Inertial Block Proximal Methods for Non-Convex Non-Smooth Optimization

L. T. K. Hien [1] N. Gillis [1] P. Patrinos [2]

[1]University of Mons

[2]KU Leuven

The 37th International Conference on Machine Learning
ICML 2020

Overview

1. Problem set up
   - Motivation
   - Block Coordinate Descent Methods
2. The proposed methods: IBP and IBPG
   - Extension to Bregman divergence
3. Convergence Analysis
   - Subsequential convergence
   - Global convergence
4. Application to NMF
5. Preliminary numerical results

Problem set up

We consider the following non-smooth non-convex optimization problem

$$\min_{x \in \mathbb{E}} F(x), \quad \text{where } F(x) := f(x) + g(x), \tag{1}$$

and

- x is partitioned into s blocks/groups of variables: $x = (x_1, \ldots, x_s) \in \mathbb{E} = \mathbb{E}_1 \times \ldots \times \mathbb{E}_s$ with $\mathbb{E}_i$, $i = 1, \ldots, s$, being finite dimensional real linear spaces equipped with the norm $\|\cdot\|_{(i)}$ and the inner product $\langle \cdot, \cdot \rangle_{(i)}$,
- $f : \mathbb{E} \to \mathbb{R}$ is a continuous but possibly non-smooth non-convex function, and
- $g(x) = \sum_{i=1}^{s} g_i(x_i)$ with $g_i : \mathbb{E}_i \to \mathbb{R} \cup \{+\infty\}$ for $i = 1, \ldots, s$ are proper and lower semi-continuous functions.

Nonnegative matrix factorization – A motivation

NMF

Given $X \in \mathbb{R}_+^{\mathbf{m} \times \mathbf{n}}$ and the integer $\mathbf{r} < \min(\mathbf{m}, \mathbf{n})$, solve

$$\min_{U \geq 0, V \geq 0} \frac{1}{2} \|X - UV\|_F^2 \text{ such that } U \in \mathbb{R}_+^{\mathbf{m} \times \mathbf{r}} \text{ and } V \in \mathbb{R}_+^{\mathbf{r} \times \mathbf{n}}.$$

NMF is a key problem in data analysis and machine learning with applications in

- image processing,
- document classification,
- hyperspectral unmixing,
- audio source separation.

Nonnegative matrix factorization – A motivation

NMF

Given $X \in \mathbb{R}_+^{\mathbf{m}\times\mathbf{n}}$ and the integer $\mathbf{r} < \min(\mathbf{m}, \mathbf{n})$, solve

$$\min_{U\geq 0, V\geq 0} \frac{1}{2}\|X - UV\|_F^2 \text{ such that } U \in \mathbb{R}_+^{\mathbf{m}\times\mathbf{r}} \text{ and } V \in \mathbb{R}_+^{\mathbf{r}\times\mathbf{n}}.$$

Let $f(U, V) = \frac{1}{2}\|X - UV\|_F^2$,
$g_1(U) = \mathbb{I}_{\mathbb{R}_+^{\mathbf{m}\times\mathbf{r}}}(U)$, and
$g_2(V) = \mathbb{I}_{\mathbb{R}_+^{\mathbf{r}\times\mathbf{n}}}(V)$.
NMF is rewritten as
$\min_{U,V} f(U, V) + g_1(U) + g_2(V)$.

Let $f(U_{:i}, V_{i:}) = \frac{1}{2}\left\|X - \sum_{i=1}^{\mathbf{r}} U_{:i}V_{i:}\right\|_F^2$,
$g_i(U_{:i}) = \mathbb{I}_{\mathbb{R}_+^{\mathbf{m}}}(U_{:i})$, $i = 1, \ldots, \mathbf{r}$, and
$g_{i+\mathbf{r}}(V_{i:}) = \mathbb{I}_{\mathbb{R}_+^{\mathbf{n}}}(V_{i:})$, $i = 1, \ldots, \mathbf{r}$.
NMF is rewritten as
$\min_{U_{:i}, V_{i:}} f(U_{:i}, V_{i:}) + \sum_{i=1}^{\mathbf{r}} g_i(U_{:i}) + \sum_{i=\mathbf{r}+1}^{2\mathbf{r}} g_i(V_{i:})$.

Non-negative approximate canonical polyadic decomposition (NCPD)

We consider the following NCPD problem: given a non-negative tensor $T \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and a specified order $\mathbf{r}$, solve

$$\min_{X^{(1)},\ldots,X^{(N)}} f := \frac{1}{2} \left\| T - X^{(1)} \circ \ldots \circ X^{(N)} \right\|_F^2 \qquad (2)$$
$$\text{such that} \quad X^{(n)} \in \mathbb{R}_+^{I_n \times \mathbf{r}}, n = 1, \ldots, N,$$

where the Frobenius norm of a tensor $T \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is defined as $\|T\|_F = \sqrt{\sum_{i_1,\ldots,i_N} T^2_{i_1 i_2 \ldots i_N}}$, and the tensor product $X = X^{(1)} \circ \ldots \circ X^{(N)}$ is defined as $X_{i_1 i_2 \ldots i_N} = \sum_{j=1}^{\mathbf{r}} X^{(1)}_{i_1 j} X^{(2)}_{i_2 j} \ldots X^{(N)}_{i_N j}$, for $i_n \in \{1, \ldots, I_n\}$, $n = 1, \ldots, N$. Here $X^{(n)}_{ij}$ is the (i,j)-th element of $X^{(n)}$. Let $g_i(X^{(i)}) = \mathbb{I}_{\mathbb{R}_+^{I_i \times \mathbf{r}}}(X^{(i)})$. NCPD is rewritten as

$$\min_{X^{(1)},\ldots,X^{(N)}} f(X^{(1)}, \ldots, X^{(N)}) + \sum_{i=1}^{N} g_i(X^{(i)}).$$

Block Coordinate Descent Methods

Initialize: Choosing initial point $x^{(0)}$ and other parameters.

for $k = 1, \dots$ **do**

  for $i = 1, \dots, s$ **do**

    Fix the latest values of the blocks $j \neq i$:
    $\left(x_1^{(k)}, \dots, x_{i-1}^{(k)}, x_i, x_{i+1}^{(k-1)}, \dots, x_s^{(k-1)}\right)$

    Update block i to get
    $\left(x_1^{(k)}, \dots, x_{i-1}^{(k)}, x_i^{(k)}, x_{i+1}^{(k-1)}, \dots, x_s^{(k-1)}\right)$

  end for

end for

Algorithm 1: General framework of BCD methods.

