First Order Algorithms for Well Structured Optimization Problems

Marc Teboulle

School of Mathematical Sciences
Tel Aviv University

SJOM – Sino-Japan Optimization Meeting
September 26-29, 2011 – Beijing, China

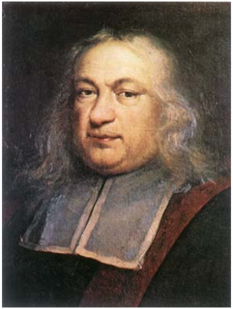
Opening Remark and Credit

About more than 380 years ago.....In 1629, Fermat suggested the following:

Opening Remark and Credit

About more than 380 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

A Wealth of Algorithms Using/Based First Order Information

......Historical Development: Some fundamental Schemes.......

- Fixed point methods [Babylonian time!/Heron for square root, Picard, Banach, Weisfield'34]
- Gauss-Seidel '1798 (coordinate descent), Alternating Minimization
- Gradient methods [Cauchy' 1846, Rosen'63, Frank-Wolfe '56, Polyak'62]
- Stochastic Gradients [Robbins and Monro '51]
- Arrow-Hurwicz ['58]; Subgradient methods [Shor'61, Polyak'64]
- Proximal-Algorithms [Martinet '70, Rockafellar '76, Fukushima-Mine'81]
- Penalty/Barrier methods [Courant'49, Fiacco-McCormick'66]
- Augmented Lagrangians and Splitting [Hestenes-Powell'69, Goldstein-Treyakov'72, Rockafellar'74, Mercier-Lions '79, Passty'79, Fortin-Glowinski'76, Bertsekas'82]
- Extragradient-methods for VI [Korpelevich '76, Konnov,'80]
- Optimal Gradient Schemes [Nemirosvki-Yudin'81, Nesterov'83]
- ....and more............................

Mainly developed as general purpose algorithms

Goals and Outline

Building and Analyzing Simple and Efficient First Order Schemes Exploiting Structures for Various Classes of Problems

Goals and Outline

Building and Analyzing Simple and Efficient First Order Schemes Exploiting Structures for Various Classes of Problems

Outline

- Gradient/Subgradient: Some Basic Algorithms and Results
- Fast Gradient-Based Schemes with Improved Convergence Rate:
- Nonconvex Models with Nice Structures

Talk based on joint works with:

A. Auslender (Lyon), A.Beck (Technion), R. Luss (Tel Aviv)

First Order/Gradient Based Methods: Why?

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

First Order/Gradient Based Methods: Why?

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

- Use minimal information, e.g., (f, f')
- Often lead to very simple and "cheap" iterative schemes.
- Complexity/iteration mildly dependent (e.g., linear) in problem's dimension, (as opposed to more sophisticated methods)
- Suitable when high accuracy is not crucial [in many large scale applications, the data is anyway corrupted or known only roughly..]

First Order/Gradient Based Methods: Why?

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

- Use minimal information, e.g., (f, f')
- Often lead to very simple and "cheap" iterative schemes.
- Complexity/iteration mildly dependent (e.g., linear) in problem's dimension, (as opposed to more sophisticated methods)
- Suitable when high accuracy is not crucial [in many large scale applications, the data is anyway corrupted or known only roughly..]

For very large scale problems with medium accuracy requirements, gradient based methods often remain the only practical alternative.... Widely used in many applications....

1. **Clustering Analysis:** *The k-means algorithm*
2. **Neuro-computing:** *The backpropagation algorithm*
3. **Statistical Estimation:** *The EM (Expectation-Maximization)* algorithm.
4. **Machine Learning:** *SVM, Regularized regression, PCA, etc...*
5. **Signal and Image Processing:** *Sparse Recovery, Denoising and Deblurring Schemes, Total Variation minimization...*
6. **...and much more...**

A Useful Optimization Model

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

- $\mathbb{E}$ is a finite dimensional Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \| = \langle \cdot, \cdot \rangle^{1/2}$.
- $g : \mathbb{E} \to (-\infty, \infty]$ is proper closed and convex, assumed subdifferentiable over $\operatorname{dom} g$ assumed closed.
- $f : \mathbb{E} \to \mathbb{R}$ is $C^{1,1}_{L(f)}$ over $\mathbb{E}$, i.e., with gradient Lipschitz:

$$\exists L(f) > 0 : \ \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L(f)\|\mathbf{x} - \mathbf{y}\|, \ \forall \mathbf{x}, \mathbf{y}.$$

- We assume that (M) is solvable, i.e.,

$$X_* := \operatorname{argmin} f \neq \emptyset, \text{ and for } \mathbf{x}^* \in X_*, \text{ set } F_* := F(\mathbf{x}^*).$$

The model (M) does already have *structural information*. It is rich enough to recover various classes of smooth/nonsmooth convex and nonconvex minimization problems.

Gradient-Based Schemes for Special Cases of (M)

Specializing model (M): $\min_{\mathbf{x}} F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})$ with $f = 0$ or $g = 0, \delta_C$

The Gradient Method $\min_{\mathbf{x}} f(\mathbf{x})$	:	$\mathbf{x}^k = \mathbf{x}^{k-1} - t_k \nabla f(\mathbf{x}^{k-1})$
The Gradient Projection $\min_{\mathbf{x}\in C} f(\mathbf{x})$	:	$\mathbf{x}^k = \Pi_C(\mathbf{x}^{k-1} - t_k \nabla f(\mathbf{x}^{k-1}))$
Subgradient Projection $\min_{\mathbf{x}\in C} g(\mathbf{x})$	:	$\mathbf{x}^k = \Pi_C(\mathbf{x}^{k-1} - t_k \gamma^{k-1}),\ \gamma^{k-1} \in \partial g(\mathbf{x}^{k-1})$
Proximal Minimization $\min_{\mathbf{x}} g(\mathbf{x})$	:	$\mathbf{x}_k = \operatorname{argmin}_{\mathbf{x}}\{g(x) + \frac{1}{2t_k}\lVert \mathbf{x} - \mathbf{x}^{k-1}\rVert^2\}$

- $t_k > 0$ is a suitable stepsize: fixed; backtracking line search; exact line search; or diminishing step-size: $t_k \to 0, \sum t_k = \infty$
- $\Pi_C(\mathbf{x}) := \operatorname{argmin}_{\mathbf{z}\in C} \lVert \mathbf{z} - \mathbf{x}\rVert^2$. is the orthogonal projection onto $C \subset \mathbb{E}$
- $\delta_C(\cdot)$ is the indicator for C

Some Typical Rate of Convergence for Gradient Schemes

Our focus is on *non-asymptotic global rate* of convergence.

1. Convex Smooth Minimization: Gradient/Gradient Projection (GP)

$$f(\mathbf{x}^k) - f(\mathbf{x}^*) = O(1/k)$$

2. Convex Nonsmooth Minimization: Subgradient Method (SM)

$$\min_{1 \le s \le k} g(\mathbf{x}_s) - g_* = O(\frac{1}{\sqrt{k}})$$

3. Nonconvex Smooth Mininization: Gradient/Gradient Projection

$$\min_{1 \le s \le k} \|\nabla f(\mathbf{x}_{s-1})\| = O(\frac{1}{\sqrt{k}})$$

- **Key Advantages:** rate nearly *independent* of problem's dimension. GP Simple, when projections are easy to compute...
- **Main Drawbacks:** GP often too slow even for low accuracy requirements...For SM, worse... needs $k \ge \epsilon^{-2}$ iterations!
- Can we improve the situation..?...

Building Gradient-Based Schemes

Our objective is to solve

$$(\mathrm{M}) \quad \min\left\{F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}) : \mathbf{x} \in \mathbb{E}\right\},\ f \text{ smooth},\ g \text{ nonsmooth}$$

Building Gradient-Based Schemes

Our objective is to solve

$$\text{(M)} \quad \min \{F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}) : \mathbf{x} \in \mathbb{E}\}, \ f \text{ smooth}, \ g \text{ nonsmooth}$$

Useful and Basic Approaches Include:

- Discretization of dynamical systems
- Local Approximation models for(M)
- Fixed point methods on corresponding optimality conditions

Less Standard: Deriving schemes for optimization via VI algorithms

Building Gradient-Based Schemes

Our objective is to solve

$$\text{(M)} \quad \min\left\{F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}) : \mathbf{x} \in \mathbb{E}\right\},\ f \text{ smooth},\ g \text{ nonsmooth}$$

Useful and Basic Approaches Include:

- Discretization of dynamical systems
- Local Approximation models for(M)
- Fixed point methods on corresponding optimality conditions

Less Standard: Deriving schemes for optimization via VI algorithms

A Key Player: The Proximal Framework

Quadratic Approximation and Fixed Point Approaches

Two key ideas from: [Fukushima-Mine'81] and [Passty'79]

Quadratic Approximation and Fixed Point Approaches

Two key ideas from: [Fukushima-Mine'81] and [Passty'79]

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

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

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

Quadratic Approximation and Fixed Point Approaches

Two key ideas from: [Fukushima-Mine'81] and [Passty'79]

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

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

That is, **leaving the nonsmooth part $g(\cdot)$ untouched**.
Then, solve the approximate model: $\mathbf{x}_k = \underset{\mathbf{x}}{\operatorname{argmin}}\, q(\mathbf{x}, \mathbf{x}_{k-1})$

Quadratic Approximation and Fixed Point Approaches

Two key ideas from: [Fukushima-Mine'81] and [Passty'79]

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

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

   That is, **leaving the nonsmooth part $g(\cdot)$ untouched**.
   Then, solve the approximate model: $\mathbf{x}_k = \operatorname{argmin}_{\mathbf{x}} q(\mathbf{x}, \mathbf{x}_{k-1})$

2. **Fixed Point via the optimality condition (Convex case):**
   $\mathbf{x}^* \in \operatorname{argmin}\{f(\mathbf{x}) + g(\mathbf{x})\}$ iff $\mathbf{0} \in \nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*)$.

