Convex Optimization in Machine Learning and Inverse Problems
Part 2: First-Order Methods

Mário A. T. Figueiredo[1] and Stephen J. Wright[2]

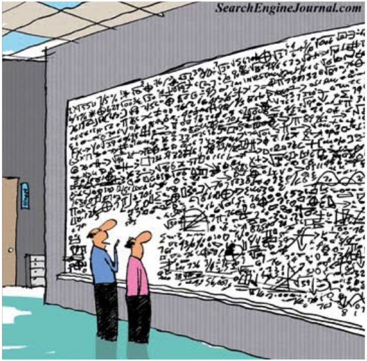
[1]Instituto de Telecomunicações,
Instituto Superior Técnico, Lisboa, Portugal

[2]Computer Sciences Department,
University of Wisconsin,
Madison, WI, USA

Condensed version of ICCOPT tutorial, Lisbon, Portugal, 2013

“In general, optimization problems are unsolvable”, Nesterov, 1984

"In general, optimization problems are unsolvable", Nesterov, 1984



Focus (Initially) on Smooth Convex Functions

Consider $\min_{x \in \mathbb{R}^n} f(x)$, with f smooth and convex.

Usually assume $\mu I \preceq \nabla^2 f(x) \preceq LI, \ \forall_x$, with $0 \leq \mu \leq L$ (thus L is a Lipschitz constant of ∇f).

If $\mu > 0$, then f is μ-strongly convex (as seen in Part 1) and

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2} \|y - x\|_2^2.$$

Define conditioning (or condition number) as $\kappa := L/\mu$.

We are often interested in convex quadratics:

$$f(x) = \frac{1}{2} x^T A x, \qquad \mu I \preceq A \preceq LI \ \text{ or}$$

$$f(x) = \frac{1}{2} \|Bx - b\|_2^2, \quad \mu I \preceq B^T B \preceq LI$$

What's the Setup?

We consider iterative algorithms

$$x_{k+1} = \Phi(x_k), \quad \text{or} \quad x_{k+1} = \Phi(x_k, x_{k-1})$$

For now, assume we can evaluate $f(x_t)$ and $\nabla f(x_t)$ at each iteration.
Later, we look at broader classes of problems:

- nonsmooth f;
- f not available (or too expensive to evaluate exactly);
- only an *estimate* of the gradient is available;
- nonsmooth regularization; *i.e.*, instead of simply $f(x)$, we want to minimize $f(x) + \tau\psi(x)$.

We focus on algorithms that can be adapted to those scenarios.

Skip to slide 29

Recall Francis Bach's talk

Steepest Descent

Steepest descent (a.k.a. gradient descent):

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \qquad \text{for some } \alpha_k > 0.$$

Different ways to select an appropriate α_k.

1. Hard: interpolating scheme with safeguarding to identify an approximate minimizing α_k.
2. Easy: backtracking. $\bar{\alpha}$, $\frac{1}{2}\bar{\alpha}$, $\frac{1}{4}\bar{\alpha}$, $\frac{1}{8}\bar{\alpha}$, ... until sufficient decrease in f is obtained.
3. Trivial: don't test for function decrease; use rules based on L and μ.

Analysis for 1 and 2 usually yields global convergence at unspecified rate. The "greedy" strategy of getting good decrease in the current search direction may lead to better practical results.

Analysis for 3: Focuses on convergence rate, and leads to accelerated multi-step methods.

Line Search

Seek α_k that satisfies Wolfe conditions: "sufficient decrease" in f:

$$f(x_k - \alpha_k \nabla f(x_k)) \le f(x_k) - c_1 \alpha_k \|\nabla f(x_k)\|^2, \qquad (0 < c_1 \ll 1)$$

while "not being too small" (significant increase in the directional derivative):

$$\nabla f(x_{k+1})^T \nabla f(x_k) \ge -c_2 \|\nabla f(x_k)\|^2, \qquad (c_1 < c_2 < 1).$$

(works for nonconvex f.) Can show that accumulation points $\bar{x}$ of $\{x_k\}$ are stationary: $\nabla f(\bar{x}) = 0$ (thus minimizers, if f is convex)

Can do one-dimensional line search for α_k, taking minima of quadratic or cubic interpolations of the function and gradient at the last two values tried. Use brackets to ensure steady convergence. Often finds suitable α within 3 attempts. (Nocedal and Wright, 2006)

Backtracking

Try $\alpha_k = \bar{\alpha}, \frac{\bar{\alpha}}{2}, \frac{\bar{\alpha}}{4}, \frac{\bar{\alpha}}{8}, \ldots$ until the sufficient decrease condition is satisfied.

No need to check the second Wolfe condition: the α_k thus identified is "within striking distance" of an α that's too large — so it is not too short.

Backtracking is widely used in applications, but doesn't work on nonsmooth problems, or when f is not available / too expensive.

Constant (Short) Steplength

By elementary use of Taylor's theorem, and since $\nabla^2 f(x) \preceq LI$,

$$f(x_{k+1}) \le f(x_k) - \alpha_k \left(1 - \frac{\alpha_k}{2} L\right) \|\nabla f(x_k)\|_2^2$$

For $\alpha_k \equiv 1/L$, $\quad f(x_{k+1}) \le f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|_2^2,$

thus $\quad \|\nabla f(x_k)\|^2 \le 2L[f(x_k) - f(x_{k+1})]$

Summing for $k = 0, 1, \ldots, N$, and telescoping the sum,

$$\sum_{k=0}^{N} \|\nabla f(x_k)\|^2 \le 2L[f(x_0) - f(x_{N+1})].$$

It follows that $\nabla f(x_k) \to 0$ if f is bounded below.

Suppose that the minimizer x^* is unique.

Another elementary use of Taylor's theorem shows that

$$\|x_{k+1} - x^*\|^2 \le \|x_k - x^*\|^2 - \alpha_k \left(\frac{2}{L} - \alpha_k \right) \|\nabla f(x_k)\|^2,$$

so that $\{\|x_k - x^*\|\}$ is decreasing.

Define for convenience: $\Delta_k := f(x_k) - f(x^*)$. By convexity, have

$$\Delta_k \le \nabla f(x_k)^T (x_k - x^*) \le \|\nabla f(x_k)\| \, \|x_k - x^*\| \le \|\nabla f(x_k)\| \, \|x_0 - x^*\|.$$

From previous page (subtracting $f(x^*)$ from both sides of the inequality), and using the inequality above, we have

$$\Delta_{k+1} \le \Delta_k - (1/2L)\|\nabla f(x_k)\|^2 \le \Delta_k - \frac{1}{2L\|x_0 - x^*\|^2} \Delta_k^2.$$

Weakly convex: $1/k$ sublinear; Strongly convex: linear

Take reciprocal of both sides and manipulate (using $(1-\epsilon)^{-1} \geq 1+\epsilon$):

$$\frac{1}{\Delta_{k+1}} \geq \frac{1}{\Delta_k} + \frac{1}{2L\|x_0 - x^*\|^2} \geq \frac{1}{\Delta_0} + \frac{k+1}{2L\|x_0 - x^*\|^2},$$

which yields

$$f(x_{k+1}) - f(x^*) \leq \frac{2L\|x_0 - x\|^2}{k+1}.$$

The classic $1/k$ convergence rate!

By assuming $\mu > 0$, can set $\alpha_k \equiv 2/(\mu + L)$ and get a linear (geometric) rate: Much better than sublinear, in the long run

$$\|x_k - x^*\|^2 \leq \left(\frac{L-\mu}{L+\mu}\right)^{2k} \|x_0 - x^*\|^2 = \left(1 - \frac{2}{\kappa+1}\right)^{2k} \|x_0 - x^*\|^2.$$

Weakly convex: $1/k$ sublinear; Strongly convex: linear

Since by Taylor's theorem we have

$$\Delta_k = f(x_k) - f(x^*) \leq (L/2)\|x_k - x^*\|^2,$$

it follows immediately that

$$f(x_k) - f(x^*) \leq \frac{L}{2}\left(1 - \frac{2}{\kappa + 1}\right)^{2k} \|x_0 - x^*\|^2.$$

Note: A geometric / linear rate is generally better than almost any sublinear ($1/k$ or $1/k^2$) rate.

Exact minimizing α_k: Faster rate?

Question: does taking α_k as the exact minimizer of f along $-\nabla f(x_k)$ yield better rate of linear convergence?

Consider $f(x) = \frac{1}{2} x^T A x$ (thus $x^* = 0$ and $f(x^*) = 0$.)

We have $\nabla f(x_k) = A x_k$. Exactly minimizing w.r.t. α_k,

$$\alpha_k = \arg\min_{\alpha} \frac{1}{2}(x_k - \alpha A x_k)^T A (x_k - \alpha A x_k) = \frac{x_k^T A^2 x_k}{x_k^T A^3 x_k} \in \left[\frac{1}{L}, \frac{1}{\mu}\right]$$

Thus

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2} \frac{(x_k^T A^2 x_k)^2}{(x_k^T A x_k)(x_k^T A^3 x_k)},$$

so, defining $z_k := A x_k$, we have

$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} \leq 1 - \frac{\|z_k\|^4}{(z_k^T A^{-1} z_k)(z_k^T A z_k)}.$$

Exact minimizing α_k: Faster rate?

Using Kantorovich inequality:

$$(z^T A z)(z^T A^{-1} z) \le \frac{(L+\mu)^2}{4L\mu} \|z\|^4.$$

Thus

$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} \le 1 - \frac{4L\mu}{(L+\mu)^2} = \left(1 - \frac{2}{\kappa+1}\right)^2,$$

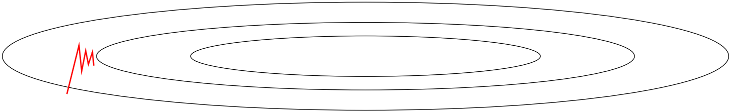
and so

$$f(x_k) - f(x^*) \le \left(1 - \frac{2}{\kappa+1}\right)^{2k} [f(x_0) - f(x^*)].$$

No improvement in the linear rate over constant steplength!

The slow linear rate is typical!

Not just a pessimistic bound!



