Simple Algorithms for Difficult Optimization Problems Illustrated

Marc Teboulle

School of Mathematical Sciences
Tel Aviv University

Based on joint works with

Jérôme Bolte, (TSE, Toulouse I), Yoel Drori (Google Research, Tel Aviv)
Ronny Luss (IBM-New York), Shoham Sabach, (Technion), Ron Shefi (Tel Aviv)

Optimization in machine learning, vision and image processing
Université Paul Sabatier, Toulouse. October 6 - 7, 2015

Outline

- Simple algorithms exploiting structures and data information
- Nonsmooth Convex – Nonconvex Smooth – Nonsmooth Nonconvex

3 ELEMENTARY PRINCIPLES

- **Approximation**
- **Regularization**
- **Decomposition**

Opening Remark and Credit

About more than 386 years ago.....In 1629, Fermat suggested the following:

Opening Remark and Credit

About more than 386 years ago.....In 1629, Fermat suggested the following:

- Given f, solve for x:
- $\left[\dfrac{f(x+d)-f(x)}{d}\right]_{d=0} = 0$

...We can hardly expect to find a more general method to get the maximum or minimum points on a curve.....

Pierre de Fermat

Simple Minimization Methods

Practical Side

- Simple computational operations: inner products; No matrix inversion.
- Minimal storage of data; exploit smartly stored data.
- Easy access to function values, gradient/subgradients.
- Explicit iterative formula involving simple operations.

Theoretical Side

- Free of unknown/heuristic choices of parameters.
- Avoid nested optimization schemes/control-correction of accumulated errors.
- Versatile mathematical analytic tools broadly applicable..and with no pains!
- Complexity/Performance: mildly dependent on dimension/reasonable medium accuracy.

Natural Candidates: Schemes based on First Order Methods

Anecdote: The World's Simplest Impossible problem

[From C. Moler (1990)]

Anecdote: The World's Simplest Impossible problem

[From C. Moler (1990)]

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

Anecdote: The World's Simplest Impossible problem

[From C. 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..

Anecdote: The World's Simplest Impossible problem

[From C. 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) !?....!...

Anecdote: The World's Simplest Impossible problem

[From C. 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!..

Anecdote: The World's Simplest Impossible problem

[From C. 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/constraints have to be specified to make it a reasonable mathematical/computational task and often lead 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 **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 Regularization Models

- $g(\mathbf{x})$ 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 $\mathcal{A}(\mathbf{x})$

$$\begin{aligned}
&\min \quad \{g(\mathbf{x}) : \ \mathcal{A}(\mathbf{x}) = \mathbf{b}, \ \mathbf{x} \in C\} \\
&\min \quad \{g(\mathbf{x}) : \ d(\mathbf{b}, \mathcal{A}(\mathbf{x})) \le \epsilon, \ \mathbf{x} \in C\} \\
&\min \quad \{d(\mathbf{b}, \mathcal{A}(\mathbf{x})) : \ g(\mathbf{x}) \le \delta, \ \mathbf{x} \in C\} \\
&\min \quad \{d(\mathbf{b}, \mathcal{A}(\mathbf{x})) + \lambda g(\mathbf{x}) : \ \mathbf{x} \in C\} \ (\lambda > 0)
\end{aligned}$$

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 **b** (observable output), and $\mathcal{A}$ are known.

Approach: via Regularization Models

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

$$\begin{aligned}
&\min \quad \{g(\mathbf{x}) : \ \mathcal{A}(\mathbf{x}) = \mathbf{b}, \ \mathbf{x} \in C\} \\
&\min \quad \{g(\mathbf{x}) : \ d(\mathbf{b}, \mathcal{A}(\mathbf{x})) \leq \epsilon, \ \mathbf{x} \in C\} \\
&\min \quad \{d(\mathbf{b}, \mathcal{A}(\mathbf{x})) : \ g(\mathbf{x}) \leq \delta, \ \mathbf{x} \in C\} \\
&\min \quad \{d(\mathbf{b}, \mathcal{A}(\mathbf{x})) + \lambda g(\mathbf{x}) : \ \mathbf{x} \in C\} \ (\lambda > 0)
\end{aligned}$$

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

• Intensive research activities over the past 50 years...Now, much more...with emerging new applications and advances 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
- Image processing: denoising, deblurring,....and much more....

Find the sparsest $\mathbf{x} \in \mathbb{R}^d$ subject to specific requirements S:

$$\min\{\|\mathbf{x}\|_0 : \quad \mathbf{x} \in S\}$$

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

Simplify design by zeroing values that are not needed: Trust topology design - bars that are not needed; Antenna Array beamforming - eliminate un-needed antenna ....etc..

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
- Image processing: denoising, deblurring,....and much more....

Find the sparsest $\mathbf{x} \in \mathbb{R}^d$ subject to specific requirements S:

$$\min\{\|\mathbf{x}\|_0 : \quad \mathbf{x} \in S\}$$

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

Simplify design by zeroing values that are not needed: Trust topology design - bars that are not needed; Antenna Array beamforming - eliminate un-needed antenna ....etc..

This is **Hard!**, (even is S is convex !).

Approaches

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

Convex Nonsmooth Composite: Lagrangians Based Methods

Nonsmooth Convex with Separable Objective

$$(P) \qquad p_* = \inf\{\varphi(x) \equiv f(x) + g(Ax) : x \in \mathbb{R}^n\},$$

Here f, g are **both nonsmooth**, $A : \mathbb{R}^n \to \mathbb{R}^m$ a given linear map.

Nonsmooth Convex with Separable Objective

$$(P) \qquad p_* = \inf\{\varphi(x) \equiv f(x) + g(Ax) : x \in \mathbb{R}^n\},$$

Here f, g are **both nonsmooth**, $A : \mathbb{R}^n \to \mathbb{R}^m$ a given linear map.

Problem (P) is equivalent to (via the standard splitting variables trick):

$$(P) \qquad p_* = \inf\{f(x) + g(z) : Ax = z, \quad x \in \mathbb{R}^n, z \in \mathbb{R}^m\}$$

Nonsmooth Convex with Separable Objective

$$(P) \qquad p_* = \inf\{\varphi(x) \equiv f(x) + g(Ax) : x \in \mathbb{R}^n\},$$

Here f, g are **both nonsmooth**, $A : \mathbb{R}^n \to \mathbb{R}^m$ a given linear map.

Problem (P) is equivalent to (via the standard splitting variables trick):

$$(P) \qquad p_* = \inf\{f(x) + g(z) : Ax = z, \quad x \in \mathbb{R}^n, z \in \mathbb{R}^m\}$$

Rockafellar ('76) has shown that the *Proximal Point Algorithm* can be applied to the dual and primal-dual formulation of (P) to produce:

- The Multipliers Method (augmented Lagrangian Method).
- **The Proximal Method of Multipliers (PMM).**
- Largely ignored over last 20 years.....Recent very strong revival in, image science, machine learning etc... within many algorithms **all being rooted in - and variants of – the PMM**.