Quadratic Approximation and Fixed Point Approaches

Two key ideas from: [Fukushima-Mine'81] and [Passty'79]

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

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

   That is, **leaving the nonsmooth part $g(\cdot)$ untouched**.
   Then, solve the approximate model: $\mathbf{x}_k = \underset{\mathbf{x}}{\operatorname{argmin}}\, q(\mathbf{x}, \mathbf{x}_{k-1})$

2. **Fixed Point via the optimality condition (Convex case):**
   $\mathbf{x}^* \in \operatorname{argmin}\{f(\mathbf{x}) + g(\mathbf{x})\}$ iff $\mathbf{0} \in \nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*)$. Fix any $t > 0$, then the following equivalent statements hold:

$$\Leftrightarrow \quad \mathbf{0} \in t\nabla f(\mathbf{x}^*) - \mathbf{x}^* \quad + \quad \mathbf{x}^* + t\partial g(\mathbf{x}^*) \quad \Leftrightarrow$$
$$(I + t\partial g)(\mathbf{x}^*) \in (I - t\nabla f)(\mathbf{x}^*) \quad \Leftrightarrow \quad \mathbf{x}^* \in (I + t\partial g)^{-1}(I - t\nabla f)(\mathbf{x}^*),$$

Quadratic Approximation and Fixed Point Approaches

Two key ideas from: [Fukushima-Mine'81] and [Passty'79]

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

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

   That is, **leaving the nonsmooth part $g(\cdot)$ untouched**.
   Then, solve the approximate model: $\mathbf{x}_k = \operatorname{argmin}_{\mathbf{x}} q(\mathbf{x}, \mathbf{x}_{k-1})$

2. **Fixed Point via the optimality condition (Convex case):**
   $\mathbf{x}^* \in \operatorname{argmin}\{f(\mathbf{x}) + g(\mathbf{x})\}$ iff $\mathbf{0} \in \nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*)$. Fix any $t > 0$, then the following equivalent statements hold:

$$\Leftrightarrow \quad \mathbf{0} \in t\nabla f(\mathbf{x}^*) - \mathbf{x}^* \quad + \quad \mathbf{x}^* + t\partial g(\mathbf{x}^*) \quad \Leftrightarrow$$
$$(I + t\partial g)(\mathbf{x}^*) \in (I - t\nabla f)(\mathbf{x}^*) \quad \Leftrightarrow \quad \mathbf{x}^* \in (I + t\partial g)^{-1}(I - t\nabla f)(\mathbf{x}^*),$$

   Through both approaches we obtain the **Proximal-Gradient Scheme**:

$$\mathbf{x}_k = \operatorname{argmin}_{\mathbf{x}} q(\mathbf{x}, \mathbf{x}_{k-1}) = \operatorname{argmin}_{\mathbf{x} \in \mathbb{E}} \left\{ g(\mathbf{x}) + \frac{1}{2t_k} \|\mathbf{x} - (\mathbf{x}_{k-1} - t_k \nabla f(\mathbf{x}_{k-1}))\|^2 \right\}$$
$$\mathbf{x}_k = (I + t_k \partial g)^{-1}(I - t_k \nabla f)(\mathbf{x}_{k-1}) := \operatorname{prox}_{t_k}(g)(I - t_k \nabla f)(\mathbf{x}_{k-1})$$

Thus, the scheme is *a proximal step at a gradient iteration* for f and reveals the fundamental role of the **proximal operator**.

The Proximal Map (Moreau - (1964))

Theorem [Moreau-(64)] Let $g : \mathbb{E} \to (-\infty, \infty]$ be closed proper convex. For any $t > 0$, let

$$g_t(\mathbf{z}) = \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{1}{2t} \|\mathbf{u} - \mathbf{z}\|^2 \right\} \qquad (*)$$

The Proximal Map (Moreau - (1964))

Theorem [Moreau-(64)] Let $g : \mathbb{E} \to (-\infty, \infty]$ be closed proper convex. For any $t > 0$, let

$$g_t(\mathbf{z}) = \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{1}{2t} \|\mathbf{u} - \mathbf{z}\|^2 \right\} \qquad (*)$$

1. $\min\{g_t(\mathbf{z}) : \ z \in \mathbb{E}\} = \min\{g(\mathbf{u}) : \ u \in \mathbb{E}\}$.
2. The minimum in (*) is attained at the *unique* point

$$\text{prox}_t(g)(\mathbf{z}) = (I + t\partial g)^{-1}(\mathbf{z}) \text{ for every } \mathbf{z} \in \mathbb{E},$$

   and the map $(I + t\partial g)^{-1}$ is single valued from $\mathbb{E}$ into itself.
3. The function $g_t(\cdot)$ is $C^{1,1}$ convex on $\mathbb{E}$ with a $\frac{1}{t}$-Lipschitz gradient:

$$\nabla g_t(\mathbf{z}) = \frac{1}{t}(I - \text{prox}_t(g)(\mathbf{z})) \text{ for every } \mathbf{z} \in \mathbb{E}.$$

The Proximal Gradient Method for (M)

The proximal gradient method with a constant stepsize rule.

Proximal Gradient Method with Constant Stepsize
Input: $L = L(f)$ - A Lipschitz constant of ∇f.
Step 0. Take $\mathbf{x}_0 \in \mathbb{E}$.
Step k. $(k \geq 1)$ Compute the prox of g

$$\mathbf{x}_k = p_L(\mathbf{x}_{k-1}) = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{E}} \left\{ g(\mathbf{x}) + \frac{L}{2}\left\|\mathbf{x} - (\mathbf{x}_{k-1} - \frac{1}{L}\nabla f(\mathbf{x}_{k-1}))\right\|^2 \right\}$$

- The Lipschitz constant $L(f)$ is not always known or not easily computable, this issue is resolved with an easy backtracking stepsize rule.
- **A drawback:** need to know how to compute efficiently $\text{prox}_t(g)(\cdot)$
- **What is the Global Rate of Convergence for PGM?**

Computing $\text{prox}_t(g)$: A Useful Example

- **Computing $\text{prox}_t(g)$ can be very hard..If at all possible..!.?..**
- But, for many useful special cases can be easy...

Computing $\text{prox}_t(g)$: A Useful Example

- **Computing $\text{prox}_t(g)$ can be very hard..If at all possible..!.?..**
- But, for many useful special cases can be easy...
- If $g \equiv \delta_C$, the indicator of C closed and convex, then

$$\begin{aligned}\text{prox}_t(g)(\mathbf{x}) &= \underset{\mathbf{u}}{\text{argmin}}\{\delta_C(\mathbf{u}) + \frac{1}{2t}\|\mathbf{u}-\mathbf{x}\|^2\} = \text{argmin}\{\frac{1}{2t}\|\mathbf{u}-\mathbf{x}\|^2 : \ \mathbf{u} \in C\} \\ &= (I + t\partial g)^{-1}(\mathbf{x}) = \Pi_C(\mathbf{x}), \text{ the ortho projection on } C\end{aligned}$$

Computing $\text{prox}_t(g)$: A Useful Example

- **Computing $\text{prox}_t(g)$ can be very hard..If at all possible..!.?..**
- But, for many useful special cases can be easy...
- If $g \equiv \delta_C$, the indicator of C closed and convex, then

$$\begin{aligned}\text{prox}_t(g)(\mathbf{x}) &= \underset{\mathbf{u}}{\text{argmin}}\{\delta_C(\mathbf{u}) + \frac{1}{2t}\|\mathbf{u}-\mathbf{x}\|^2\} = \text{argmin}\{\frac{1}{2t}\|\mathbf{u}-\mathbf{x}\|^2 : \ \mathbf{u} \in C\} \\ &= (I + t\partial g)^{-1}(\mathbf{x}) = \Pi_C(\mathbf{x}), \text{ the ortho projection on } C\end{aligned}$$

For some useful sets C easy to compute Π_C:

- Affine sets, Simple Polyhedral Sets (halfspace, $\mathbb{R}^n_+$, $[l,u]^n$),
- l_2, l_1, l_∞ - Balls,
- Ice Cream Cone, Semidefinite Cone S^n_+,
- Simplex and Spectrahedron (Simplex in S^n).

This covers many interesting models + equally easy for $g = \delta^*_C$ the support function of C. Some more useful examples....