Multistep Methods: The Heavy-Ball

Enhance the search direction using a contribution from the previous step. (known as heavy ball, momentum, or two-step)

Consider first a constant step length α, and a second parameter β for the "momentum" term:

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$$

Analyze by defining a composite iterate vector:

$$w_k := \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix}.$$

Thus

$$w_{k+1} = B w_k + o(\|w_k\|), \qquad B := \begin{bmatrix} -\alpha \nabla^2 f(x^*) + (1+\beta) I & -\beta I \\ I & 0 \end{bmatrix}.$$

Multistep Methods: The Heavy-Ball

Matrix B has same eigenvalues as

$$\begin{bmatrix} -\alpha\Lambda + (1+\beta)I & -\beta I \\ I & 0 \end{bmatrix}, \qquad \Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$

where λ_i are the eigenvalues of $\nabla^2 f(x^*)$.

Choose α, β to explicitly minimize the max eigenvalue of B, obtain

$$\alpha = \frac{4}{L}\frac{1}{(1+1/\sqrt{\kappa})^2}, \quad \beta = \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^2.$$

Leads to linear convergence for $\|x_k - x^*\|$ with rate approximately

$$\left(1 - \frac{2}{\sqrt{\kappa}+1}\right).$$

Summary: Linear Convergence, Strictly Convex f

- Steepest descent: Linear rate approx $\left(1 - \frac{2}{\kappa}\right)$;
- Heavy-ball: Linear rate approx $\left(1 - \frac{2}{\sqrt{\kappa}}\right)$.

Big difference! To reduce $\|x_k - x^*\|$ by a factor ϵ, need k large enough that

$$\left(1 - \frac{2}{\kappa}\right)^k \le \epsilon \quad \Leftarrow \quad k \ge \frac{\kappa}{2} |\log \epsilon| \quad \text{(steepest descent)}$$

$$\left(1 - \frac{2}{\sqrt{\kappa}}\right)^k \le \epsilon \quad \Leftarrow \quad k \ge \frac{\sqrt{\kappa}}{2} |\log \epsilon| \quad \text{(heavy-ball)}$$

A factor of $\sqrt{\kappa}$ difference; e.g. if $\kappa = 1000$ (not at all uncommon in inverse problems), need ~ 30 times fewer steps.

Conjugate Gradient

Basic conjugate gradient (CG) step is

$$x_{k+1} = x_k + \alpha_k p_k, \qquad p_k = -\nabla f(x_k) + \gamma_k p_{k-1}.$$

Can be identified with heavy-ball, with $\beta_k = \dfrac{\alpha_k \gamma_k}{\alpha_{k-1}}$.

However, CG can be implemented in a way that doesn't require knowledge (or estimation) of L and μ.

- Choose α_k to (approximately) miminize f along p_k;
- Choose γ_k by a variety of formulae (Fletcher-Reeves, Polak-Ribiere, etc), all of which are equivalent if f is convex quadratic. e.g.

$$\gamma_k = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}$$

Conjugate Gradient

Nonlinear CG: Variants include Fletcher-Reeves, Polak-Ribiere, Hestenes.

Restarting periodically with $p_k = -\nabla f(x_k)$ is useful (e.g. every n iterations, or when p_k is not a descent direction).

For quadratic f, convergence analysis is based on eigenvalues of A and Chebyshev polynomials, min-max arguments. Get

- Finite termination in as many iterations as there are distinct eigenvalues;
- Asymptotic linear convergence with rate approx $1 - \frac{2}{\sqrt{\kappa}}$. (like heavy-ball.)

(Nocedal and Wright, 2006)

Accelerated First-Order Methods

Accelerate the rate to $1/k^2$ for weakly convex, while retaining the linear rate (related to $\sqrt{\kappa}$) for strongly convex case.

Nesterov (1983) describes a method that requires κ.

Initialize: Choose x_0, $\alpha_0 \in (0,1)$; set $y_0 \leftarrow x_0$.

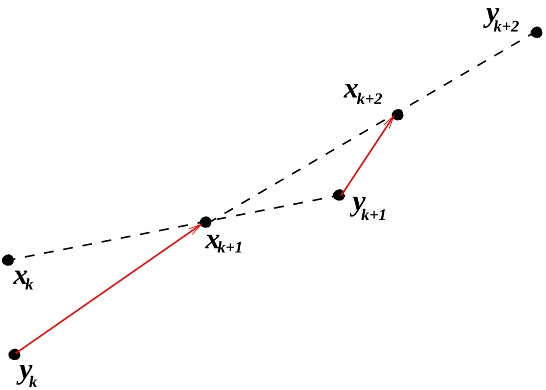
Iterate: $x_{k+1} \leftarrow y_k - \frac{1}{L}\nabla f(y_k)$; (*short-step*)

find $\alpha_{k+1} \in (0,1)$: $\alpha_{k+1}^2 = (1-\alpha_{k+1})\alpha_k^2 + \frac{\alpha_{k+1}}{\kappa}$;

set $\beta_k = \dfrac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$;

set $y_{k+1} \leftarrow x_{k+1} + \beta_k(x_{k+1} - x_k)$.

Still works for weakly convex ($\kappa = \infty$).



Separates the "gradient descent" and "momentum" step components.

Convergence Results: Nesterov

If $\alpha_0 \geq 1/\sqrt{\kappa}$, have

$$f(x_k) - f(x^*) \leq c_1 \min \left(\left(1 - \frac{1}{\sqrt{\kappa}} \right)^k, \frac{4L}{(\sqrt{L} + c_2 k)^2} \right),$$

where constants c_1 and c_2 depend on x_0, α_0, L.

- Linear convergence "heavy-ball" rate for strongly convex f;
- $1/k^2$ sublinear rate otherwise.

In the special case of $\alpha_0 = 1/\sqrt{\kappa}$, this scheme yields

$$\alpha_k \equiv \frac{1}{\sqrt{\kappa}}, \qquad \beta_k \equiv 1 - \frac{2}{\sqrt{\kappa} + 1}.$$

Beck and Teboulle

Beck and Teboulle (2009) propose a similar algorithm, with a fairly short and elementary analysis (though still not intuitive).

Initialize: Choose x_0; set $y_1 = x_0$, $t_1 = 1$;

Iterate:
$$x_k \leftarrow y_k - \tfrac{1}{L}\nabla f(y_k);$$
$$t_{k+1} \leftarrow \tfrac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right);$$
$$y_{k+1} \leftarrow x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1}).$$

For (weakly) convex f, converges with $f(x_k) - f(x^*) \sim 1/k^2$.

When L is not known, increase an estimate of L until it's big enough.

Beck and Teboulle (2009) do the convergence analysis in 2-3 pages; elementary, but "technical."

A Non-Monotone Gradient Method: Barzilai-Borwein

Barzilai and Borwein (1988) (BB) proposed an unusual choice of α_k. Allows f to increase (sometimes a lot) on some steps: non-monotone.

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \qquad \alpha_k := \arg\min_{\alpha} \|s_k - \alpha z_k\|^2,$$

where

$$s_k := x_k - x_{k-1}, \qquad z_k := \nabla f(x_k) - \nabla f(x_{k-1}).$$

Explicitly, we have

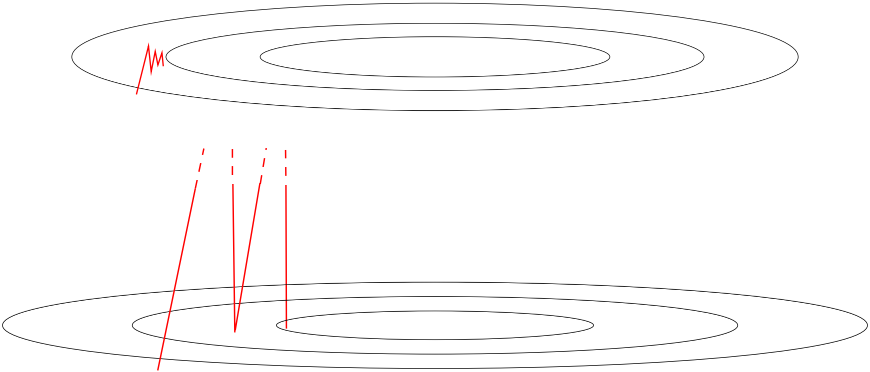
$$\alpha_k = \frac{s_k^T z_k}{z_k^T z_k}.$$

Note that for $f(x) = \frac{1}{2}x^T Ax$, we have

$$\alpha_k = \frac{s_k^T A s_k}{s_k^T A^2 s_k} \in \left[\frac{1}{L}, \frac{1}{\mu}\right].$$

BB can be viewed as a quasi-Newton method, with the Hessian approximated by $\alpha_k^{-1} I$.

Comparison: BB vs Greedy Steepest Descent



There Are Many BB Variants

- use $\alpha_k = s_k^T s_k / s_k^T z_k$ in place of $\alpha_k = s_k^T z_k / z_k^T z_k$;
- alternate between these two formulae;
- hold α_k constant for a number (2, 3, 5) of successive steps;
- take α_k to be the steepest descent step from the previous iteration.

Nonmonotonicity appears essential to performance. Some variants get global convergence by requiring a sufficient decrease in f over the worst of the last M (say 10) iterates.

The original 1988 analysis in BB's paper is nonstandard and illuminating (just for a 2-variable quadratic).

In fact, most analyses of BB and related methods are nonstandard, and consider only special cases. The precursor of such analyses is Akaike (1959). More recently, see Ascher, Dai, Fletcher, Hager and others.

Extending to the Constrained Case: $x \in \Omega$

How to change these methods to handle the constraint $x \in \Omega$?
(assuming that Ω is a closed convex set)

Some algorithms and theory stay much the same,
...if we can involve the constraint $x \in \Omega$ explicity in the subproblems.

Example: Nesterov's constant step scheme requires just one calculation to be changed from the unconstrained version.

Initialize: Choose x_0, $\alpha_0 \in (0,1)$; set $y_0 \leftarrow x_0$.

Iterate: $x_{k+1} \leftarrow \arg\min_{y \in \Omega} \frac{1}{2}\|y - [y_k - \frac{1}{L}\nabla f(y_k)]\|_2^2$;
find $\alpha_{k+1} \in (0,1)$: $\alpha_{k+1}^2 = (1-\alpha_{k+1})\alpha_k^2 + \frac{\alpha_{k+1}}{\kappa}$;
set $\beta_k = \frac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$;
set $y_{k+1} \leftarrow x_{k+1} + \beta_k(x_{k+1} - x_k)$.