Block Coordinate Descent Methods

Denote $f_i^{(k)}(x_i) := f\left(x_1^{(k)}, \ldots, x_{i-1}^{(k)}, x_i, x_{i+1}^{(k-1)}, \ldots, x_s^{(k-1)}\right)$.
(**First order**) BCD methods can typically be classified into three categories:

1. Classical BCD methods update each block of variables as follows

$$x_i^{(k)} = \operatorname*{argmin}_{x_i \in \mathbb{E}_i} f_i^{(k)}\left(x_i\right) + g_i\left(x_i\right).$$

   - $\oplus$ converge to a stationary point under suitable convexity assumptions.
   - $\ominus$ fails to converge for some non-convex problems.

Block Coordinate Descent Methods

2. Proximal BCD methods update each block of variables as follows

$$x_i^{(k)} = \operatorname*{argmin}_{x_i \in \mathbb{E}_i} f_i^{(k)}(x_i) + g_i(x_i) + \frac{1}{2\beta_i^{(k)}} \left\| x_i - x_i^{(k-1)} \right\|^2 .$$

- The authors in [1] established, for the first time, the convergence of $\{x^{(k)}\}$ to a critical point of F with non-convex setting and $s = 2$.

[1] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka - Lojasiewicz inequality. Mathematics of Operations Research, 35(2) : 438–457, 2010.

Block Coordinate Descent Methods

3. Proximal gradient BCD methods update each block of variables as follows

$$x_i^{(k)} = \operatorname*{argmin}_{x_i \in \mathbb{E}_i} \left\langle \nabla f_i^{(k)} \left(x_i^{(k-1)} \right), x_i - x_i^{(k-1)} \right\rangle + g_i\left(x_i\right) + \frac{1}{2\beta_i^{(k)}} \left\| x_i - x_i^{(k-1)} \right\|^2 .$$

When $g_i(x_i) = \mathbb{I}_{X_i}(x_i)$ and $\|\cdot\|$ is Frobenius norm, we have

$$x_i^{(k)} = \mathrm{Proj}_{X_i}(x_i^{(k-1)} - \beta_i^{(k)} \nabla f_i^{(k)}(x_i^{(k-1)})).$$

⊕ In the general non-convex setting, Bolte et al in [2] proved the convergence of $\left\{x^{(k)}\right\}$ to a critical point of F when $s = 2$.

[2] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Mathematical Programming, 146(1) : 459–494, Aug 2014.

Gradient descent method

When $\mathbb{E} = \mathbb{R}^n$, $s = 1$, $g(x) = 0$ and $\|\cdot\|$ is Frobenius norm, proximal gradient BCD amounts to gradient descent method for unconstrained optimization problem $\min_{x \in \mathbb{R}^n} f(x)$:

$$x_{k+1} = x_k - \beta_k \nabla f(x_k).$$

Some remarks

- It is a descent method when β_k is appropriately chosen.
- In the convex setting, the method does not have the optimal convergence rate.

Acceleration by extrapolation

Heavy-ball method of Polyak [3]:

$$x_{k+1} = x_k - \beta_k \nabla f(x_k) + \theta_k(x_k - x_{k-1}).$$

Accelerated gradient method of Nesterov [4]:

$$\begin{aligned} y_k &= x_k + \theta_k(x_k - x_{k-1}) \\ x_{k+1} &= y_k - \beta_k \nabla f(y_k) = x_k - \beta_k \nabla f(y_k) + \theta_k(x_k - x_{k-1}) \end{aligned}$$

Some remarks:

- they are not descent methods,
- in the convex setting, these methods are proved to achieve the optimal convergence rate.

[3] B. Polyak. Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics, 4(5) : 1–17, 1964.

[4] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Soviet Mathematics Doklady, 27(2), 1983.

Let's recall

1. Classical BCD

$$x_i^{(k)} = \underset{x_i \in \mathbb{E}_i}{\operatorname{argmin}}\, f_i^{(k)}\left(x_i\right) + g_i\left(x_i\right).$$

2. Proximal BCD

$$x_i^{(k)} \;=\; \underset{x_i \in \mathbb{E}_i}{\operatorname{argmin}}\, f_i^{(k)}\left(x_i\right) + g_i\left(x_i\right) + \frac{1}{2\beta_i^{(k)}}\left\|x_i - x_i^{(k-1)}\right\|^2.$$

3. Proximal gradient BCD

$$x_i^{(k)} = \underset{x_i \in \mathbb{E}_i}{\operatorname{argmin}} \left\langle \nabla f_i^{(k)}\left(x_i^{(k-1)}\right), x_i \right\rangle + g_i\left(x_i\right) + \frac{1}{2\beta_i^{(k)}}\left\|x_i - x_i^{(k-1)}\right\|^2.$$

The proposed methods: IBP and IBPG

Initialize: Choose $\tilde{x}^{(0)} = \tilde{x}^{(-1)}$.
for $k = 1, \dots$ **do**
$x^{(k,0)} = \tilde{x}^{(k-1)}$.
for $j = 1, \dots, T_k$ **do**
Choose $i \in \{1, \dots, s\}$. Let y_i be the value of the ith block before it was updated to $x_i^{(k,j-1)}$.
Extrapolate

$$\hat{x}_i = x_i^{(k,j-1)} + \alpha_i^{(k,j)} \left(x_i^{(k,j-1)} - y_i \right), \tag{3}$$

and compute

$$x_i^{(k,j)} = \operatorname*{argmin}_{x_i} F_i^{(k,j)}(x_i) + \frac{1}{2\beta_i^{(k,j)}} \|x_i - \hat{x}_i\|^2 . \tag{4}$$