Some Calculus Rules for Computing $\text{prox}_t(g)$

$$\text{prox}_t(g)(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{1}{2t}\|\mathbf{u}-\mathbf{x}\|^2 \right\}.$$

$g(\mathbf{u})$	$\text{prox}_t(g)(\mathbf{x})$
$\delta_C(\mathbf{u})$	$\Pi_C(\mathbf{x})$
$\delta_C^*(\mathbf{u})$ -support function-	$\mathbf{x} - \Pi_C(\mathbf{x})$
$d_C(\mathbf{u})$	$\begin{cases} \mathbf{x} + \frac{(\Pi_C(\mathbf{x})-\mathbf{x})}{t d_C(\mathbf{x})} & \text{if } d_C(\mathbf{x}) > 1/t \\ \mathbf{x} & \text{otherwise} \end{cases}$
$\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2/2, \mathbf{A} \in \mathbb{R}^{m\times n}$	$(I + t^{-1}\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{x} + t^{-1}\mathbf{A}^T\mathbf{b})$
$\|\mathbf{u}\|_1$	(-shrinkage-) $\text{sgn}\,(x_j)\max\{\lvert x_j\rvert - t, 0\}$
$\|\mathbf{u}\|$	$\begin{cases} \|\mathbf{x}\|^2/2t & \text{if } \|\mathbf{x}\| \le t \\ \|\mathbf{x}\| - t/2 & \text{otherwise} \end{cases}$
$\|\mathbf{U}\|_*,\ \mathbf{U} \in \mathbb{R}^{m\times n},\ (m \ge n)$	$\mathbf{P}\,\text{diag}(\mathbf{s})\mathbf{Q}^T$

- $\sigma_1(\mathbf{U}) \ge \sigma_2(\mathbf{U}) \ge \ldots$ singular values of $\mathbf{U}$
- Nuclear norm $\|\mathbf{U}\|_* = \sum_j \sigma_j(\mathbf{U})$
- Singular value decomposition

$$\mathbf{U} = \mathbf{P}\,\text{diag}(\sigma)\mathbf{Q}^T, \text{ then shrinkage } s_j = \text{sgn}\,(\sigma_j)\max\{|\sigma_j| - t, 0\}.$$

Rate of Convergence of Prox-Grad for Convex (M)

Theorem - Rate of Convergence of Prox-Grad Let $\{\mathbf{x}_k\}$ be the sequence generated by the prox-grad. Then for every $k \geq 1$:

$$F(\mathbf{x}_k) - F(\mathbf{x}) \leq \frac{\alpha L(f)\|\mathbf{x}_0 - \mathbf{x}\|^2}{2k}, \ \forall x \in X_*$$

- Thus the prox grad method converges at a *sublinear rate* in function values, namely like there were **no nonsmooth term**.

Rate of Convergence of Prox-Grad for Convex (M)

Theorem - Rate of Convergence of Prox-Grad Let $\{\mathbf{x}_k\}$ be the sequence generated by the prox-grad. Then for every $k \geq 1$:

$$F(\mathbf{x}_k) - F(\mathbf{x}) \leq \frac{\alpha L(f)\|\mathbf{x}_0 - \mathbf{x}\|^2}{2k}, \ \forall x \in X_*$$

- Thus the prox grad method converges at a *sublinear rate* in function values, namely like there were **no nonsmooth term**.
- Special Cases: With $g \equiv 0$ and $g = \delta_C$, our model (M) recovers results for the basic gradient and gradient projection methods respectively.
- With $f = 0$ in (M), recovers the *Proximal Minimization Algorithm* (Martinet 70) and its sublinear complexity rate (Guler 90).

Rate of Convergence of Prox-Grad for Convex (M)

Theorem - Rate of Convergence of Prox-Grad Let $\{\mathbf{x}_k\}$ be the sequence generated by the prox-grad. Then for every $k \geq 1$:

$$F(\mathbf{x}_k) - F(\mathbf{x}) \leq \frac{\alpha L(f) \|\mathbf{x}_0 - \mathbf{x}\|^2}{2k}, \ \forall x \in X_*$$

- Thus the prox grad method converges at a *sublinear rate* in function values, namely like there were **no nonsmooth term**.
- Special Cases: With $g \equiv 0$ and $g = \delta_C$, our model (M) recovers results for the basic gradient and gradient projection methods respectively.
- With $f = 0$ in (M), recovers the *Proximal Minimization Algorithm* (Martinet 70) and its sublinear complexity rate (Guler 90).
- This is "Better" than Subgrad Scheme...But in general non-implementable, unless g is **"simple"**.... Nevertheless, very useful when combined with duality: $\longrightarrow$ **Augmented Lagrangians Methods**
- **Note:** The sequence $\{\mathbf{x}_k\}$ can also be proven to *converge* to global solution $\mathbf{x}^*$ provided a step size is in $(0, 2/L)$ (Combettes-Wajs (05)).

The Nonconvex Case in (M): F=f+g

When f is nonconvex, the global convergence rate results are of course weaker:

- Convergence to a global minimum is out of reach.
- Convergence of the sequence to a stationary point is measured by the quantity $\|\mathbf{x} - p_L(\mathbf{x})\|$. **No global results on $\{\mathbf{x}_k\}$ or even $\{F(\mathbf{x}_k)\}$.!..**

The Nonconvex Case in (M): F=f+g

When f is nonconvex, the global convergence rate results are of course weaker:

- Convergence to a global minimum is out of reach.
- Convergence of the sequence to a stationary point is measured by the quantity $\|\mathbf{x} - p_L(\mathbf{x})\|$. **No global results on $\{\mathbf{x}_k\}$ or even $\{F(\mathbf{x}_k)\}$.!..**

Theorem (Global Rate of Convergence for γ_n)

Let $\{\mathbf{x}_k\}$ be the sequence generated by the proximal gradient method with either a constant or a backtracking stepsize rule. Then for every $n \geq 1$ we have

$$\gamma_n \leq \frac{1}{\sqrt{n}} \left(\frac{2(F(\mathbf{x}_0) - F_*)}{\beta L(f)} \right)^{1/2},$$

where

$$\gamma_n := \min_{1 \leq k \leq n} \|\mathbf{x}_{k-1} - p_{L_k}(\mathbf{x}_{k-1})\|.$$

Moreover, $\|\mathbf{x}_{k-1} - p_{L_k}(\mathbf{x}_{k-1})\| \to 0$ as $k \to \infty$.

Improving Complexity–Fast Gradient Schemes

Previous **explicit** methods are simple but are often too slow.

- For Prox-Grad and Gradient methods: a complexity rate of $O(1/k)$
- For Subgradient Methods: complexity rate of $O(1/\sqrt{k})$.

Improving Complexity–Fast Gradient Schemes

Previous **explicit** methods are simple but are often too slow.

- For Prox-Grad and Gradient methods: a complexity rate of $O(1/k)$
- For Subgradient Methods: complexity rate of $O(1/\sqrt{k})$.
- Can we do better to solve the convex nonsmooth problem (M)?

$$(M) \qquad \min\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) : \ \mathbf{x} \in \mathbb{E}\}.$$

- Can we devise a method with:
  ♠ the *same computational effort/simplicty as Prox-Grad* .
  ♠ a *Faster* global rate of convergence.

Yes we Can...

Yes we Can...

- **Answer: Yes**, through an "equally simple" scheme
  Let $Q_L(\mathbf{x},\mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{x}-\mathbf{y}, \nabla f(\mathbf{y})\rangle + \frac{1}{2L}\|\mathbf{x}-\mathbf{y}\|^2 + g(\mathbf{x}), L > 0$

  $$\clubsuit \;\; \mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} Q_L(\mathbf{x}, \mathbf{y}_k), \hookleftarrow \;\; \mathbf{y}_k \text{ instead of } \mathbf{x}_k$$

  The new point $\mathbf{y}_k$ will be smartly chosen and **easy** to compute.

Yes we Can...

- **Answer: Yes**, through an "equally simple" scheme
  Let $Q_L(\mathbf{x},\mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{x}-\mathbf{y}, \nabla f(\mathbf{y})\rangle + \frac{1}{2L}\|\mathbf{x}-\mathbf{y}\|^2 + g(\mathbf{x}), L > 0$

  $$\clubsuit \ \mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} Q_L(\mathbf{x},\mathbf{y}_k), \longleftrightarrow \ \mathbf{y}_k \text{ instead of } \mathbf{x}_k$$

  The new point $\mathbf{y}_k$ will be smartly chosen and **easy** to compute.

- **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)) with complexity $O(1/k^2)$

Yes we Can...

- **Answer: Yes**, through an "equally simple" scheme
  Let $Q_L(\mathbf{x},\mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{x}-\mathbf{y}, \nabla f(\mathbf{y})\rangle + \frac{1}{2L}\|\mathbf{x}-\mathbf{y}\|^2 + g(\mathbf{x}), L > 0$

  $$\clubsuit \quad \mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}}\, Q_L(\mathbf{x},\mathbf{y}_k), \hookleftarrow \ \mathbf{y}_k \text{ instead of } \mathbf{x}_k$$

  The new point $\mathbf{y}_k$ will be smartly chosen and **easy** to compute.
- **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)) with complexity $O(1/k^2)$
- But, here problem (M) is **nonsmooth**. Yet, we can also derive a fast algorithm for the general NSO problem (M), namely *"as if the nonsmooth part can be neutralized"*

* Y. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, (1983)

A Fast Prox-Grad Algorithm - FISTA [Beck-Teboulle' 09]

An equally simple algorithm as prox-grad. (Here $L(f)$ is known).

