Block Coordinate Descent for Regularized Multi-convex Optimization

Yangyang Xu and Wotao Yin

CAAM Department, Rice University

February 12, 2013

Outline

Multi-convex optimization

- Model definition
- Applications

Algorithms and existing convergence results

- Algorithm framework
- Three block-update schemes

Convergence

- Subsequence convergence
- Global convergence and rate
- Kurdyka-Łojasiewicz (KL) property

Numerical experiments

- Nonnegative matrix factorization
- Nonnegative 3-way tensor factorization
- Nonnegative 3-way tensor completion

Conclusions

Slides, paper, and matlab demos at: www.caam.rice.edu/~optimization/bcu

Regularized multi-convex optimization

Model

$$\underset{\mathbf{x}}{\text{minimize}}\, F(\mathbf{x}_1, \cdots, \mathbf{x}_s) \equiv f(\mathbf{x}_1, \cdots, \mathbf{x}_s) + \sum_{i=1}^{s} r_i(\mathbf{x}_i),$$

where

1. f is differentiable and multi-convex, generally non-convex; e.g., $f(x_1, x_2) = x_1^2 x_2^2 + 2x_1^2 + x_2$;
2. each r_i is convex, possibly non-smooth; e.g., $r_i(\mathbf{x}_i) = \|\mathbf{x}_i\|_1$;
3. r_i is defined on $\mathbb{R} \cup \infty$; it can enforce $\mathbf{x}_i \in \mathcal{X}_i$ by setting

$$r_i(\mathbf{x}_i) = \delta_{\mathcal{X}_i}(\mathbf{x}_i) = \begin{cases} 0, & \text{if } \mathbf{x}_i \in \mathcal{X}_i, \\ \infty, & \text{otherwise.} \end{cases}$$

Applications

- Low-rank matrix recovery (Recht et. al, 2010)

$$\underset{\mathbf{X},\mathbf{Y}}{\text{minimize}} \ \|\mathcal{A}(\mathbf{XY}) - \mathcal{A}(\mathbf{M})\|^2 + \alpha\|\mathbf{X}\|_F^2 + \beta\|\mathbf{Y}\|_F^2$$

- Sparse dictionary learning (Mairal et. al, 2009)

$$\underset{\mathbf{D},\mathbf{X}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{DX} - \mathbf{Y}\|_F^2 + \lambda \sum_i \|\mathbf{x}_i\|_1, \text{ subject to } \|\mathbf{d}_j\|_2 \leq 1, \forall j;$$

- Blind source separation (Zibulevsky and Pearlmutter, 2001)

$$\underset{\mathbf{A},\mathbf{Y}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{AYB} - \mathbf{X}\|_F^2 + \lambda\|\mathbf{Y}\|_1, \text{ subject to } \|\mathbf{a}^j\|_2 \leq 1, \forall j;$$

- Nonnegative matrix factorization (Lee and Seung, 1999)

$$\underset{\mathbf{X},\mathbf{Y}}{\text{minimize}} \ \|\mathbf{M} - \mathbf{XY}\|_F^2, \text{ subject to } \mathbf{X} \geq 0, \mathbf{Y} \geq 0;$$

- Nonnegative tensor factorization (Welling and Weber, 2001)

$$\underset{\mathbf{A}_1,\cdots,\mathbf{A}_N \geq 0}{\text{minimize}} \ \|\boldsymbol{\mathcal{M}} - \mathbf{A}_1 \circ \mathbf{A}_2 \circ \cdots \circ \mathbf{A}_N\|_F^2;$$

Challenges

Non-convexity and non-smoothness cause

1. tricky convergence analysis;

2. expensive updates to all variables simultaneously.

Challenges

Non-convexity and non-smoothness cause

1. tricky convergence analysis;
2. expensive updates to all variables simultaneously.

Goal: to develop an efficient algorithm with simple update and global convergence (of course, to a stationary point)

Framework of block coordinate descent (BCD)[1]

$$\underset{\mathbf{x}}{\text{minimize}}\, F(\mathbf{x}_1, \cdots, \mathbf{x}_s) \equiv f(\mathbf{x}_1, \cdots, \mathbf{x}_s) + \sum_{i=1}^{s} r_i(\mathbf{x}_i)$$

Algorithm 1 Block coordinate descent

Initialization: choose $(\mathbf{x}_1^0, \cdots, \mathbf{x}_s^0)$
for $k = 1, 2, \cdots$ **do**
  for $i = 1, 2, \cdots, s$ **do**
    update $\mathbf{x}_i^k$ with all other blocks fixed
  end for
  if stopping criterion is satisfied **then**
    return $(\mathbf{x}_1^k, \cdots, \mathbf{x}_s^k)$.
  end if
end for

Throughout iterations, each block $\mathbf{x}_i$ is updated by one of the three update schemes (coming next...)