Let $x_{i'}^{(k,j)} = x_{i'}^{(k,j-1)}$ for $i' \neq i$.
end for
Update $\tilde{x}^{(k)} = x^{(k,T_k)}$.
end for

Algorithm 2: IBP

Initialize: Choose $\tilde{x}^{(0)} = \tilde{x}^{(-1)}$.
for $k = 1, \dots$ **do**
$x^{(k,0)} = \tilde{x}^{(k-1)}$.
for $j = 1, \dots, T_k$ **do**
Choose $i \in \{1, \dots, s\}$. Let y_i be the value of the ith block before it was updated to $x_i^{(k,j-1)}$.
Extrapolate

$$\begin{aligned} \hat{x}_i &= x_i^{(k,j-1)} + \alpha_i^{(k,j)} \left(x_i^{(k,j-1)} - y_i \right), \\ \grave{x}_i &= x_i^{(k,j-1)} + \gamma_i^{(k,j)} \left(x_i^{(k,j-1)} - y_i \right), \end{aligned} \tag{5}$$

and compute

$$\begin{aligned} x_i^{(k,j)} = \operatorname*{argmin}_{x_i} &\langle \nabla f_i^{(k,j)}(\grave{x}_i), x_i - x_i^{(k,j-1)} \rangle \\ &+ g_i(x_i) + \frac{1}{2\beta_i^{(k,j)}} \|x_i - \hat{x}_i\|^2. \end{aligned} \tag{6}$$

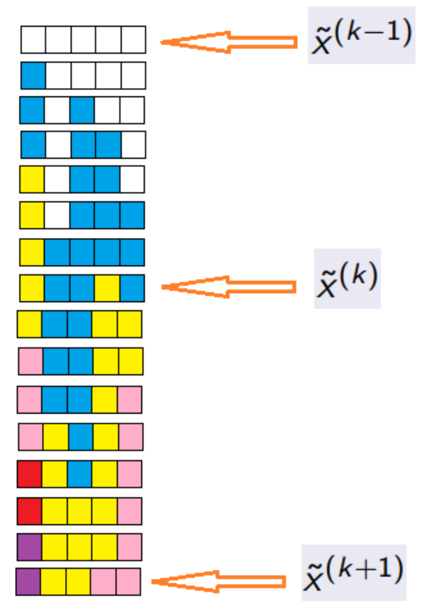
Let $x_{i'}^{(k,j)} = x_{i'}^{(k,j-1)}$ for $i' \neq i$.
end for
Update $\tilde{x}^{(k)} = x^{(k,T_k)}$.
end for

Algorithm 3: IBPG

Assumption 1

For all k, all blocks are updated after the T_k iterations performed within the kth outer loop, and there exists a positive constant $\bar{T}$ such that $s \leq T_k \leq \bar{T}$.

An illustration



Notations

Table: Notation

Notation	Definition
$x^{(k,j)}$	x at the jth iteration within the kth outer loop
$\tilde{x}^{(k)}$	the main generated sequence (the output)
T_k	number of iterations within the kth outer loop
$f_i^{(k,j)}(x_i)$	a function of the ith block while fixing the latest updated values of the other blocks, i.e.,
$= f(x_1^{(k,j-1)}, \ldots,$	$x_{i-1}^{(k,j-1)}, x_i, x_{i+1}^{(k,j-1)}, \ldots, x_s^{(k,j-1)})$
$F_i^{(k,j)}(x_i)$	$F_i^{(k,j)}(x_i) = f_i^{(k,j)}(x_i) + g_i(x_i)$
$\bar{x}_i^{(k,m)}$	the value of block i after it has been updated m times during the kth outer loop
d_i^k	the total number of times the ith block is updated during the kth outer loop
$\bar{\alpha}_i^{(k,m)}$	the values of $\alpha_i^{(k,j)}$,
$\bar{\beta}_i^{(k,m)}$	the values of $\beta_i^{(k,j)}$,
$\bar{\gamma}_i^{(k,m)}$	and the values of $\gamma_i^{(k,j)}$ that are used in (3), (4), (5), (6), (7) and (8) to update block i from $\bar{x}_i^{(k,m-1)}$ to $\bar{x}_i^{(k,m)}$
$\{\bar{x}_i^{(k,m)}\}_{k\geq 1}$	the sequence that contains the updates of the ith block, i.e., $\{\ldots, \bar{x}_i^{(k,1)}, \ldots, \bar{x}_i^{(k,d_i^k)}, \ldots\}$

Extension to Bregman divergence

Definition (Bregman distance)

Let $H_i : \mathbb{E}_i \to \mathbb{R}$ be a strictly convex function that is continuously differentiable. The Bregman distance associated with H_i is defined as:

$$D_i(u, v) = H_i(u) - H_i(v) - \langle \nabla H_i(v), u - v \rangle, \forall u, v \in \mathbb{E}_i.$$

Example:

- Let $H_i(u) = \frac{1}{2}\|u\|_2^2$, we have $D_i(u, v) = \frac{1}{2}\|u - v\|_2^2$.