Here with constant stepsize
Input: $L = L(f)$ - A Lipschitz constant of ∇f.
Step 0. Take $\mathbf{y}_1 = \mathbf{x}_0 \in \mathbb{E}$, $t_1 = 1$.
Step k. $(k \geq 1)$ Compute

$$\mathbf{x}_k = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{E}} \left\{ g(\mathbf{x}) + \frac{L}{2}\|\mathbf{x} - (\mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k))\|^2 \right\}$$

$$\mathbf{x}_k \equiv p_L(\mathbf{y}_k), \quad \hookleftarrow \text{ main computation as Prox-Grad}$$

$$\bullet \quad t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2},$$

$$\bullet\bullet \quad \mathbf{y}_{k+1} = \mathbf{x}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}_k - \mathbf{x}_{k-1}).$$

Additional computation in (•) and (••) is clearly marginal.
Knowledge of $L(f)$ is not Necessary, can use BLS.

With $g = 0$, this is the smooth Fast Gradient of Nesterov (83);
With $t_k \equiv 1, \forall k$ we recover ProxGrag (PG).

An Improved $O(1/k^2)$ Global Rate of Convergence for (M)

Theorem – [B-T' 09] Let $\{\mathbf{x}_k\}$ be generated by FISTA. Then for any $k \geq 1$

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{2L(f)\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{(k+1)^2},$$

- # of iterations to reach $F(\tilde{\mathbf{x}}) - F_* \leq \varepsilon$ is $\sim O(1/\sqrt{\varepsilon})$.
- Clearly improves Prox Grad by **a square root factor**.

An Improved $O(1/k^2)$ Global Rate of Convergence for (M)

Theorem – [B-T' 09] Let $\{\mathbf{x}_k\}$ be generated by FISTA. Then for any $k \geq 1$

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{2L(f)\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{(k+1)^2},$$

- $\#$ of iterations to reach $F(\tilde{\mathbf{x}}) - F_* \leq \varepsilon$ is $\sim O(1/\sqrt{\varepsilon})$.
- Clearly improves Prox Grad by **a square root factor**.
- On the practical side this theoretical rate is achieved.
- Many computational studies have confirmed the efficiency of FISTA for solving several interesting models in *Signal/image recovery* and in *Machine learning*

  e.g., image denoising/deblurring, nuclear matrix norm regularization, matrix completion problems, multi-task learning, matrix classification, etc..

Applications/Limitations of FISTA for (M)

$$(M) \min\{f(\mathbf{x}) + g(\mathbf{x}) : \mathbf{x} \in \mathbb{E}\}$$

$f \in C^{1,1}$ convex can be of any type with available gradient

- **FISTA is not a monotone method!..** But can be made monotone.
- As long as the **prox** of the nonsmooth function g

$$p_L(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{E}} \left\{ g(\mathbf{x}) + \frac{L}{2}\|\mathbf{x} - (\mathbf{y} - \frac{1}{L}\nabla f(\mathbf{y}))\|^2 \right\}$$

can be computed analytically or easily/efficiently, via some other approach (e.g., dual for TV); FISTA (MFISTA) is useful and quite efficient.

Applications/Limitations of FISTA for (M)

$$(M) \min\{f(\mathbf{x}) + g(\mathbf{x}) : \mathbf{x} \in \mathbb{E}\}$$

$f \in C^{1,1}$ convex can be of any type with available gradient

- **FISTA is not a monotone method!..** But can be made monotone.
- As long as the **prox** of the nonsmooth function g

$$p_L(\mathbf{y}) = \underset{\mathbf{x}\in\mathbb{E}}{\operatorname{argmin}} \left\{ g(\mathbf{x}) + \frac{L}{2}\left\|\mathbf{x} - (\mathbf{y} - \frac{1}{L}\nabla f(\mathbf{y}))\right\|^2 \right\}$$

  can be computed analytically or easily/efficiently, via some other approach (e.g., dual for TV); FISTA (MFISTA) is useful and quite efficient.
- **Caveat:** Many inverse problems solve the Penalized Model:

$$\min\{f(\mathbf{x}) + \lambda g(\mathbf{x})\}; \ \lambda > 0 \text{ tradeoff -}\textbf{unknown penalty parameter}$$

  FISTA **does not** resolve the issue on how to pick the unknown λ! Continuation, or heuristic techniques can be used.

  Many other algorithms suffer the same problem with the unknown parameter and require "tuning".

Gradient Schemes with Non-Euclidean Distances

- All previous schemes were based on using the squared Euclidean distance for measuring proximity of two points in $\mathbb{E}$
- It is useful to exploit the *geometry of the constraints set* X
- This is done by selecting a "distance-like" function that sometimes can reduce computational costs and even improve the rate of convergence.

Gradient Schemes with Non-Euclidean Distances

- All previous schemes were based on using the squared Euclidean distance for measuring proximity of two points in $\mathbb{E}$
- It is useful to exploit the *geometry of the constraints set* X
- This is done by selecting a "distance-like" function that sometimes can reduce computational costs and even improve the rate of convergence.

1. Mirror Descent Algorithms
2. More on Fast Gradient Schemes
3. Building Gradient Schemes via Algorithms for Variational Inequalities

A Proximal Distance-Like Function

Exploiting the Geometry of the constraints

- Usual gradient method reads:

$$y = \operatorname*{argmin}_{\boldsymbol{\xi} \in X} \{ t\langle \boldsymbol{\xi}, \nabla f(\mathbf{x}) \rangle + \frac{1}{2} \|\boldsymbol{\xi} - \mathbf{x}\|^2 \}, \ t > 0.$$

A Proximal Distance-Like Function

Exploiting the Geometry of the constraints

- Usual gradient method reads:

$$y = \operatorname*{argmin}_{\boldsymbol{\xi} \in X} \{ t \langle \boldsymbol{\xi}, \nabla f(\mathbf{x}) \rangle + \frac{1}{2} \|\boldsymbol{\xi} - \mathbf{x}\|^2 \}, \; t > 0.$$

- Replace $\| \cdot \|^2$ by some **distance-like** $d(\cdot, \cdot)$ that better exploits C (e.g., allows for deriving ***explicit and simple*** formula) through a **Projection-Like Map:**

$$p(\mathbf{g}, \mathbf{x}) := \operatorname*{argmin}_{\mathbf{v}} \{ \langle \mathbf{v}, \mathbf{g} \rangle + d(\mathbf{v}, \mathbf{x}) \}.$$

A Proximal Distance-Like Function

Exploiting the Geometry of the constraints

- Usual gradient method reads:

$$y = \operatorname*{argmin}_{\boldsymbol{\xi} \in X} \{ t \langle \boldsymbol{\xi}, \nabla f(\mathbf{x}) \rangle + \frac{1}{2} \|\boldsymbol{\xi} - \mathbf{x}\|^2 \}, \; t > 0.$$

- Replace $\| \cdot \|^2$ by some **distance-like** $d(\cdot, \cdot)$ that better exploits C (e.g., allows for deriving ***explicit and simple*** formula) through a **Projection-Like Map:**

$$p(\mathbf{g}, \mathbf{x}) := \operatorname*{argmin}_{\mathbf{v}} \{ \langle \mathbf{v}, \mathbf{g} \rangle + d(\mathbf{v}, \mathbf{x}) \}.$$

- **Minimal required properties for d:**

  $d(\cdot, \mathbf{v})$ is a convex function, $\forall \mathbf{v}$
  $d(\cdot, \cdot) \geq 0$, and $d(\mathbf{u}, \mathbf{v}) = 0$ iff $\mathbf{u} = \mathbf{v}$ $\forall \mathbf{u}, \mathbf{v}$.
  • d **is not a distance**: no symmetry or/and triangle inequality

Two Generic Families for Proximal Distances d

- Bregman type distances - based on kernel ψ:

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

- Φ-divergence type distances - based on 1-d kernel ϕ convex

$$d_\varphi(\mathbf{x},\mathbf{y}) := \sum_{j=1}^{n} y_j^r \varphi(\frac{x_j}{y_j}) + \frac{\sigma}{2}\|\mathbf{x}-\mathbf{y}\|^2, \ r = 1,2$$

The choice of d should be dictated to

♠ best match the constraints of a given problem
♠ simplify the projection-like computation for given class of "Simple Constraints with Special Structures"
♣ **What are Simple Constraints...?..**

Simple Constraints

"Simple" but also fundamental.. $X := \overline{C} \cap V$, $\overline{C}$ closure of C with

$$C \text{ open convex}, V := \{\mathbf{x} \in \mathbb{R}^n : \mathcal{A}(\mathbf{x}) = \mathbf{b}\}, \ \mathcal{A} \text{ linear}, \ \mathbf{b} \in \mathbb{R}^m.$$

- $\mathbb{R}^n_+$,
- unit ball, box constraints,
- Δ_n the simplex in $\mathbb{R}^n$,
- S^n_+ (symmetric semidefinite positive matrices),
- L^n_+ the Lorentz cone,
- the Spectrahedron (Simplex in S^n)

Examples of couple (d, H)