[1] block coordinate *update* (BCU) is perhaps a more accurate name

Scheme 1: block minimization

The most-often used update:

$$\mathbf{x}_i^k = \operatorname*{argmin}_{\mathbf{x}_i} F(\mathbf{x}_{<i}^k,\ \mathbf{x}_i,\ \mathbf{x}_{>i}^{k-1});$$

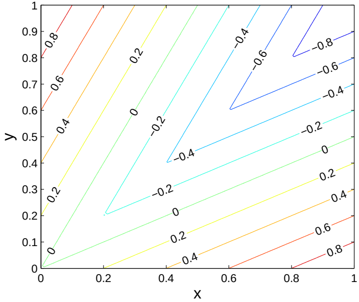
Existing results for differentiable convex F:

- Differentiable F and bounded level set $\Rightarrow$ objective converges to optimal value (Warga'63);
- Further with strict convexity $\Rightarrow$ sequence converges (Luo and Tseng'92);

Scheme 1: block minimization

Existing results for non-differentiable convex F:

- Non-differentiable F can cause stagnation at a non-critical point (Warga'63):



$$F(x, y) = |x - y| - \min(x, y), 0 \leq x, y \leq 1$$

Given y, minimizing F over x gives $x = y$. Starting from any (x^0, y^0) and cyclically updating $x, y, x, y, \cdots$ produces

$$x^k = y^k = y^0, \ k \geq 1.$$

- Non-smooth part is *separable* $\Rightarrow$ subsequence convergence (i.e., exists a limit point) (Tseng'93)

Scheme 1: block minimization

Existing results for non-convex F:

- May cycle or stagnate at a non-critical point (Powell'73):

$$F(x_1, x_2, x_3) = -x_1x_2 - x_2x_3 - x_3x_1 + \sum_{i=1}^{3}\left[(x_i - 1)_+^2 + (-x_i - 1)_+^2\right]$$

  Each $F(x_i)$ has the form $(-a)x_i + \left[(x_i - 1)_+^2 + (-x_i - 1)_+^2\right]$.

  Its minimizer: $x_i^* = \text{sign}(a)(1 + 0.5|a|)$.

  Starting from $(-1 - \epsilon, 1 + \frac{1}{2}\epsilon, -1 - \frac{1}{4}\epsilon)$ with $\epsilon > 0$, minimizing F over $x_1, x_2, x_3, x_1, x_2, x_3, \cdots$ produces:

$$\xrightarrow{x_1} (1 + \frac{1}{8}\epsilon, 1 + \frac{1}{2}\epsilon, -1 - \frac{1}{4}\epsilon) \quad \xrightarrow{x_2} (1 + \frac{1}{8}\epsilon, -1 - \frac{1}{16}\epsilon, -1 - \frac{1}{4}\epsilon)$$

$$\xrightarrow{x_3} (1 + \frac{1}{8}\epsilon, -1 - \frac{1}{16}\epsilon, 1 + \frac{1}{32}\epsilon) \quad \xrightarrow{x_1} (-1 - \frac{1}{64}\epsilon, -1 - \frac{1}{16}\epsilon, 1 + \frac{1}{32}\epsilon)$$

$$\xrightarrow{x_2} (-1 - \frac{1}{64}\epsilon, 1 + \frac{1}{128}\epsilon, 1 + \frac{1}{32}\epsilon) \xrightarrow{x_3} (-1 - \frac{1}{64}\epsilon, 1 + \frac{1}{128}\epsilon, -1 - \frac{1}{256}\epsilon)$$

Scheme 1: block minimization

Remedies for non-convex F:

- F is differentiable and *strictly quasiconvex* over each block $\Rightarrow$ limit point is a critical point (Grippo and Sciandrone'00);

  $$\text{quasiconvex: } F(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \le \max(F(\mathbf{x}), F(\mathbf{y})),\ \forall \lambda \in [0,1]$$

- F is *pseudoconvex* over every two blocks and non-differentiable part is separable $\Rightarrow$ limit point is a critical point (Tseng'01);

  $$\text{pseudoconvex: } \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle \ge 0,\ \text{some } \mathbf{g} \in \partial F(\mathbf{x}) \ \Rightarrow\ F(\mathbf{x}) \le F(\mathbf{y})$$

There is not global convergence result.