The PMM–Proximal Method of Multipliers – Rockafellar (76)

PMM Generate (x^k, z^k) and dual multiplier y^k via

$$(x^{k+1}, z^{k+1}) \in \operatorname*{argmin}_{x,z}\{f(x) + g(z) + \langle y^k, Ax - z\rangle + \frac{c}{2}\|Ax - z\|^2 + q_k(x,z)\}$$

$$y^{k+1} = y^k + c(Ax^{k+1} - z^{k+1}), \quad (c > 0).$$

- The Augmented Lagrangian $=$ Penalized Lagrangian

$$L_c(x,z,y) := \overbrace{f(x) + g(z) + \langle y, Ax - z\rangle}^{\text{Lagrangian}} + \frac{c}{2}\|Ax - z\|^2, \ (c > 0).$$

- $q_k(x,z) := \frac{1}{2}\left(\|x - x^k\|^2_{M_1} + \|z - z^k\|^2_{M_2}\right)$ is the additional *primal proximal* term.
- The choice of $M_1 \in \mathbb{S}^n_+, M_2 \in \mathbb{S}^m_+$ is used to conveniently describe/analyze several variants of the PMM.
- $M_1 = M_2 \equiv 0$, recovers the Multiplier Methods (PPA on the dual).

Proximal Method of Multipliers–Key Difficulty

- Main computational step in PMM: to minimize w.r.t (x, z) the proximal Augmented Lagrangian:

$$f(x) + g(z) + \langle y^k, Ax - z\rangle + \frac{c}{2}\|Ax - z\|^2 + q_k(x, z).$$

- **The quadratic coupling term $\|Ax - z\|^2$, destroys the separability between x and z, preventing separate minimization in (x, z).**
- In many applications, separate minimization is often much easier.....

Proximal Method of Multipliers–Key Difficulty

- Main computational step in PMM: to minimize w.r.t (x, z) the proximal Augmented Lagrangian:

$$f(x) + g(z) + \langle y^k, Ax - z\rangle + \frac{c}{2}\|Ax - z\|^2 + q_k(x, z).$$

- **The quadratic coupling term $\|Ax - z\|^2$, destroys the separability between x and z, preventing separate minimization in (x, z).**
- In many applications, separate minimization is often much easier.....

Many Strategies available to overcome this difficulty:

- Approximate Minimization – linearized the quad term $\|Ax - z\|^2$ wrt (x, z).
- Alternating Minimization – à la "Gauss-Seidel" in (x, z).
- Mixture of the above – *Partial Linearization* with respect to one variable, combined with *Alternating Minimization* of the other variable.
- Result in various useful variants of the PMM.

Main Tool for Analysis: - Via a Unified PMM Scheme

Unified Scheme U

Start with (x^0, z^0, y^0) and for all $k \geq 0$, and generate the sequence $\{x^k, z^k, y^k\}$ as follows

$$x^{k+1} \in \operatorname{argmin}\left\{ f(x) + \frac{c}{2}\|Ax - z^k + c^{-1}y^k\|^2 + \frac{1}{2}\|x - x^k\|_{M_1}^2 \right\}, \tag{1}$$

$$z^{k+1} = \operatorname{argmin}\left\{ g(z) + \frac{c}{2}\|A\eta^k - z + c^{-1}y^k\|^2 + \frac{1}{2}\|z - z^k\|_{M_2}^2 \right\}, \tag{2}$$

$$y^{k+1} = y^k + c(Ax^{k+1} - z^{k+1}), \tag{3}$$

where we define:

$$\eta^k := \begin{cases} x^k, & \text{Parallel Steps,} \\ x^{k+1}, & \text{Alternating Steps.} \end{cases} \tag{4}$$

- Adequate choices of $M_1, M_2 \succeq 0$ allows to derive various algorithms along announced strategies.
- Allows to derive convergence and efficiency estimates via a simple unifying analysis for many – old and new – PMM based algorithms.

Examples I – Parallel Schemes $\eta^k \equiv x^k$

Well definiteness of scheme, convergence and complexity ensured with:

$$\clubsuit \quad M_1 - cA^T A \succeq 0, \text{ and } M_2 - cI_m \succeq 0.$$

- **Example 1:** $M_1 := \tau^{-1} I_n - cA^T A, \quad M_2 := (\sigma^{-1} - c) I_m$ for any $\tau, \sigma > 0$.
- Condition $\clubsuit \;\Rightarrow\; 2c \le \min\{\sigma^{-1}, \tau^{-1}\|A\|^2\}$.
- This recovers the PCPM algorithm [Chen-T. (94)]. Here we also establish its complexity.

- **Example 2:**
- Pick $M_1 := \tau^{-1} I_n - cA^T A, \quad M_2 := cI_m$.
- Condition $\clubsuit \;\Rightarrow\; \quad 2c\tau\|A\|^2 \le 1$.
- This appears to be a novel scheme.

Alernating Steps $\eta^k \equiv x^{k+1}$ - A Prototype : Alternating Direction of Proximal Method of Multipliers

Eliminate the coupling (x, z) via alternating minimization steps.

Glowinski-Marocco (75), Gabay-Mercier (76), Fortin-Glowinski (83), Ecsktein-Bertsekas (91) ...... the so-called *Alternating Direction of Mulipliers* (ADM),(based on the Multiplier Methods, i.e., $M_1 = M_2 \equiv 0$.)

(AD-PMM) Alternating Direction Proximal Method of Multipliers

1. Start with any $(x^0, z^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ and $c > 0$
2. For $k = 0, 1, \ldots$ generate the sequence $\{x^k, z^k, y^k\}$ as follows:

$$\begin{aligned} x^{k+1} &\in \operatorname{argmin}\left\{ f(x) + \frac{c}{2}\|Ax - z^k + c^{-1}y^k\|^2 + \frac{1}{2}\|x - x^k\|^2_{M_1} \right\}, \\ z^{k+1} &= \operatorname{argmin}\left\{ g(z) + \frac{c}{2}\|Ax^{k+1} - z + c^{-1}y^k\|^2 + \frac{1}{2}\|z - z^k\|^2_{M_2} \right\}, \\ y^{k+1} &= y^k + c(Ax^{k+1} - z^{k+1}). \end{aligned}$$

Nicely exploits separable f, g.
Useful when $(\mathbf{x}, \mathbf{z})$ steps are "easy" to implement *exactly* or *inexactly* (e.g., via strategies just mentioned).

Examples – Alternate Schemes $\eta^k \equiv x^{k+1}$

Well definiteness, convergence and complexity ensured with <u>any</u> $M_1, M_2 \succeq 0$.

(1) Classical ADM (Alternating Direction of Multipliers): $M_1 = M_2 = 0$ Glowinski-Marocco (75), Gabay-Mercier (76), Fortin-Glowinski (83), Ecsktein-Bertsekas (91) ...