$C \cap \mathcal{V}$	$d(\mathbf{x}, \mathbf{y})$	$H(\mathbf{x}, \mathbf{y})$
$\mathbb{R}^n_{++}$	$\sum_{j=1}^n -y_j^2 \log \frac{x_j}{y_j} + x_j y_j - y_j^2 + \frac{\sigma}{2}\|\mathbf{x}-\mathbf{y}\|^2$	$\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2$
S^n_{++}	$-\log\det(\mathbf{x}\mathbf{y}^{-1}) + \mathrm{tr}(\mathbf{x}\mathbf{y}^{-1}) + \sigma\,\mathrm{tr}(\mathbf{x}-\mathbf{y})^2 - n$	$H = d$
L^n_{++}	$-\log \frac{\mathbf{x}^T D_n \mathbf{x}}{\mathbf{y}^T D_n \mathbf{y}} + \frac{2\mathbf{x}^T D_n \mathbf{y}}{\mathbf{y}^T D_n \mathbf{y}} - 2 + \frac{\sigma}{2}\|\mathbf{x}-\mathbf{y}\|^2$	$H = d$
Δ_n	$\sum_{j=1}^n x_j \log \frac{x_j}{y_j} + y_j - x_j$	$H = d$
Σ_n	$\mathrm{tr}(\mathbf{x}\log\mathbf{x} - \mathbf{x}\log\mathbf{y} + \mathbf{y} - \mathbf{x})$	$H = d$

$$\Delta_n := \{\mathbf{x} \in \mathbb{R}^n \mid \sum_{j=1}^n x_j = 1, x > 0\},\ \Sigma_n := \{\mathbf{x} \in S_n \mid \mathrm{tr}(x) = 1, \mathbf{x} \succ 0\}.$$

$$L^n_{++} := \{\mathbf{x} \in \mathbb{R}^n \mid x_n > (x_1^2 + \ldots + x_{n-1}^2)^{1/2}\},\ D_n \equiv \mathrm{diag}(-1, \ldots, -1, 1).$$

$C_n = \{\mathbf{x} \in \mathbb{R}^n :\ a_j < x_j < b_j \quad j = 1 \ldots n\}$ similar to $\mathbb{R}^n_{++}$ (log quad)

Corresponding Projections $p(\mathbf{g}, \mathbf{x})$ can be obtained analytically in these cases

Note: $H(\cdot, \cdot)$ is another proximity measure used to prove convergence results

Computing Explicit Projections $p(\mathbf{g}, \mathbf{x})$

$C \cap \mathcal{V}$	$p(\mathbf{g}, \mathbf{x})$ or $p_j(\mathbf{g}, \mathbf{x}), j = 1, \ldots, n$
$\mathbb{R}^n_{++}$	$x_j(\varphi^*)'(-g_j x_j^{-1})$
S^n_{++}	$(2\sigma)^{-1}(A(\mathbf{g}, \mathbf{x}) + \sqrt{A(\mathbf{g}, \mathbf{x})^2 + 4\sigma I})$
L^n_{++}	$\frac{1}{2\sigma}\left((1 + \frac{w_n}{\zeta})\bar{w}, (w_n + \zeta)\right)$
Δ_n	$\frac{x_j \exp(-g_j)}{\sum_{i=1}^n x_i \exp(-g_i)}$
Σ_n	via eigenvalue decomp. reduces to similar comp. as Δ_n

$$
\begin{aligned}
(\varphi^*)'(s) &= (2\sigma)^{-1}\{(\sigma - 1) + s + \sqrt{((\sigma - 1) + s)^2 + 4\sigma}\} \\
A(\mathbf{g}, \mathbf{x}) &= \sigma \mathbf{x} - \mathbf{g} - \mathbf{x}^{-1}, \tau(\mathbf{x}) = \mathbf{x}^T D_n \mathbf{x} \\
w &= (-2\tau(\mathbf{x})^{-1} D_n \mathbf{x} + 2\sigma \mathbf{x} - \mathbf{g})/2, \ \mathbf{w} = (\bar{\mathbf{w}}, w_n) \in \mathbb{R}^{n-1} \times \mathbb{R} \\
\zeta &= \left(\frac{\|\mathbf{w}\|^2 + 4\sigma + \sqrt{(\|\mathbf{w}\|^2 + 4\sigma)^2 - 4w_n^2\|\bar{\mathbf{w}}\|^2}}{2}\right)^{1/2}.
\end{aligned}
$$

1. The Mirror Descent Algorithm-MDA

$$\min\{g(\mathbf{x}) : \mathbf{x} \in C\} \quad \text{Convex Nonsmooth}$$

- Originated from functional analytic arguments in infinite dimensional setting between primal-dual spaces.
  A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization* Wiley-Interscience Publication, (1983).

1. The Mirror Descent Algorithm-MDA

$$\min\{g(\mathbf{x}) : \mathbf{x} \in C\} \quad \text{Convex Nonsmooth}$$

- Originated from functional analytic arguments in infinite dimensional setting between primal-dual spaces.
  A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization* Wiley-Interscience Publication, (1983).
- In (Beck-Teboulle-2003) we have shown that the (MDA) can be simply viewed as a **subgradient method** with a strongly convex Bregman proximal distance:

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\text{argmin}}\{\langle \mathbf{x}, \mathbf{v}_k \rangle + \frac{1}{t_k} D_\psi(\mathbf{x}, \mathbf{x}_k)\}, \; \mathbf{v}_k \in \partial g(\mathbf{x}_k), \; t_k > 0.$$

1. The Mirror Descent Algorithm-MDA

$$\min\{g(\mathbf{x}) : \mathbf{x} \in C\} \quad \text{Convex Nonsmooth}$$

- Originated from functional analytic arguments in infinite dimensional setting between primal-dual spaces.
  A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization* Wiley-Interscience Publication, (1983).
- In (Beck-Teboulle-2003) we have shown that the (MDA) can be simply viewed as a **subgradient method** with a strongly convex Bregman proximal distance:

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}}\{\langle \mathbf{x}, \mathbf{v}_k\rangle + \frac{1}{t_k} D_\psi(\mathbf{x}, \mathbf{x}_k)\},\ \mathbf{v}_k \in \partial g(\mathbf{x}_k),\ t_k > 0.$$

- **Exploiting *geometry* of constraints can improve performance of SM.**
- **Example: Convex Minimization over the Unit Simplex** Δ_n that uses the *entropy kernel* defined on Δ_n (is 1-strongly convex w.r.t $\|\cdot\|_1$).

Convex Minimization over the Unit Simplex Δ_n

$$\inf\{g(\mathbf{x}) : \ x \in \Delta_n\}, \ \Delta_n = \{\mathbf{x} \in \mathbb{R}^n : \ e^T\mathbf{x} = 1, \mathbf{x} \geq 0\}$$

- **EMDA:** Start with $\mathbf{x}^0 = n^{-1}e$. For $k \geq 1$ generate

$$x_j^k = \frac{x_j^{k-1}\exp(-t_k v_j^{k-1})}{\sum_{i=1}^n x_i^{k-1}\exp(-t_k v_i^{k-1})}, \ j = 1, \ldots, n \ t_k := \frac{\sqrt{2\log n}}{L_g\sqrt{k}},$$

where $\mathbf{v}^{k-1} := (v_1^{k-1}, \ldots, v_n^{k-1}) \in \partial g(\mathbf{x}_{k-1})$.

Theorem The sequence generated by EMDA satisfies for all $k \geq 1$

$$\min_{1 \leq s \leq k} g(\mathbf{x}^s) - \min_{\mathbf{x} \in \Delta} g(\mathbf{x}) \leq \sqrt{2\log n}\frac{\max_{1 \leq s \leq k} ||\mathbf{v}^s||_\infty}{\sqrt{k}}$$

This outperforms the classical subgradient (based on $\|\cdot\|^2$), by a factor of $(n/\log n)^{1/2}$, which for large n can make a huge difference!....

2. A Fast Non-Euclidean Gradient Method

For the nonsmooth convex case $\min\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\},\ f \in C^{1,1}$.
Easily obtained by extending the smooth case of [Auslender-Teboulle'06)] along the proof techniques of Beck-Teboulle'09 for FISTA.

A Fast Non-Euclidean Gradient Method with Bregman Distance D_ψ

Input: $L = L(f), \sigma > 0, \psi, \sigma$-strongly convex.

Step 0: $\mathbf{x}_0, \mathbf{z}_0 \in \text{ri}(\text{dom}\,\psi), t_0 = 1$

$$
\begin{aligned}
\textbf{Step k: } \mathbf{y}_k &= (1 - t_k^{-1})\mathbf{x}_k + t_k^{-1}\mathbf{z}_k \leftarrow \\
\mathbf{z}_{k+1} &= \operatorname*{argmin}_{\mathbf{x}} \left\{ \langle \mathbf{x}, \nabla f(\mathbf{y}_k) \rangle + g(\mathbf{x}) + \frac{L}{\sigma t_k} D_\psi(\mathbf{x}, \mathbf{z}_k) \right\}, \\
\mathbf{x}_{k+1} &= (1 - t_k^{-1})\mathbf{x}_k + t_k^{-1}\mathbf{z}_{k+1}, \\
t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2},
\end{aligned}
$$

As simple as FISTA, just requires the simple additional update $\mathbf{y}_k$.

Complexity of Non-Euclidean Fast Gradient

Theorem For the sequence $\{\mathbf{x}_k\}$ generated by the previous algorithm:

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{4LD_\psi(\mathbf{x}^*, \mathbf{x}_0)}{\sigma(k+1)^2}, \ \forall k \geq 1.$$

Thus, we have an $O(1/k^2)$ scheme for Non-Euclidean Distance to solve (M).

