First Order Methods for Well Structured Optimization Problems

Marc Teboulle

School of Mathematical Sciences

Tel Aviv University

SIAM Meeting – July 2016 – Boston

Partners in the Adventures...

Amir Beck, Technion, Haifa

Jerôme Bolte, Toulouse University, Toulouse

Yoel Drori, Google's R&D Center, Tel Aviv

Ronny Luss, IBM Thomas J. Watson Research Center, Yorktown Hights

Shoham Sabach, Technion, Haifa

A Quote with Very Good News for Optimizers!

A Quote with Very Good News for Optimizers!

...Nothing at all takes place in the universe in which some rule of maximum or minimum does not appear...

Leonhard Euler

A Quote with Very Good News for Optimizers!

...Nothing at all takes place in the universe in which some rule of maximum or minimum does not appear...

Leonhard Euler

Optimization Everywhere...

- **From** laws of Nature, Biology, Physics, Chemistry... **To ...**
- Management Operations, Resource Allocation, Logistic...(started with LP)
- Finance, Economics, Human behavior...
- Engineering: Mechanical, Structural design, Chemical,...
- Machine Learning, Classification, Pattern Recognition, Data Networks/Mining...
- Signal Processing, Communication Systems, Imaging Science, Tomography...
- Modern Era: Facebook, Google....
- **...and in Mathematics itself.**

Goal

- **Deriving simple and efficient methods capable of solving very large scale problems**
- **Amenable to theoretical analysis: Convergence/Complexity**

Exploit problem structures and data information: Convex and Nonconvex Models

3 ELEMENTARY PRINCIPLES

• Approximation • Regularization • Decomposition

Simple Minimization Methods

Practical Side – Simplicity/Scalability

- Simple computational operations: additions - multiplications
- Explicit iterations.
- Avoid nested optimization schemes/control-correction of accumulated errors.
- Minimal storage of data

Theoretical Side – Convergence/Complexity Analysis

- Free from heuristic choices of extra parameters.
- Versatile mathematical analytic tools broadly applicable..and with no pains!
- Complexity: <u>nearly independent on dimension</u>.
- Performance: reasonable for medium accuracy.

Simple Minimization Methods

Practical Side – Simplicity/Scalability

- Simple computational operations: additions - multiplications
- Explicit iterations.
- Avoid nested optimization schemes/control-correction of accumulated errors.
- Minimal storage of data

Theoretical Side – Convergence/Complexity Analysis

- Free from heuristic choices of extra parameters.
- Versatile mathematical analytic tools broadly applicable..and with no pains!
- Complexity: nearly independent on dimension.
- Performance: reasonable for medium accuracy.

Natural Candidates: Schemes based on First Order Methods

First-Order Methods

First-Order methods are iterative algorithms that only exploit information on the objective function and its gradient (sub-gradient).

First-Order Methods

First-Order methods are iterative algorithms that only exploit information on the objective function and its gradient (sub-gradient).

- **A main drawback:** Can be very slow for producing high accuracy solutions....But **share many advantages**:

First-Order Methods

First-Order methods are iterative algorithms that only exploit information on the objective function and its gradient (sub-gradient).

- **A main drawback:** Can be very slow for producing high accuracy solutions....But **share many advantages**:
- Requires minimal data information
- Often lead to very simple and "cheap" iterative schemes
- Provable complexity/efficiency nearly independent of dimension
- Suitable for large-scale problems when high accuracy is not crucial. [In many large scale applications, the data is anyway corrupted or known only roughly.]

First Order-Based Algorithms

Widely used in applications....

- **Clustering Analysis:** *The k-means algorithm*
- **Neuro-computing:** *The backpropagation algorithm*
- **Statistical Estimation:** *The EM (Expectation-Maximization)* algorithm.
- **Machine Learning:** *SVM, Regularized regression, etc...*
- **Signal and Image Processing:** *Sparse Recovery, Denoising/Deblurring ...*
- **Matrix minimization Problems....and much more...**

Some Basic Optimization Models, First Order Algorithms and Rate of Convergence Results: A Short Tour

The World's Simplest Impossible Problem - Moler (1990)

The World's Simplest Impossible Problem - Moler (1990)

Problem: Given the average of two numbers is 3. What are the numbers?

The World's Simplest Impossible Problem - Moler (1990)

Problem: Given the average of two numbers is 3. What are the numbers?

- Typical answers: (2,4), (1,5), (-3,9)......These already ask for "structure":..least equal distance from average.. integer numbers..

The World's Simplest Impossible Problem - Moler (1990)

Problem: Given the average of two numbers is 3. What are the numbers?

- Typical answers: (2,4), (1,5), (-3,9)......These already ask for "structure":..least equal distance from average.. integer numbers..
- Why not (2.71828, 3.28172) !?....!...

The World's Simplest Impossible Problem - Moler (1990)

Problem: Given the average of two numbers is 3. What are the numbers?

- Typical answers: (2,4), (1,5), (-3,9)......These already ask for "structure":..least equal distance from average.. integer numbers..
- Why not (2.71828, 3.28172) !?....!...
- A nice one: (3,3) ....is with **"minimal norm" and its unique!**
- Simplest: (6,0) or (0,6)?...**A sparse one!** .... here lack of uniqueness!..

The World's Simplest Impossible Problem - Moler (1990)

Problem: Given the average of two numbers is 3. What are the numbers?

- Typical answers: (2,4), (1,5), (-3,9)......These already ask for "structure":..least equal distance from average.. integer numbers..
- Why not (2.71828, 3.28172) !?....!...
- A nice one: (3,3) ....is with **"minimal norm" and its unique!**
- Simplest: (6,0) or (0,6)?...**A sparse one!** .... here lack of uniqueness!..

This simple problem captures the essence of many Ill-posed/underdetermined problems in applications.

Additional requirements have to be specified to make it a reasonable mathematical/computational task, leading to interesting optimization models.