- Alternates minimization of the standard Augmented Lagrangian L_c.
- Converges of the primal sequence $\{x^k\}$ is ensured with A has full column rank.

(2) AD-PMM with $M_1 = c^{-1}\mu_1 I_n$; $M_2 = c^{-1}\mu_2 I_m$ with $c, \mu_1, \mu_2 > 0$, [Eckstein (94)].

Examples – Alternate Schemes $\eta^k \equiv x^{k+1}$

Well definiteness, convergence and complexity ensured with <u>any</u> $M_1, M_2 \succeq 0$.

(1) Classical ADM (Alternating Direction of Multipliers): $M_1 = M_2 = 0$ Glowinski-Marocco (75), Gabay-Mercier (76), Fortin-Glowinski (83), Ecsktein-Bertsekas (91) ...

- Alternates minimization of the standard Augmented Lagrangian L_c.
- Converges of the primal sequence $\{x^k\}$ is ensured with A has full column rank.

(2) AD-PMM with $M_1 = c^{-1}\mu_1 I_n; M_2 = c^{-1}\mu_2 I_m$ with $c, \mu_1, \mu_2 > 0$, [Eckstein (94)].

(3) Partial Regularized ADMM: $M_1 \succ 0, M_2 \succeq 0$

- For example, one can use $M_1 := \tau^{-1} I_n$, and $M_2 = 0$.
- Allows to prove the convergence of the sequence $\{x^k\}$ without any assumption on the matrix A.

Examples – Alternate Schemes $\eta^k \equiv x^{k+1}$

Well definiteness, convergence and complexity ensured with <u>any</u> $M_1, M_2 \succeq 0$.

(1) Classical ADM (Alternating Direction of Multipliers): $M_1 = M_2 = 0$ Glowinski-Marocco (75), Gabay-Mercier (76), Fortin-Glowinski (83), Ecsktein-Bertsekas (91) ...

- Alternates minimization of the standard Augmented Lagrangian L_c.
- Converges of the primal sequence $\{x^k\}$ is ensured with A has full column rank.

(2) AD-PMM with $M_1 = c^{-1}\mu_1 I_n; M_2 = c^{-1}\mu_2 I_m$ with $c, \mu_1, \mu_2 > 0$, [Eckstein (94)].

(3) Partial Regularized ADMM: $M_1 \succ 0, M_2 \succeq 0$

- For example, one can use $M_1 := \tau^{-1} I_n$, and $M_2 = 0$.
- Allows to prove the convergence of the sequence $\{x^k\}$ without any assumption on the matrix A.

(4) Mixed Srategy: Linearize and Alternating Minimization

- Linearization wrt x, combined with AM in z. This is achieved by choosing:
- $M_1 := \tau^{-1} I_n - cA^T A \succeq 0, \Leftrightarrow c\tau\|A\|^2 \leq 1; \qquad M_2 := 0.$
- This recovers the recent PD algorithm [Chambolle-Pock (2010)].

Global Rate of Convergence Results - [Shefi-T. (2014)]

The proposed unified simple framework covers/extends many schemes/results.

For all resulting schemes we have:

- $O(1/N)$ **Ergodic convergence rate** in primal-dual gap (bounded domains) and in function values (when g-Lipschitz continuous).
- $O(1/\sqrt{n})$ **Non-ergodic rate** for the residual's norm sequence/constraint violations.

Global Rate of Convergence Results - [Shefi-T. (2014)]

The proposed unified simple framework covers/extends many schemes/results.

For all resulting schemes we have:

- $O(1/N)$ **Ergodic convergence rate** in primal-dual gap (bounded domains) and in function values (when g-Lipschitz continuous).
- $O(1/\sqrt{n})$ **Non-ergodic rate** for the residual's norm sequence/constraint violations.

PMM based schemes are not free of potential problems ..raising practical and theoretical issues:

- **The penalty parameter c is unknown:** trial/error runs, fine tuning, heuristics, ...
- Iteration complexity bounds **depend on c !**
- The (x, z) steps are not always "easy"..**Prox of composition with affine map**...Nested optimization
- Difficult to extend for sum of $m > 2$ convex composite functions with linear maps.

Any alternatives?..

For any sequence $\{\mathbf{w}^n\}$, any $N \geq 1$, the ergodic sequence $\mathbf{w}_N := \frac{1}{N}\sum_{n=1}^{N} \mathbf{w}^n$

A Convex-Concave Saddle-point Approach

$$\min_{u \in U} \max_{v \in V} \{K(u,v) := f(u) + \langle Au, v\rangle - g(v)\},\ U, V \text{ closed convex.}$$

f, g are convex functions, A is a linear map.

Obviously, recovers and extends the previous composite convex model.

Current methods which admit an $O(1/\varepsilon)$ efficiency estimate

- **PMM-Based:** Just discussed with its potential drawbacks.
- **Extragradient** [Korpelevitch, (1976), Nemirovsky (04), Auslender-T. (05)] (can also handle general variational inequalities).
  - Requires smooth data: f and g have Lipschitz-continuous gradients.
- **Smoothing/First Order Methods:** [Moreau (64)...Nesterov's (05), Beck-T. (12)]
  - Assume partial smoothness/compactness: $f \in C_L^{1,1}$, V compact.
  - Require a smoothing parameter in term of the accuracy fixed in advance.

Goal

An algorithm for a broader class of structured nonsmooth convex-concave saddle-point problem that achieves the nonasymptotic efficiency estimate $O(1/\varepsilon)$:

- Removes difficulties with current methods.
- Flexible enough to be applied to more general scenarios.
- Involves simple computational tasks.

A Class of Structured Convex-Concave Saddle-Point Model

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

Data Information

(i) $f : \mathbb{R}^n \to \mathbb{R}$ is convex $C^{1,1}_{L_f}$: $\|\nabla f(u_1) - \nabla f(u_2)\| \leq L_f \|u_1 - u_2\|$, $\forall u_1, u_2$.

(ii) $g_i : \mathbb{R}^{d_i} \to (-\infty, +\infty]$, $i = 1, 2, \ldots, m$, is a proper, (lsc) and convex function (possibly nonsmooth), and we let $g : \mathbb{R}^d \to (-\infty, +\infty]$

$$g(v) := \sum_{i=1}^{m} g_i(v); \; d := \sum_{i=1}^{m} d_i; \; v := (v_1, v_2, \ldots, v_m) \in \mathbb{R}^d.$$

(iii) $A_i : \mathbb{R}^{d_i} \to \mathbb{R}^n$, $i = 1, 2, \ldots, m$, is a linear map and we let $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^n$ be the linear map defined by $\mathcal{A}v = \sum_{i=1}^{m} A_i v_i$.

We assume that $K(\cdot, \cdot)$ has a saddle-point, i.e., there exists $(u^*, v^*) \in \mathbb{R}^n \times \mathbb{R}^d$ such that

$$K(u^*, v) \leq K(u^*, v^*) \leq K(u, v^*), \quad \forall \, u \in \mathbb{R}^n, \; v \in \mathbb{R}^d.$$

A Proximal Alternating Predictor Corrector (PAPC) for (M)
Drori -Sabach -T. (2015) – Advertising Time!