Moreover, as in Mirror Descent, the advantage of using Non Euclidean distance adequately exploiting the constraints allows to:

1. Simplify the prox computation for the given constraints set
2. Improve the constant in the complexity bound

Complexity of Non-Euclidean Fast Gradient

Theorem For the sequence $\{\mathbf{x}_k\}$ generated by the previous algorithm:

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{4LD_\psi(\mathbf{x}^*, \mathbf{x}_0)}{\sigma(k+1)^2}, \ \forall k \geq 1.$$

Thus, we have an $O(1/k^2)$ scheme for Non-Euclidean Distance to solve (M).

Moreover, as in Mirror Descent, the advantage of using Non Euclidean distance adequately exploiting the constraints allows to:

1. Simplify the prox computation for the given constraints set
2. Improve the constant in the complexity bound

Two other schemes :

- One requires past history of all gradients + 2 prox: one quadratic, and one based on ψ;
- the other also requires past history of all gradients, and 2 prox based on ψ.

  See, Nesterov. Smooth minimization of non-smooth functions. *Math. Program. Series A*, Vol. 103, 127–152, (2005); Gradient methods for minimizing composite objective function. CORE Technical report,(2007).

3. Gradient Schemes via Variational Inequalities

- $X \ \subset \mathbb{R}^n$ closed convex set
- $F : X \to \mathbb{R}^n$ monotone map on X, i.e.,

$$\langle F(\mathbf{x}) - F(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq 0, \ \forall \mathbf{x}, \mathbf{y} \in X.$$

VI Problem

Find $\mathbf{x}^* \in X$ such that $\langle F(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0 \quad \forall \mathbf{x} \in X$.

- VI extend and encompass a broad spectrum of problems: Complementarity, Optimization, Saddle point, Equilibrium...

3. Gradient Schemes via Variational Inequalities

- $X \subset \mathbb{R}^n$ closed convex set
- $F : X \to \mathbb{R}^n$ monotone map on X, i.e.,

$$\langle F(\mathbf{x}) - F(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq 0, \ \forall \mathbf{x}, \mathbf{y} \in X.$$

VI Problem

Find $\mathbf{x}^* \in X$ such that $\langle F(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0 \quad \forall \mathbf{x} \in X.$

- VI extend and encompass a broad spectrum of problems: Complementarity, Optimization, Saddle point, Equilibrium...
- Here, X is assumed ***"simple"*** for the VI.
- This is exploited to derive schemes **with explicit formulas** for general constrained smooth convex problems as well as some structured nonsmooth problems.

Starting Idea: The Extra-Gradient Method

Korpelevich, G. M. Extrapolation gradient methods and their relation to modified Lagrange functions. *Ekonom. i Mat. Metody*, **19** (1976), no. 4, 694–703.

- Provides a "simple cure" to difficulties, and strong assumptions needed in the usual *Projection methods for VI* (e.g., F strongly monotone on X)

$$\mathbf{x}^k = \Pi_X(\mathbf{x}^{k-1} - t_k F(\mathbf{x}^{k-1})), \ t_k > 0.$$

Starting Idea: The Extra-Gradient Method

Korpelevich, G. M. Extrapolation gradient methods and their relation to modified Lagrange functions. *Ekonom. i Mat. Metody*, **19** (1976), no. 4, 694–703.

- Provides a "simple cure" to difficulties, and strong assumptions needed in the usual *Projection methods for VI* (e.g., F strongly monotone on X)

$$\mathbf{x}^k = \Pi_X(\mathbf{x}^{k-1} - t_k F(\mathbf{x}^{k-1})),\ t_k > 0.$$

- **Extragradient Method-Korpelevich (76):**

$$\mathbf{y}^{k-1} = \Pi_X(\mathbf{x}^{k-1} - \beta_k F(\mathbf{x}^{k-1})), \quad \mathbf{x}^k = \Pi_X(\mathbf{x}^{k-1} - \alpha_k F(\mathbf{y}^{k-1})),$$

  with $\beta_k = \alpha_k = \frac{1}{L}$ (L is the Lipschtiz constant for F)

- **No complexity results.../or potential implications to solve NSO/constrained problems.**
- **Does not exploit the geometry of set X.**

Basic Model Algorithm is Very Simple

- Pick some suitable prox-distance $d(\cdot,\cdot)$ and let

$$p(\mathbf{g},\mathbf{x}) = \operatorname*{argmin}_{\mathbf{v}}\{\langle \mathbf{v},\mathbf{g}\rangle + d(\mathbf{v},\mathbf{x})\}.$$

- **Extra-Gradient-Like: EGL**
  Given $x^1 \in C \cap V$, compute:

$$\begin{aligned} \mathbf{y}^k &= p(\beta^k F(\mathbf{x}^k), \mathbf{x}^k) \\ \mathbf{x}^{k+1} &= p(\alpha^k F(\mathbf{y}^k), \mathbf{x}^k) \\ \mathbf{z}^k &= \sum_{l=1}^{k} \frac{\alpha^l \mathbf{y}^l}{\sum_{l=1}^k \alpha^l} \quad \leftarrow \text{average comp.} \end{aligned}$$

  with $\alpha^k,\ \beta^k > 0$ determined within algorithm, or fixed in terms of L.

- **Main Computational Object: The Projection-Like Map $p(\cdot,\cdot)$ with respect to the choice of $d(\cdot,\cdot)$.**

Convergence Results for EGL

Convergence Result (Auslender-Teboulle (05)

Let $\{\mathbf{x}^k, \mathbf{y}^k, \mathbf{z}^k\}$ the sequences generated by EGL. Then,

1. The sequences $\{\mathbf{x}^k\}$, $\{\mathbf{z}^k\}$ are bounded and each limit point of $\{\mathbf{z}^k\}$ is a solution of (VI).
2. If $H(\mathbf{x}, \mathbf{y}) = \frac{\sigma}{2}\|\mathbf{x} - \mathbf{y}\|^2$ for Φ-div. distance, then the **whole sequence** $\{\mathbf{x}^k\}$ converges to a solution of (VI).

Convergence Results for EGL

Convergence Result (Auslender-Teboulle (05)

Let $\{\mathbf{x}^k, \mathbf{y}^k, \mathbf{z}^k\}$ the sequences generated by EGL. Then,

1. The sequences $\{\mathbf{x}^k\}$, $\{\mathbf{z}^k\}$ are bounded and each limit point of $\{\mathbf{z}^k\}$ is a solution of (VI).
2. If $H(\mathbf{x}, \mathbf{y}) = \frac{\sigma}{2}\|\mathbf{x} - \mathbf{y}\|^2$ for Φ-div. distance, then the **whole sequence** $\{\mathbf{x}^k\}$ converges to a solution of (VI).
3. If F is L-Lipschitz on X, we have the complexity estimate

$$\theta(\mathbf{z}^k) = O(\frac{1}{k}),$$

- where $\theta(\mathbf{z}) = \sup\{\langle F(\boldsymbol{\xi}), z - \boldsymbol{\xi}\rangle : \boldsymbol{\xi} \in X\}$ is the gap function.

Related independent result (only with $d(\cdot, \cdot) \equiv$ Bregman and for rate of convergence), Nemirovsky (04).

Applying EGL to Convex Minimization

$$(P) \quad f_* = \inf\{f(\mathbf{x}) : \; -G(\mathbf{x}) \in K, \; \mathbf{A}\mathbf{x} = \mathbf{a}, \mathbf{x} \in S\}.$$

- $\mathbb{R}^n$, $\mathbb{R}^m$, and $\mathbb{R}^p$ finite dim. v.s. with inner products, $\langle \cdot, \cdot \rangle_{n,m,p}$
- f convex; $G : \mathbb{R}^n \to \mathbb{R}^p$, $K-$ convex; $\mathbf{a} \in \mathbb{R}^p$, $\mathbf{A} : \mathbb{R}^n \to \mathbb{R}^p$
- S **"simple"** closed convex
- K closed convex cone, int $K \neq \emptyset$; *e.g.*, $K = \mathbb{R}^m_+, S^m_+, L^m_+$

Applying EGL to Convex Minimization

$$(P) \quad f_* = \inf\{f(\mathbf{x}) : \; -G(\mathbf{x}) \in K, \; \mathbf{A}\mathbf{x} = \mathbf{a}, \mathbf{x} \in S\}.$$

- $\mathbb{R}^n$, $\mathbb{R}^m$, and $\mathbb{R}^p$ finite dim. v.s. with inner products, $\langle \cdot, \cdot \rangle_{n,m,p}$
- f convex; $G : \mathbb{R}^n \to \mathbb{R}^p$, $K-$ convex; $\mathbf{a} \in \mathbb{R}^p$, $\mathbf{A} : \mathbb{R}^n \to \mathbb{R}^p$
- S **"simple"** closed convex
- K closed convex cone, int $K \neq \emptyset$; *e.g.*, $K = \mathbb{R}^m_+, S^m_+, L^m_+$

- Possible, thanks to the *theory of duality for variational inequalities*.
- Produce methods with explicit formulas at each iteration ***that does not require the solution of any subproblem***.
- Yields algorithms with low computational cost very easy to implement, and with improved iteration complexity bounds.
- Naturally applied to Structured and Nonsmooth Convex Problems: SDP, SOC, Saddle point/minimax
- Again, "structure" helps to get better complexity results with EGL with a complexity estimate $\sim O(\frac{1}{k})$ for various NSO.