Linear Inverse Problems

Problem: Find $\mathbf{x} \in C \subset \mathbb{E}$ which "best" solves $\mathcal{A}(\mathbf{x}) \approx \mathbf{b}$, $\mathcal{A} : \mathbb{E} \to \mathbb{F}$,
where $\mathbf{b}$ (observable output), and $\mathcal{A}$ are known.

Linear Inverse Problems

Problem: Find $\mathbf{x} \in C \subset \mathbb{E}$ which "best" solves $\mathcal{A}(\mathbf{x}) \approx \mathbf{b}$, $\mathcal{A} : \mathbb{E} \to \mathbb{F}$,
where $\mathbf{b}$ (observable output), and $\mathcal{A}$ are known.

Approach via Optimization – Regularization Models

- $\rho(\mathbf{x})$ is a "regularizer" (one – or sum of functions, convex or nonconvex)
- $d(\mathbf{b}, \mathcal{A}(\mathbf{x}))$ some "proximity" measure from $\mathbf{b}$ to $\mathcal{A}(\mathbf{x})$

▷ $\min\{\rho(\mathbf{x}) : \mathcal{A}(\mathbf{x}) = \mathbf{b}, \ \mathbf{x} \in C\}$ or $\min\{\rho(\mathbf{x}) : d(\mathbf{b}, \mathcal{A}(\mathbf{x})) \leq \epsilon, \ \mathbf{x} \in C\}$

▷ $\min\{d(\mathbf{b}, \mathcal{A}(\mathbf{x})) : \rho(\mathbf{x}) \leq \delta, \mathbf{x} \in C\}$ or $\min\{d(\mathbf{b}, \mathcal{A}(\mathbf{x})) + \mu\rho(\mathbf{x}) : \mathbf{x} \in C\}, \mu > 0$

Linear Inverse Problems

Problem: Find $\mathbf{x} \in C \subset \mathbb{E}$ which "best" solves $\mathcal{A}(\mathbf{x}) \approx \mathbf{b}$, $\mathcal{A} : \mathbb{E} \to \mathbb{F}$,
where $\mathbf{b}$ (observable output), and $\mathcal{A}$ are known.

Approach via Optimization – Regularization Models

- $\rho(\mathbf{x})$ is a "regularizer" (one – or sum of functions, convex or nonconvex)
- $d(\mathbf{b}, \mathcal{A}(\mathbf{x}))$ some "proximity" measure from $\mathbf{b}$ to $\mathcal{A}(\mathbf{x})$

▷ $\min\{\rho(\mathbf{x}) : \ \mathcal{A}(\mathbf{x}) = \mathbf{b}, \ \mathbf{x} \in C\}$ or $\min\{\rho(\mathbf{x}) : \ d(\mathbf{b}, \mathcal{A}(\mathbf{x})) \leq \epsilon, \ \mathbf{x} \in C\}$

▷ $\min\{d(\mathbf{b}, \mathcal{A}(\mathbf{x})) : \rho(\mathbf{x}) \leq \delta, \mathbf{x} \in C\}$ or $\min\{d(\mathbf{b}, \mathcal{A}(\mathbf{x})) + \mu\rho(\mathbf{x}) : \mathbf{x} \in C\}, \mu > 0$

- Choices for $\rho(\cdot)$, $d(\cdot, \cdot)$ depends on the application at hand.
- **Nonsmooth and Nonconvex** regularizers ρ useful to describe desired features.

- Intensive research activities over the past 50 years.
- Today more with emerging new technologies and increase in computer power.

Example: Sparsity is a Common Desired Feature/Structure

Arises in Many Applications

- Sparse learning: feature selection, support vector machines, PCA,...
- Compressive sensing: recover a signal from few measurements ...
- Trust topology design: remove bars that are not needed...
- Image processing: denoising, deblurring,....and much more....

Example Let $d(\mathbf{b}, \mathcal{A}(\mathbf{x})) := \|\mathbf{b} - \mathcal{A}(\mathbf{x})\|^2, \quad \rho(\mathbf{x}) := \|\mathbf{x}\|_0$.

Find $\mathbf{x} \in \mathbb{R}^d$ which is sparsest or at least δ-sparse

$$\min\{\|\mathbf{x}\|_0 : \ \|\mathbf{b} - \mathcal{A}(\mathbf{x})\|^2 \leq \epsilon, \mathbf{x} \in \mathbb{R}^d\}; \quad \min\{\|\mathbf{b} - \mathcal{A}(\mathbf{x})\|^2 : \ \|\mathbf{x}\|_0 \leq \delta, \ \in \mathbb{R}^d\}$$

where $\|\mathbf{x}\|_0$ denotes the number of nonzero component of $\mathbf{x}$.

This can be **Hard** (despite the convex objective/constraint!).

Example: Sparsity is a Common Desired Feature/Structure

Arises in Many Applications

- Sparse learning: feature selection, support vector machines, PCA,...
- Compressive sensing: recover a signal from few measurements ...
- Trust topology design: remove bars that are not needed...
- Image processing: denoising, deblurring,....and much more....

Example Let $d(\mathbf{b}, \mathcal{A}(\mathbf{x})) := \|\mathbf{b} - \mathcal{A}(\mathbf{x})\|^2, \quad \rho(\mathbf{x}) := \|\mathbf{x}\|_0.$

Find $\mathbf{x} \in \mathbb{R}^d$ which is sparsest or at least δ-sparse

$$\min\{\|\mathbf{x}\|_0 : \ \|\mathbf{b} - \mathcal{A}(\mathbf{x})\|^2 \leq \epsilon, \mathbf{x} \in \mathbb{R}^d\}; \quad \min\{\|\mathbf{b} - \mathcal{A}(\mathbf{x})\|^2 : \ \|\mathbf{x}\|_0 \leq \delta, \ \in \mathbb{R}^d\}$$

where $\|\mathbf{x}\|_0$ denotes the number of nonzero component of $\mathbf{x}$.