Convergence theory is unchanged.

Regularized Optimization

How to change these methods to handle regularized optimization?

$$\min_x f(x) + \tau\psi(x),$$

where f is convex and smooth, while ψ is convex but usually **nonsmooth**.

Regularized Optimization

How to change these methods to handle regularized optimization?

$$\min_x f(x) + \tau\psi(x),$$

where f is convex and smooth, while ψ is convex but usually **nonsmooth**.

Often, all that is needed is to change the update step to

$$x_k = \arg\min_x \|x - \Phi(x_k)\|_2^2 + \lambda\psi(x).$$

where $\Phi(x_k)$ is a gradient descent step

Regularized Optimization

How to change these methods to handle regularized optimization?

$$\min_x \; f(x) + \tau\psi(x),$$

where f is convex and smooth, while ψ is convex but usually **nonsmooth**.

Often, all that is needed is to change the update step to

$$x_k = \arg\min_x \|x - \Phi(x_k)\|_2^2 + \lambda\psi(x).$$

where $\Phi(x_k)$ is a gradient descent step or something more complicated; *e.g.*, an accelerated method with $\Phi(x_k, x_{k-1})$ instead of $\Phi(x_k)$

Regularized Optimization

How to change these methods to handle regularized optimization?

$$\min_x f(x) + \tau\psi(x),$$

where f is convex and smooth, while ψ is convex but usually **nonsmooth**.

Often, all that is needed is to change the update step to

$$x_k = \arg\min_x \|x - \Phi(x_k)\|_2^2 + \lambda\psi(x).$$

where $\Phi(x_k)$ is a gradient descent step or something more complicated; *e.g.*, an accelerated method with $\Phi(x_k, x_{k-1})$ instead of $\Phi(x_k)$

This is the shrinkage/tresholding step; how to solve it with nonsmooth ψ? That's the topic of the following slides.

Handling Nonsmoothness (e.g. ℓ_1 Norm)

Convexity $\Rightarrow$ continuity (on the domain of the function).

Convexity $\not\Rightarrow$ differentiability (e.g., $\psi(x) = \|x\|_1$).

Subgradients generalize gradients for general convex functions:

Handling Nonsmoothness (e.g. ℓ_1 Norm)

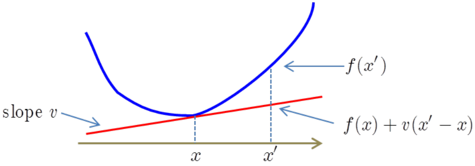
Convexity $\Rightarrow$ continuity (on the domain of the function).

Convexity $\not\Rightarrow$ differentiability (e.g., $\psi(x) = \|x\|_1$).

Subgradients generalize gradients for general convex functions:

$$v \text{ is a subgradient of } f \text{ at } x \text{ if } \; f(x') \geq f(x) + v^T(x' - x)$$

Subdifferential: $\partial f(x) = \{\text{all subgradients of } f \text{ at } x\}$



linear lower bound

Handling Nonsmoothness (e.g. ℓ_1 Norm)

Convexity $\Rightarrow$ continuity (on the domain of the function).

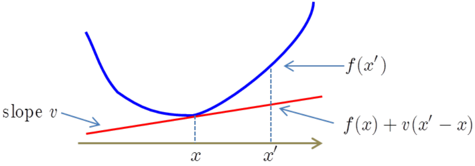
Convexity $\not\Rightarrow$ differentiability (e.g., $\psi(x) = \|x\|_1$).

Subgradients generalize gradients for general convex functions:

v is a subgradient of f at x if $f(x') \geq f(x) + v^T(x' - x)$

Subdifferential: $\partial f(x) = \{\text{all subgradients of } f \text{ at } x\}$

If f is differentiable, $\partial f(x) = \{\nabla f(x)\}$



linear lower bound

Handling Nonsmoothness (e.g. ℓ_1 Norm)

Convexity $\Rightarrow$ continuity (on the domain of the function).

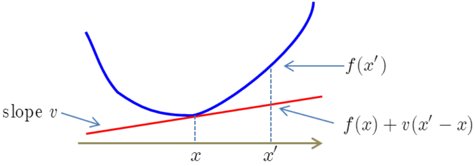
Convexity $\not\Rightarrow$ differentiability (e.g., $\psi(x) = \|x\|_1$).

Subgradients generalize gradients for general convex functions:

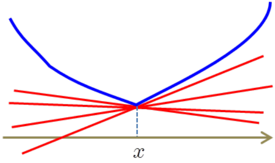
v is a subgradient of f at x if $f(x') \geq f(x) + v^T(x' - x)$

Subdifferential: $\partial f(x) = \{\text{all subgradients of } f \text{ at } x\}$

If f is differentiable, $\partial f(x) = \{\nabla f(x)\}$



linear lower bound



nondifferentiable case

More on Subgradients and Subdifferentials

The subdifferential is a set-valued function:

$$f : \mathbb{R}^d \to \mathbb{R} \quad \Rightarrow \quad \partial f : \mathbb{R}^d \to 2^{\mathbb{R}^d} \;\; \text{(power set of } \mathbb{R}^d\text{)}$$

More on Subgradients and Subdifferentials

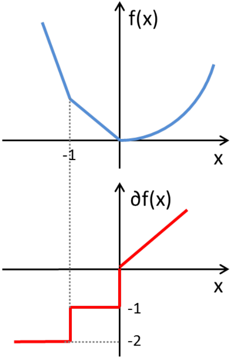
The subdifferential is a set-valued function:

$$f : \mathbb{R}^d \to \mathbb{R} \quad \Rightarrow \quad \partial f : \mathbb{R}^d \to 2^{\mathbb{R}^d} \quad \text{(power set of } \mathbb{R}^d\text{)}$$

Example:

$$f(x) = \begin{cases} -2x - 1, & x \leq -1 \\ -x, & -1 < x \leq 0 \\ x^2/2, & x > 0 \end{cases}$$

$$\partial f(x) = \begin{cases} \{-2\}, & x < -1 \\ [-2, -1], & x = -1 \\ \{-1\}, & -1 < x < 0 \\ [-1, 0], & x = 0 \\ \{x\}, & x > 0 \end{cases}$$



More on Subgradients and Subdifferentials

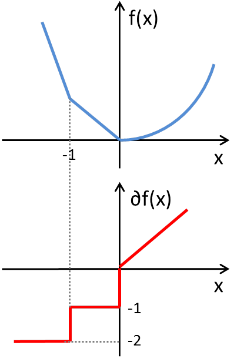
The subdifferential is a set-valued function:

$$f : \mathbb{R}^d \to \mathbb{R} \quad \Rightarrow \quad \partial f : \mathbb{R}^d \to 2^{\mathbb{R}^d} \quad \text{(power set of } \mathbb{R}^d\text{)}$$

Example:

$$f(x) = \begin{cases} -2x - 1, & x \leq -1 \\ -x, & -1 < x \leq 0 \\ x^2/2, & x > 0 \end{cases}$$

$$\partial f(x) = \begin{cases} \{-2\}, & x < -1 \\ [-2, \, -1], & x = -1 \\ \{-1\}, & -1 < x < 0 \\ [-1, \, 0], & x = 0 \\ \{x\}, & x > 0 \end{cases}$$



Fermat's Rule: $x \in \arg\min_x f(x) \;\Leftrightarrow\; 0 \in \partial f(x)$

A Key Tool: Moreau's Proximity Operators

Moreau (1962) proximity operator

$$\boxed{\widehat{x} \in \arg\min_{x} \frac{1}{2}\|x - y\|_2^2 + \psi(x) =: \text{prox}_{\psi}(y)}$$

...well defined for convex ψ, since $\| \cdot - y\|_2^2$ is coercive and strictly convex.

A Key Tool: Moreau's Proximity Operators

Moreau (1962) proximity operator

$$\boxed{\widehat{x} \in \arg\min_{x} \frac{1}{2}\|x - y\|_2^2 + \psi(x) =: \text{prox}_{\psi}(y)}$$

...well defined for convex ψ, since $\|\cdot - y\|_2^2$ is coercive and strictly convex.

Example: (seen above) $\text{prox}_{\tau|\cdot|}(y) = \text{soft}(y, \tau) = \text{sign}(y) \max\{|y| - \tau, 0\}$

A Key Tool: Moreau's Proximity Operators

Moreau (1962) proximity operator

$$\widehat{x} \in \arg\min_x \frac{1}{2}\|x - y\|_2^2 + \psi(x) =: \operatorname{prox}_\psi(y)$$

...well defined for convex ψ, since $\|\cdot - y\|_2^2$ is coercive and strictly convex.

Example: (seen above) $\operatorname{prox}_{\tau|\cdot|}(y) = \operatorname{soft}(y, \tau) = \operatorname{sign}(y) \max\{|y| - \tau, 0\}$

Block separability: $x = (x_1, ..., x_N)$ (a partition of the components of x)

$$\psi(x) = \sum_i \psi_i(x_i) \;\Rightarrow\; (\operatorname{prox}_\psi(y))_i = \operatorname{prox}_{\psi_i}(y_i)$$

Relationship with subdifferential: $z = \operatorname{prox}_\psi(y) \;\Leftrightarrow\; z - y \in \partial\psi(z)$

A Key Tool: Moreau's Proximity Operators

Moreau (1962) proximity operator

$$\widehat{x} \in \arg\min_x \frac{1}{2}\|x - y\|_2^2 + \psi(x) =: \text{prox}_\psi(y)$$

...well defined for convex ψ, since $\|\cdot - y\|_2^2$ is coercive and strictly convex.

Example: (seen above) $\text{prox}_{\tau|\cdot|}(y) = \text{soft}(y, \tau) = \text{sign}(y) \max\{|y| - \tau, 0\}$

Block separability: $x = (x_1, ..., x_N)$ (a partition of the components of x)

$$\psi(x) = \sum_i \psi_i(x_i) \;\Rightarrow\; (\text{prox}_\psi(y))_i = \text{prox}_{\psi_i}(y_i)$$

Relationship with subdifferential: $z = \text{prox}_\psi(y) \;\Leftrightarrow\; z - y \in \partial\psi(z)$

Resolvent: $z = \text{prox}_\psi(y) \;\Leftrightarrow\; 0 \in \partial\psi(z) + (z - y) \;\Leftrightarrow\; y \in (\partial\psi + I)z$

$$\text{prox}_\psi(y) = (\partial\psi + I)^{-1} y$$

Important Proximity Operators

- Soft-thresholding is the proximity operator of the ℓ_1 norm.