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

Algorithm based on fundamental and old ideas: **it blends duality, predictor-corrector steps, and proximal operation**.

A Proximal Alternating Predictor Corrector (PAPC) for (M) Drori -Sabach -T. (2015) – Advertising Time!

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

Algorithm based on fundamental and old ideas: **it blends duality, predictor-corrector steps, and proximal operation**.

Features of PAPC - Fully exploits structures of a problem.

- PAPC **avoids the computational difficult task:** the prox of the composition with a linear map $(g \circ \mathcal{A})(\mathbf{x}) = g(\mathcal{A}\mathbf{x})$. **Only ask prox of $g(\cdot)$.**
- Can be easily applied to minimization problems **with sum of such composite terms in objective/constraints**.
- Constraints on the variable v, built-in thanks to g being extended valued.
- Constraints on the variable u can be easily handled via **The Dual Transportation Trick**, (see details in Paper).

The PAPC Method

PAPC

Initialization. $\left(u^0, v^0\right) \in \mathbb{R}^n \times \mathbb{R}^d$, $\tau > 0$, and $\{\sigma_i\}_{i=1}^m > 0$.
For $k = 1, 2, \ldots,$:

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

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

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

Output: $\bar{u}^N = \frac{1}{N}\sum_{k=1}^N u^k$, $\bar{v}^N = \frac{1}{N}\sum_{k=1}^N v^k$.

$\clubsuit$ v step "decomposes" according to structure; **only** prox for each $g_i(\cdot)$, **not of** $g(A_i x)$.

The parameters (τ, σ_i) are defined in terms of problem's data L_f, A_i.

Each iteration requires *one* application of $\mathcal{A}$ and of $\mathcal{A}^T$ and one evaluation of ∇f.

The PAPC Method – Main Convergence Results -Drori-Sabach-T. (2015)

Shares the best theoretical rate $0(1/N)$ for convex-concave saddle point.

Let $\left\{\left(p^k, u^k, v^k\right)\right\}_{k\in\mathbb{N}}$ be a sequence generated by the PAPC algorithm with $\tau L_f \leq 1$ and $\sigma\tau\sum_{i=1}^m \|A_i\|^2 \leq 1$.

1. **Global Rate of Convergence -Ergodic**

$$K\left(\bar{u}^N, v\right) - K\left(u, \bar{v}^N\right) = O(1/N).$$

Bound constant in terms of Data (L_f, A_i)– Parameters free.

2. **Sequential Convergence:** The sequence $\{(u^k, v^k)\}_{k\in\mathbb{N}}$ converges to a saddle-point (u^*, v^*) of K.

PAPC Applies to Many Important Models

♣ Convex Problems with Sum of Composite Convex Functions with Linear Maps

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

For all these models, PAPC

- **Decomposes nicely according to given structure.**
- **Removes the difficult task of "computing prox of convex function composed with an affine map".**
- **Parameters are determined from problem's data info: L_f and A_i.**
- **Performs well in applications:** Image processing, Learning (Fused lasso)...

Non-Convex Smooth Models

Sparse PCA

Principal Component Analysis solves

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

while Sparse Principal Component Analysis solves

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

$\|x\|_0$ counts the number of nonzero entries of x

Issues:

1. Maximizing a Convex objective.
2. Hard Nonconvex Constraint $\|x\|_0 \leq k$.

Possible Approaches:

1. SDP Convex Relaxations
2. Approximation/Modified formulations: Many proposed approaches

Sparse PCA via Penalization/Relaxation/Approx.

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

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

Sparse PCA via Penalization/Relaxation/Approx.

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

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

♠ Literature has focused on solving various modifications:

- l_0**-penalized PCA** $\max\{x^T Ax - s\|x\|_0 : \|x\|_2 = 1\},\ s > 0$
- **Relaxed** l_1**-constrained PCA** $\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_1 \leq \sqrt{k}\}$
- **Relaxed** l_1**-penalized PCA** $\max\{x^T Ax - s\|x\|_1 : \|x\|_2 = 1\}$
- **Approx-Penalized** $\max\{x^T Ax - sg_p(|x|) : \|x\|_2 = 1\}$ $g_p(x) \simeq \|x\|_0$
- **SDP-Convex Relaxations** $\max\{\mathrm{tr}(AX) :\ \mathrm{tr}\,(X) = 1, X \succeq 0, \|X\|_1 \leq k\}$

Sparse PCA via Penalization/Relaxation/Approx.

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

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

♠ Literature has focused on solving various modifications:

- l_0**-penalized PCA** $\max\{x^T Ax - s\|x\|_0 : \|x\|_2 = 1\},\ s > 0$
- **Relaxed** l_1**-constrained PCA** $\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_1 \leq \sqrt{k}\}$
- **Relaxed** l_1**-penalized PCA** $\max\{x^T Ax - s\|x\|_1 : \|x\|_2 = 1\}$
- **Approx-Penalized** $\max\{x^T Ax - sg_p(|x|) : \|x\|_2 = 1\}$ $g_p(x) \simeq \|x\|_0$
- **SDP-Convex Relaxations** $\max\{\mathrm{tr}(AX) : \mathrm{tr}(X) = 1, X \succeq 0, \|X\|_1 \leq k\}$

- SDP-relaxations often too computationally expensive for large problems.
- No algorithm give bounds to the optimal solution of the **original problem**.
- Even when "Simple", these algorithms are for modifications:
  ♣ **do not solve the original problem of interest**
  ♣ **do require unknown penalty parameter s to be tuned**.

Quick Highlight of Simple Algorithms for "Modified Problems"

