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, UK

Abstract

- Higher-order Methods (HoM) for factorization models
- Two new efficient & implementable proximal HoMs

Setup: minimizing p-times differentiable convex function

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

- $\mathbb{E} \subset \mathbb{R}^n$ a vector space (with inner product & norm)
- $f : \mathbb{E} \to \mathbb{R}$ closed, convex, p-times differentiable with bounded derivative
$$\sup_{x\in\mathbb{E}} \|D^p f(x)\| =: M_p < +\infty, \tag{2}$$
  - $D^p f(x)[h_1, ..., h_p]$: pth-order directional derivative of $f(x)$ along direction $h = [h_1, ..., h_p]$
  - $\|D^p f(x)\| = \max_h \{|D^p f(x)[h]^p| : \|h\| \le 1\}$

Higher-order proximal point methods

- pth-order prox operator $\text{prox}^p_{f,\lambda}(\bar{x}) := \operatorname*{argmin}_{x\in\mathbb{E}} \left\{ f^p_{\bar{x},\lambda} := f(x) + \lambda d_{p+1}(x - \bar{x}) \right\}$ (3)
  - $\lambda \in \mathbb{R}^+$, $p \in \mathbb{N}$, $f^p_{\bar{x},\lambda}$ is the *pth-order model*, $d_{p+1}(a) = \frac{1}{p+1}\|a\|^{p+1}$
- (3) generalizes the prox operator $\text{prox}_{f,\lambda}(\bar{x}) := \operatorname*{argmin}_{x\in\mathbb{E}} \left\{ f(x) + \frac{1}{2\lambda}\|x - \bar{x}\|^2 \right\}$
- Inexact proximal method
  - Often (3) can't be computed efficiently $\implies$ compute an approximate sol.
  - HoM achieves fast convergence even if (3) is not solved exactly
  - (Nesterov21) defines a set of acceptable sol. to (3) as
$$\mathcal{A}^p_{\lambda,f}(\bar{x}, \beta) = \left\{ x \in \mathbb{E} : \|\nabla f^p_{\bar{x},\lambda}(x)\| \le \beta \|\nabla f(x)\| \right\}, \tag{4}$$
  where $\beta \in [0, 1)$ is a tolerance parameter.
    - If 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 (3) is solved exactly.

AIPPA: Accelerated Inexact Pth-order Proximal Algorithm

Algorithm 1 AIPPA.

Input: $x_0 \in \mathbb{E}$, $\beta \in [0, 1)$, $\lambda > 0$, $\Phi_0(x) := d_{p+1}(x - x_0)$
Output: An approximate solution to Problem (1)
1: **for** $k = 0, 1, ...$ **do**
2: $v_k := \operatorname*{argmin}_{x\in\mathbb{E}} \Phi_k(x)$ and $y_k := \frac{A_k}{A_{k+1}} x_k + \frac{a_{k+1}}{A_{k+1}} v_k$
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**

$A_k := \frac{2(1-\beta)}{\lambda}\left(\frac{k}{2p+2}\right)^{p+1}$

$a_{k+1} := A_{k+1} - A_k$

$\Phi_0(x) := d_{p+1}(x - x_0)$

We pick $\beta = \frac{1}{3}$ for CPD.

- **Convergence** (Nesterov21): $\{x_k\}$ from AIPPA satisfies $f(x_k) - f^\star \le O(\frac{1}{k^{p+1}})$
- (4) can be used as stopping criterion for procedure used to get T_k given y_k
- **BLUM Bi-level framework**: has two levels:
  - up-lv corresponds to a chosen pth-order proximal algo
  - low-lv where an algo running on low-order derivative is used to *approximately* solve Step 3 in AIPPA.
- Step 2 in AIPPA: $v_k = \operatorname*{argmin}_{x\in\mathbb{E}} \Phi_{k-1}(x) + a_k\Big(f(T_k) + \langle \nabla f(T_k), x - T_k\rangle\Big)$

  Let $g_0 = 0$, $g_k = g_{k-1} + a_k \nabla f(T_k)$, the problem is simplified to
$$v_k = \operatorname*{argmin}_{x\in\mathbb{E}} \; g_k^T x + d_{p+1}(x - x_k),$$
  which has optimal sol $v_k^\star = x_k - g_k/(\|g_k\|^{1-1/p})$

3rd-order algo for CPD

- Rank-R nonnegative CPD of a tensor $\mathcal{T}$
$$\operatorname*{argmin}_{\substack{U^{(n)}\ge 0\\ 1\le n\le N}} D\Big(\mathcal{T} | \mathcal{I} \times_1 U^{(1)} \times_2 \cdots \times_N U^{(N)}\Big) := F\Big(\{U^{(n)}\}\Big), \tag{5}$$
- We solve (5) by BCD, where each subproblem is solved using AIPPA.
- Why BCD: all-at-once approach is too expensive for HoM.
- We consider two functions D, denoted as $f(X)$:

2-norm-power-2 : $$\min_X f(X) := \frac{1}{2}\|V - WX\|_F^2 - \gamma \sum_{i,j} \log(X_{ij}) \tag{6}$$

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

  - X is the matrix variable of mode-n factor
  - W is the Khatri-Rao product of other factors
  - V is input tensor $\mathcal{T}$ in mode-n unfolding
  - negative log is for nonnegativity constraint and γ is a parameter

For example, (6) can be cast as $\min_{x\in\mathbb{E}} f(x) := \frac{1}{2}\|v - Wx\|_F^2 - \gamma \sum_i \log(x_i)$, which belongs to problem class (1).

Step 3 in AIPPA

- **Determine M (upper bound of directional derivative in (2))** For the 4-norm-power-4 (7), computing M for the 4th derivative boils down to eigenvalue problem for 4th-order tensor $\mathcal{I} \times_1 W^T \times_2 W^T \times_3 W^T \times_4 W^T$.
- **Solving high-order prox** Computing $T_k \in \mathcal{A}_{H,f}(y_k, \frac{1}{3})$ is equivalent to
$$\text{prox}_{f,3M_4}(y_k) := \operatorname*{argmin}_{x\in\mathbb{E}} \left\{ \bar{f}(x) := f(x) + 3M_4 d_4(x - y_k) \right\}, \tag{8}$$
where f is the cost function of (6) or (7).

We use Bregman gradient descent to compute T_k: update x_{i+1} via minimizing the linearized f (the $\bar{f}$ in (8))

Algorithm 2 Bregman gradient descent (BGD)

Input: Given y_k, β, M_4, $\gamma \ge 0$, 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: $x_{i+1} := \operatorname*{argmin}_{x\in\mathbb{E}} \langle \nabla \bar{f}(x_i), x - x_i\rangle + L\beta_{\rho_{y_k}}(x_i, x)$
3: **end while**

Step 2 of BGD is a **quartic minimization problem**
$$x_{i+1} = \operatorname*{argmin}_x \frac{(x - y_k)^T Q (x - y_k)}{2} + \frac{2g_{ki}^T x}{3} + \frac{3M_4}{4}\|x - y_k\|^4,$$
where $Q = \nabla^2 f(y_k)$ is Hessian of f, and the linear term $g_{ki} = \nabla f(y_k) - \frac{3}{2}Q(x_i - y_k) - \frac{3M_4}{4}\|x_i - y_k\|^2(x_i - y_k)$.

- **Solving quartic problem** f convex $\implies$ Hessian Q has eigenvalue decomposition $Q = U\text{diag}(\sigma)U^T$. Let $c = U^T g_{ki}$, then optimal $x^\star_{i+1}$
$$x^\star_{i+1} = y_k - \frac{2}{3}U\frac{c}{\sigma_i + \lambda^\star},$$
where $\lambda^\star$ is the unique nonnegative sol of scalar problem
$$\lambda^\star = \operatorname*{argmin}_\lambda \frac{M_4}{3}\left(\sum_n \frac{c_n^2}{(\sigma_n + \lambda)^2}\right)^2 - \sum_n \frac{c_n^2(\lambda + 1/2s_n)}{(\sigma_n + \lambda)^2},$$
can be solved numerically.

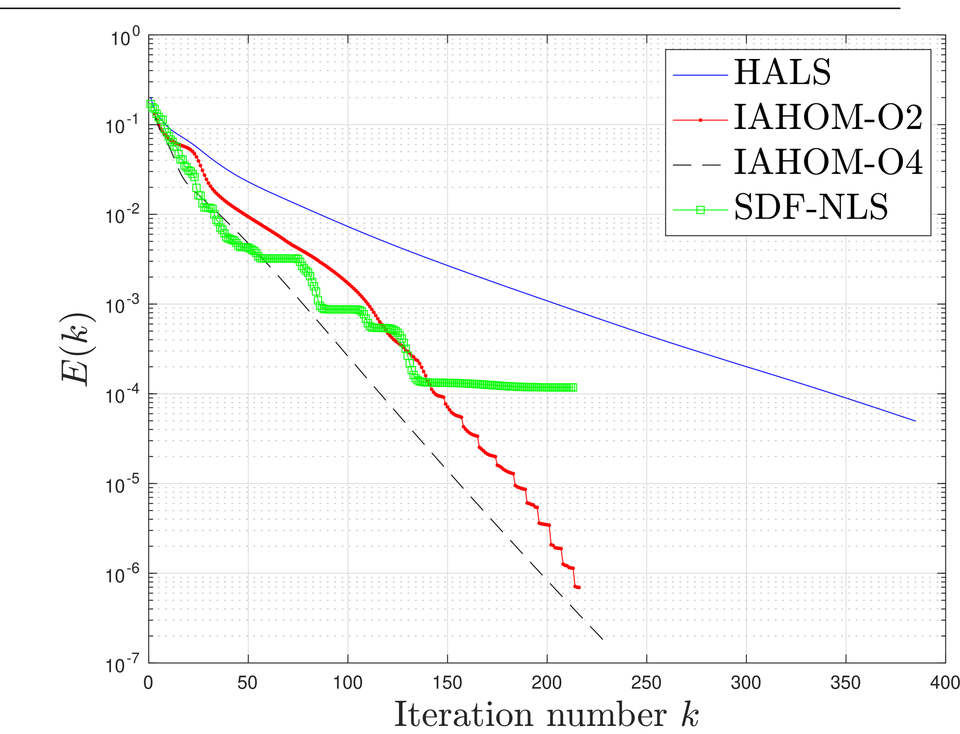
IAHOM: Inexact Accelerated HoM (= AIPPA for nonneg. CPD)