Important Proximity Operators

- Soft-thresholding is the proximity operator of the ℓ_1 norm.
- Consider the indicator $\iota_{\mathcal{S}}$ of a convex set $\mathcal{S}$;

$$\text{prox}_{\iota_{\mathcal{S}}}(u) = \arg\min_{x} \frac{1}{2}\|x-u\|_2^2 + \iota_{\mathcal{S}}(x) = \arg\min_{x\in\mathcal{S}} \frac{1}{2}\|x-y\|_2^2 = P_{\mathcal{S}}(u)$$

...the Euclidean projection on $\mathcal{S}$.

Important Proximity Operators

- Soft-thresholding is the proximity operator of the ℓ_1 norm.
- Consider the indicator $\iota_{\mathcal{S}}$ of a convex set $\mathcal{S}$;

$$\text{prox}_{\iota_{\mathcal{S}}}(u) = \arg\min_x \frac{1}{2}\|x-u\|_2^2 + \iota_{\mathcal{S}}(x) = \arg\min_{x\in\mathcal{S}} \frac{1}{2}\|x-y\|_2^2 = P_{\mathcal{S}}(u)$$

...the Euclidean projection on $\mathcal{S}$.

- Squared Euclidean norm (separable, smooth):

$$\text{prox}_{\tau\|\cdot\|_2^2}(y) = \arg\min_x \|x-y\|_2^2 + \tau\|x\|_2^2 = \frac{y}{1+\tau}$$

Important Proximity Operators

- Soft-thresholding is the proximity operator of the ℓ_1 norm.
- Consider the indicator $\iota_{\mathcal{S}}$ of a convex set $\mathcal{S}$;

$$\text{prox}_{\iota_{\mathcal{S}}}(u) = \arg\min_x \frac{1}{2}\|x-u\|_2^2 + \iota_{\mathcal{S}}(x) = \arg\min_{x\in\mathcal{S}} \frac{1}{2}\|x-y\|_2^2 = P_{\mathcal{S}}(u)$$

...the Euclidean projection on $\mathcal{S}$.

- Squared Euclidean norm (separable, smooth):

$$\text{prox}_{\tau\|\cdot\|_2^2}(y) = \arg\min_x \|x-y\|_2^2 + \tau\|x\|_2^2 = \frac{y}{1+\tau}$$

- Euclidean norm (not separable, nonsmooth):

$$\text{prox}_{\tau\|\cdot\|_2}(y) = \begin{cases} \frac{x}{\|x\|_2}(\|x\|_2 - \tau), & \text{if } \|x\|_2 > \tau \\ 0 & \text{if } \|x\|_2 \le \tau \end{cases}$$

More Proximity Operators

	$\phi(x)$	$\mathrm{prox}_\phi x$
i	$\iota_{[\underline{\omega},\overline{\omega}]}(x)$	$P_{[\underline{\omega},\overline{\omega}]}\, x$
ii	$\sigma_{[\underline{\omega},\overline{\omega}]}(x) = \begin{cases} \underline{\omega}x & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ \overline{\omega}x & \text{otherwise} \end{cases}$	$\mathrm{soft}_{[\underline{\omega},\overline{\omega}]}(x) = \begin{cases} x - \underline{\omega} & \text{if } x < \underline{\omega} \\ 0 & \text{if } x \in [\underline{\omega},\overline{\omega}] \\ x - \overline{\omega} & \text{if } x > \overline{\omega} \end{cases}$
iii	$\psi(x) + \sigma_{[\underline{\omega},\overline{\omega}]}(x)$ $\psi \in \Gamma_0(\mathbb{R})$ differentiable at 0 $\psi'(0) = 0$	$\mathrm{prox}_\psi\big(\mathrm{soft}_{[\underline{\omega},\overline{\omega}]}(x)\big)$
iv	$\max\{\lvert x\rvert - \omega, 0\}$	$\begin{cases} x & \text{if } \lvert x\rvert < \omega \\ \mathrm{sign}(x)\omega & \text{if } \omega \le \lvert x\rvert \le 2\omega \\ \mathrm{sign}(x)(\lvert x\rvert - \omega) & \text{if } \lvert x\rvert > 2\omega \end{cases}$
v	$\kappa\lvert x\rvert^q$	$\mathrm{sign}(x)p$, where $p \ge 0$ and $p + q\kappa p^{q-1} = \lvert x\rvert$
vi	$\begin{cases} \kappa x^2 & \text{if } \lvert x\rvert \le \omega/\sqrt{2\kappa} \\ \omega\sqrt{2\kappa}\lvert x\rvert - \omega^2/2 & \text{otherwise} \end{cases}$	$\begin{cases} x/(2\kappa+1) & \text{if } \lvert x\rvert \le \omega(2\kappa+1)/\sqrt{2\kappa} \\ x - \omega\sqrt{2\kappa}\,\mathrm{sign}(x) & \text{otherwise} \end{cases}$
vii	$\omega\lvert x\rvert + \tau\lvert x\rvert^2 + \kappa\lvert x\rvert^q$	$\mathrm{sign}(x)\mathrm{prox}_{\kappa\lvert\cdot\rvert^q/(2\tau+1)} \dfrac{\max\{\lvert x\rvert - \omega, 0\}}{2\tau+1}$
viii	$\omega\lvert x\rvert - \ln(1 + \omega\lvert x\rvert)$	$(2\omega)^{-1}\,\mathrm{sign}(x)\Big(\omega\lvert x\rvert - \omega^2 - 1 + \sqrt{\lvert\omega\lvert x\rvert - \omega^2 - 1\rvert^2 + 4\omega\lvert x\rvert}\Big)$
[illegible]	[illegible] if $x \ge 0$ $+\infty$ otherwise	$x - \omega$ if x [illegible] ω [illegible] otherwise
x	[illegible] x [illegible] 0 $+\infty$ otherwise	p^1 [illegible] where [illegible]
xi	$\begin{cases} \omega x^{-q} & \text{if } x > 0 \\ +\infty & \text{otherwise} \end{cases}$	$p > 0$ such that $p^{q+2} - xp^{q+1} = \omega q$
xii	$\begin{cases} x\ln(x) & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ +\infty & \text{otherwise} \end{cases}$	$W(e^{x-1})$, where W is the Lambert W-function
xiii	$\begin{cases} -\ln(x - \underline{\omega}) + \ln(-\underline{\omega}) & \text{if } x \in]\underline{\omega}, 0] \\ -\ln(\overline{\omega} - x) + \ln(\overline{\omega}) & \text{if } x \in]0, \overline{\omega}[\\ +\infty & \text{otherwise} \end{cases}$ $\underline{\omega} < 0 < \overline{\omega}$	$\begin{cases} \frac{1}{2}\Big(x + \underline{\omega} + \sqrt{\lvert x - \underline{\omega}\rvert^2 + 4}\Big) & \text{if } x < 1/\underline{\omega} \\ \frac{1}{2}\Big(x + \overline{\omega} - \sqrt{\lvert x - \overline{\omega}\rvert^2 + 4}\Big) & \text{if } x > 1/\overline{\omega} \\ 0 & \text{otherwise} \end{cases}$ (see Figure 1)
xiv	$\begin{cases} -\kappa\ln(x) + \tau x^2/2 + \alpha x & \text{if } x > 0 \\ +\infty & \text{otherwise} \end{cases}$	$\dfrac{1}{2(1+\tau)}\Big(x - \alpha + \sqrt{\lvert x - \alpha\rvert^2 + 4\kappa(1+\tau)}\Big)$
xv	$\begin{cases} -\kappa\ln(x) + \alpha x + \omega x^{-1} & \text{if } x > 0 \\ +\infty & \text{otherwise} \end{cases}$	$p > 0$ such that $p^3 + (\alpha - x)p^2 - \kappa p = \omega$
xvi	$\begin{cases} -\kappa\ln(x) + \omega x^q & \text{if } x > 0 \\ +\infty & \text{otherwise} \end{cases}$	$p > 0$ such that $q\omega p^q + p^2 - xp = \kappa$
xvii	$\begin{cases} -\underline{\kappa}\ln(x - \underline{\omega}) - \overline{\kappa}\ln(\overline{\omega} - x) & \text{if } x \in]\underline{\omega}, \overline{\omega}[\\ +\infty & \text{otherwise} \end{cases}$	$p \in]\underline{\omega}, \overline{\omega}[$ such that $p^3 - (\underline{\omega} + \overline{\omega} + x)p^2 +$ $(\underline{\omega}\overline{\omega} - \underline{\kappa} - \overline{\kappa} + (\underline{\omega} + \overline{\omega})x)p = \underline{\omega}\overline{\omega}x - \underline{\omega}\overline{\kappa} - \overline{\omega}\underline{\kappa}$

Many others!

(Combettes and Pesquet, 2011)

Another Key Tool: Fenchel-Legendre Conjugates

The Fenchel-Legendre conjugate of a proper convex function f — denoted by $f^* : \mathbb{R}^n \to \bar{\mathbb{R}}$ — is defined by

$$\boxed{f^*(u) = \sup_x x^T u - f(x)}$$

Another Key Tool: Fenchel-Legendre Conjugates

The Fenchel-Legendre conjugate of a proper convex function f — denoted by $f^* : \mathbb{R}^n \to \bar{\mathbb{R}}$ — is defined by

$$\boxed{f^*(u) = \sup_x x^T u - f(x)}$$

Main properties and relationship with proximity operators:

- Biconjugation: if f is convex and proper, $f^{**} = f$.

Another Key Tool: Fenchel-Legendre Conjugates

The Fenchel-Legendre conjugate of a proper convex function f — denoted by $f^* : \mathbb{R}^n \to \bar{\mathbb{R}}$ — is defined by

$$\boxed{f^*(u) = \sup_x x^T u - f(x)}$$

Main properties and relationship with proximity operators:

- Biconjugation: if f is convex and proper, $f^{**} = f$.
- Moreau's decomposition: $\text{prox}_f(u) + \text{prox}_{f^*}(u) = u$

  ...meaning that, if you know prox_f, you know prox_{f^*}, and vice-versa.

