I} \times_1 U_k^{(1)} \times_2 \cdots \times_N U_k^{(N)}\|_F}{\|\mathcal{T}\|_F}$$

along iterations k. The results for the different datasets (differing in rank R and dimensions) are shown on Figs. 1-3. IAHOM-O2 and IAHOM-O4 converged faster than HALS and SDF-NLS in all the cases.

V. Conclusions and further works

We presented a first application of 3rd-order proximal methods within the BLUM framework. We extended the BLUM framework to solve constrained minimization problems, particularly with nonnegativity constraints by using log barrier. We provide two tractable algorithms to solve approximately the 3rd-order proximal operators and ultimately to compute low-rank approximation of tensors under the BCD framework.

In the experiment on synthetic data sets, we showed that the proposed algorithms can be used efficiently for computing the nonnegative CPD of an input tensor with moderate sizes. We showed that the algorithms enjoy a faster convergence than the state-of-the-art method.

This work is only preliminary and further works will focus:

- developing new approaches for more constraints,
- developing efficient routines for the fast estimation of the uniform bound M, which is critical for the algorithm to have a faster convergence,
- the theoretical analysis of BLUM framework for constrained minimization problems.

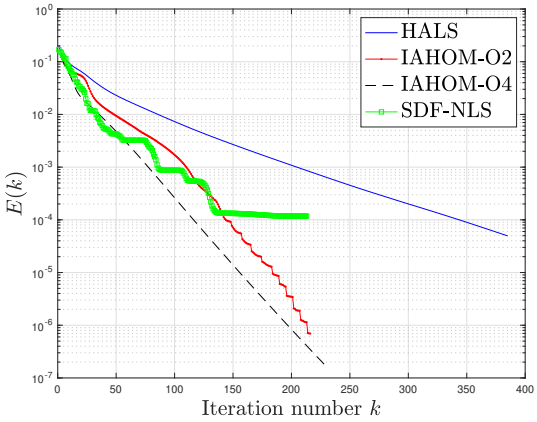

Fig. 1. Results for $[I_1, I_2, I_3, R] = [50, 50, 50, 5]$.

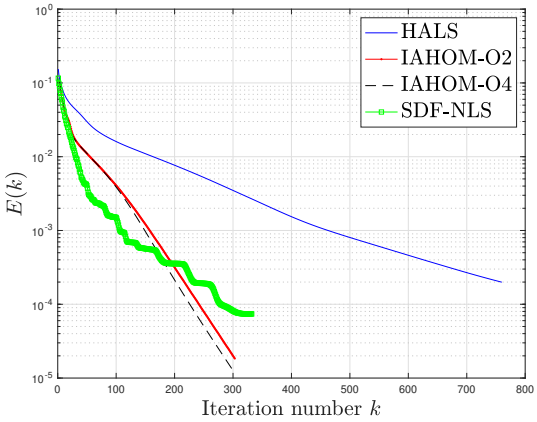

Fig. 2. Results for $[I_1, I_2, I_3, R] = [50, 50, 50, 10]$.

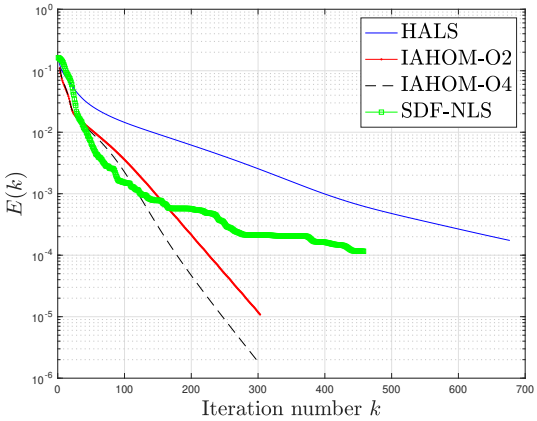

Fig. 3. Results for $[I_1, I_2, I_3, R] = [100, 100, 100, 10]$.

REFERENCES

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[2] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. John Wiley & Sons, 2009.

[3] N. Gillis, "The why and how of nonnegative matrix factorization," *Connections*, vol. 12, no. 2, 2014.

[4] A. M. S. Ang and N. Gillis, "Algorithms and comparisons of nonnegative matrix factorizations with volume regularization for hyperspectral unmixing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 4843–4853, 2019.

[5] N. Gillis, *Nonnegative matrix factorization*. SIAM, 2020.

[6] V. Leplat, N. Gillis, and A. M. Ang, "Blind audio source separation with minimum-volume beta-divergence nmf," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3400–3410, 2020.

[7] Y. Nesterov, "Superfast second-order methods for unconstrained convex optimization," *J Optim Theory Appl*, vol. 191, pp. 1–30, 2021.

[8] ——, "Implementable tensor methods in unconstrained convex optimization," *Math. Program.*, vol. 186, p. 157–183, 2021.

[9] G. Grapiglia and Y. Nesterov, "On inexact solution of auxiliary problems in tensor methods for convex optimization," *Optimization Methods and Software*, vol. 36, no. 1, pp. 145–170, 2021.

[10] ——, "Tensor methods for finding approximate stationary points of convex functions," *Optimization Methods and Software*, 2020.

[11] Y. Nesterov, "Inexact accelerated high-order proximal-point methods," *Math. Program.*, 2021.

[12] ——, "Inexact high-order proximal-point methods with auxiliary search procedure," *SIOPT*, vol. 31, no. 4, pp. 2807–2828, 2021.

[13] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM journal on control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[14] N. Parikh and S. Boyd, "Proximal algorithms." *Found.Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2013.

[15] P. Paatero, "A weighted non-negative least squares algorithm for three-way 'parafac'factor analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 38, no. 2, pp. 223–242, 1997.

[16] A. Cichocki, R. Zdunek, and S. Amari, "Hierarchical als algorithms for nonnegative matrix and 3d tensor factorization," in *Independent Component Analysis and Signal Separation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 169–176.

[17] L. Sorber, M. Van Barel, and L. De Lathauwer, "Structured data fusion," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 586–600, 2015.

[18] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab 3.0. mar. 2016. available online." https://www.tensorlab.net, 2016.