Type	Iteration	Per-Iteration Complexity	References
l_1-constrained	$x_i^{j+1} = \frac{\mathrm{sgn}(((A+\frac{\sigma}{2})x^j)_i)(\lvert((A+\frac{\sigma}{2})x^j)_i\rvert - \lambda^j)_+}{\sqrt{\sum_h (\lvert((A+\frac{\sigma}{2})x^j)_h\rvert - \lambda^j)_+^2}}$	$O(n^2)$, $O(mn)$	Witten et al. (2009)
l_1-constrained	$x_i^{j+1} = \frac{\mathrm{sgn}((Ax^j)_i)(\lvert(Ax^j)_i\rvert - s^j)_+}{\sqrt{\sum_h (\lvert(Ax^j)_h\rvert - s^j)_+^2}}$ where s^j is $(k+1)$-largest entry of vector $\lvert Ax^j\rvert$	$O(n^2)$, $O(mn)$	Sigg-Buhman (2008)
l_0-penalized	$z^{j+1} = \frac{\sum_i [\mathrm{sgn}((b_i^T z^j)^2 - s)]_+ (b_i^T z^j) b_i}{\lVert \sum_i [\mathrm{sgn}((b_i^T z^j)^2 - s)]_+ (b_i^T z^j) b_i \rVert_2}$	$O(mn)$	Shen-Huang (2008), Journee et al. (2010)
l_0-penalized	$x_i^{j+1} = \frac{\mathrm{sgn}(2(Ax^j)_i)(\lvert 2(Ax^j)_i\rvert - s\varphi_p'(\lvert x_i^j\rvert))_+}{\sqrt{\sum_h (\lvert 2(Ax^j)_h\rvert - s\varphi_p'(\lvert x_h^j\rvert))_+^2}}$	$O(n^2)$	Sriperumbudur et al. (2010)
l_1-penalized	$y^{j+1} = \operatorname{argmin}_y \{\sum_i \lVert b_i - x^j y^T b_i \rVert_2^2 + \lambda \lVert y \rVert_2^2 + s \lVert y \rVert_1\}$ $x^{j+1} = \frac{(\sum_i b_i b_i^T) y^{j+1}}{\lVert (\sum_i b_i b_i^T) y^{j+1} \rVert_2}$		Zou et al. (2006)
l_1-penalized	$z^{j+1} = \frac{\sum_i (\lvert b_i^T z^j\rvert - s)_+ \mathrm{sgn}(b_i^T z^j) b_i}{\lVert \sum_i (\lvert b_i^T z^j\rvert - s)_+ \mathrm{sgn}(b_i^T z^j) b_i \rVert_2}$	$O(mn)$	Shen-Huang (2008), Journee et al. (2010)

A Plethora of Models/Algorithms Revisited - [Luss-Teboulle (2013)]

All previous listed algorithms have been derived from various disparate approaches/motivations to solve **modifications** of SPCA: Expectation Maximization; Majorization-Mininimization techniques; DC programming; Alternating minimization etc...

1. **Are all these algorithms different? Any connection?**
2. **Is it possible to tackle the difficult sparse PCA problem "as is"**

A Plethora of Models/Algorithms Revisited - [Luss-Teboulle (2013)]

All previous listed algorithms have been derived from various disparate approaches/motivations to solve **modifications** of SPCA: Expectation Maximization; Majorization-Mininimization techniques; DC programming; Alternating minimization etc...

1. **Are all these algorithms different? Any connection?**
2. **Is it possible to tackle the difficult sparse PCA problem "as is"**

We have shown that:

- All the previously listed algorithms are a particular realization of a **"Father Algorithm": ConGradU** (based on the well-known Conditional Gradient Algorithm)
- **ConGradU CAN be applied directly to the original problem!**

Maximizing a Convex function over a Compact Nonconvex set

Classic Conditional Gradient Algorithm [Frank-Wolfe'56, Polyak'63, Dunn'79..]

$$\text{solves}: \max\{F(x) : x \in C\}, \quad \text{with} \quad F \text{ is } C^1;\ C \text{ convex compact}$$

$$\begin{aligned} x^0 \in C,\ p^j &= \operatorname{argmax}\{\langle x - x^j, \nabla F(x^j)\rangle : x \in C\} \\ x^{j+1} &= x^j + \alpha^j(p^j - x^j),\ \alpha^j \in (0,1] \text{ stepsize} \end{aligned}$$

♠ **Here : F is convex, possibly nonsmooth**; C is compact but **nonconvex**

Idea goes back to Mangasarian (96) developed for C a polyhedral set.

ConGradU – Conditional Gradient with Unit Step Size

$$x^0 \in C,\ x^{j+1} \in \operatorname{argmax}\{\langle x - x^j, F'(x^j)\rangle : x \in C\}$$

Notes:

1. F is not assumed to be differentiable and $F'(x)$ is a subgradient of F at x.
2. Useful when $\max\{\langle x - x^j, F'(x^j)\rangle : x \in C\}$ is *easy* to solve

Solving Original l_0-constrained PCA via ConGradU

Applying **ConGradU** directly to $\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$ results in

$$\begin{aligned} x^{j+1} &= \operatorname{argmax}\{x^{jT} Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k\} = \frac{T_k(Ax^j)}{\|T_k(Ax^j)\|_2} \\ T_k(a) &:= \operatorname*{argmin}_y\{\|x - a\|_2^2 : \|x\|_0 \leq k\} \end{aligned}$$

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.

Solving Original l_0-constrained PCA via ConGradU

Applying **ConGradU** directly to $\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \le k,\ x \in \mathbf{R}^n\}$ results in

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

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.

- **Convergence:** Every limit point of $\{x^j\}$ converges to a stationary point.
- **Complexity:** $O(kn)$ or $O(mn)$

- **Thus, original problem can be solved using ConGradU with the same complexity as when applied to modifications!**
- Penalized/Modified problems require tuning **an unknown tradeoff penalty parameter** This can be very computationally expensive and not needed here.

ConGradU for a General Class of Problems

$$(G) \qquad \max_x \{f(x) + g(|x|) : x \in C\}$$

$$\begin{array}{ll} f : \mathbf{R}^n \to \mathbf{R} & \text{is convex, } C \subseteq \mathbf{R}^n \text{ is a compact set.} \\ g : \mathbf{R}^n_+ \to \mathbf{R} & \text{is convex differentiable and montonote decreasing} \end{array}$$

- Particularly useful for handling *approximate* l_0-penalized problems.

ConGradU for a General Class of Problems

$$(G) \qquad \max_x \{f(x) + g(|x|) : x \in C\}$$

$$\begin{array}{ll} f : \mathbf{R}^n \to \mathbf{R} & \text{is convex, } C \subseteq \mathbf{R}^n \text{ is a compact set.} \\ g : \mathbf{R}^n_+ \to \mathbf{R} & \text{is convex differentiable and montonote decreasing} \end{array}$$

- Particularly useful for handling *approximate* l_0-penalized problems.
- **CondGradU** applied to (G) produces the following simple:

Weighted l_1-norm maximization problem:

$$x^0 \in C, \ x^{j+1} = \text{argmax}\{\langle a^j, x\rangle - \sum_i w_i^j |x_i| : x \in C\}, \ j = 0, \dots,$$

where $w^j := -g'(|x^j|) > 0$ and $a^j := f'(x^j) \in \mathbf{R}^n$.

For *penalized/approximate penalized SPCA*, C is a unit ball, and above admits a **closed form solution**:

$$x^{j+1} = \frac{S_{w^j}(f'(x^j))}{\|S_{w^j}(f'(x^j))\|}, \ j = 0, \dots; \quad S_w(a) := (|a| - w)_+ \text{sgn}(a), \ \text{(Soft Threshold)}.$$

Non-Convex and NonSmooth

Goal and Results

Derive a simple self-contained convergence analysis framework for a broad class of nonconvex and nonsmooth minimization problems.

- A "Recipe" for proving global convergence to a critical point.
- An Example of a Simple/Useful Algorithm: PALM.
- Many Applications: phase retrieval for diffractive imaging, dictionary learning,... .... **Sparse nonnegative matrix factorization** ...and much more...