Scheme 2: block proximal descent

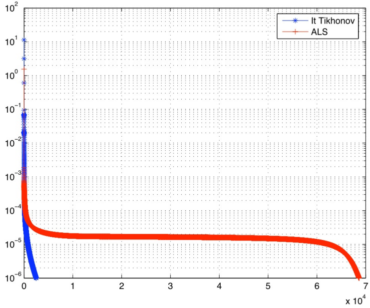
Adding $\|\mathbf{x}_i - \mathbf{x}_i^{k-1}\|_2^2$ gives better stability:

$$\mathbf{x}_i^k = \underset{\mathbf{x}_i}{\operatorname{argmin}}\, F(\mathbf{x}_{<i}^k,\ \mathbf{x}_i,\ \mathbf{x}_{>i}^{k-1}) + \frac{L_i^{k-1}}{2}\|\mathbf{x}_i - \mathbf{x}_i^{k-1}\|^2;$$

Convergence results require fewer assumptions on F:

- F is convex $\Rightarrow$ objective converges to optimal value (Auslender'92);
- F is non-convex $\Rightarrow$ limit point is stationary (Grippo and Sciandrone'00);

  Non-smooth terms must still be separable. No global convergence for non-convex F.



Also, it can reduce the "swamp effect" of scheme 1 on tensor decomposition (Navasca et. al, '08)

Scheme 3: block proximal linear

Linearize f over block i and add $\frac{L_i^{k-1}}{2}\|\mathbf{x}_i - \hat{\mathbf{x}}_i^{k-1}\|^2$:

$$\mathbf{x}_i^k = \underset{\mathbf{x}_i}{\operatorname{argmin}} \ \langle \nabla_i f(\mathbf{x}_{<i}^k, \hat{\mathbf{x}}_i^{k-1}, \mathbf{x}_{>i}^{k-1}), \mathbf{x}_i - \hat{\mathbf{x}}_i^{k-1} \rangle + r_i(\mathbf{x}_i) + \frac{L_i^{k-1}}{2}\|\mathbf{x}_i - \hat{\mathbf{x}}_i^{k-1}\|^2;$$

- Extrapolate $\hat{\mathbf{x}}_i^{k-1} = \mathbf{x}_i^{k-1} + \omega_i^{k-1}(\mathbf{x}_i^{k-1} - \mathbf{x}_i^{k-2})$ with weight $\omega_i^{k-1} \geq 0$;
- Much easier than schemes 1 & 2; may have closed-form solutions for simple r_i;
- Used in randomized BCD for differentiable convex problems (Nesterov'12);
- The update is less greedy than schemes 1 & 2, causes more iterations, but may save total time;
- Empirically, the "relaxation" tend to avoid "shallow-puddle" local minima better than schemes 1 & 2.

Comparisons

1. Block coordinate minimization (scheme 1) is mostly used
   - May generally cycle or stagnate at a non-critical point (Powell'73);
   - Globally convergent for strictly convex problem (Luo and Tseng'92);
   - For non-convex problem, each limit point is a critical point if each subproblem has unique solution and objective is regular (Tseng'01);
   - Global convergence for non-convex problems is unknown;
2. Block proximal (scheme 2) can stabilize iterates
   - Each limit point is a critical point (Grippo and Sciandrone'00);
   - Global convergence for non-convex problems is unknown;
3. Block proximal linearization (scheme 3) is often easiest
   - Very few works use this scheme for non-convex problems yet;
   - Related to the coordinate gradient descent method (Tseng and Yun'09).

Why different update schemes?

- They deal with subproblems of different properties;
- Implementations are easier for many applications;
- Schemes 2 & 3 may save total time than scheme 1;
- Convergence can be analyzed in a unified way.

Example: sparse dictionary learning

$$\underset{\mathbf{D},\mathbf{X}}{\text{minimize}}\ \frac{1}{2}\|\mathbf{D}\mathbf{X}-\mathbf{Y}\|_F^2+\|\mathbf{X}\|_1,\ \text{subject to } \|\mathbf{D}\|_F\le 1$$

apply scheme 1 to $\mathbf{D}$ and scheme 3 to $\mathbf{X}$; both are closed-form.