Another Key Tool: Fenchel-Legendre Conjugates

The Fenchel-Legendre conjugate of a proper convex function f — denoted by $f^* : \mathbb{R}^n \to \bar{\mathbb{R}}$ — is defined by

$$f^*(u) = \sup_x x^T u - f(x)$$

Main properties and relationship with proximity operators:

- Biconjugation: if f is convex and proper, $f^{**} = f$.
- Moreau's decomposition: $\quad \text{prox}_f(u) + \text{prox}_{f^*}(u) = u$

  ...meaning that, if you know prox_f, you know prox_{f^*}, and vice-versa.
- Conjugate of indicator: if $f(x) = \iota_C(x)$, where C is a convex set,

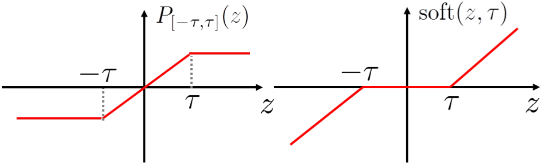
  $$f^*(u) = \sup_x x^T u - \iota_C(x) = \sup_{x \in C} x^T u \equiv \sigma_C(u)$$ (support function of C).

From Conjugates to Proximity Operators

Notice that $|u| = \sup_{x\in[-1,1]} x^T u = \sigma_{[-1,1]}(u)$, thus $|\cdot|^* = \iota_{[-1,1]}$.

Using Moreau's decomposition, we easily derive the soft-threshold:

$$\text{prox}_{\tau|\cdot|} = 1 - \text{prox}_{\iota_{[-\tau,\tau]}} = 1 - P_{[-\tau,\tau]} = \text{soft}(\cdot, \tau)$$



Conjugate of a norm: if $f(x) = \tau\|x\|_p$ then $f^* = \iota_{\{x:\|x\|_q\leq\tau\}}$,

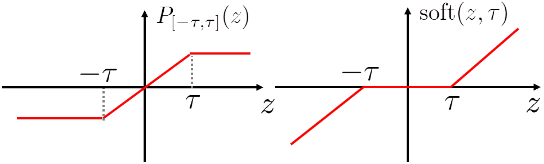
where $\frac{1}{q} + \frac{1}{p} = 1$ (a Hölder pair, or Hölder conjugates).

From Conjugates to Proximity Operators

Notice that $|u| = \sup_{x\in[-1,1]} x^T u = \sigma_{[-1,1]}(u)$, thus $|\cdot|^* = \iota_{[-1,1]}$.

Using Moreau's decomposition, we easily derive the soft-threshold:

$$\text{prox}_{\tau|\cdot|} = 1 - \text{prox}_{\iota_{[-\tau,\tau]}} = 1 - P_{[-\tau,\tau]} = \text{soft}(\cdot, \tau)$$



Conjugate of a norm: if $f(x) = \tau\|x\|_p$ then $f^* = \iota_{\{x:\|x\|_q\leq\tau\}}$,

where $\frac{1}{q} + \frac{1}{p} = 1$ (a Hölder pair, or Hölder conjugates).

That is, $\|\cdot\|_p$ and $\|\cdot\|_q$ are dual norms:

$$\|z\|_q = \sup\{x^T z : \|x\|_p \leq 1\} = \sup_{x\in B_p(1)} x^T z = \sigma_{B_p(1)}(z)$$

From Conjugates to Proximity Operators

- Proximity of norm:

$$\boxed{\text{prox}_{\tau\|\cdot\|_p} = I - P_{B_q(\tau)}}$$

where $B_q(\tau) = \{x : \|x\|_q \le \tau\}$ and $\frac{1}{q} + \frac{1}{p} = 1$.

From Conjugates to Proximity Operators

- Proximity of norm:

$$\boxed{\text{prox}_{\tau\|\cdot\|_p} = I - P_{B_q(\tau)}}$$

where $B_q(\tau) = \{x : \|x\|_q \leq \tau\}$ and $\frac{1}{q} + \frac{1}{p} = 1$.

- Example: computing $\text{prox}_{\|\cdot\|_\infty}$ (notice ℓ_∞ is not separable):

Since $\frac{1}{\infty} + \frac{1}{1} = 1$,

$$\text{prox}_{\tau\|\cdot\|_\infty} = I - P_{B_1(\tau)}$$

... the proximity operator of ℓ_∞ norm is the residual of the projection on an ℓ_1 ball.

From Conjugates to Proximity Operators

- Proximity of norm:

$$\boxed{\text{prox}_{\tau\|\cdot\|_p} = I - P_{B_q(\tau)}}$$

where $B_q(\tau) = \{x : \|x\|_q \leq \tau\}$ and $\frac{1}{q} + \frac{1}{p} = 1$.

- Example: computing $\text{prox}_{\|\cdot\|_\infty}$ (notice ℓ_∞ is not separable):

Since $\frac{1}{\infty} + \frac{1}{1} = 1$,

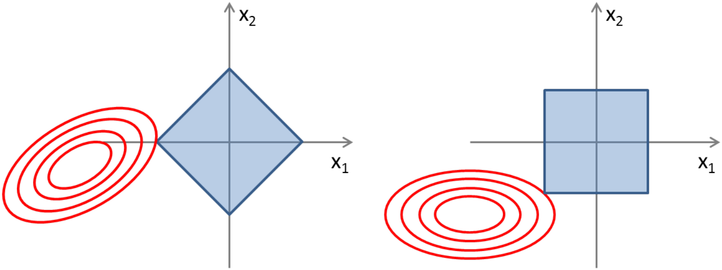
$$\text{prox}_{\tau\|\cdot\|_\infty} = I - P_{B_1(\tau)}$$

... the proximity operator of ℓ_∞ norm is the residual of the projection on an ℓ_1 ball.

- Projection on ℓ_1 ball has no closed form, but there are efficient (linear cost) algorithms (Brucker, 1984), (Maculan and de Paula, 1989).

Geometry and Effect of $\text{prox}_{\ell_\infty}$

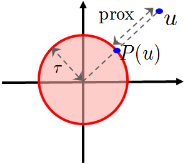
Whereas ℓ_1 promotes sparsity, ℓ_∞ promotes equality (in absolute value).



From Conjugates to Proximity Operators

The dual of the ℓ_2 norm is the ℓ_2 norm.

$$\text{prox}_{\tau\|\cdot\|_2}(u) = u - P_{\{x:\|x\|_2\leq\tau\}}(u)$$



$$= u - \begin{cases} u & \Leftarrow \quad \|u\|_2 \leq \tau \\ \tau\, u/\|u\|_2 & \Leftarrow \quad \|u\|_2 > \tau \end{cases}$$

$$= \frac{u}{\|u\|_2}\max\{0, \|u\|_2 - \tau\}$$

vector soft thresholding

Group Norms and their Prox Operators

Group-norm regularizer: $\psi(x) = \sum_{m=1}^{M} \lambda_m \|x_{G_m}\|_p$.

In the non-overlapping case ($G_1, ..., G_m$ is a partition of $\{1, ..., n\}$), simply use separability:

$$\big(\text{prox}_\psi(u)\big)_{G_m} = \text{prox}_{\lambda_m \|\cdot\|_p}\big(u_{G_m}\big).$$

Group Norms and their Prox Operators

Group-norm regularizer: $\psi(x) = \sum_{m=1}^{M} \lambda_m \|x_{G_m}\|_p$.

In the non-overlapping case ($G_1, ..., G_m$ is a partition of $\{1, ..., n\}$), simply use separability:

$$\big(\text{prox}_\psi(u)\big)_{G_m} = \text{prox}_{\lambda_m \|\cdot\|_p}\big(u_{G_m}\big).$$

In the tree-structured case, can get a complete ordering of the groups: $G_1 \preceq G_2 ... \preceq G_M$, where $(G \preceq G') \Leftrightarrow (G \subset G')$ or $(G \cap G' = \emptyset)$.

Group Norms and their Prox Operators

Group-norm regularizer: $\psi(x) = \sum_{m=1}^{M} \lambda_m \|x_{G_m}\|_p$.

In the non-overlapping case ($G_1, ..., G_m$ is a partition of $\{1, ..., n\}$), simply use separability:

$$\big(\text{prox}_\psi(u)\big)_{G_m} = \text{prox}_{\lambda_m\|\cdot\|_p}\big(u_{G_m}\big).$$

In the tree-structured case, can get a complete ordering of the groups: $G_1 \preceq G_2 ... \preceq G_M$, where $(G \preceq G') \Leftrightarrow (G \subset G')$ or $(G \cap G' = \emptyset)$.

Define $\Pi_m : \mathbb{R}^n \to \mathbb{R}^N$:

$$(\Pi_m(u))_{G_m} = \text{prox}_{\lambda_m\|\cdot\|_p}(u_{G_m}),$$
$$(\Pi_m(u))_{\bar{G}_m} = u_{\bar{G}_m}, \quad \text{where } \bar{G}_m = \{1, ..., n\} \setminus G_m$$

Then

$$\text{prox}_\psi = \Pi_M \circ \cdots \circ \Pi_2 \circ \Pi_1$$

...only valid for $p \in \{1, 2, \infty\}$ (Jenatton et al., 2011).