This can be **Hard** (despite the convex objective/constraint!).

Approaches

- **Convex Relaxation/Approximation:** Replace $\|\mathbf{x}\|_0$ by a more tractable object. The l_1-norm $\|\mathbf{x}\|_1$ has been well known (since 70's) to promote sparsity. Nonconvex (concave) approximations are also relevant.
- **Tackle directly the nonconvex problem "as is"?**. More on this soon...

A Basic and Useful Model: Composite Minimization

$$\text{(M)} \quad \min\left\{F(\mathbf{x}) \equiv f(\mathbf{x}) + g(\mathbf{x}) : \mathbf{x} \in \mathbb{E}\right\}.$$

- $\mathbb{E}$ is a finite dimensional Euclidean space
- $f : \mathbb{E} \to \mathbb{R}$ is smooth: $C_L^{1,1}$ (L-Lipschitz continuous gradient)
- $g : \mathbb{E} \to (-\infty, \infty]$ is **nonsmooth extended valued** (allowing constraints)
- With a constraint set C, replace g by $g + \delta_C$, the indicator of C:

$$\delta_C(x) = \begin{cases} 0, & \text{if } x \in C, \\ +\infty, & \text{otherwise.} \end{cases}$$

A Basic and Useful Model: Composite Minimization

$$\text{(M)} \quad \min \left\{ F(\mathbf{x}) \equiv f(\mathbf{x}) + g(\mathbf{x}) : \mathbf{x} \in \mathbb{E} \right\}.$$

- $\mathbb{E}$ is a finite dimensional Euclidean space
- $f : \mathbb{E} \to \mathbb{R}$ is smooth: $C_L^{1,1}$ (L-Lipschitz continuous gradient)
- $g : \mathbb{E} \to (-\infty, \infty]$ is **nonsmooth extended valued** (allowing constraints)
- With a constraint set C, replace g by $g + \delta_C$, the indicator of C:

$$\delta_C(x) = \begin{cases} 0, & \text{if } x \in C, \\ +\infty, & \text{otherwise.} \end{cases}$$

This "simple" model (M) has **structural information**, and captures various classes of smooth/nonsmooth/convex/nonconvex minimization problems.

We are interested in solving (M) *approximately* to a given accuracy $\varepsilon > 0$:

$$F(\hat{\mathbf{x}}) - F(\mathbf{x}^*) \le \varepsilon.$$

Building First Order Based Schemes: Basic Old Idea

$\boxed{\textbf{Pick an adequate approximate model}}$

Building First Order Based Schemes: Basic Old Idea

Pick an adequate approximate model

1. **Linearize + regularize:** Given some $\mathbf{y}$, approximate $f(\mathbf{x}) + g(\mathbf{x})$ via:

$$q(\mathbf{x}, \mathbf{y}) = f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{1}{2t}\|\mathbf{x} - \mathbf{y}\|^2 + g(\mathbf{x}), \quad (t > 0)$$

That is, **leaving the nonsmooth part $g(\cdot)$ untouched**.

Building First Order Based Schemes: Basic Old Idea

Pick an adequate approximate model

1. **Linearize + regularize:** Given some $\mathbf{y}$, approximate $f(\mathbf{x}) + g(\mathbf{x})$ via:

$$q(\mathbf{x}, \mathbf{y}) = f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{1}{2t} \|\mathbf{x} - \mathbf{y}\|^2 + g(\mathbf{x}), \quad (t > 0)$$

   That is, **leaving the nonsmooth part $g(\cdot)$ untouched**.

2. **Linearize only + use info on C:** e.g., C compact, $g := \delta_C$

$$q(\mathbf{x}, \mathbf{y}) = f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle$$

Building First Order Based Schemes: Basic Old Idea

Pick an adequate approximate model

1. **Linearize + regularize:** Given some $\mathbf{y}$, approximate $f(\mathbf{x}) + g(\mathbf{x})$ via:

$$q(\mathbf{x}, \mathbf{y}) = f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{1}{2t} \|\mathbf{x} - \mathbf{y}\|^2 + g(\mathbf{x}), \quad (t > 0)$$

   That is, **leaving the nonsmooth part $g(\cdot)$ untouched**.

2. **Linearize only + use info on C:** e.g., C compact, $g := \delta_C$

$$q(\mathbf{x}, \mathbf{y}) = f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle$$

Solve "some how", the resulting approximate model:

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{x}} q(\mathbf{x}, \mathbf{x}^k), k = 0, \ldots$$

.

Examples $\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\text{argmin}}\, q(\mathbf{x}, \mathbf{x}^k)$

1. The Proximal-Gradient - [Passty'79, Lions-Mercier'79]

$$\mathbf{x}^{k+1} = \underset{\mathbf{x}\in\mathbb{E}}{\text{argmin}} \left\{ g(\mathbf{x}) + \frac{1}{2t_k}\|\mathbf{x} - (\mathbf{x}^k - t_k \nabla f(\mathbf{x}^k))\|^2 \right\} \equiv \text{prox}_{t_k g}(\mathbf{x}^k - t_k \nabla f(\mathbf{x}^k))$$

$$\text{prox}_{\mathbf{g}}(\mathbf{z}) := \underset{\mathbf{u}}{\text{argmin}} \left\{ \mathbf{g}(\mathbf{u}) + \frac{\mathbf{1}}{\mathbf{2}}\|\mathbf{u} - \mathbf{z}\|^{\mathbf{2}} \right\}$$ [Moreau 64]

The Prox-Grad scheme covers: gradient ($g \equiv 0$); projected gradient, ($g \equiv \delta_C$); proximal minimization ($f \equiv 0$).

Useful when projection/prox step easy to compute.