Definition (Bregman proximal map)

For a given $v \in \mathbb{E}_i$, and a positive number β, the Bregman proximal map of a function ϕ is defined by

$$\mathrm{prox}_{\beta,\phi}^{H_i}(v) := \operatorname{argmin}\left\{\phi(u) + \frac{1}{\beta} D_i(u, v) : u \in \mathbb{E}_i\right\}.$$

Definition

For given $u_1 \in \operatorname{int}\operatorname{dom}\varphi$, $u_2 \in \mathbb{E}_i$ and $\beta > 0$, the Bregman proximal gradient map of a pair of non-convex function (ϕ, φ) (φ is continuously differentiable) is defined by

$$\mathrm{Gprox}_{\beta,\phi,\varphi}^{H_i}(u_1, u_2) := \operatorname{argmin}\left\{\phi(u) + \langle \nabla\varphi(u_1), u\rangle + \frac{1}{\beta} D_i(u, u_2) : u \in \mathbb{E}_i\right\}$$

Extension to Bregman divergence

Initialize: Choose $\tilde{x}^{(0)} = \tilde{x}^{(-1)}$.
for $k = 1, \ldots$ **do**
  $x^{(k,0)} = \tilde{x}^{(k-1)}$.
  for $j = 1, \ldots, T_k$ **do**
    Choose $i \in \{1, \ldots, s\}$ such that Assumption 1 is satisfied.
    Update of IBP: extrapolate as in (3) and compute

$$x_i^{(k,j)} \in \operatorname{prox}^{H_i}_{\beta_i^{(k,j)}, F_i^{(k,j)}} (\hat{x}_i) . \tag{7}$$

    Update of IBPG: extrapolate as in (5) and compute

$$x_i^{(k,j)} \in \operatorname{Gprox}^{H_i}_{\beta_i^{(k,j)}, g_i, f_i^{(k,j)}} (\grave{x}_i, \hat{x}_i) . \tag{8}$$

    Let $x_{i'}^{(k,j)} = x_{i'}^{(k,j-1)}$ for $i' \neq i$.
  end for
  Update $\tilde{x}^{(k)} = x^{(k,T_k)}$.
end for

Algorithm 4: IBP and IBPG with Bregman divergence

Convergence Analysis

Assumptions

- The function H_i, $i = 1, \ldots, s$, is σ_i-strongly convex, continuously differentiable and ∇H_i is L_{H_i}-Lipschitz continuous.
  Examples: The Euclidean distance (or, more generally, a quadratic entropy distance) is a typical example of a Bregman distance that satisfies this assumption. A non-typical simple example of H_i is $x \in \mathbb{R} \mapsto \log(x + \sqrt{1 + x^2}) + x^2$.
- The proximal maps are well-defined.
- The function F is bounded from below.
- Considering Algorithm IBPG, we need to assume that $\nabla f_i^{(k,j)}$ is $L_i^{(k,j)}$-Lipschitz continuous, with $L_i^{(k,j)} > 0$. For notational clarity, we correspondingly use $\overline{L}_i^{(k,m)}$ for $L_i^{(k,j)}$.

Subsequential convergence of IBP

Choosing parameters for IBP: Let $0 < \nu < 1$. For $m = 1, \ldots, d_i^k$ and $i = 1, \ldots, s$, denote $\theta_i^{(k,m)} = \frac{\left(L_{H_i}\bar{\alpha}_i^{(k,m)}\right)^2}{2\nu\sigma_i\bar{\beta}_i^{(k,m)}}$. Let $\theta_i^{(k,d_i^k+1)} = \theta_i^{(k+1,1)}$. We choose $\bar{\alpha}_i^{(k,m)}$ and $\bar{\beta}_i^{(k,m)}$ satisfying $\frac{(1-\nu)\sigma_i}{2\bar{\beta}_i^{(k,m)}} \geq \delta\theta_i^{(k,m+1)}$, for $m = 1, \ldots, d_i^k$, where $\delta > 1$.

Assumption

There exist positive numbers W_1, $\overline{\alpha}$ and $\underline{\beta}$ such that $\theta_i^{(k,m)} \geq W_1$, $\bar{\alpha}_i^{(k,m)} \leq \overline{\alpha}$ and $\underline{\beta} \leq \bar{\beta}_i^{(k,m)}$ for all $k \in \mathbb{N}$, $m = 1, \ldots, d_i^k$ and $i = 1, \ldots, s$.

Theorem

If F is regular then every limit point of $\left\{\tilde{x}^{(k)}\right\}_{k\in\mathbb{N}}$ is a critical point type I of F. If f is continuously differentiable then every limit point of $\left\{\tilde{x}^{(k)}\right\}_{k\in\mathbb{N}}$ is a critical point type II of F.

Some definitions

- For any $x \in \operatorname{dom} \varphi$, and $d \in \mathbb{E}$, we denote the directional derivative of φ at x in the direction d by

$$\varphi'(x; d) = \liminf_{\tau \downarrow 0} \frac{\varphi(x + \tau d) - \varphi(x)}{\tau}.$$

- For each $x \in \operatorname{dom} \varphi$, we denote $\hat{\partial}\varphi(x)$ as the Frechet subdifferential of φ at x which contains vectors $v \in \mathbb{E}$ satisfying

$$\liminf_{y \neq x, y \to x} \frac{1}{\|y - x\|} \left(\varphi(y) - \varphi(x) - \langle v, y - x \rangle\right) \geq 0.$$

  If $x \notin \operatorname{dom} \varphi$, then we set $\hat{\partial}\varphi(x) = \emptyset$.

- The limiting-subdifferential $\partial\varphi(x)$ of φ at $x \in \operatorname{dom} \varphi$ is

$$\partial\varphi(x) := \Big\{ v \in \mathbb{E} : \exists x^{(k)} \to x,\ \varphi\big(x^{(k)}\big) \to \varphi(x),\ v^{(k)} \in \hat{\partial}\varphi\big(x^{(k)}\big),$$
$$v^{(k)} \to v \Big\}.$$

Some definitions

- We say that $x^* \in \mathrm{dom}\, F$ is a critical point type I of F if $F'(x^*; d) \geq 0, \forall\, d$.
- We say that F is regular at $x \in \mathrm{dom}\, F$ if for all $d = (d_1, \ldots, d_s)$ such that $F'\left(z; (0, \ldots, d_i, \ldots, 0)\right) \geq 0, i = 1, \ldots, s$, then $F'(x; d) \geq 0$.
- We call $x^* \in \mathrm{dom}\, \mathrm{F}$ a critical point type II of F if $0 \in \partial F\left(x^*\right)$.

We note that if x^* is a minimizer of F then x^* is a critical point type I and type II of F.