Convergence results

Assumptions

$$\underset{\mathbf{x}}{\text{minimize}}\, F(\mathbf{x}_1, \cdots, \mathbf{x}_s) \equiv f(\mathbf{x}_1, \cdots, \mathbf{x}_s) + \sum_{i=1}^{s} r_i(\mathbf{x}_i)$$

Assumption 1. Continuous, lower-bounded, and $\exists$ a stationary point.

Assumption 2. Each block uses only one update scheme throughout, and

1. block using scheme 1: subproblem is *strongly convex* with modulus L_i^k;
2. block using scheme 3: subproblem has *Lipschitz continuous gradient*.

Assumption 3. $\exists\ 0 < \ell \leq L < \infty$ such that $\ell \leq L_i^k \leq L, \forall i, k$.

Assumptions 1–3 are assumed for all results below.

Convergence results

Lemma 2.2 Let $\{\mathbf{x}^k\}$ be the sequence generated by BCD. If block i is updated by scheme 3, the extrapolation weight is controlled as $0 \leq \omega_i^k \leq \delta_\omega \sqrt{\frac{L_i^{k-1}}{L_i^k}}$ with $\delta_\omega < 1$ for all k. Then,

$$\sum_{i=1}^{\infty} \|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2 < \infty.$$

Theorem 2.1 (Limit point is stationary point) Under the assumptions of Lemma 2.2, any limit point of $\{\mathbf{x}^k\}$ is a stationary point.

As a trivial extension:

Theorem 2.2 (Isolated stationary points) If $\{\mathbf{x}^k\}$ is bounded and the stationary points are isolated, then $\mathbf{x}^k$ converges to a stationary point.

Remark: The isolation condition of Theorem 2.2 is difficult to check. Existing results considering non-convexity and/or non-smoothness have only subsequence convergence. We need a better tool for global convergence.

Global convergence and rate
(using the Kurdyka-Łojasiewicz property)

Theorem 2.3: Let $\{\mathbf{x}^k\}$ be the sequence of BCD. If block i is updated by Scheme 3, assume $0 \le \omega_i^k \le \delta_\omega \sqrt{\frac{L_i^{k-1}}{L_i^k}}$ with $\delta_\omega < 1$ for all k. Assume $F(\mathbf{x}^k) \le F(\mathbf{x}^{k-1})$. If $\{\mathbf{x}^k\}$ has a finite limit point $\bar{\mathbf{x}}$ and

$$\frac{|F(\mathbf{x}) - F(\bar{\mathbf{x}})|^\theta}{\text{dist}(\mathbf{0}, \partial F(\mathbf{x}))} \text{ is bounded around } \bar{\mathbf{x}} \text{ for } \theta \in [0,1), \tag{1}$$

then

$$\mathbf{x}^k \to \bar{\mathbf{x}}.$$

Theorem 2.4 (rate of convergence): In addition, in (1),

1. if $\theta = 0$, $\mathbf{x}^k$ converges to $\bar{\mathbf{x}}$ *in finitely many iterations*;
2. if $\theta \in (0, \frac{1}{2}]$, $\|\mathbf{x}^k - \bar{\mathbf{x}}\| \le C\tau^k, \ \forall k$, for certain $C > 0, \ \tau \in [0,1)$;
3. if $\theta \in (\frac{1}{2}, 1)$, $\|\mathbf{x}^k - \bar{\mathbf{x}}\| \le Ck^{-(1-\theta)/(2\theta-1)}, \ \forall k$, for certain $C > 0$.

The Kurdyka-Łojasiewicz (KL) property

Definition 2.9. (Łojasiewicz'93) $\psi(\mathbf{x})$ has the Kurdyka-Łojasiewicz (KL) property if there exists $\theta \in [0, 1)$ such that

$$\frac{|\psi(\mathbf{x}) - \psi(\bar{\mathbf{x}})|^\theta}{\mathrm{dist}(\mathbf{0}, \partial\psi(\mathbf{x}))} \tag{2}$$

is bounded around $\bar{\mathbf{x}}$.

History:

- Introduced by (Łojasiewicz'93) on *real analytic functions*, for which the term with $\theta \in [\frac{1}{2}, 1)$ in (2) is bounded around any critical point $\bar{\mathbf{x}}$.
- (Kurdyka'98) extended the properties to functions on the o*-minimal structure*.
- (Bolte et. al '07) extended the property to *nonsmooth sub-analytic functions*.