Examples $\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}}\, q(\mathbf{x}, \mathbf{x}^k)$

1. The Proximal-Gradient - [Passty'79, Lions-Mercier'79]

$$\mathbf{x}^{k+1} = \underset{\mathbf{x} \in \mathbb{E}}{\operatorname{argmin}} \left\{ g(\mathbf{x}) + \frac{1}{2t_k} \|\mathbf{x} - (\mathbf{x}^k - t_k \nabla f(\mathbf{x}^k))\|^2 \right\} \equiv \operatorname{prox}_{t_k g}(\mathbf{x}^k - t_k \nabla f(\mathbf{x}^k))$$

$$\operatorname{prox}_{\mathbf{g}}(\mathbf{z}) := \underset{\mathbf{u}}{\operatorname{argmin}} \left\{ \mathbf{g}(\mathbf{u}) + \frac{\mathbf{1}}{\mathbf{2}} \|\mathbf{u} - \mathbf{z}\|^2 \right\} \quad \text{[Moreau 64]}$$

The Prox-Grad scheme covers: gradient ($g \equiv 0$); projected gradient, ($g \equiv \delta_C$); proximal minimization ($f \equiv 0$).
Useful when projection/prox step easy to compute.

2. The Conditional-Gradient Method - $g := \delta_C$ the indicator of C, compact
[Frank-Wolfe'56, Polyak'63, Dunn'78]

$$\odot \quad \mathbf{p}^k = \operatorname{argmin}\{\langle \mathbf{x}, \nabla f(\mathbf{x}^k)\rangle : \ \mathbf{x} \in C\}, \ \mathbf{x}^{k+1} = (1 - t_k)\mathbf{x}^k + t_k \mathbf{p}^k, \ t_k \in (0, 1].$$

Useful when "linear oracles" $\odot$ can be efficiently solved.
Schemes widely used in the convex setting.
But also relevant in the **Nonconvex setting**. More on this soon!

Global Rate of Convergence/Complexity for Convex FOM

Global Rate (Nonasymptotic) of Convergence Results for $F(x^k) - F_*$

- For Prox-Grad and Gradient methods: $O(1/k)$
- For Subgradient Methods: $O(1/\sqrt{k})$.
- Can we find a faster method?

Global Rate of Convergence/Complexity for Convex FOM

Global Rate (Nonasymptotic) of Convergence Results for $F(x^k) - F_*$

- For Prox-Grad and Gradient methods: $O(1/k)$
- For Subgradient Methods: $O(1/\sqrt{k})$.
- Can we find a faster method? Yes we can..!

Global Rate of Convergence/Complexity for Convex FOM

Global Rate (Nonasymptotic) of Convergence Results for $F(x^k) - F_*$

- For Prox-Grad and Gradient methods: $O(1/k)$
- For Subgradient Methods: $O(1/\sqrt{k})$.
- Can we find a faster method? Yes we can..!

Idea: From an old algorithm of Nesterov (1983) designed for minimizing **a smooth** convex function, and proven to be an *"optimal"* first order method (Yudin-Nemirovsky (80)).

But, here our composite problem (M) is nonsmooth. Yet, we can derive a faster algorithm than Prox-Grad, and **equally simple**.

A Fast Prox-Grad Algorithm (FISTA)- [Beck-Teboulle (2009)]

Algorithm as simple as "prox-grad", but **with the rate $O(1/k^2)$**.

Fast Prox-Grad Algorithm (FISTA)
For $k \geq 1$, compute a prox at auxiliary $\mathbf{y}^k$:

$$
\begin{aligned}
\mathbf{x}_k &= \operatorname{prox}_{\frac{g}{L}}\left(\mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k)\right), \hookleftarrow \text{ main computation as Prox-Grad} \\
\bullet\ t_{k+1} &= 2^{-1}\left(1 + \sqrt{1 + 4t_k^2}\right); \quad s_k = t_{k+1}^{-1}(t_k - 1) \\
\bullet\bullet\ \mathbf{y}_{k+1} &= \mathbf{x}_k + s_k(\mathbf{x}_k - \mathbf{x}_{k-1}).
\end{aligned}
$$

1. Additional computation in ($\bullet$) and ($\bullet\bullet$) is marginal.
2. Knowledge of L is not necessary. (Use a backtracking procedure).
3. Extensive testing in the literature confirms the efficiency of FISTA in many applications e.g.,:
   image denoising/deblurring, nuclear matrix norm regularization, matrix completion problems, multi-task learning, matrix classification, etc..

An Example: l_1-Image Deblurring

$$\min_{\mathbf{x}}\{\|\mathbf{Ax}-\mathbf{b}\|^2+\|\mathbf{x}\|_1\}$$

Comparing ISTA versus FISTA on Problems

- dimension d like $d = 256 \times 256 = 65,536,$ or/and $512 \times 512 = 262,144.$
- The $d \times d$ matrix $\mathbf{A}$ is **dense**

(Gaussian blurring times inverse of two-stage Haar wavelet transform).

- All problems with Gaussian noise.

Example l_1 Image Deblurring

original

blurred and noisy

1000 Iterations of ISTA versus 200 of FISTA

ProxGrad=ISTA: **1000 Iterations**

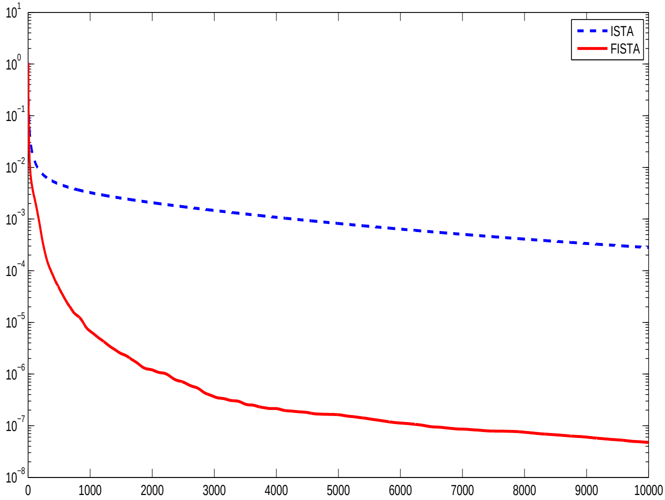
FastPG=FISTA: **200 Iterations**

Original Versus Deblurring via FISTA

Original

FISTA:1000 Iterations

Function Values errors $F(\mathbf{x}_k) - F(\mathbf{x}^*)$



Extension: FOM with Non-Euclidean Distances

- All previous schemes were based on using the squared Euclidean distance
- It is useful to exploit the *geometry of the constraints set X*
- This is done by selecting a "distance-like" function