Subsequential convergence of IBPG

Choosing parameters for IBPG: Choose $\bar{\beta}_i^{(k,m)} = \frac{\sigma_i}{\kappa \bar{L}_i^{(k,m)}}$ with $\kappa > 1$.

Let $0 < \nu < 1$. For $m = 1, \ldots, d_i^k$, and $i = 1, \ldots, s$ denote $\lambda_i^{(k,m)} = \frac{1}{2}\left(\bar{\gamma}_i^{(k,m)} + \frac{\kappa L_{H_i} \bar{\alpha}_i^{(k,m)}}{\sigma_i}\right)^2 \frac{\bar{L}_i^{(k,m)}}{\nu(\kappa-1)}$. Let $\lambda_i^{(k,d_i^k+1)} = \lambda_i^{(k+1,1)}$. We choose $\bar{\alpha}_i^{(k,m)}$, $\bar{\beta}_i^{(k,m)}$ and $\bar{\gamma}_i^{(k,m)}$ satisfying $\frac{(1-\nu)(\kappa-1)\bar{L}_i^{(k,m)}}{2} \geq \delta \lambda_i^{(k,m+1)}$, for $m = 1, \ldots, d_i^k$, where $\delta > 1$.

Assumption

There exist positive numbers W_1, $\overline{L}$, $\overline{\alpha}$ and $\overline{\gamma}$ such that $\lambda_i^{(k,m)} \geq W_1$, $\bar{L}_i^{(k,m)} \leq \overline{L}$, $\bar{\alpha}_i^{(k,m)} \leq \overline{\alpha}$ and $\bar{\gamma}_i^{(k,m)} \leq \overline{\gamma}$ for all $k \in \mathbb{N}$, $m = 1, \ldots, d_i^k$ and $i = 1, \ldots, s$.

Theorem

Every limit point of $\left\{\tilde{x}^{(k)}\right\}_{k \in \mathbb{N}}$ is a critical point type II of F.

Relaxing conditions for block-convex F

For IBP, if F is block-wise convex then we can choose $\bar{\alpha}_i^{(k,m)}$ and $\bar{\beta}_i^{(k,m)}$ satisfying

$$\frac{2(1-\nu)\sigma_i}{\bar{\beta}_i^{(k,m)}} \geq \delta\theta_i^{(k,m+1)}, \quad \text{for } m = 1, \ldots, d_i^k. \tag{9}$$

This condition allows larger values of $\bar{\alpha}_i^{(k,m)}$ when using the same $\bar{\beta}_i^{(k,m)}$.

Relaxing conditions for convex g_i's

For IBPG, if the functions g_i's are convex we can use

$$\bar{\beta}_i^{(k,m)} = \sigma_i/\bar{L}_i^{(k,m)}, \qquad \lambda_i^{(k,m)} = \frac{1}{2}\left(\bar{\gamma}_i^{(k,m)} + \frac{L_{H_i}\bar{\alpha}_i^{(k,m)}}{\sigma_i}\right)^2 \frac{\bar{L}_i^{(k,m)}}{\nu},$$

and choose $\bar{\alpha}_i^{(k,m)}$ and $\bar{\gamma}_i^{(k,m)}$ satisfying $\frac{(1-\nu)\bar{L}_i^{(k,m)}}{2} \geq \delta\lambda_i^{(k,m+1)}$ for $m = 1, \ldots, d_i^k$,. This condition allows a larger stepsize.

Relaxing conditions for block-convex f and convex g_i's

For IBPG, if the g_i's are convex and $f(x)$ is block-wise convex, then we can use larger extrapolation parameters. Specifically, we choose $H_i(x_i) = \frac{1}{2}\|x_i\|^2$ and let $\bar{\beta}_i^{(k,m)} = 1/\bar{L}_i^{(k,m)}$ and

$$\lambda_i^{(k,m)} = \left(\left(\bar{\gamma}_i^{(k,m)}\right)^2 + \frac{\left(\bar{\gamma}_i^{(k,m)} - \bar{\alpha}_i^{(k,m)}\right)^2}{\nu} \right) \frac{\bar{L}_i^{(k,m)}}{2},$$

where $0 < \nu < 1$, and choose $\bar{\alpha}_i^{(k,m)}$ and $\bar{\gamma}_i^{(k,m)}$ satisfying

$$\frac{1-\nu}{2} \bar{L}_i^{(k,m)} \geq \delta \lambda_i^{(k,m+1)}, \text{for } m = 1, \ldots, d_i^k.$$

Global convergence

We modify the proof recipe proposed by J. Bolte, S. Sabach, and M. Teboulle (*Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Mathematical Programming, 146(1) : 459–494, Aug 2014*) so that it is applicable to our proposed methods.

Definition (KL function)

A function $\phi(x)$ is said to have the Kurdyka-Łojasiewicz (KL) property at $\bar{x} \in \operatorname{dom} \partial \phi$ if there exists $\eta \in (0, +\infty]$, a neighborhood U of $\bar{x}$ and a concave function $\xi : [0, \eta) \to \mathbb{R}_+$ that is continuously differentiable on $(0, \eta)$, continuous at 0, $\xi(0) = 0$, and $\xi'(s) > 0$ for all $s \in (0, \eta)$, such that for all $x \in U \cap [\phi(\bar{x}) < \phi(x) < \phi(\bar{x}) + \eta]$, the following inequality holds

$$\xi' \left(\phi(x) - \phi(\bar{x}) \right) \operatorname{dist} \left(0, \partial \phi(x) \right) \geq 1.$$

If $\phi(x)$ satisfies the KL property at each point of $\operatorname{dom} \partial \phi$ then ϕ is a KL function.

Some noticeable examples include real analytic functions, semi-algebraic functions, locally strongly convex functions.