Functions satisfying the KL property

Real analytic functions (some $\theta \in [\frac{1}{2}, 1)$): $\varphi(t)$ is analytic if $\left(\frac{\varphi^{(k)}(t)}{k!}\right)^{\frac{1}{k}}$ is bounded for all k and on any compact set $\mathcal{D} \subset \mathbb{R}$. $\psi(\mathbf{x})$ on $\mathbb{R}^n$ is analytic if $\varphi(t) \triangleq \psi(\mathbf{x} + t\mathbf{y})$ is so for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Examples:

- Polynomial functions: $\|\mathbf{XY} - \mathbf{M}\|_F^2$ and $\|\boldsymbol{\mathcal{M}} - \mathbf{A}_1 \circ \mathbf{A}_2 \circ \cdots \circ \mathbf{A}_N\|_F^2$;
- $L_q(\mathbf{x}) = \sum_{i=1}^n (x_i^2 + \varepsilon^2)^{q/2} + \frac{1}{2\lambda}\|\mathbf{Ax} - \mathbf{b}\|^2$ with $\varepsilon > 0$;
- Logistic loss function

$$\psi(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-c_i(\mathbf{a}_i^\top \mathbf{x} + b)}\right)$$

.

Locally strongly convex functions ($\theta = \frac{1}{2}$): $\psi(\mathbf{x})$ is strongly convex in a neighborhood $\mathcal{D}$ with modulus μ, if for any $\gamma(\mathbf{x}) \in \partial\psi(\mathbf{x})$ and $\mathbf{x}, \mathbf{y} \in \mathcal{D}$

$$\psi(\mathbf{y}) \geq \psi(\mathbf{x}) + \langle \gamma(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2$$

Example:

- Logistic loss function: $\log(1 + e^{-t})$;

Semi-algebraic functions

$\mathcal{D} \subset \mathbb{R}^n$ is a semi-algebraic set if it can be represented as

$$\mathcal{D} = \bigcup_{i=1}^{s} \bigcap_{j=1}^{t} \{\mathbf{x} \in \mathbb{R}^n : p_{ij}(\mathbf{x}) = 0, q_{ij}(\mathbf{x}) > 0\},$$

where p_{ij}, q_{ij} are real polynomial functions for $1 \leq i \leq s$, $1 \leq j \leq t$.

ψ is a semi-algebraic function if its graph

$$\mathrm{Gr}(\psi) \triangleq \{(\mathbf{x}, \psi(\mathbf{x})) : \mathbf{x} \in \mathrm{dom}(\psi)\}$$

is a semi-algebraic set.

Properties of semi-algebraic sets and functions:

1. If a set $\mathcal{D}$ is semi-algebraic, so is its closure $\mathrm{cl}(\mathcal{D})$.
2. If $\mathcal{D}_1$ and $\mathcal{D}_2$ are both semi-algebraic, so are $\mathcal{D}_1 \cup \mathcal{D}_2$, $\mathcal{D}_1 \cap \mathcal{D}_2$ and $\mathbb{R}^n \backslash \mathcal{D}_1$.
3. Indicator functions of semi-algebraic sets are semi-algebraic.
4. Finite sums and products of semi-algebraic functions are semi-algebraic.
5. The composition of semi-algebraic functions is semi-algebraic.

Functions satisfying the KL property (cont.)

Semi-algebraic functions: some $\theta \in [0, 1)$ in (2)

- Indicator functions of polyhedral sets: $\{\mathbf{x} : \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$;
- Polynomial functions: $\|\mathbf{X}\mathbf{Y} - \mathbf{M}\|_F^2$ and $\|\boldsymbol{\mathcal{M}} - \mathbf{A}_1 \circ \mathbf{A}_2 \circ \cdots \circ \mathbf{A}_N\|_F^2$;
- ℓ_1-norm $\|\mathbf{x}\|_1$, sup-norm $\|\mathbf{x}\|_\infty$, and Euclidean norm $\|\mathbf{x}\|$;
- TV semi-norm $\|\mathbf{x}\|_{TV}$;
- Indicator functions of set of *positive semidefinite matrices*
- Finite sum, product or composition of all these functions.