Matrix Nuclear Norm and its Prox Operator

- Recall the trace/nuclear norm: $\|X\|_* = \displaystyle\sum_{i=1}^{\min\{m,n\}} \sigma_i$.
- The dual of a Schatten p-norm is a Schatten q-norm, with $\frac{1}{q} + \frac{1}{p} = 1$. Thus, the dual of the nuclear norm is the spectral norm:

$$\|X\|_\infty = \max\left\{\sigma_1, ..., \sigma_{\min\{m,n\}}\right\}.$$

- If $Y = U\Lambda V^T$ is the SVD of Y, we have

$$\begin{aligned}\text{prox}_{\tau\|\cdot\|_*}(Y) &= U\Lambda V^T - P_{\{X:\max\{\sigma_1,...,\sigma_{\min\{m,n\}}\}\leq\tau\}}(U\Lambda V^T)\\ &= U\,\text{soft}\big(\Lambda, \tau\big)\,V^T.\end{aligned}$$

Atomic Norms: A Unified View

vectors			matrices		
norm	prox	atomic set	norm	prox	atomic set
ℓ_1 $\|x\|_1$	component soft thresholding	$\mathcal{A} = \{\pm e_i\}$ $\lvert\mathcal{A}\rvert = 2N$	nuclear $\|X\|_*$	singular value thresholding	$\mathcal{A} =$ set of all rank 1, norm 1 matrices
ℓ_∞ $\|x\|_\infty$	residual of projection on ℓ_1 ball	$\mathcal{A} = \{\pm 1\}^N$ $\lvert\mathcal{A}\rvert = 2^N$	spectral $\|X\|_2$	residual of s.v. proj. on ℓ_1 ball	$\mathcal{A} =$ set of all orthogonal matrices
ℓ_2 $\|x\|_2$	vector soft thresholding	$\mathcal{A} =$ set of all vectors with norm 1 $\lvert\mathcal{A}\rvert = \infty$	Frobenius $\|X\|_F$	matrix soft threshold.	$\mathcal{A} =$ all matrices of unit Frobenius norm.

Another Use of Fenchel-Legendre Conjugates

- The original problem: $\min_x f(x) + \psi(x)$

Another Use of Fenchel-Legendre Conjugates

- The original problem: $\min_x f(x) + \psi(x)$
- Often this has the form: $\min_x g(Ax) + \psi(x)$

Another Use of Fenchel-Legendre Conjugates

- The original problem: $\min_x f(x) + \psi(x)$
- Often this has the form: $\min_x g(Ax) + \psi(x)$
- Using the definition of conjugate $g(Ax) = \sup_u u^T Ax - g^*(u)$

$$\begin{aligned}
\min_x g(Ax) + \psi(x) &= \inf_x \sup_u u^T Ax - g^*(u) + \psi(x) \\
&= \sup_u (-g^*(u)) + \inf_x \; u^T Ax + \psi(x) \\
&= \sup_u (-g^*(u)) - \underbrace{\sup_x \; -x^T A^T u - \psi(x)}_{\psi^*(-A^T u)} \\
&= -\inf_u g^*(u) + \psi^*(-A^T u)
\end{aligned}$$

Another Use of Fenchel-Legendre Conjugates

- The original problem: $\min_x f(x) + \psi(x)$
- Often this has the form: $\min_x g(Ax) + \psi(x)$
- Using the definition of conjugate $g(Ax) = \sup_u u^T Ax - g^*(u)$

$$\begin{aligned}
\min_x g(Ax) + \psi(x) &= \inf_x \sup_u u^T Ax - g^*(u) + \psi(x) \\
&= \sup_u (-g^*(u)) + \inf_x \; u^T Ax + \psi(x) \\
&= \sup_u (-g^*(u)) - \underbrace{\sup_x \; -x^T A^T u - \psi(x)}_{\psi^*(-A^T u)} \\
&= -\inf_u g^*(u) + \psi^*(-A^T u)
\end{aligned}$$

- The problem $\inf_u g^*(u) + \psi^*(-A^T u)$ is sometimes easier to handle.

Basic Proximal-Gradient Algorithm

Use basic structure:

$$x_k = \arg\min_x \|x - \Phi(x_k)\|_2^2 + \psi(x).$$

with $\Phi(x_k)$ a simple gradient descent step, thus

$$\boxed{x_{k+1} = \text{prox}_{\alpha_k \psi}\big(x_k - \alpha_k \nabla f(x_k)\big)}$$

This approach goes by different names, such as

- "proximal gradient algorithm" (PGA),
- "iterative shrinkage/thresholding" (IST or ISTA),
- "forward-backward splitting" (FBS)

It has been proposed in different communities: optimization, PDEs, convex analysis, signal processing, machine learning.

Convergence of the Proximal-Gradient Algorithm

- Basic algorithm: $x_{k+1} = \text{prox}_{\alpha_k \psi}\big(x_k - \alpha_k \nabla f(x_k)\big)$

Convergence of the Proximal-Gradient Algorithm

- Basic algorithm: $x_{k+1} = \text{prox}_{\alpha_k \psi}\big(x_k - \alpha_k \nabla f(x_k)\big)$

- generalized (possibly inexact) version:

$$x_{k+1} = (1-\lambda_k)x_k + \lambda_k\Big(\text{prox}_{\alpha_k\psi}\big(x_k - \alpha_k \nabla f(x_k) + b_k\big) + a_k\Big)$$

where a_k and b_k are "errors" in computing the prox and the gradient; λ_k is an over-relaxation parameter.

Convergence of the Proximal-Gradient Algorithm

- Basic algorithm: $x_{k+1} = \text{prox}_{\alpha_k \psi}\big(x_k - \alpha_k \nabla f(x_k)\big)$

- generalized (possibly inexact) version:

$$x_{k+1} = (1 - \lambda_k) x_k + \lambda_k \Big(\text{prox}_{\alpha_k \psi}\big(x_k - \alpha_k \nabla f(x_k) + b_k\big) + a_k \Big)$$

  where a_k and b_k are "errors" in computing the prox and the gradient; λ_k is an over-relaxation parameter.

- Convergence is guaranteed (Combettes and Wajs, 2006) if

  ✓ $0 < \inf \alpha_k \leq \sup \alpha_k < \frac{2}{L}$

  ✓ $\lambda_k \in (0, 1]$, with $\inf \lambda_k > 0$

  ✓ $\sum_k^\infty \|a_k\| < \infty$ and $\sum_k^\infty \|b_k\| < \infty$

Proximal-Gradient Algorithm: Quadratic Case

- Consider the quadratic case (of great interest): $f(x) = \frac{1}{2}\|B\,x - b\|_2^2$.

Proximal-Gradient Algorithm: Quadratic Case

- Consider the quadratic case (of great interest): $f(x) = \frac{1}{2}\|B\,x - b\|_2^2$.
- Here, $\nabla f(x) = B^T(B\,x - b)$ and the IST/PGA/FBS algorithm is

$$x_{k+1} = \text{prox}_{\alpha_k \psi}\big(x_k - \alpha_k B^T(B\,x - b)\big)$$

requires only matrix-vector multiplications with B and B^T.

Proximal-Gradient Algorithm: Quadratic Case

- Consider the quadratic case (of great interest): $f(x) = \frac{1}{2}\|Bx - b\|_2^2$.
- Here, $\nabla f(x) = B^T(Bx - b)$ and the IST/PGA/FBS algorithm is

$$x_{k+1} = \text{prox}_{\alpha_k \psi}\big(x_k - \alpha_k B^T(Bx - b)\big)$$

requires only matrix-vector multiplications with B and B^T.

Very important in signal/image processing, where fast algorithms exist to compute these products (*e.g.* FFT, DWT)

Proximal-Gradient Algorithm: Quadratic Case

- Consider the quadratic case (of great interest): $f(x) = \frac{1}{2}\|Bx - b\|_2^2$.
- Here, $\nabla f(x) = B^T(Bx - b)$ and the IST/PGA/FBS algorithm is

$$x_{k+1} = \text{prox}_{\alpha_k \psi}\big(x_k - \alpha_k B^T(Bx - b)\big)$$

  requires only matrix-vector multiplications with B and B^T.

  Very important in signal/image processing, where fast algorithms exist to compute these products (*e.g.* FFT, DWT)

- In this case, some more refined convergence results are available.

Proximal-Gradient Algorithm: Quadratic Case

- Consider the quadratic case (of great interest): $f(x) = \frac{1}{2}\|B\,x - b\|_2^2$.
- Here, $\nabla f(x) = B^T(B\,x - b)$ and the IST/PGA/FBS algorithm is

$$x_{k+1} = \text{prox}_{\alpha_k \psi}\big(x_k - \alpha_k B^T(B\,x - b)\big)$$

  requires only matrix-vector multiplications with B and B^T.

  Very important in signal/image processing, where fast algorithms exist to compute these products (*e.g.* FFT, DWT)

- In this case, some more refined convergence results are available.
- Even more refined results are available if $\psi(x) = \tau\|x\|_1$

More on IST/FBS/PGA for the ℓ_2-ℓ_1 Case

- Problem: $\widehat{x} \in G = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2}\|B\,x - b\|_2^2 + \tau\|x\|_1$ (recall $B^T B \preceq LI$)
- IST/FBS/PGA becomes $\boxed{x_{k+1} = \text{soft}\big(x_k - \alpha B^T(B\,x - b), \alpha\tau\big)}$ with $\alpha < 2/L$.

More on IST/FBS/PGA for the ℓ_2-ℓ_1 Case

- Problem: $\widehat{x} \in G = \arg\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Bx - b\|_2^2 + \tau\|x\|_1$ (recall $B^T B \preceq LI$)

- IST/FBS/PGA becomes $\boxed{x_{k+1} = \text{soft}\big(x_k - \alpha B^T(Bx - b), \alpha\tau\big)}$ with $\alpha < 2/L$.

- The zero set: $\mathcal{Z} \subseteq \{1, ..., n\}$: $\widehat{x} \in G \Rightarrow \widehat{x}_{\mathcal{Z}} = 0$

- Zeros are found in a finite number of iterations (Hale et al., 2008): after a finite number of iterations $(x_k)_{\mathcal{Z}} = 0$.

More on IST/FBS/PGA for the ℓ_2-ℓ_1 Case

- Problem: $\widehat{x} \in G = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2}\|B\,x - b\|_2^2 + \tau\|x\|_1$ (recall $B^T B \preceq LI$)
- IST/FBS/PGA becomes $\boxed{x_{k+1} = \text{soft}\big(x_k - \alpha B^T(B\,x - b), \alpha\tau\big)}$ with $\alpha < 2/L$.
- The zero set: $\mathcal{Z} \subseteq \{1, ..., n\}$: $\widehat{x} \in G \Rightarrow \widehat{x}_{\mathcal{Z}} = 0$
- Zeros are found in a finite number of iterations (Hale et al., 2008): after a finite number of iterations $(x_k)_{\mathcal{Z}} = 0$.
- After that, if $B_{\mathcal{Z}}^T B_{\mathcal{Z}} \succeq \mu I$, with $\mu > 0$ (thus $\kappa(B_{\mathcal{Z}}^T B_{\mathcal{Z}}) = L/\mu < \infty$), we have linear convergence

$$\|x_{k+1} - \widehat{x}\|_2 \leq \frac{1-\kappa}{1+\kappa}\|x_k - \widehat{x}\|_2$$

for the optimal choice $\alpha = 2/(L+\mu)$ (see unconstrained theory).