Theorem (Global convergence recipe)

Let $\Phi : \mathbb{R}^N \to (-\infty, +\infty]$ be a proper and lower semicontinuous function which is bounded from below. Let $\mathcal{A}$ be a generic algorithm which is assumed to generate a bounded sequence $\left\{z^{(k)}\right\}_{k\in\mathbb{N}}$ by

$$z^{(0)} \in \mathbb{R}^N, z^{(k+1)} \in \mathcal{A}\left(z^{(k)}\right), \quad k = 0, 1, \ldots .$$

Assume that there exist positive constants ρ_1, ρ_2 and ρ_3 and a nonnegative sequence $\{\zeta_k\}_{k\in\mathbb{N}}$ such that the following conditions are satisfied

(B1) **Sufficient decrease property**:

$$\rho_1 \left\|z^{(k)} - z^{(k+1)}\right\|^2 \le \rho_2 \zeta_k^2 \le \Phi\left(z^{(k)}\right) - \Phi\left(z^{(k+1)}\right), \quad \forall k = 0, 1, \ldots$$

(B2) **Boundedness of subgradient**:

$$\left\|w^{(k+1)}\right\| \le \rho_3 \zeta_k, \quad w^{(k)} \in \partial\Phi\left(z^{(k)}\right), \quad \forall k = 0, 1, \ldots$$

Furthermore, assume that

(B3) **KL property**: *Φ is a KL function.*

(B4) **A continuity condition**: *If a subsequence $\left\{z^{(k_n)}\right\}_{n\in\mathbb{N}}$ of $\left\{z^{(k)}\right\}$ converges to $\bar{z}$ then $\Phi\left(z^{(k_n)}\right) \to \Phi(\bar{z})$ as $n \to \infty$.*

Then we have $\sum_{k=1}^{\infty} \zeta_k < \infty$, and $\left\{z^{(k)}\right\}$ converges to a critical point type II of Φ.

Convergence rate

The following theorem establish the convergence rate under Łojasiewicz property.

Theorem

Suppose Φ is a KL function and $\xi(a)$ of the KL function definition has the form $\xi(a) = Ca^{1-\omega}$ for some $C > 0$ and $\omega \in [0, 1)$. Then we have

(i) *If $\omega = 0$ then $\{z^{(k)}\}$ converges after a finite number of steps.*

(ii) *If $\omega \in (0, 1/2]$ then there exists $\omega_1 > 0$ and $\omega_2 \in [0, 1)$ such that $\left\| z^{(k)} - \bar{z} \right\| \leq \omega_1 \omega_2^k$.*

(iii) *If $\omega \in (1/2, 1)$ then there exists $\omega_1 > 0$ such that $\left\| z^{(k)} - \bar{z} \right\| \leq \omega_1 k^{-(1-\omega)/(2\omega-1)}$.*

Theorem (Global convergence of IBP and IBPG)

Assumption

- *The sequences* $\left\{\tilde{x}^{(k)}\right\}_{k\in\mathbb{N}}$ *generated by IBP and IBPG are bounded. (Note: this condition is satisfied when* F *has bounded level sets).*
- f *is continuously differentiable and* ∇f *is Lipschitz continuous on bounded subsets of* $\mathbb{E}$*.*
- *There exists a constant* W_2 *such that, for all* $k\in\mathbb{N}$, $m=1,\dots,d_i^k$ *and* $i=1,\dots,s$*, we have* $\theta_i^{(k,m)}\le W_2$ *for IBP,* $\lambda_i^{(k,m)}\le W_2$ *for IBPG and* $\delta>(L_H W_2)/(\sigma W_1)$*.*
- *Assume* F *is a KL-function.*

Then the whole sequence $\left\{\tilde{x}^{(k)}\right\}_{k\in\mathbb{N}}$ *generated by IBP or IBPG converges to a critical point type II of* F*.*

Applying IBPG to solve NMF with $s = 2$

$$\min_{U,V} \frac{1}{2} \|X - UV\|_F^2 + \mathbb{I}_{\mathbb{R}_+^{\mathbf{m}\times\mathbf{r}}}(U) + \mathbb{I}_{\mathbb{R}_+^{\mathbf{r}\times\mathbf{n}}}(V).$$

- We choose the Frobenius norm for (6). We have $\nabla_U f = UVV^T - XV^T$ and $\nabla_V f = U^T UV - U^T X$, hence (6) is a projected gradient step.
- IBPG should update U or V several times before updating the other one. This strategy accelerates the algorithm compared to the pure cyclic update rule, see [5].

Choosing parameters

We have $\bar{L}_1^{(k,m)} = \tilde{L}_1^{(k)} = \left\| \left(\tilde{V}^{(k-1)}\right)^T \tilde{V}^{(k-1)} \right\|$, and $\bar{L}_2^{(k,m)} = \tilde{L}_2^{(k)} = \left\| \left(\tilde{U}^{(k)}\right)^T \tilde{U}^{(k)} \right\|$ for $m \geq 1$.

We choose $\bar{\beta}_i^{(k,m)} = 1/\tilde{L}_i^{(k)}$, $\bar{\gamma}_i^{(k,m)} = \min\left\{ \frac{\tau_k - 1}{\tau_k}, \breve{\gamma}\sqrt{\frac{\tilde{L}_i^{(k-1)}}{\tilde{L}_i^{(k)}}} \right\}$, and $\bar{\alpha}_i^{(k,m)} = \breve{\alpha}\bar{\gamma}_i^{(k,m)}$, where $\tau_0 = 1$, $\tau_k = \frac{1}{2}(1 + \sqrt{1 + 4\tau_{k-1}^2})$, $\breve{\gamma} = 0.99$ and $\breve{\alpha} = 1.01$.

The parameters satisfy the relaxing conditions for block-convex f and convex g_i's. IBPG for NMF guarantees a subsequential convergence.

---

[5] N. Gillis and F. Glineur. Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. Neural Computation, 24(4):10851105, 2012.