Primal-Dual Variational Inequality Associated to (P)

$$(P) \quad f_* = \inf\{f(\mathbf{x}) : \; -G(\mathbf{x}) \in K, \; \mathbf{A}\mathbf{x} = \mathbf{a}, \mathbf{x} \in S\}$$

One can show: $\mathbf{x}^*$ solves (P) iff $\exists(\mathbf{u}^*, \mathbf{v}^*)$ s.t. $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*)$ solves (PDVI):

$$\text{Find } \mathbf{z}^* = (\mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*) \in \Omega : \; \langle T(\mathbf{z}^*), \mathbf{z} - \mathbf{z}^* \rangle \geq 0, \; \forall \mathbf{z} \in \Omega$$

Primal-Dual Variational Inequality Associated to (P)

$$(P) \quad f_* = \inf\{f(\mathbf{x}) : \ -G(\mathbf{x}) \in K, \ \mathbf{A}\mathbf{x} = \mathbf{a}, \mathbf{x} \in S\}$$

One can show: $\mathbf{x}^*$ solves (P) iff $\exists(\mathbf{u}^*, \mathbf{v}^*)$ s.t. $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*)$ solves (PDVI):

$$\text{Find } \mathbf{z}^* = (\mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*) \in \Omega : \ \langle T(\mathbf{z}^*), \mathbf{z} - \mathbf{z}^* \rangle \geq 0, \ \forall \mathbf{z} \in \Omega$$

- $\Omega := S \times (K \times \mathbb{R}^p)$= "simple" $\times$ "Hard" $\times$ "Affine"
- The primal-dual operator is defined by

$$\begin{aligned} T(\mathbf{z}) &:= (\nabla f(\mathbf{x}) + \langle \mathbf{u}, \nabla G(\mathbf{x}) \rangle_m + \mathbf{A}^* \mathbf{v}, -G(\mathbf{x}), -(\mathbf{A}\mathbf{x} - \mathbf{a})) \\ &\equiv (T_1(\mathbf{z}), T_2(\mathbf{z}), T_3(\mathbf{z})). \end{aligned}$$

• Given $\mathbf{z} = (\mathbf{x}, \mathbf{u}, \mathbf{v}) \in \Omega, \ \ \Omega \equiv S \times (K \times \mathbb{R}^p)$
• let $Z := (X, U, W) = T(\bar{\mathbf{z}})$ for some other given $\bar{\mathbf{z}} \in \Omega$.
To apply EGL for solving (PDVI), and hence for solving (P) **all we need is to compute the projection-like map**

$$\mathbf{z}^+ := p(Z, \mathbf{z}) = \mathop{\mathbf{argmin}}_{\zeta} \{\langle Z, \zeta \rangle + d(\zeta, \mathbf{z})\}$$

for some chosen distance $d(\zeta, \mathbf{z})$.

Projection-like Map $\mathbf{z}^+ := p(Z, \mathbf{z})$ is Easy to Compute!

We choose d defined by:

$$d(\mathbf{z}', \mathbf{z}) := d_1(\mathbf{x}', \mathbf{x}) + d_2(\mathbf{u}', \mathbf{u}) + \frac{1}{2}\|\mathbf{v}' - \mathbf{v}\|^2,$$

1. d_1 captures the "simple" constraints described by S
2. d_2 captures the "hard" constraints through projections-like maps on K
3. Last distance captures the affine equality constraints (if any).
4. Since d is *separable*, the computation of p decomposed accordingly, and hence $\mathbf{z}^+ = (\mathbf{x}^+, \mathbf{u}^+, \mathbf{v}^+)$ are computed independently and easily as follows.

Projection-like Map $\mathbf{z}^+ := p(Z, \mathbf{z})$ is Easy to Compute!

We choose d defined by:

$$d(\mathbf{z}', \mathbf{z}) := d_1(\mathbf{x}', \mathbf{x}) + d_2(\mathbf{u}', \mathbf{u}) + \frac{1}{2}\|\mathbf{v}' - \mathbf{v}\|^2,$$

1. d_1 captures the "simple" constraints described by S
2. d_2 captures the "hard" constraints through projections-like maps on K
3. Last distance captures the affine equality constraints (if any).
4. Since d is *separable*, the computation of p decomposed accordingly, and hence $\mathbf{z}^+ = (\mathbf{x}^+, \mathbf{u}^+, \mathbf{v}^+)$ are computed independently and easily as follows.

$$\begin{aligned}
\mathbf{x}^+ = p_1(T_1(\bar{\mathbf{z}}), \mathbf{x}) := p_1(X, \mathbf{x}) &= \operatorname{argmin}\{\langle \mathbf{w}, X\rangle + d_1(\mathbf{w}, \mathbf{x}) : \ \mathbf{w} \in S\},\\
\mathbf{u}^+ = p_2(T_2(\bar{\mathbf{z}}), u) := p_2(U, \mathbf{u}) &= \operatorname{argmin}\{\langle \mathbf{w}, U\rangle + d_2(\mathbf{w}, \mathbf{u}) : \ \mathbf{w} \in K\},\\
\mathbf{v}^+ = p_3(T_3(\bar{\mathbf{z}}), v) := p_3(W, \mathbf{v}) &= \operatorname{argmin}\{\langle \mathbf{w}, W\rangle + \frac{1}{2}\|\mathbf{w} - \mathbf{v}\|^2 : \ \mathbf{w} \in \mathbb{R}^p\}
\end{aligned}$$

In particular, note that one always has: $\mathbf{v}^+ = \mathbf{v} - W$.

- For computing $\mathbf{x}^+, \mathbf{u}^+$ we use the results given in the previous tables, e.g. for $S = \mathbb{R}^n, \mathbb{R}^n_+, \ S^n_+, L^n_+$. Similarly, for $K = \mathbb{R}^n_+, \ S^n_+$, and L^n_+.
- **No matter how complicated the constraints are in the ground set $S \cap Q$**, the resulting projections-like maps are **given by analytical formulas!**

Nonsmooth and Nonconvex Problems

...No miracles here...!....

Again, look for problems with special structures that can beneficially exploited.

- The Single Source Sensor Localization Problem
- Sparse PCA Problems
- Nonconvex Affine Feasibility Problems

The Source Localization Problem

- **SL Problem:** Locate a single radiating source from noisy range measurements collected using a network of passive sensors.
- The SL problem has received significant attention in the signal processing literature, specifically in the field of mobile phones localization.

The Source Localization Problem

- **SL Problem:** Locate a single radiating source from noisy range measurements collected using a network of passive sensors.
- The SL problem has received significant attention in the signal processing literature, specifically in the field of mobile phones localization.
- Consider an array of m sensors with
  1. $\mathbf{a}_j \in \mathbb{R}^n$ coordinates of the jth sensor (in practical applications $n = 2$ or 3)
  2. $d_j > 0$ the noisy observation of range between source and jth sensor:

$$d_j = \|\mathbf{x} - \mathbf{a}_j\| + \varepsilon_j, \quad j = 1, \ldots, m,$$

  $\mathbf{x} \in \mathbb{R}^n$ is the unknown source's coordinate vector; ε unknown noise vector.

Many possible mathematical formulations. Given the observed range measurements $d_j > 0$, find a "good" approximation of the source $\mathbf{x}$. A natural and common optimization formulation:

$$\text{(SL)} \quad \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ f(\mathbf{x}) \equiv \sum_{j=1}^{m} \left(\|\mathbf{x} - \mathbf{a}_j\| - d_j \right)^2 \right\}.$$

The Source Localization Problem

- **SL Problem:** Locate a single radiating source from noisy range measurements collected using a network of passive sensors.
- The SL problem has received significant attention in the signal processing literature, specifically in the field of mobile phones localization.
- Consider an array of m sensors with
  1. $\mathbf{a}_j \in \mathbb{R}^n$ coordinates of the jth sensor (in practical applications $n = 2$ or 3)
  2. $d_j > 0$ the noisy observation of range between source and jth sensor:

$$d_j = \|\mathbf{x} - \mathbf{a}_j\| + \varepsilon_j, \quad j = 1, \ldots, m,$$

$\mathbf{x} \in \mathbb{R}^n$ is the unknown source's coordinate vector; ε unknown noise vector.

Many possible mathematical formulations. Given the observed range measurements $d_j > 0$, find a "good" approximation of the source $\mathbf{x}$. A natural and common optimization formulation:

$$\text{(SL)} \quad \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ f(\mathbf{x}) \equiv \sum_{j=1}^{m} (\|\mathbf{x} - \mathbf{a}_j\| - d_j)^2 \right\}.$$

Has also a statistical interpretation: when ϵ follows a Gaussian distribution with a covariance matrix $\sim I_d$, the optimal solution of (SL) is in fact a maximum likelihood estimate.

The SL problem is a **nonsmooth nonconvex** problem and as such, not easy to solve.