Sum of real analytic and semi-algebraic functions: some $\theta \in [0, 1)$ in (2)

- Sparse logistic regression: $\frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-c_i(\mathbf{a}_i^\top \mathbf{x} + b)}\right) + \lambda\|\mathbf{x}\|_1$;

Examples of global convergence by BCD

- Low-rank matrix recovery (Recht et. al, 2010)

$$\min_{\mathbf{X},\mathbf{Y}} \|\mathcal{A}(\mathbf{XY}) - \mathcal{A}(\mathbf{M})\|_2^2 + \alpha\|\mathbf{X}\|_F^2 + \beta\|\mathbf{Y}\|_F^2$$

- Sparse dictionary learning (Mairal et. al, 2009)

$$\min_{\mathbf{D},\mathbf{X}} \frac{1}{2}\|\mathbf{DX} - \mathbf{Y}\|_F^2 + \|\mathbf{X}\|_1 + \delta_{\mathcal{D}}(\mathbf{D});\ \mathcal{D} = \{\mathbf{D} : \|\mathbf{d}_j\|_2^2 \le 1, \forall j\}$$

- Blind source separation (Zibulevsky and Pearlmutter, 2001)

$$\min_{\mathbf{A},\mathbf{Y}} \frac{\lambda}{2}\|\mathbf{AYB} - \mathbf{X}\|_F^2 + \|\mathbf{Y}\|_1 + \delta_{\mathcal{A}}(\mathbf{A});\ \mathcal{A} = \{\mathbf{A} : \|\mathbf{a}^j\|_2^2 \le 1, \forall j\}$$

- Nonnegative matrix factorization (Lee and Seung, 1999)

$$\min_{\mathbf{X},\mathbf{Y}} \|\mathbf{M} - \mathbf{XY}\|_F^2 + \delta_{\mathbb{R}_+^{m\times r}}(\mathbf{X}) + \delta_{\mathbb{R}_+^{r\times n}}(\mathbf{Y});$$

- Nonnegative tensor factorization (Welling and Weber, 2001)

$$\min_{\mathbf{A}_1,\cdots,\mathbf{A}_N} \frac{1}{2}\|\boldsymbol{\mathcal{M}} - \mathbf{A}_1 \circ \mathbf{A}_2 \circ \cdots \circ \mathbf{A}_N\|_F^2 + \sum_{n=1}^{N} \delta_{\mathbb{R}_+^{I_n\times r}}(\mathbf{A}_n);$$

Numerical results

Part I: nonnegative matrix factorization (NMF)

Model:

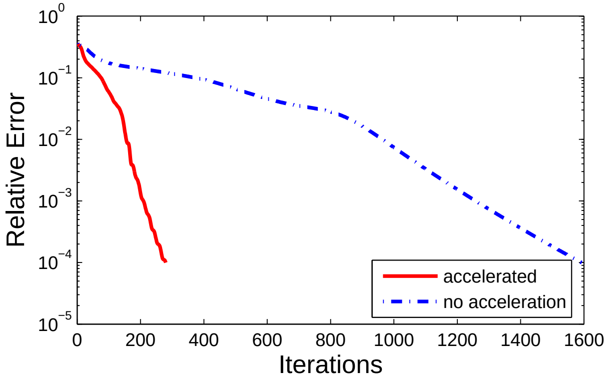
$$\underset{\mathbf{X},\mathbf{Y}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{M}-\mathbf{X}\mathbf{Y}\|_F^2, \ \text{subject to } \mathbf{X}\in\mathbb{R}_+^{m\times r}, \mathbf{Y}\in\mathbb{R}_+^{r\times n}$$

Algorithms compared:

1. APG-MF (proposed): BCD with scheme 3, $\omega_i^k = \min\left(\hat{\omega}_k, \sqrt{\frac{L_i^{k-1}}{L_i^k}}\right), i=1,2,$ where $\hat{\omega}_k = \frac{t_{k-1}-1}{t_k}$ and $t_0=1, t_k=\frac{1}{2}\sqrt{1+4t_{k-1}^2}$; $\hat{\omega}_k$ used in FISTA (Beck and Teboulle'09);
2. ADM-MF: alternating direction method for NMF (Y. Zhang'10);
3. Blockpivot-MF: BCD with block minimization (scheme 1); subproblems solved by block principle pivoting method (Kim and Park'08);
4. Als-MF and Mult-MF: Matlab's implementation.