Heavy Ball Acceleration: FISTA

- FISTA (*fast iterative shrinkage-thresholding algorithm*) is heavy-ball-type acceleration of IST (based on Nesterov (1983)) (Beck and Teboulle, 2009).

Initialize: Choose $\alpha \leq 1/L$, x_0; set $y_1 = x_0$, $t_1 = 1$;

Iterate:

$$x_k \leftarrow \text{prox}_{\tau\alpha\psi}\big(y_k - \alpha \nabla f(y_k)\big);$$

$$t_{k+1} \leftarrow \tfrac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right);$$

$$y_{k+1} \leftarrow x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1}).$$

Heavy Ball Acceleration: FISTA

- FISTA (*fast iterative shrinkage-thresholding algorithm*) is heavy-ball-type acceleration of IST (based on Nesterov (1983)) (Beck and Teboulle, 2009).

Initialize: Choose $\alpha \le 1/L$, x_0; set $y_1 = x_0$, $t_1 = 1$;

Iterate:
$$x_k \leftarrow \text{prox}_{\tau\alpha\psi}\big(y_k - \alpha\nabla f(y_k)\big);$$
$$t_{k+1} \leftarrow \tfrac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right);$$
$$y_{k+1} \leftarrow x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1}).$$

- Acceleration:

$$\text{FISTA:}\ \ f(x_k) - f(\widehat{x}) \sim O\left(\frac{1}{k^2}\right) \qquad \text{IST:}\ \ f(x_k) - f(\widehat{x}) \sim O\left(\frac{1}{k}\right).$$

Heavy Ball Acceleration: FISTA

- FISTA (*fast iterative shrinkage-thresholding algorithm*) is heavy-ball-type acceleration of IST (based on Nesterov (1983)) (Beck and Teboulle, 2009).

Initialize: Choose $\alpha \leq 1/L$, x_0; set $y_1 = x_0$, $t_1 = 1$;

Iterate:
$$x_k \leftarrow \text{prox}_{\tau\alpha\psi}\big(y_k - \alpha\nabla f(y_k)\big);$$
$$t_{k+1} \leftarrow \tfrac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right);$$
$$y_{k+1} \leftarrow x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1}).$$

- Acceleration:

$$\text{FISTA: } f(x_k) - f(\widehat{x}) \sim O\left(\frac{1}{k^2}\right) \qquad \text{IST: } f(x_k) - f(\widehat{x}) \sim O\left(\frac{1}{k}\right).$$

- When L is not known, increase an estimate of L until it's big enough.

Heavy Ball Acceleration: TwIST

- TwIST (*two-step iterative shrinkage-thresholding* (Bioucas-Dias and Figueiredo, 2007)) is a heavy-ball-type acceleration of IST, for

$$\min_{x} \tfrac{1}{2}\|Bx - b\|_2^2 + \tau\psi(x)$$

- Iterations (with $\alpha < 2/L$)

$$x_{k+1} = (\gamma - \beta)\, x_k + (1 - \gamma) x_{k-1} + \beta \, \text{prox}_{\alpha\tau\psi}\big(x_k - \alpha\, B^T(Bx - b)\big)$$

Heavy Ball Acceleration: TwIST

- TwIST (*two-step iterative shrinkage-thresholding* (Bioucas-Dias and Figueiredo, 2007)) is a heavy-ball-type acceleration of IST, for

$$\min_x \tfrac{1}{2}\|B\,x - b\|_2^2 + \tau\psi(x)$$

- Iterations (with $\alpha < 2/L$)

$$x_{k+1} = (\gamma - \beta)\,x_k + (1-\gamma)x_{k-1} + \beta\,\mathrm{prox}_{\alpha\tau\psi}\big(x_k - \alpha\,B^T(B\,x - b)\big)$$

- Analysis in the strongly convex case: $\mu I \preceq B^T B \preceq LI$, with $\mu > 0$. Condition number $\kappa = L/\mu < \infty$.

Heavy Ball Acceleration: TwIST

- TwIST (*two-step iterative shrinkage-thresholding* (Bioucas-Dias and Figueiredo, 2007)) is a heavy-ball-type acceleration of IST, for

$$\min_x \tfrac{1}{2}\|Bx - b\|_2^2 + \tau\psi(x)$$

- Iterations (with $\alpha < 2/L$)

$$x_{k+1} = (\gamma - \beta)\, x_k + (1-\gamma)x_{k-1} + \beta\, \mathrm{prox}_{\alpha\tau\psi}\big(x_k - \alpha\, B^T(Bx - b)\big)$$

- Analysis in the strongly convex case: $\mu I \preceq B^T B \preceq LI$, with $\mu > 0$. Condition number $\kappa = L/\mu < \infty$.
- Optimal parameters: $\gamma = \rho^2 + 1$, $\beta = \frac{2\alpha}{\mu+L}$, where $\rho = \frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}$, yield linear convergence

$$\|x_{k+1} - \widehat{x}\|_2 \leq \frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\|x_k - \widehat{x}\|_2$$

Heavy Ball Acceleration: TwIST

- TwIST (*two-step iterative shrinkage-thresholding* (Bioucas-Dias and Figueiredo, 2007)) is a heavy-ball-type acceleration of IST, for

$$\min_x \tfrac{1}{2}\|Bx - b\|_2^2 + \tau\psi(x)$$

- Iterations (with $\alpha < 2/L$)

$$x_{k+1} = (\gamma - \beta)\, x_k + (1-\gamma) x_{k-1} + \beta\, \text{prox}_{\alpha\tau\psi}\big(x_k - \alpha\, B^T(Bx - b)\big)$$

- Analysis in the strongly convex case: $\mu I \preceq B^T B \preceq LI$, with $\mu > 0$. Condition number $\kappa = L/\mu < \infty$.

- Optimal parameters: $\gamma = \rho^2 + 1$, $\beta = \frac{2\alpha}{\mu + L}$, where $\rho = \frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}$, yield linear convergence

$$\|x_{k+1} - \widehat{x}\|_2 \leq \frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\|x_k - \widehat{x}\|_2 \quad \left(\text{versus } \frac{1-\kappa}{1+\kappa} \text{ for IST}\right)$$

Illustration of the TwIST Acceleration

original

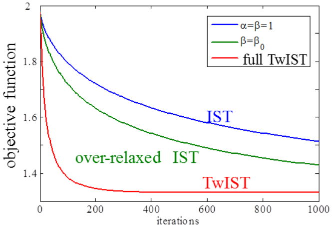
Blurred (B), 9x9, 40db noise

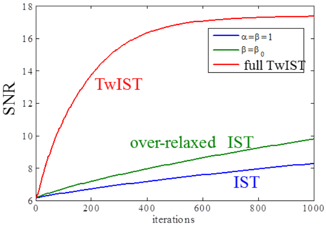
restored

$$\widehat{x} \in \arg\min_{x\in\mathbb{R}^n} \frac{1}{2}\|B\Psi x - u\|_2^2 + \tau\|x\|_1$$

representation coefficients

dictionary (e.g, wavelet basis, frame, ...)





Acceleration via Larger Steps: SpaRSA

- The standard step-size $\alpha_k \leq \frac{2}{L}$ in IST too timid

Acceleration via Larger Steps: SpaRSA

- The standard step-size $\alpha_k \leq \frac{2}{L}$ in IST too timid
- The SpARSA (sparse reconstruction by separable approximation) framework proposes bolder choices of α_k (Wright et al., 2009):
  - ✓ Barzilai-Borwein, to mimic Newton steps.
  - ✓ keep increasing α_k until monotonicity is violated: backtrack.

Acceleration via Larger Steps: SpaRSA

- The standard step-size $\alpha_k \leq \frac{2}{L}$ in IST too timid
- The SpARSA (sparse reconstruction by separable approximation) framework proposes bolder choices of α_k (Wright et al., 2009):
  - ✓ Barzilai-Borwein, to mimic Newton steps.
  - ✓ keep increasing α_k until monotonicity is violated: backtrack.
- Convergence to critical points (minima in the convex case) is guaranteed for a safeguarded version: ensure sufficient decrease w.r.t. the worst value in previous M iterations.

Another Approach: Gradient Projection

- $\min_x \frac{1}{2}\|Bx - b\|_2^2 + \tau\|x\|_1$ can be written as a standard QP:

$$\min_{u,v} \frac{1}{2}\|B(u-v) - b\|_2^2 + \tau u^T 1 + \tau u^T 1 \quad \text{s.t. } u \geq 0,\ v \geq 0,$$

where $u_i = \max\{0, x_i\}$ and $v_i = \max\{0, -x_i\}$.

Another Approach: Gradient Projection

- $\min_x \frac{1}{2}\|Bx - b\|_2^2 + \tau\|x\|_1$ can be written as a standard QP:

$$\min_{u,v} \frac{1}{2}\|B(u-v) - b\|_2^2 + \tau u^T 1 + \tau u^T 1 \quad \text{s.t. } \; u \geq 0, \; v \geq 0,$$

where $u_i = \max\{0, x_i\}$ and $v_i = \max\{0, -x_i\}$.

- With $z = \begin{bmatrix} u \\ v \end{bmatrix}$, problem can be written in canonical form

$$\min_z \frac{1}{2} z^T Q z + c^T z \quad \text{s.t. } \; z \geq 0$$

Another Approach: Gradient Projection

- $\min_x \frac{1}{2}\|B\,x - b\|_2^2 + \tau\|x\|_1$ can be written as a standard QP:

$$\min_{u,v} \frac{1}{2}\|B(u-v) - b\|_2^2 + \tau u^T 1 + \tau u^T 1 \quad \text{s.t.} \;\; u \geq 0, \;\; v \geq 0,$$

  where $u_i = \max\{0, x_i\}$ and $v_i = \max\{0, -x_i\}$.