Typical example: Bregman type distances - based on kernel ψ:

$$D_\psi(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}) - \psi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla\psi(\mathbf{y})\rangle, \ \psi \text{ strongly convex}$$

Extension: FOM with Non-Euclidean Distances

- All previous schemes were based on using the squared Euclidean distance
- It is useful to exploit the *geometry of the constraints set X*
- This is done by selecting a "distance-like" function

Typical example: Bregman type distances - based on kernel ψ:

$$D_\psi(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}) - \psi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla\psi(\mathbf{y})\rangle, \ \psi \text{ strongly convex}$$

Advantages: can exploit geometry of the constraints and allows to:

1. Simplify the prox computation for the given constraint, with adequate D_ψ
2. Preserve Complexity rate $O(1/k^2)$
3. Often improve the **constant** in the complexity bound.

Studied in various frameworks: *Mirror descent algorithms, extragradient-like, Lagrangians, smoothing, dual fast-prox-grad...*

[Nemirovsky-Yudin (80), Teboulle (92), Beck-Teboulle (03), Nemirovsky (04), Nesterov (05), Auslender-Teboulle (05), Beck-Teboulle.(12,14)...]

More General Convex Nonsmooth Composite: Saddle Point Based Methods

A Class of Structured Convex-Concave Saddle-Point Model

Extends the previous model, and allows for handling more general problems

A Class of Structured Convex-Concave Saddle-Point Model

Extends the previous model, and allows for handling more general problems

$$\text{(SP)} \qquad \min_{u \in \mathbb{R}^n} \max_{v \in \mathbb{R}^d} \left\{ K(u, v) := f(u) + \langle u, \mathcal{A}v \rangle - g(v) \right\},$$

Data Information

(i) $f : \mathbb{R}^n \to \mathbb{R}$ is convex, smooth $: C^{1,1}_{L_f}$

(ii) $g : \mathbb{R}^d \to (-\infty, +\infty]$, is convex nonsmooth

(iii) $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^n$ is a linear map.

The model handles general scenarios with:

$$g(v_1, \ldots, v_m) := \sum_{i=1}^{m} g_i(v_i); \; \mathcal{A}v = \sum_{i=1}^{m} A_i v_i, \; v_i \in \mathbb{R}^{d_i}, d = \sum_{i=1}^{m} d_i$$

A Simple Algorithm for the Convex-Concave SP Drori -Sabach -T. (2015)

Relies on fundamental ideas: **it combines duality, predictor-corrector steps, and proximal operation within very simple iterations.**

A Simple Algorithm for the Convex-Concave SP
Drori -Sabach -T. (2015)

Relies on fundamental ideas: **it combines duality, predictor-corrector steps, and proximal operation within very simple iterations.**

PAPC – Proximal Alternating Predictor Corrector

For $k \geq 1$ compute:

$$p^k = u^{k-1} - \tau\left(\mathcal{A}v^{k-1} + \nabla f\left(u^{k-1}\right)\right)$$

$$v_i^k = \mathrm{prox}_{\sigma_i}^{g_i}\left(v_i^{k-1} + \sigma_i A_i^T p^k\right), \quad i = 1, 2, \ldots, m,$$

$$u^k = u^{k-1} - \tau\left(\mathcal{A}v^k + \nabla f\left(u^{k-1}\right)\right).$$

- $\oplus$ The - v step **"decomposes" according to structure**
- $\oplus$ **Only** prox for each $g_i(\cdot)$, **and not for the difficult composite** $g_i \circ A_i$.
- $\oplus$ **The parameters** (τ, σ_i) **are defined in terms of problem's data** L_f, A_i.

PAPC – Convergence Results and Features

1. **Global Rate of Convergence** Shares the best known estimate $O(1/\varepsilon)$ for primal-dual gap. **Complexity bound constant in terms of data** (L_f, A_i)
2. **Convergence:** $\{(u^k, v^k)\}_{k\in\mathbb{N}}$ converges to a saddle-point (u^*, v^*) of K.

PAPC – Convergence Results and Features

1. **Global Rate of Convergence** Shares the best known estimate $O(1/\varepsilon)$ for primal-dual gap. **Complexity bound constant in terms of data** (L_f, A_i)
2. **Convergence:** $\{(u^k, v^k)\}_{k\in\mathbb{N}}$ converges to a saddle-point (u^*, v^*) of K.

Features of PAPC - Fully exploits given structures of a problem

- Free of heuristic/extra parameters: No tuning necessary, etc...
- Constraints on the variable u and **presence of nonsmooth f can be easily handled** via **The Dual Transportation Trick**. (Details in Paper).
- Performs well in applications: Image processing, Machine Learning ... and can be applied to many important optimization models which cannot be tackled by other current methods with same rate:

- $\min_u \left\{F(u) + \sum_{i=1}^m H_i(B_i u)\right\}$
- $\min_{x_i} \left\{\sum_{i=1}^m \psi(x_i) : \sum_{i=1}^m M_i x_i = b\right\}$
- $\min_{u\in\mathbb{R}^p} \left\{F(u) : \sum_{i=1}^m H_i(B_i u) \le \alpha\right\}$.