Applying IBP to solve NMF with $s = 2\mathbf{r}$

$$\min_{U_{:i}, V_{i:}} \frac{1}{2}\|X - \sum_{i=1}^{\mathbf{r}} U_{:i} V_{i:}\|_F^2 + \sum_{i=1}^{\mathbf{r}} \mathbb{I}_{\mathbb{R}_+^{\mathbf{m}}}(U_{:i}) + \sum_{i=\mathbf{r}+1}^{2\mathbf{r}} \mathbb{I}_{\mathbb{R}_+^{\mathbf{n}}}(V_{i:}).$$

Applying IBP:

- We choose the Frobenius norm for (4). Equation (4) has the closed form solution

$$\begin{aligned} \operatorname*{argmin}_{U_{:i} \geq 0} \sum \frac{1}{2}\Big\|X - \sum_{q=1}^{i-1} U_{:q} V_{q:} - \sum_{q=i+1}^{r} U_{:q} V_{q:} - U_{:i} V_{i:}\Big\|^2 \\ + \frac{1}{2\beta_i}\Big\|U_{:i} - \hat{U}_{:i}\Big\|^2 \\ = \max\left(0, \frac{X V_{i:}^T - (UV) V_{i:}^T + U_{:i} V_{i:} V_{i:}^T + 1/\beta_i \hat{U}_{:i}}{V_{i:} V_{i:}^T + 1/\beta_i}\right), \end{aligned}$$

- IBP should update the columns of U and the rows of V several times before doing so for the other one.

Choosing parameters

We choose $1/\beta_i^{(k,m)} = 0.001$ and $\alpha_i^{(k,m)} = \tilde{\alpha}^{(k)} = \min(\bar{\beta}, \gamma\tilde{\alpha}^{(k-1)})$, with $\bar{\beta} = 1$, $\gamma = 1.01$ and $\tilde{\alpha}^{(1)} = 0.6$.

These parameters satisfy the global convergence conditions, hence IBP for NMF guarantees a global convergence.

Preliminary numerical results

We use the following notations for NMF algorithms:

- IBP: this is our proposed IBP algorithm.
- IBPG: this is our proposed IBPG algorithm when U and V are cyclically updated.
- IBPG-A: this is our proposed IBPG algorithm when we update U several times before updating V, and vice versa.
- iPALM: the inertial proximal alternating linearized minimization method proposed in [6].
- A-HALS: the accelerated hierarchical alternating least squares algorithm in [7].
- E-A-HALS: the acceleration version of A-HALS using extrapolation points proposed in [8]. This algorithm was experimentally shown to outperform A-HALS. This is, as far as we know, one of the most efficient NMF algorithms. Note that E-A-HALS is a heuristic with no convergence guarantees.
- APGC: the accelerated proximal gradient coordinate descent method proposed in [9].

---

[6] T. Pock and S. Sabach. Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. SIAM Journal on Imaging Sciences, 9(4):1756–1787, 2016.

[7] N. Gillis and F. Glineur. Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. Neural Computation, 24(4):1085–1105, 2012.

[8] A. M. S. Ang and N. Gillis. Accelerating nonnegative matrix factorization algorithms using extrapolation. Neural Computation, 31(2):417–439, 2019.

[9] Y. Xu and W. Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM Journal on Imaging Sciences, 6(3):1758–1789, 2013.

We define relative errors

$$\mathrm{relerror}_k = \frac{\left\| X - \tilde{U}^{(k)} \tilde{V}^{(k)} \right\|_F}{\|X\|_F}.$$

We let

- $e_{\min} = 0$ for the experiments with low-rank synthetic data sets, and
- in the other experiments, $e_{\min}$ is the lowest relative error obtained by any algorithms with any initializations

We define

$$E(k) = \mathrm{relerror}_k - e_{\min}.$$

Low-rank synthetic data sets

- Two low-rank matrices of size 200×200 and 200×500 are generated by letting $X = UV$, where U and V are generated by MATLAB commands $rand(\mathbf{m}, \mathbf{r})$ and $rand(\mathbf{r}, \mathbf{n})$ respectively, with $\mathbf{r} = 20$.
- For each matrix X, we run all algorithms with the same 50 random initializations $W_0 = rand(\mathbf{m}, \mathbf{r})$ and $V_0 = rand(\mathbf{r}, \mathbf{n})$, and for each initialization we run each algorithm for 20 seconds.

Low-rank synthetic data sets

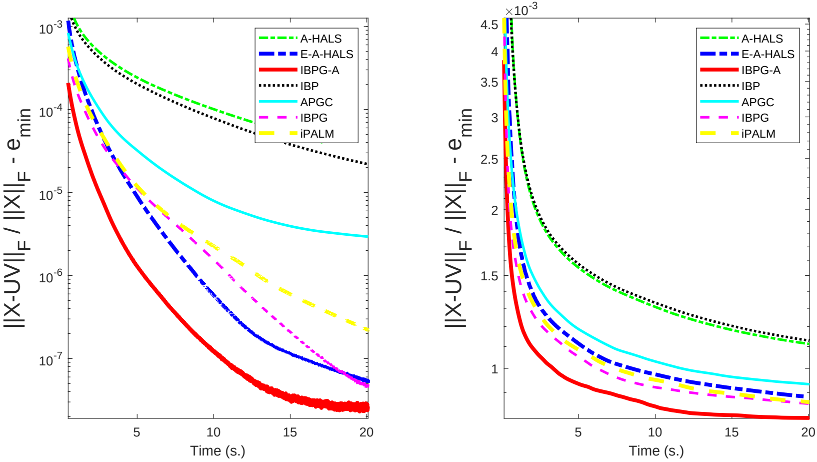


Figure: Average value of $E(k)$ with respect to time on 2 random low-rank matrices: 200×200 (left) and 200×500 (right).

Low-rank synthetic data sets

To compare the accuracy of the solutions, we generate 80 random low-rank $\mathbf{m} \times \mathbf{n}$ matrices, $\mathbf{m}$ and $\mathbf{n}$ are random integer numbers in the interval [200,500]. For each X we run the algorithms for 20 seconds with 1 random initialization.

Table: Average, standard deviation and ranking of the value of $E(k)$ at the last iteration among the different runs on the low-rank synthetic data sets. The best performance is highlighted in bold.