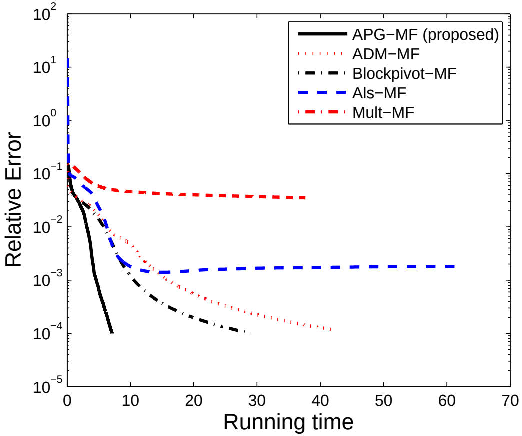
Extrapolation accelerates convergence

- Extrapolation acceleration: $\omega_i^k = \min\left(\hat{\omega}_k, \sqrt{\frac{L_i^{k-1}}{L_i^k}}\right), i = 1, 2,$ where $\hat{\omega}_k = \frac{t_{k-1}-1}{t_k}$ and $t_0 = 1, t_k = \frac{1}{2}\sqrt{1 + 4t_{k-1}^2}$;
- No acceleration: $\omega_i^k = 0, i = 1, 2;$



Comparison on synthetic data

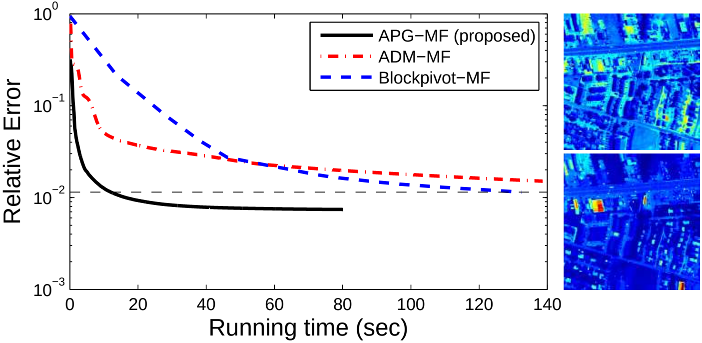
- Random $\mathbf{M} = \mathbf{LR}$ and $\mathbf{L} \in \mathbb{R}_+^{500\times 30}, \mathbf{R} \in \mathbb{R}_+^{30\times 1000}$;
- $\text{relerr} = \frac{\|\mathbf{M}-\mathbf{XY}\|_F}{\|\mathbf{M}\|_F}$ and running time (sec)



running time is second

Comparison on hyperspectral data

- $163 \times 150 \times 150$ hyperspectral cube is reshaped to 22500×163 matrix $\mathbf{M}$



Part II: Nonnegative 3-way tensor factorization

Model:

$$\underset{\mathbf{A}_1,\mathbf{A}_2,\mathbf{A}_3}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{\mathcal{M}} - \mathbf{A}_1 \circ \mathbf{A}_2 \circ \mathbf{A}_3\|_F^2, \ \text{subject to } \mathbf{A}_n \in \mathbb{R}_+^{I_n \times r}, \forall n.$$

Compared algorithms

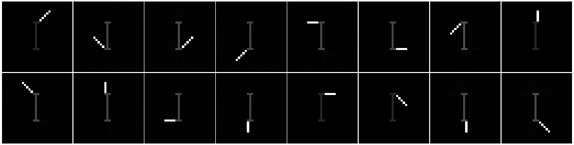
1. APG-TF (proposed) : BCD with scheme 3, $\omega_i^k = \min\left(\hat{\omega}_k, \sqrt{\frac{L_i^{k-1}}{L_i^k}}\right)$, $i = 1, 2, 3$, where $\hat{\omega}_k = \frac{t_{k-1}-1}{t_k}$ and $t_0 = 1, t_k = \frac{1}{2}\sqrt{1 + 4t_{k-1}^2}$;
2. AS-TF: BCD with scheme 1) subproblems solved by active set method (Kim et. al, '08);
3. Blockpivot-TF: BCD with scheme 1; subproblems solved by block principle pivoting method (Kim and Park '12);

Swimmer dataset[2]

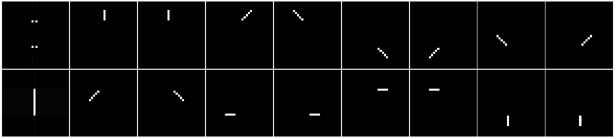
Shashua and Hazan'05: NMF tends to form invariant parts as ghosts while NTF can correctly resolve all parts