Nonconvex Smooth Models

Principal Component Analysis (PCA) – Pearson(1901)

- PCA is a tool for analyzing data. The way it works: project high dimensional data to a lower dimension in such a way that the amount of variance captured by the low dimensional data is maximized.
- PCA can be done by eigenvalue decomposition of a data covariance matrix:

$$\max\{\mathbf{x}^T A\mathbf{x} : \|\mathbf{x}\|_2 = 1,\ \mathbf{x} \in \mathbf{R}^n\},\ (A \succeq 0).$$

- **Problem with PCA:** Each data point is taken as a linear combination of all original features. Allows for nicely separating data but **we don't have an interpretation as to what separates the data?**
- **This is where sparsity helps:** Sparse PCA solves a similar problem to PCA but forces the factors to be a linear combinations of a limited number of the original features.

Sparse PCA

Principal Component Analysis solves

$$(PCA) \quad \max\{\mathbf{x}^T A\mathbf{x} : \|\mathbf{x}\|_2 = 1,\ \mathbf{x} \in \mathbb{R}^n\},\ (A \succeq 0)$$

while Sparse Principal Component Analysis solves

$$(SPCA) \quad \max\{\mathbf{x}^T A\mathbf{x} : \|\mathbf{x}\|_2 = 1,\ \|\mathbf{x}\|_\mathbf{0} \leq \mathbf{k},\ \mathbf{x} \in \mathbb{R}^n\},\ k \in (1, n] \text{ sparsity}$$

$\|\mathbf{x}\|_0$ counts the number of nonzero entries of x

Issues in SPCA:

1. Maximizing a convex objective.
2. Hard nonconvex constraint $\|\mathbf{x}\|_0 \leq k$.

Current Approaches:

1. **SDP Convex Relaxations** – too expensive for large problems.
2. **Solve modification/approximations** of SPCA.

Sparse PCA via Penalization/Relaxation/Approx.

♠ The problem of interest is the difficult sparse PCA problem **as is**

$$\max\{\mathbf{x}^T A\mathbf{x} : \|\mathbf{x}\|_2 = 1,\ \|\mathbf{x}\|_0 \leq k,\ \mathbf{x} \in \mathbf{R}^n\}$$

♠ Literature has focused on solving various relaxation/Approximations:

- **l_0-penalized PCA**

$$\max\{\mathbf{x}^T A\mathbf{x} - s\|\mathbf{x}\|_0 : \|x\|_2 = 1\},\ s > 0$$

- **Relaxed l_1-constrained PCA**

$$\max\{\mathbf{x}^T A\mathbf{x} : \|\mathbf{x}\|_2 = 1,\ \|x\|_1 \leq \sqrt{k}\}$$

- **Relaxed l_1-penalized PCA**

$$\max\{\mathbf{x}^T A\mathbf{x} - s\|\mathbf{x}\|_1 : \|\mathbf{x}\|_2 = 1\}$$

- **Approximated-Penalized**

$$\max\{\mathbf{x}^T A\mathbf{x} - sg_p(\mathbf{x}) : \|\mathbf{x}\|_2 = 1\} \text{ where } g_p(\mathbf{x}) \simeq \|\mathbf{x}\|_0\}$$

Many algorithms from various disparate approaches/motivations to solve **modifications/appproximations** of SPCA: Expectation Maximization; Majorization-Mininimization techniques; DC programming.. etc..

A Plethora of Algorithms for Modified/Approximate SPCA

A Plethora of Algorithms for Modified/Approximate SPCA

1. **Are all current algorithms for modified SPCA different?**
2. **Can we tackle directly the sparse PCA problem "as is"?**

Sparse PCA Revisited - [Luss and T. (2013)]

- Current algorithms for **modified SPCA** are just a particular realization of **the well-known Conditional Gradient Algorithm!** with unit step size.

Sparse PCA Revisited - [Luss and T. (2013)]

- Current algorithms for **modified SPCA** are just a particular realization of **the well-known Conditional Gradient Algorithm!** with unit step size.
- **ConGradU CAN be applied directly to the original problem!**

Sparse PCA Revisited - [Luss and T. (2013)]

- Current algorithms for **modified SPCA** are just a particular realization of **the well-known Conditional Gradient Algorithm!** with unit step size.
- **ConGradU CAN be applied directly to the original problem!**

Solving Original Sparse PCA: $\max\{\mathbf{x}^T A\mathbf{x} : \|\mathbf{x}\|_2 = 1,\ \|\mathbf{x}\|_0 \le k,\ \mathbf{x} \in \mathbb{R}^n\}$

ConGradU generates the sequence $\{x^j\}$ via

$$
\begin{aligned}
x^{j+1} &= \frac{T_k(Ax^j)}{\|T_k(Ax^j)\|_2},\ j = 0, \ldots \\
T_k(a) &:= \operatorname*{argmin}_u \{\|u - a\|_2^2 : \|x\|_0 \le k\}
\end{aligned}
$$

Despite the hard constraint, easy to compute: $(T_k(a))_i = a_i$ for the k largest entries (in absolute value) of a and $(T_k(x))_i = 0$ otherwise.

Sparse PCA Revisited - [Luss and T. (2013)]

- Current algorithms for **modified SPCA** are just a particular realization of **the well-known Conditional Gradient Algorithm!** with unit step size.
- **ConGradU CAN be applied directly to the original problem!**

Solving Original Sparse PCA: $\max\{\mathbf{x}^T A\mathbf{x} : \|\mathbf{x}\|_2 = 1,\ \|\mathbf{x}\|_0 \le k,\ \mathbf{x} \in \mathbb{R}^n\}$

ConGradU generates the sequence $\{x^j\}$ via

$$x^{j+1} = \frac{T_k(Ax^j)}{\|T_k(Ax^j)\|_2},\ j = 0, \ldots$$

$$T_k(a) := \underset{u}{\operatorname{argmin}}\{\|u - a\|_2^2 : \|x\|_0 \le k\}$$

Despite the hard constraint, easy to compute: $(T_k(a))_i = a_i$ for the k largest entries (in absolute value) of a and $(T_k(x))_i = 0$ otherwise.