Algorithm	mean ± std	ranking
A-HALS	$1.227\,10^{-3} \pm 7.365\,10^{-4}$	(1, 0, 3, 4, 7, 24, 41)
E-A-HALS	$8.501\,10^{-4} \pm 6.882\,10^{-4}$	(16, 10, 12, 13, 17, 3, 9)
IBPG-A	$\mathbf{5.036\,10^{-4}} \pm 5.522\,10^{-4}$	(**39**, 10, 14, 10, 3, 2, 2)
IPG	$1.209\,10^{-3} \pm 7.386\,10^{-4}$	(0, 3, 5, 7, 15, 39, 11)
APGC	$8.726\,10^{-4} \pm 6.561\,10^{-4}$	(3, 10, 14, 22, 18, 3, 10)
IBPG	$6.621\,10^{-4} \pm 6.371\,10^{-4}$	(17, 17, 15, 11, 14, 2, 4)
iPALM	$6.759\,10^{-4} \pm 6.302\,10^{-4}$	(17, 22, 13, 12, 6, 7, 3)

Full-rank synthetic data sets

- Two full-rank matrices of size 200×200 and 200×500 are generated by MATLAB command $X = rand(m, n)$. We take $\mathbf{r} = 20$.
- For each matrix X, we run all algorithms with the same 50 random initializations $W_0 = rand(\mathbf{m}, \mathbf{r})$ and $V_0 = rand(\mathbf{r}, \mathbf{n})$, and for each initialization we run each algorithm for 20 seconds.

Full-rank synthetic data sets

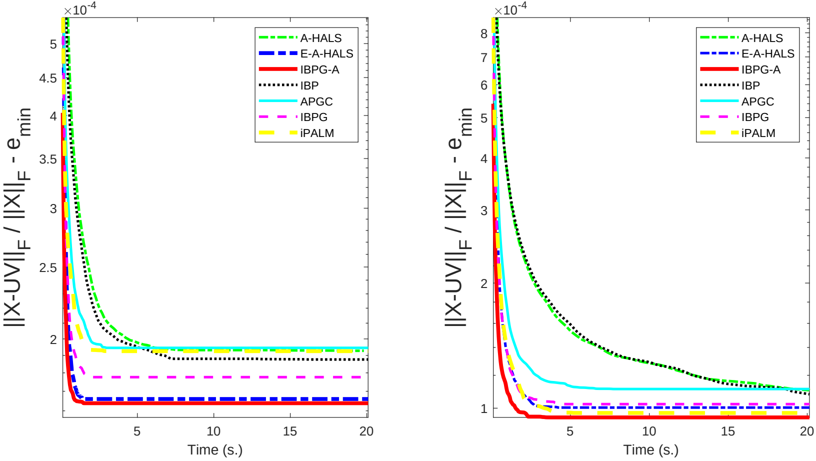


Figure: Average value of $E(k)$ with respect to time on 2 random full-rank matrices: 200×200 (left) and 200×500 (right).

Full-rank synthetic data sets

We then generate 80 full-rank matrices $X = rand(m, n)$, with **m** and **n** being random integer numbers in the interval [200,500]. For each matrix X, we run the algorithms for 20 seconds with a single random initialization.

Table: Average, standard deviation and ranking of the value of $E(k)$ at the last iteration among the different runs on full-rank synthetic data sets. The best performance is highlighted in bold.

Algorithm	mean $\pm$ std	ranking
A-HALS	$0.450056 \pm 7.688\,10^{-3}$	(5, 17, 11, 10, 10, 11, 16)
E-A-HALS	$0.450055 \pm 7.684\,10^{-3}$	(13, 11, 8, 17, 8, 7, 16)
IBPG-A	$\mathbf{0.450052} \pm 7.682\,10^{-3}$	(**25**, 5, 11, 7, 7, 16, 9)
IPG	$0.450057 \pm 7.686\,10^{-3}$	(14, 14, 10, 10, 11, 16, 5)
APGC	$0.450060 \pm 7.682\,10^{-3}$	(7, 7, 18, 12, 12, 9, 15)
IBPG	$0.450062 \pm 7.671\,10^{-3}$	(13, 10, 10, 10, 18, 7, 12)
iPALM	$0.450060 \pm 7.683\,10^{-3}$	(4, 15, 12, 15, 15, 12, 7)

Experiments with real data sets

We test the algorithms on Urban and San Diego data sets. We choose the rank $\mathbf{r} = 10$. For each data set, we generate 35 random initializations and for each initialization we run each algorithm for 200 seconds.

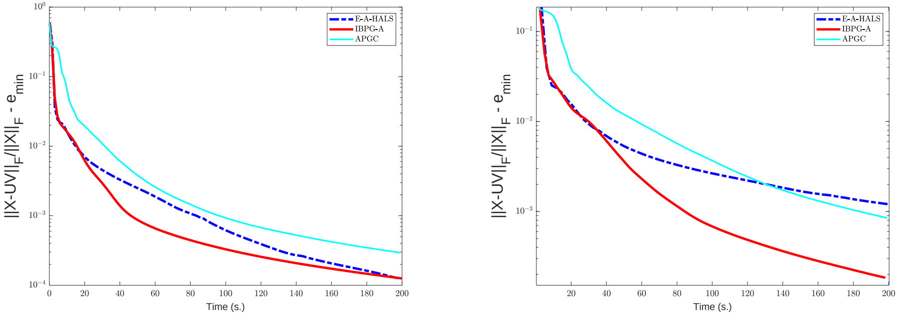


Figure: Average value of $E(k)$ with respect to time on 2 hyperspectral images: urban (the left) and SanDiego (the right).

Dense hyperspectral images

Table: Average error, standard deviation and ranking among the different runs for urban and SanDiego data sets.

Algorithm	mean $\pm$ std	ranking
E-A-HALS	$0.018823 \pm 6.739\ 10^{-4}$	(17, 28, 25)
IBPG-A	$\mathbf{0.018316} \pm 9.745\ 10^{-4}$	(**53**, 15, 2)
APGC	$0.018728 \pm 7.779\ 10^{-4}$	(0, 27, 43)

More experiments on NMF and NCPD can be found in the supplementary material of our paper.

Thank you!