The Problem : An Abstract Formulation

Let $\Psi : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper, lsc and bounded from below function.

$$(P) \qquad \inf \left\{ \Psi(z) : \; z \in \mathbb{R}^d \right\}.$$

Suppose $\mathcal{A}$ is a generic algorithm which generates a sequence $\left\{z^k\right\}_{k \in \mathbb{N}}$ via:

$$z^0 \in \mathbb{R}^d, z^{k+1} \in \mathcal{A}(z^k), \; k = 0, 1, \ldots.$$

Goal: Prove that the whole sequence $\left\{z^k\right\}_{k \in \mathbb{N}}$ converges to a critical point z^* of Ψ, i.e., $0 \in \partial\Psi(z^*)$.

Recall [Rockafellar-Wets (98)]

- (Limiting) Subdifferential $\partial\Psi(x)$:

$$\begin{aligned} x^* \in \partial\Psi(x) \quad &\text{iff} \quad (x_k, x^*) \to (x, x^*) \text{ s.t. } \Psi(x_k) \to \Psi(x) \text{ and} \\ \Psi(u) \quad &\geq \quad \Psi(x_k) + \langle x_k^*, u - x_k \rangle + o(\|u - x_k\|) \end{aligned}$$

- $x \in \mathbb{R}^d$ is a critical point of Ψ if $\partial\Psi(x) \ni 0$.

A General Recipe with 3 Main Steps

C1 **Sufficient decrease property:** Find a positive constant ρ_1 such that

$$\rho_1 \|z^{k+1} - z^k\|^2 \leq \Psi(z^k) - \Psi(z^{k+1}), \quad \forall k = 0, 1, \ldots.$$

C2 **A subgradient lower bound for the iterates gap:** Assume that $\left\{z^k\right\}_{k \in \mathbb{N}}$ is bounded. Find another positive constant ρ_2, such that

$$\left\|w^k\right\| \leq \rho_2 \|z^{k+1} - z^k\|, \quad w^k \in \partial\Psi(z^k), \quad \forall k = 0, 1, \ldots.$$

- These two steps are typical for *any descent* type algorithms but lead **ONLY to convergence of limit points.** [Ostrowski 1966].

A General Recipe with 3 Main Steps

C1 **Sufficient decrease property:** Find a positive constant ρ_1 such that

$$\rho_1 \|z^{k+1} - z^k\|^2 \leq \Psi(z^k) - \Psi(z^{k+1}), \quad \forall k = 0, 1, \ldots .$$

C2 **A subgradient lower bound for the iterates gap:** Assume that $\left\{z^k\right\}_{k \in \mathbb{N}}$ is bounded. Find another positive constant ρ_2, such that

$$\left\| w^k \right\| \leq \rho_2 \|z^{k+1} - z^k\|, \quad w^k \in \partial \Psi(z^k), \quad \forall k = 0, 1, \ldots .$$

- These two steps are typical for *any descent* type algorithms but lead **ONLY to convergence of limit points.** [Ostrowski 1966].
- To get global convergence to a critical point ...We need more info on problem's data.
- To prove the result, we need an additional mathematical tool. **This is the third step of the recipe**.

The Third Main Step of the Recipe

C3. The Kurdyka-Łojasiewicz property: Assume that Ψ 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!

The Third Main Step of the Recipe

C3. The Kurdyka-Łojasiewicz property: Assume that Ψ 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!

This general recipe

- Singles out the 3 main ingredients at play to derive global convergence in the nonconvex and nonsmooth setting.
- In particular, thanks to a uniformization Lemma of the KL property, [Bolte, Sabach, T. (2014)] it is **applicable to any descent algorithm** without the need of going through the KL machinery for each particular algorithm.

The Third Main Step of the Recipe

C3. The Kurdyka-Łojasiewicz property: Assume that Ψ 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!

This general recipe

- Singles out the 3 main ingredients at play to derive global convergence in the nonconvex and nonsmooth setting.
- In particular, thanks to a uniformization Lemma of the KL property, [Bolte, Sabach, T. (2014)] it is **applicable to any descent algorithm** without the need of going through the KL machinery for each particular algorithm.

The remaining questions

- What is the KL property ?Łojasiewicz (68), Kurdyka (98), Bolte et al. (06,07,10)
- Are there many functions satisfying KL?

The Third Main Step of the Recipe

C3. The Kurdyka-Łojasiewicz property: Assume that Ψ 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!

This general recipe

- Singles out the 3 main ingredients at play to derive global convergence in the nonconvex and nonsmooth setting.
- In particular, thanks to a uniformization Lemma of the KL property, [Bolte, Sabach, T. (2014)] it is **applicable to any descent algorithm** without the need of going through the KL machinery for each particular algorithm.

The remaining questions

- What is the KL property ?Łojasiewicz (68), Kurdyka (98), Bolte et al. (06,07,10)
- Are there many functions satisfying KL?

Theorem 1 (Bolte-Daniilidis-Lewis (2006))

Let $\sigma : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper and lsc function. If σ is semi-algebraic then it satisfies the KL property at any point of $\operatorname{dom}\sigma$.

Global Convergence to a Critical Point [Bolte-Sabach-T. 2014]

Global Convergence Result

Let $\Psi : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper lsc and **semi-algebraic function** with $\inf \Psi > -\infty$. Assume that $\left\{\mathbf{z}^k\right\}_{k\in\mathbb{N}}$ is a sequence produced **by any algorithm** satisfying conditions C1 and C2. Let $\omega\left(\mathbf{z}^0\right)$ be the set of all limit points of the sequence $\left\{\mathbf{z}^k\right\}_{k\in\mathbb{N}}$.

If $\emptyset \neq \omega\left(\mathbf{z}^0\right) \subset \operatorname{crit}\Psi$, then the sequence $\left\{\mathbf{z}^k\right\}_{k\in\mathbb{N}}$ converges to a critical point $\mathbf{z}^*$ of Ψ.

Recall: Semi-algebraic sets and functions

(i) A semialgebraic subset of $\mathbb{R}^d$ is a finite union of sets

$$\{x \in \mathbb{R}^d : \ p_i(x) = 0, \ q_j(x) < 0, \ i \in I, \ j \in J\}$$

where $p_i, q_j : \mathbb{R}^d \to \mathbb{R}$ are real polynomial functions and I, J are finite.

(ii) A function σ is semi-algebraic if its graph

$$\left\{(u, t) \in \mathbb{R}^{n+1} : \ \sigma(u) = t\right\}$$

is a semi-algebraic subset of $\mathbb{R}^{n+1}$.

There is a Wealth of Semi-Algebraic Functions!

Some Semi-Algebraic Sets/Functions .."Starring" in Optimization/Applications

- Real polynomial functions.
- Indicator functions of semi-algebraic sets.
- In matrix theory: cone of PSD matrices, constant rank matrices, Stiefel manifolds...
- The function $x \to \operatorname{dist}(x, S)^2$ is semi-algebraic whenever S is a nonempty semi-algebraic subset of $\mathbb{R}^n$.
- $\|\cdot\|_0$ is semi-algebraic.
- $\|\cdot\|_p$ is semi-algebraic whenever $p > 0$ is rational.