A Simple Gradient-Based Algorithm

- The derivation is inspired from Weiszfeld's algorithm (1939) for the classical *convex* location problem

A Simple Gradient-Based Algorithm

- The derivation is inspired from Weiszfeld's algorithm (1939) for the classical *convex* location problem

Algorithm SWLS:

$$\mathbf{x}_{k+1} \in \underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}} \sum_{j=1}^{m} \left(\frac{\|\mathbf{x}-\mathbf{a}_j\|^2}{\|\mathbf{x}_k-\mathbf{a}_j\|} - d_j \right)^2 .$$

- Can be re-formulated for each k as a Weighted Least Squares (WLS)
- Denote the set of sensors by $\mathcal{A} := \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$.
- The scheme **is not well defined if $\mathbf{x}_k \in \mathcal{A}$ for some k !**

A Simple Gradient-Based Algorithm

- The derivation is inspired from Weiszfeld's algorithm (1939) for the classical *convex* location problem

Algorithm SWLS:

$$\mathbf{x}_{k+1} \in \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{m} \left(\frac{\|\mathbf{x}-\mathbf{a}_j\|^2}{\|\mathbf{x}_k-\mathbf{a}_j\|} - d_j \right)^2.$$

- Can be re-formulated for each k as a Weighted Least Squares (WLS)
- Denote the set of sensors by $\mathcal{A} := \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$.
- The scheme **is not well defined if $\mathbf{x}_k \in \mathcal{A}$ for some k !**
- Eliminate non-smoothness difficulty by choosing a "good" initial point!

$$(G) \;\; \exists \mathbf{x}_0 \;\text{ s.t. }\; f(\mathbf{x}_0) < \frac{1}{4} \min_{j=1,\ldots,m} d_j^2$$

The analysis is quite unusual...[Beck-Teboulle'(08)]

Convergence of SWLS

Theorem Let $\{\mathbf{x}_k\}$ be generated by SWLS such that $\mathbf{x}_0$ satisfies (G). Then,

(a) $\mathbf{x}_k \notin \mathcal{A}$ for every $k \geq 0$.

(b) The sequence $\{\mathbf{x}_k\}$ is bounded. Any limit point of $\{\mathbf{x}_k\}$ is a stationary point of f.

(c) The sequence of function values $\{f(\mathbf{x}_k)\}$ converges to f_*, where f_* is the function value at some stationary point of f.

(d) Assuming all stationary points are isolated, i.e.,
$\mathbf{x}^*$ is an isolated s.p. of f if there are no other s.p. in some $N(\mathbf{x}^*)$,
the sequence $\{\mathbf{x}_k\}$ converges to a stationary point.

Convergence of SWLS

Theorem Let $\{\mathbf{x}_k\}$ be generated by SWLS such that $\mathbf{x}_0$ satisfies (G). Then,

(a) $\mathbf{x}_k \notin \mathcal{A}$ for every $k \geq 0$.

(b) The sequence $\{\mathbf{x}_k\}$ is bounded. Any limit point of $\{\mathbf{x}_k\}$ is a stationary point of f.

(c) The sequence of function values $\{f(\mathbf{x}_k)\}$ converges to f_*, where f_* is the function value at some stationary point of f.

(d) Assuming all stationary points are isolated, i.e.,
$\mathbf{x}^*$ is an isolated s.p. of f if there are no other s.p. in some $N(\mathbf{x}^*)$,
the sequence $\{\mathbf{x}_k\}$ converges to a stationary point.

We have performed Monte Carlo runs and observed

- The algorithm appears very robust: $\#$ of iterations constant ≈ 30, independently of size (m, n) with stopping rule $\|\nabla f(\mathbf{x}_k)\| \leq 10^{-5}$
- Convergence to a "global minimum" was almost always observed..
- A probabilistic analysis of the algorithm seems worthwhile......

Sparse PCA

Principal Component Analysis solves

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

while sparse Principal Component Analysis solves

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

$\|x\|_0$ 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 [D'aspremont et al. 2008]
2. Approximation/Modified formulations: Many proposed approaches

Sparse PCA: The Big Picture

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

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

Sparse PCA: The Big Picture

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

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

♠ 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** $\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_1 \le \sqrt{k}\}$
- **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** $\max\{\mathrm{tr}(AX) :\ \mathrm{tr}\,(X) = 1, X \succeq 0, \|X\|_1 \le k\}$

Sparse PCA: The Big Picture

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

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

♠ 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** $\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_1 \le \sqrt{k}\}$
- **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** $\max\{\mathrm{tr}(AX) :\ \mathrm{tr}\,(X) = 1, X \succeq 0, \|X\|_1 \le k\}$

- Convex relaxations are too computationally expensive for large problems.
- No algorithm give bounds to the optimal solution of the **original problem**.
- Even when "Simple", the algorithms 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 SPCA

Type	Iteration	Per-Iteration Complexity	References
l_1-constrained	$x_i^{j+1} = \frac{\text{sgn}(((A+\frac{\sigma}{2})x^j)_i)(\|((A+\frac{\sigma}{2})x^j)_i\|-\lambda^j)_+}{\sqrt{\sum_h (\|((A+\frac{\sigma}{2})x^j)_h\|-\lambda^j)_+^2}}$	$O(n^2)$, $O(mn)$	Witten et al. (2009)
l_1-constrained	$x_i^{j+1} = \frac{\text{sgn}((Ax^j)_i)(\|(Ax^j)_i\|-s^j)_+}{\sqrt{\sum_h (\|(Ax^j)_h\|-s^j)_+^2}}$ where s^j is $(k+1)$-largest entry of vector $\|Ax^j\|$	$O(n^2)$, $O(mn)$	Sigg-Buhman (2008)
l_0-penalized	$z^{j+1} = \frac{\sum_i [\text{sgn}((b_i^T z^j)^2 - s)]_+ (b_i^T z^j) b_i}{\\|\sum_i [\text{sgn}((b_i^T z^j)^2 - s)]_+ (b_i^T z^j) b_i\\|_2}$	$O(mn)$	Shen-Huang (2008), Journee et al. (2010)
l_0-penalized	$x_i^{j+1} = \frac{\text{sgn}(2(Ax^j)_i)(\|2(Ax^j)_i\| - s\varphi_p'(\|x_i^j\|))_+}{\sqrt{\sum_h (\|2(Ax^j)_h\| - s\varphi_p'(\|x_h^j\|))_+^2}}$	$O(n^2)$	Sriperumbudur et al. (2010)
l_1-penalized	$y^{j+1} = \underset{y}{\text{argmin}} \{ \sum_i \\|b_i - x^j y^T b_i\\|_2^2 + \lambda\\|y\\|_2^2 + s\\|y\\|_1 \}$ $x^{j+1} = \frac{(\sum_i b_i b_i^T) y^{j+1}}{\\|(\sum_i b_i b_i^T) y^{j+1}\\|_2}$		Zou et al. (2006)
l_1-penalized	$z^{j+1} = \frac{\sum_i (\|b_i^T z^j\| - s)_+ \text{sgn}(b_i^T z^j) b_i}{\\|\sum_i (\|b_i^T z^j\| - s)_+ \text{sgn}(b_i^T z^j) b_i\\|_2}$	$O(mn)$	Shen-Huang (2008), Journee et al. (2010)

Table: Cheap sparse PCA algorithms for modified problems.

The Big Picture Revisited

1. All previous listed algorithms have been derived from various disparate approaches/motivations to solve **modifications** of SPCA.

Any connection?

2. Is is possible to tackle the difficult sparse PCA problem **as is**?

The Big Picture Revisited

1. All previous listed algorithms have been derived from various disparate approaches/motivations to solve **modifications** of SPCA.

   Any connection?

2. Is is possible to tackle the difficult sparse PCA problem **as is**?

Very recently we have shown that:(Details in Luss-Teboulle (2011))

- 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**

Maximizing a Convex function over a Compact Nonconvex set

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

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

$$x^0 \in C,\ p^j = \operatorname{argmax}\{\langle x - x^j, \nabla F(x^j)\rangle : x \in C\}$$

$$x^{j+1} = x^j + \alpha^j(p^j - x^j),\ \alpha^j \in (0,1] \text{ stepsize}$$

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

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

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

Despite the hard constraint, very 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 $\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$ 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, very 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.

- **Iterations are cheap** (e.g., in comparison to SDP convex relaxations which require eigenvalue decompositions at every iteration)
- **Convergence:** Every limit point of $\{x^j\}$ converges to a stationary point.
- **Complexity:** $O(kn)$ or $O(mn)$
- **The 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** to get the desired sparsity. This can be very computationally expensive and not needed here.
- For Numerical results and Comparisons, see Luss-Teboulle (2011), available on arXiv.

Extensions

Again the special problem structures beneficially exploited to build a simple scheme **ConGradU**:

- that encompasses all currently known cheap methods for sparse PCA
- can easily be applied to the solve **original l_0-constrained problem**

Our tools can be easily extended to produce other novel simple algorithms for other similar problems:

1. Sparse Singular Value Decomposition:

$$\max\{x^T By : \|x\|_2 = 1,\ \|y\|_2 = 1,\ \|x\|_0 \leq k_1,\ \|y\|_0 \leq k_2\}$$

2. Sparse Canonical Correlation Analysis:

$$\max\{x^T B^T Cy : x^T B^T Bx = 1\ y^T C^T Cy = 1,\ \|x\|_0 \leq k_1,\ \|y\|_0 \leq k_2\}$$

3. Sparse nonnegative Principal Component Analysis:

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \geq 0\}$$

Summary on First Order Schemes

- Powerful for constructing cheap iterations
- Efficient algorithms in many applied optimization models with structures.
- Further research needed for simple and efficient schemes that can cope with **curse of dimensionality and Nonconvex/Nonsmooth settings**.

Summary on First Order Schemes

- Powerful for constructing cheap iterations
- Efficient algorithms in many applied optimization models with structures.
- Further research needed for simple and efficient schemes that can cope with **curse of dimensionality and Nonconvex/Nonsmooth settings**.

Thank you for listening!