- **Convergence:** Every limit point of $\{x^j\}$ converges to a critical point.
- **Computationally Cheap:** Handles very large-scale SPCA problems (limited only by storage of data matrix.)

Nonconvex and NonSmooth Models

A Broad Class of Nonsmooth Nonconvex Problems

A Useful Block Optimization Model

$$(B) \qquad \text{minimize}_{x,y} \Psi(x,y) := f(x) + g(y) + H(x,y)$$

- $f : \mathbb{R}^n \to (-\infty, +\infty]$ and $g : \mathbb{R}^m \to (-\infty, +\infty]$ proper and lsc.
- $H : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a C^1 function.
- Partial gradients of H are smooth $C^{1,1}$

♠ **NO convexity** assumed in the objective and the constraints (built-in through f and g extended valued).

Two blocks is only for the sake of simplicity. Same for the p-blocks case:

$$\text{minimize}_{x_1,\ldots,x_p} H(x_1, x_2, \ldots, x_p) + \sum_{i=1}^{p} f_i(x_i), \; x_i \in \mathbb{R}^{n_i}, n = \sum_{i=1}^{p} n_i$$

PALM: Proximal Alternating Linearized Minimization

PALM "blends" old spices:

- $\oplus$ **Space decomposition [á la Gauss-Seidel]**
- $\oplus$ **Composite decomposition [á la Prox-Gradient].**

PALM: Proximal Alternating Linearized Minimization

PALM "blends" old spices:

$\oplus$ **Space decomposition [á la Gauss-Seidel]**

$\oplus$ **Composite decomposition [á la Prox-Gradient].**

PALM Algorithm

1. Take $\gamma_1 > 1$, set $c_k = \gamma_1 L_1\left(y^k\right)$ and compute

$$x^{k+1} \in \operatorname{prox}_{c_k}^{f}\left(x^k - \frac{1}{c_k}\nabla_x H\left(x^k, y^k\right)\right).$$

2. Take $\gamma_2 > 1$, set $d_k = \gamma_2 L_2\left(x^{k+1}\right)$ and compute

$$y^{k+1} \in \operatorname{prox}_{d_k}^{g}\left(y^k - \frac{1}{d_k}\nabla_y H\left(x^{k+1}, y^k\right)\right).$$

Stepsizes c_k^{-1}, d_k^{-1} are in $\left]0, 1/L_2(y^k)\right[\quad \& \quad \left]0, 1/L_1(x^{k+1})\right[$.

Main computational step: Computing the prox of a nonconvex function.

Convergence of PALM and More...

Theorem (Bolte–Sabach–T. 2014)

Assume f, g, H real semi-algebraic. Any bounded PALM sequence $\left\{z^k\right\}_{k\in\mathbb{N}}$ converges to a critical point $z^* = \left(x^*, y^*\right)$ of Ψ.

Moreover there exists $\gamma > 0, C > 0$ such that

$$\|z^k - z^*\| \leq C\; k^{-\gamma}$$

Convergence of PALM and More...

Theorem (Bolte–Sabach–T. 2014)

Assume f, g, H real semi-algebraic. Any bounded PALM sequence $\{z^k\}_{k\in\mathbb{N}}$ converges to a critical point $z^* = (x^*, y^*)$ of Ψ.

Moreover there exists $\gamma > 0, C > 0$ such that

$$\|z^k - z^*\| \leq C\, k^{-\gamma}$$

1. Are there many semi-algebraic functions?
2. What is behind these results ?

Answer to 2 $\Longrightarrow$

A general convergence framework for any descent algorithm.

A General Recipe in 3 Main Steps for Descent Methods

A sequence z^k is called *a descent sequence* for $F : \mathbb{R}^n \to (-\infty, +\infty]$ if

C1. Sufficient decrease property

$$\exists \rho_1 > 0 \quad \text{with} \quad \rho_1 \|z^{k+1} - z^k\|^2 \leq F(z^k) - F(z^{k+1}), \quad \forall k \geq 0$$

C2. Iterates gap For each k there exists $w^k \in \partial F(z^k)$ such that:

$$\exists \rho_2 > 0 \quad \text{with} \quad \left\| w^{k+1} \right\| \leq \rho_2 \|z^{k+1} - z^k\|, \forall k \geq 0.$$

- These two steps are typical for **any descent** type algorithms but lead **only to subsequential convergence** [Ostrowski 1966].

A General Recipe in 3 Main Steps for Descent Methods

A sequence z^k is called *a descent sequence* for $F : \mathbb{R}^n \to (-\infty, +\infty]$ if

C1. Sufficient decrease property

$$\exists \rho_1 > 0 \quad \text{with} \quad \rho_1 \|z^{k+1} - z^k\|^2 \le F(z^k) - F(z^{k+1}), \quad \forall k \ge 0$$

C2. Iterates gap For each k there exists $w^k \in \partial F(z^k)$ such that:

$$\exists \rho_2 > 0 \quad \text{with} \quad \left\|w^{k+1}\right\| \le \rho_2 \|z^{k+1} - z^k\|, \forall k \ge 0.$$

- These two steps are typical for **any descent** type algorithms but lead **only to subsequential convergence** [Ostrowski 1966].
- To get **global convergence** to a critical point, we need a deep mathematical tool.[Łojasiewicz (68), Kurdyka (98)]

C3. The Kurdyka-Łojasiewicz property: Assume that F satisfies the KL property. Use this to prove that the generated sequence $\left\{z^k\right\}_{k \in \mathbb{N}}$ is a *Cauchy sequence*, and thus converges!