Semi-Algebraic Property is Preserved under Many Operations

- Finite sums and product of semi-algebraic functions; Composition of semi-algebraic functions;
- Sup/Inf type function, *e.g.*, $\sup \{g(u, v) : \; v \in C\}$ is semi-algebraic when g is a semi-algebraic function and C a semi-algebraic set.

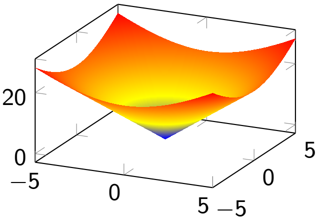
Sharpness: A Geometric Snapshot toward KL

Definition 2 (Sharpness)

A function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is called sharp on the slice $[r_0 < f < r_1] := \left\{x \in \mathbb{R}^d : \ r_0 < f(x) < r_1\right\}$, if there exists $c > 0$ such that

$$\|\partial f(x)\|_{-} := \min\{\|\xi\| : \ \xi \in \partial f(x)\} \geq c > 0 \quad \forall x \in [r_0 < f < r_1].$$

Basic Example: $f(x) = \|x\|$.



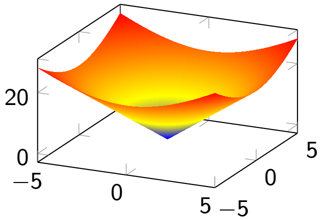
Sharpness: A Geometric Snapshot toward KL

Definition 2 (Sharpness)

A function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is called sharp on the slice $[r_0 < f < r_1] := \left\{x \in \mathbb{R}^d : \; r_0 < f(x) < r_1\right\}$, if there exists $c > 0$ such that

$$\|\partial f(x)\|_{-} := \min\{\|\xi\| : \; \xi \in \partial f(x)\} \geq c > 0 \quad \forall x \in [r_0 < f < r_1].$$

Basic Example: $f(x) = \|x\|$.



KL Property Informal: A KL function is a function whose values can be re-parametrized in the neighborhood of each of its critical point so that the resulting function becomes sharp.

KL Property [Łojasiewicz (68), Kurdyka (98), Bolte et al. (06,07,10)]

$$\Phi_\eta := \left\{\varphi \in C\left([0,\eta), \mathbb{R}_+\right) \text{concave}: \ \varphi \in C^1\left((0,\eta)\right), \varphi' > 0, \varphi(0) = 0\right\}, \eta \in (0,+\infty].$$

Definition 3 (Kurdyka-Łojasiewicz property)

Let $\sigma : \mathbb{R}^d \to (-\infty, +\infty]$ be proper and lsc.

(i) The function σ has the Kurdyka-Łojasiewicz (KL) property at $\overline{u}$ if there exist a neighborhood U of $\overline{u}$, and a function $\varphi \in \Phi_\eta$, such that the following inequality holds:

$$\varphi'\left(\sigma(u) - \sigma(\overline{u})\right) \operatorname{dist}\left(0, \partial\sigma(u)\right) \geq 1.$$

for all

$$u \in U \cap \left[\sigma(\overline{u}) < \sigma(u) < \sigma(\overline{u}) + \eta\right].$$

(ii) If σ satisfy the KL property at each point of $\operatorname{dom}\partial\sigma$ then σ is called a KL function.

The relevant aspect of this property is when $\overline{u}$ is critical, i.e., $0 \in \partial\sigma(\overline{u})$. In that case:

- it warrants that σ is *sharp* up to re-parametrization of its values.
- The re-parametrization function is called the *desingularizing function* of σ at $\overline{u}$.

Illustration on a Useful Optimization Model

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

Illustration on a Useful Optimization Model

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

Assumption 1

(i) $f : \mathbb{R}^n \to (-\infty, +\infty]$ *and* $g : \mathbb{R}^m \to (-\infty, +\infty]$ *proper and lsc functions.*

(ii) $H : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ *is a* C^1 *function.*

(iii) *Partial gradients of* H *are Lipshitz continuous:* $H(\cdot, y) \in C^{1,1}_{L(y)}$ *and* $H(x, \cdot) \in C^{1,1}_{L(x)}$.

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

Illustration on a Useful Optimization Model

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

Assumption 1

(i) $f : \mathbb{R}^n \to (-\infty, +\infty]$ *and* $g : \mathbb{R}^m \to (-\infty, +\infty]$ *proper and lsc functions.*

(ii) $H : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ *is a* C^1 *function.*

(iii) *Partial gradients of* H *are Lipshitz continuous:* $H(\cdot, y) \in C^{1,1}_{L(y)}$ *and* $H(x, \cdot) \in C^{1,1}_{L(x)}$.

- **NO convexity** is assumed in the objective and the constraints (built-in through f and g extended valued).
- The choice of two blocks of variables is only for the sake of simplicity of exposition. Same for the p-blocks case:

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

- This optimization model covers many applications: signal/image processing, machine learning, etc....Vast Literature..

The Algorithm: Proximal Alternating Linearization Minimization (PALM)

Cocktail Time! PALM simply "blends" old spices: AM and Prox-Gradient.

1. Initialization: start with any $\left(x^0, y^0\right) \in \mathbb{R}^n \times \mathbb{R}^m$.
2. For each $k = 0, 1, \ldots$ generate a sequence $\left\{\left(x^k, y^k\right)\right\}_{k \in \mathbb{N}}$:

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

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

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

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

Main computational step: prox of a "nonconvex" function.

Application to a Broad Class of Matrix Factorization Problems

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

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

Where

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

Here R_1 and R_2 are lsc functions and $\alpha, \beta \in \mathbb{R}_+$ are given parameters. R_1 (R_2) are often used to describe some additional features of X (Y).

(MF) covers a very large number of problems in applications...

The Optimization Approach

We adopt the Constrained Nonconvex Nonsmooth Formulation

$$(MF) \qquad \min \left\{ d\left(A, XY\right) : \; X \in \mathcal{K}_{m,r} \cap \mathcal{F}, Y \in \mathcal{K}_{r,n} \cap \mathcal{G} \right\},$$

- $d : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \to \mathbb{R}_+$ stands as a proximity function.
- Measures the quality of the approximation, satisfies $d\left(U, V\right) = 0$ if and only if $U = V$.

This formulation fits our general nonsmooth nonconvex model (M) with obvious identifications for H, f, g.

We now illustrate with semi-algebraic data on two important models.