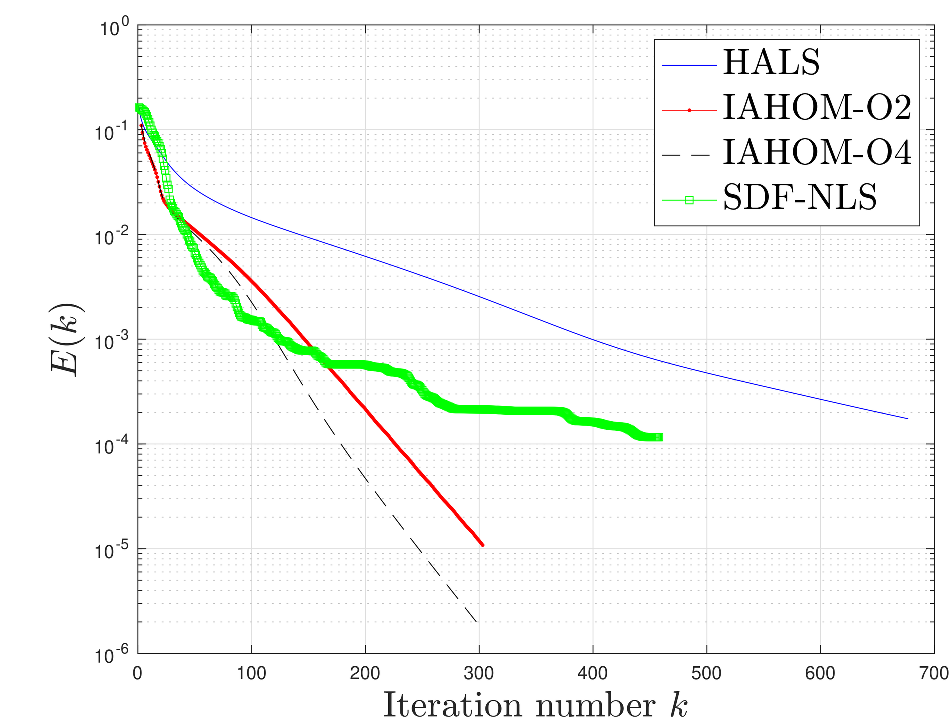
Algorithm 3 IAHOM for nonnegative CPD.

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

Numerical results

- IAHOM-O2 = AIPPA for (6)
- IAHOM-O4 = AIPPA for (7)
- compare with
  - HALS (hierarchical alternating least squares)
  - SDF-NLS (a L-BFGS method)
- $E(k)$ relative fitting error
$$\frac{\|\mathcal{T} - \mathcal{I} \times_1 U_k^{(1)} \times_2 \cdots \times_N U_k^{(N)}\|_F}{\|\mathcal{T}\|_F}$$
- Test on order-3 tensor
- Data generated $\mathcal{U}[0, 1]$ for all factor matrices
- Test cases $[I_1, I_2, I_3, R]$
  - $[50, 50, 50, 5]$
  - $[100, 100, 100, 10]$
- IAHOM-O2 & O4 are faster in all cases.





Other information

Contact
V.Leplat@skoltech.ru a.phan@skoltech.ru andersen.ang@soton.ac.uk

22nd IEEE Statistical Signal Processing Workshop 2-5 July 2023, Hanoi, Vietnam

IEEE Advancing Technology for Humanity IEEE Signal Processing Society CELEBRATING 75 YEARS IEEE SSP 2023 HÀ NỘI VIỆT NAM