Impact of KL in optimization:

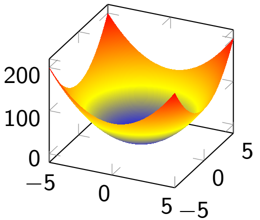
[Bolte et al. (06,07,10), Attouch-Bolte et al. (09,10,12)]

The KL Property Informal: A Geometric Snapshot

Let $\bar{\mathbf{z}}$ be critical, with $F(\bar{\mathbf{z}}) = 0$ (true up to translation); $\mathcal{L}_\eta := \{z \in \mathbb{R}^d : 0 < F(z) < \eta\}$

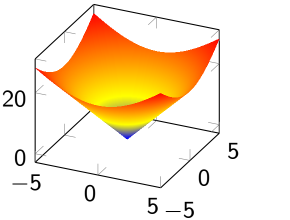
Definition [Sharpness] A function $F : \mathbb{R}^d \to (-\infty, +\infty]$ is called sharp on $\mathcal{L}_\eta$ if there exists $c > 0$ such that $\min\{\|\xi\| : \ \xi \in \partial F(z)\} \geq c > 0 \quad \forall z \in \mathcal{L}_\eta$.

KL warrants F amenable to sharpness



$\longrightarrow$

Sharp reparameterization $\varphi \circ F$



- Sharpness implies excellent convergence properties.

Theorem [Bolte-Daniilidis-Lewis (2006)]
KL property holds for all semi-algebraic functions.

The KL Property: (Łojasiewicz (68), Kurdyka (98))

- $\varphi : \mathbb{R} \to \mathbb{R}_+$ **a desingularizing function on** $(0, \eta)$**:**

$$\varphi \in C[0,\eta),\ \textbf{concave},\ \varphi \in C^1(0,\eta), \varphi' > 0, \varphi(0) = 0.$$

- $\mathcal{L}_\eta := \{z \in \mathbb{R}^d : 0 < F(z) < \eta\}$

The KL Property F has the KL property on $\mathcal{L}_\eta$ if there exists a desingularizing function φ such that

$$\operatorname{dist}(0, \partial(\varphi \circ F)(x)) \geq 1. \quad \forall x \in \mathcal{L}_\eta.$$

Meaning: Subgradients of $\varphi \circ F$ have a norm bounded away from zero, no matter how close is z to the critical point $\bar{\mathbf{z}}$ – **This is sharpness.**

Answer to 1 - There is a Wealth of Semi-Algebraic Functions!

Semi-algebraic Sets/Functions

- Semi-algebraic objects: defined by finitely polynomials.
- Semi-algebraic property is very stable and preserved under many operations : Finite sums and product, composition, ...

Some Examples - "Starring" in Optimization/Applications

- Real polynomial functions.
- Standard Cones: $\mathbb{R}^d_+$, SDP, Lorentz..
- Rank, $\|\cdot\|_0$ and l_p-norms (p rational or $p = \infty$)
- Indicator functions of semi-algebraic sets...

Application: Nonnegative Matrix Factorization Problems

The NMF Problem: Given $A \in \mathbb{R}^{m \times n}$ and $r \ll \min\{m, n\}$.
Find $X \in \mathbb{R}^{m \times r}$ and $Y \in \mathbb{R}^{r \times n}$ such that

$$A \approx XY, \;\; X \in \mathcal{K}_{m,r} \cap \mathcal{F}, \;\; Y \in \mathcal{K}_{r,n} \cap \mathcal{G},$$

$$\begin{aligned} \mathcal{K}_{p,q} &= \left\{ M \in \mathbb{R}^{p \times q} : \; M \geq 0 \right\} \\ \mathcal{F} &= \left\{ X \in \mathbb{R}^{m \times r} : \; R_1(X) \leq \alpha \right\} \\ \mathcal{G} &= \left\{ Y \in \mathbb{R}^{r \times n} : \; R_2(Y) \leq \beta \right\}. \end{aligned}$$

$R_1(\cdot)$ and $R_2(\cdot)$ are functions used to describe some additional/required features of X, Y.

(NMF) covers a very large number of problems in applications: Text Mining (data clusters in documents); Audio-Denoising (speech dictionnary); Bio-informatics (clustering gene expression); Medical Imaging,...Vast Literature.

Example: Applying PALM on NMF Problems

I. Nonnegative Matrix Factorization (NMF): $\mathcal{F} \equiv \mathbb{R}^{m \times r}$; $\mathcal{G} \equiv \mathbb{R}^{r \times n}$.

$$\min \left\{ \frac{1}{2} \|A - XY\|_F^2 : X \geq 0, Y \geq 0 \right\}.$$

II. Sparsity Constrained NMF: Useful in many applications

$$\min \left\{ \frac{1}{2} \|A - XY\|_F^2 : \|X\|_0 \leq \alpha, \|Y\|_0 \leq \beta,\ X \geq 0, Y \geq 0 \right\}.$$

Sparsity measure of matrix: $\|X\|_0 := \sum_i \|x_i\|_0$, ($x_i$ column vector of X).

Example: Applying PALM on NMF Problems

I. Nonnegative Matrix Factorization (NMF): $\mathcal{F} \equiv \mathbb{R}^{m \times r}$; $\mathcal{G} \equiv \mathbb{R}^{r \times n}$.

$$\min \left\{ \frac{1}{2} \|A - XY\|_F^2 : X \geq 0, Y \geq 0 \right\}.$$

II. Sparsity Constrained NMF: Useful in many applications

$$\min \left\{ \frac{1}{2} \|A - XY\|_F^2 : \|X\|_0 \leq \alpha, \|Y\|_0 \leq \beta, \; X \geq 0, Y \geq 0 \right\}.$$

Sparsity measure of matrix: $\|X\|_0 := \sum_i \|x_i\|_0$, ($x_i$ column vector of X).

For both models:

- **The data is semi-algebraic**, and fit our block model (B):

$$H(X,Y) \equiv 2^{-1} \|A - XY\|_F^2 ; \; f \text{ and } g \equiv \delta_{U \geq 0} + \delta_{\|U\|_0 \leq s}$$

- **PALM** produces very simple practical schemes, proven to globally converge. [Bolte-Sabach-T. (2014)].

For More Details and Results....

http://www.math.tau.ac.il/~teboulle

THANK YOU FOR YOUR ATTENTION!