Model I – Nonnegative Matrix Factorization Problems

Let the proximity measure be defined via the Frobenius norm

$$\begin{aligned} d(A, XY) &:= H(X, Y) = \frac{1}{2}\|A - XY\|_F^2, \text{ and} \\ \mathcal{F} &\equiv \mathbb{R}^{m\times r}; \quad \mathcal{G} \equiv \mathbb{R}^{r\times n}. \end{aligned}$$

The Problem (MF) reduces to the so called Nonnegative Matrix Factorization (NMF)

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

Model I – Nonnegative Matrix Factorization Problems

Let the proximity measure be defined via the Frobenius norm

$$d(A, XY) \quad := \quad H(X, Y) = \frac{1}{2}\|A - XY\|_F^2, \text{ and}$$
$$\mathcal{F} \quad \equiv \quad \mathbb{R}^{m\times r}; \quad \mathcal{G} \equiv \mathbb{R}^{r\times n}.$$

The Problem (MF) reduces to the so called Nonnegative Matrix Factorization (NMF)

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

- H is a real polynomial function hence semi-algebraic.
- $X \to H(X, Y)$ (for fixed Y) and $Y \to H(X, Y)$ (for fixed X), are $C^{1,1}$ with $L_1(Y) \equiv \left\|YY^T\right\|_F$, $L_2(X) \equiv \left\|X^TX\right\|_F$.
- H is C^2 on bounded subsets.

Model I – Nonnegative Matrix Factorization Problems

Let the proximity measure be defined via the Frobenius norm

$$
\begin{aligned}
d(A, XY) \quad &:= \quad H(X, Y) = \frac{1}{2}\|A - XY\|_F^2, \text{ and} \\
\mathcal{F} \quad &\equiv \quad \mathbb{R}^{m\times r}; \quad \mathcal{G} \equiv \mathbb{R}^{r\times n}.
\end{aligned}
$$

The Problem (MF) reduces to the so called Nonnegative Matrix Factorization (NMF)

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

- H is a real polynomial function hence semi-algebraic.
- $X \to H(X, Y)$ (for fixed Y) and $Y \to H(X, Y)$ (for fixed X), are $C^{1,1}$ with $L_1(Y) \equiv \left\|YY^T\right\|_F$, $L_2(X) \equiv \left\|X^TX\right\|_F$.
- H is C^2 on bounded subsets.

Thus we can PALM it! The two computational steps reduce to projection onto the nonnegative cone of matrices–Trivial!..

$$
P_+(U) := \operatorname{argmin}\{\|U - V\|_F^2 : \ V \in \mathbb{R}^{m\times n}, V \geq 0\} = \max\{0, U\}.
$$

Model II - Sparse Constraints in Nonnegative Matrix Factorization

Consider in NMF the overall sparsity measure of a matrix defined by

$$R_1(X) = \|X\|_0 := \sum_i \|x_i\|_0 \,, \ (x_i \text{ column vector of } X) \quad ; R_2(Y) = \|Y\|_0 \,.$$

To apply PALM all we need is to compute the **prox of** $f := \delta_{X \geq 0} + \delta_{\|X\|_0 \leq s}$.
It turns out that this can be simply done!

Model II - Sparse Constraints in Nonnegative Matrix Factorization

Consider in NMF the overall sparsity measure of a matrix defined by

$$R_1(X) = \|X\|_0 := \sum_i \|x_i\|_0, \ (x_i \text{ column vector of } X) \quad ; R_2(Y) = \|Y\|_0 .$$

To apply PALM all we need is to compute the **prox of** $f := \delta_{X \geq 0} + \delta_{\|X\|_0 \leq s}$.
It turns out that this can be simply done!

Proposition 1 (Proximal map formula for $f = \delta_{X \geq 0} + \delta_{\|X\|_0 \leq s}$)

Let $U \in \mathbb{R}^{m \times n}$. Then

$$\mathrm{prox}_1^f(U) = \mathit{argmin}\left\{\frac{1}{2}\|X - U\|_F^2 : X \geq 0, \|X\|_0 \leq s\right\} = T_s(P_+(U))$$

where

$$T_s(U) := \underset{V \in \mathbb{R}^{m \times n}}{\mathit{argmin}}\left\{\|U - V\|_F^2 : \ \|U\|_0 \leq s\right\}.$$

Computing T_s simply requires determining the s-th largest numbers of mn numbers. This can be done in $O(mn)$ time, and zeroing out the proper entries in one more pass of the mn numbers.

PALM for Sparse NMF

1. Initialization: Select random nonnegative $X^0 \in \mathbb{R}^{m \times r}$ and $Y^0 \in \mathbb{R}^{r \times n}$.
2. For each $k = 0, 1, \ldots$ generate a sequence $\left\{ \left(X^k, Y^k \right) \right\}_{k \in \mathbb{N}}$:

2.1. Take $\gamma_1 > 1$, set $c_k = \gamma_1 \left\| Y^k \left(Y^k \right)^T \right\|_F$ and compute

$$U^k = X^k - \frac{1}{c_k} \left(X^k Y^k - A \right) \left(Y^k \right)^T; \quad X^{k+1} \in \operatorname{prox}_{c_k}^{R_1} \left(U^k \right) = T_\alpha \left(P_+ \left(U^k \right) \right).$$

2.2. Take $\gamma_2 > 1$, set $d_k = \gamma_2 \left\| X^{k+1} \left(X^{k+1} \right)^T \right\|_F$ and compute

$$V^k = Y^k - \frac{1}{d_k} \left(X^{k+1} \right)^T \left(X^{k+1} Y^k - A \right); \quad Y^{k+1} \in \operatorname{prox}_{d_k}^{R_2} \left(V^k \right) = T_\beta \left(P_+ \left(V^k \right) \right).$$

- Applying our main Theorem, we get the global convergence result to a critical point.
- The algorithm is simple and appears to be efficient in practice.

For More Details, Results....

- R. Shefi and M. Teboulle. Rate of Convergence Analysis of Decomposition Methods Based on the Proximal Method of Multipliers for Convex Minimization. *SIAM J. Optimization*, **24**, 269–297, (2014).
- Y. Drori, S. Sabach and M. Teboulle. A simple algorithm for a class of nonsmooth convex-concave saddle-point problems. *Operations Research Letters*, **43**, 209–214, (2015).
- R. Luss and M. Teboulle. Conditional Gradient Algorithms for Rank One Matrix Approximations with a Sparsity Constraint. *SIAM Review*, **55**, 65-98, (2013).
- J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming, Series A*, **146**, 459–494, (2014).

For More Details, Results....

- R. Shefi and M. Teboulle. Rate of Convergence Analysis of Decomposition Methods Based on the Proximal Method of Multipliers for Convex Minimization. *SIAM J. Optimization*, **24**, 269–297, (2014).
- Y. Drori, S. Sabach and M. Teboulle. A simple algorithm for a class of nonsmooth convex-concave saddle-point problems. *Operations Research Letters*, **43**, 209–214, (2015).
- R. Luss and M. Teboulle. Conditional Gradient Algorithms for Rank One Matrix Approximations with a Sparsity Constraint. *SIAM Review*, **55**, 65-98, (2013).
- J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming, Series A*, **146**, 459–494, (2014).

THANK YOU FOR YOUR ATTENTION!