8 among 256 images

factors by NMF



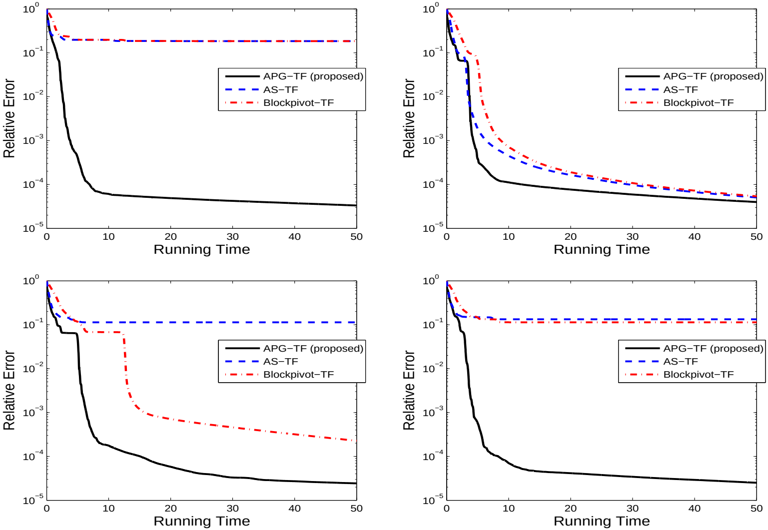
factors by NTF



[2]Donoho and Stodden'03, When does non-negative matrix factorization give a correct decomposition into parts

Comparison on the Swimmer dataset

$32 \times 32 \times 256$ nonnegative tensor $\boldsymbol{\mathcal{M}}$; run to 50 seconds; r set to 60;



Part III: Nonnegative 3-way tensor completion

Compared algorithms

- APG-TC (proposed) solves

$$\min_{\mathbf{A},\boldsymbol{\mathcal{X}}} \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \mathbf{A}_1 \circ \mathbf{A}_2 \circ \mathbf{A}_3\|_F^2, \text{ s.t. } \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{M}}), \mathbf{A}_n \in \mathbb{R}_+^{I_n \times r}, \forall n.$$

  BCD with scheme 3 applied to $\mathbf{A}$-subproblems and scheme 1 to $\boldsymbol{\mathcal{X}}$-subproblem;

- FaLRTC and HaLRTC (Liu et. al, '12) solve

$$\min_{\boldsymbol{\mathcal{X}}} \sum_{n=1}^{3} \alpha_n \|\mathbf{X}_{(n)}\|_*, \text{ subject to } \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{M}}) \qquad (3)$$

  - FaLRTC first smoothes (3) and then applies an accelerated proximal gradient method;
  - HaLRTC applies an alternating direction method to (3).

Comparison on synthetic data

- Random $\mathcal{M} = \mathbf{L} \circ \mathbf{C} \circ \mathbf{R}$ with $\mathbf{L}, \mathbf{C} \in \mathbb{R}_+^{50\times 20}$ and $\mathbf{R} \in \mathbb{R}_+^{500\times 20}$;
- Compare $\text{relerr} = \frac{\|\mathbf{A}_1 \circ \mathbf{A}_2 \circ \mathbf{A}_3 - \mathcal{M}\|_F}{\|\mathcal{M}\|_F}$ for APG-TC and $\text{relerr} = \frac{\|\mathcal{X} - \mathcal{M}\|_F}{\|\mathcal{M}\|_F}$ for FaLRTC and HaLRTC; running time is in second

	APG-TC (pros'd) $r = 20$		APG-TC (pros'd) $r = 25$		FaLRTC		HaLRTC	
SR	relerr	time	relerr	time	relerr	time	relerr	time
0.10	1.65e-4	2.25e1	3.87e-4	4.62e1	3.13e-1	1.40e2	3.56e-1	2.55e2
0.30	1.06e-4	1.38e1	1.69e-4	3.65e1	1.73e-2	1.53e2	1.42e-3	2.24e2
0.50	1.01e-4	1.33e1	1.14e-4	3.46e1	1.14e-2	1.07e2	1.95e-4	1.17e2

Observation: APG-TC (proposed) gives lower errors and runs faster.

Summary

- Multi-convex optimization has very interesting applications;
- A 3-scheme block-coordinate descent method is introduced;
  - The three schemes allow easy implementation and fast running time on many applications;
- Global convergence and rate are established; the assumptions are met by many applications;
- Applied BCD with prox-linear scheme to nonnegative matrix factorization, nonnegative tensor factorization, and completion;
  - Extrapolation significantly speeds up convergence;
  - BCD based on scheme 3 (or hybrid schemes 1 & 3) is much faster than the current state-of-the-art solvers and achieves lower objectives.