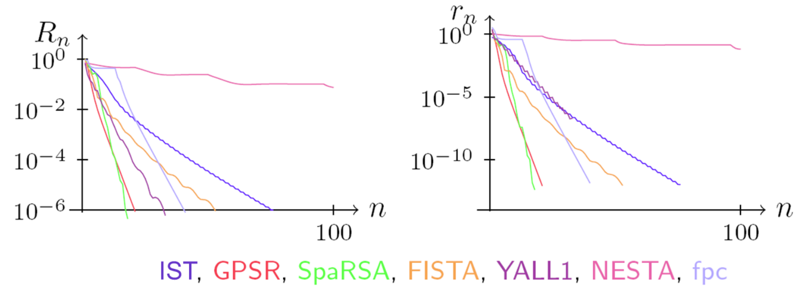
- With $z = \begin{bmatrix} u \\ v \end{bmatrix}$, problem can be written in canonical form

$$\min_{z} \frac{1}{2} z^T Q\, z + c^T z \quad \text{s.t.} \;\; z \geq 0$$

- Solving this problem with projected gradient using Barzilai-Borwein steps: GPSR (gradient projection for sparse reconstruction) (Figueiredo et al., 2007).

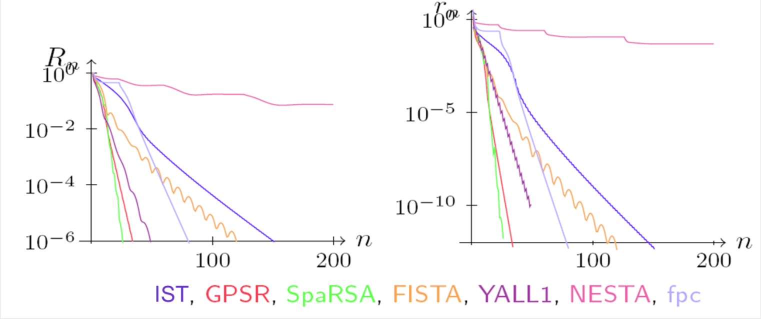
- Lorenz (2011) proposed a way of generating problem instances with known solution $\widehat{x}$: useful for speed comparison.
- Define: $R_k = \frac{\|x_k - \widehat{x}\|_2}{\|\widehat{x}\|_2}$ and $r_k = \frac{L(x_k) - L(\widehat{x})}{L(\widehat{x})}$ (where $L(x) = f(x) + \tau\psi(x)$).

Typical CS example: $\mathbf{A} = [\mathbf{I}\ \mathbf{U}]$ (512 x 1024), $\widehat{\mathbf{x}}$ has 80 non-zeros, $\tau = 0.1$



More Speed Comparisons

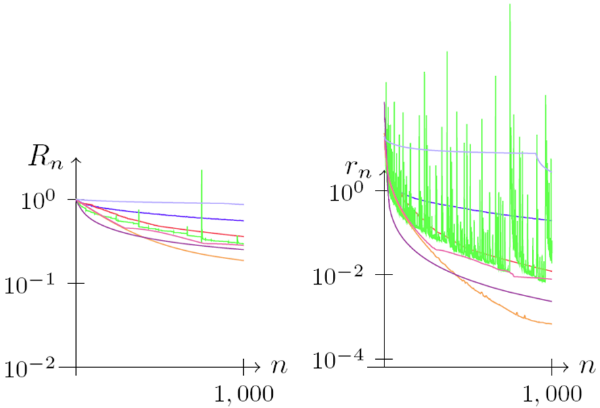
Typical CS example: $\mathbf{A} = [\mathbf{I}\ \mathbf{U}\ \mathbf{R}]$ (512 x 1536), $\widehat{\mathbf{x}}$ has 120 non-zeros, $\tau = 0.1$



Even More Speed Comparisons

A difficult problem: $\mathbf{A}$ is very coherent, τ is small $\tau = 10^{-3}$

All the solvers struggle...



IST, GPSR, SpaRSA, FISTA, YALL1, NESTA, fpc

Acceleration by Continuation

- IST/FBS/PGA can be very slow if τ is very small and/or f is poorly conditioned.

Acceleration by Continuation

- IST/FBS/PGA can be very slow if τ is very small and/or f is poorly conditioned.

- A very simple acceleration strategy: continuation/homotopy

Initialization: Set $\tau_0 \gg \tau$, starting point $\bar{x}$, factor $\sigma \in (0, 1)$, and $k = 0$.

Iterations: Find approx solution $x(\tau_k)$ of $\min_x \; f(x) + \tau_k \psi(x)$, starting from $\bar{x}$;

if $\tau_k = \tau$ STOP;

Set $\tau_{k+1} \leftarrow \max(\tau, \sigma\tau_k)$ and $\bar{x} \leftarrow x(\tau_k)$;

Acceleration by Continuation

- IST/FBS/PGA can be very slow if τ is very small and/or f is poorly conditioned.
- A very simple acceleration strategy: continuation/homotopy

Initialization: Set $\tau_0 \gg \tau$, starting point $\bar{x}$, factor $\sigma \in (0,1)$, and $k = 0$.

Iterations: Find approx solution $x(\tau_k)$ of $\min_x f(x) + \tau_k \psi(x)$, starting from $\bar{x}$;

if $\tau_k = \tau$ STOP;

Set $\tau_{k+1} \leftarrow \max(\tau, \sigma\tau_k)$ and $\bar{x} \leftarrow x(\tau_k)$;

- Often the solution path $x(\tau)$, for a range of values of τ is desired, anyway (e.g., within an outer method to choose an optimal τ)

Acceleration by Continuation

- IST/FBS/PGA can be very slow if τ is very small and/or f is poorly conditioned.

- A very simple acceleration strategy: continuation/homotopy

Initialization: Set $\tau_0 \gg \tau$, starting point $\bar{x}$, factor $\sigma \in (0,1)$, and $k = 0$.

Iterations: Find approx solution $x(\tau_k)$ of $\min_x f(x) + \tau_k \psi(x)$, starting from $\bar{x}$;
if $\tau_k = \tau$ STOP;
Set $\tau_{k+1} \leftarrow \max(\tau, \sigma\tau_k)$ and $\bar{x} \leftarrow x(\tau_k)$;

- Often the solution path $x(\tau)$, for a range of values of τ is desired, anyway (e.g., within an outer method to choose an optimal τ)

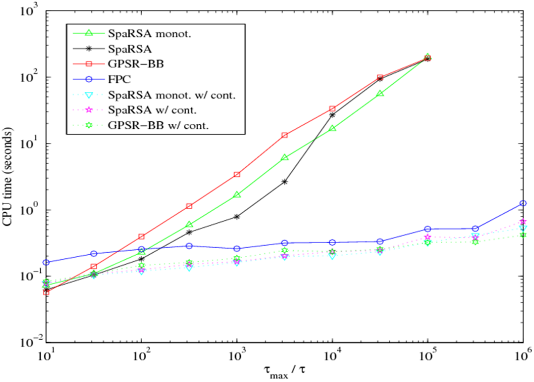
- Shown to be very effective in practice (Hale et al., 2008; Wright et al., 2009). Recently analyzed by Xiao and Zhang (2012).

Acceleration by Continuation: An Example

Classical sparse reconstruction problem (Wright et al., 2009)

$$\widehat{x} \in \arg\min_x \tfrac{1}{2}\|Bx - b\|_2^2 + \tau\|x\|_1$$

with $B \in \mathbb{R}^{1024\times 4096}$ (thus $x \in \mathbb{R}^{4096}$ and $b \in \mathbb{R}^{1024}$).



A Final Touch: Debiasing

Consider problems of the form $\widehat{x} \in \arg\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Bx - b\|_2^2 + \tau\|x\|_1$

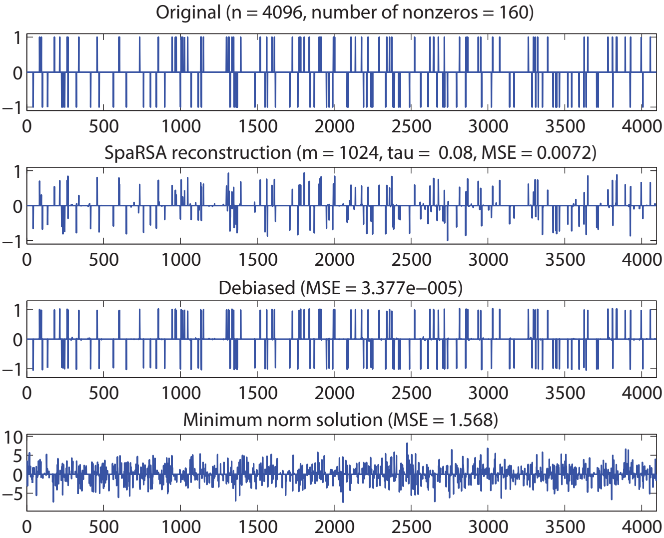
Often, the original goal was to minimize the quadratic term, after the support of x had been found. But the ℓ_1 term can cause the nonzero values of x_i to be "suppressed."

Debiasing:

- ✓ find the zero set (complement of the support of $\widehat{x}$): $\mathcal{Z}(\widehat{x}) = \{1, ..., n\} \setminus \text{supp}(\widehat{x})$.
- ✓ solve $\min_x \|Bx - b\|_2^2$ s.t. $x_{\mathcal{Z}(\widehat{x})} = 0$. (Fix the zeros and solve an unconstrained problem over the support.)

  Often, this problem has to be solved using an algorithm that only involves products by B and B^T, since this matrix cannot be partitioned.

Effect of Debiasing



Example: Matrix Recovery (Toh and Yun, 2010)

$$\widehat{M} \in \arg \min_{M \in \mathbb{R}^{n \times n}} \frac{1}{2} \|\Phi(M) - U\|_F^2 + \mu \|M\|_*$$

linear operator

...its adjoint

The proximal algorithm (IST) is as before:

$$X_{k+1} = \text{svt}_{\mu\,\beta_k}\Big(X_k - \beta_k\, \Phi^*\big(\Phi(X_k) - U\big)\Big)$$

Matrix completion: $\Phi(X) = X_\Omega$ (subset of entries) $|\Omega| = p$

Unknown M				IST			APG (FISTA)		
n/r	p	p/d_r	μ	iter	#sv	error	iter	#sv	error
100/10	5666	3	8.21e-03	7723	61	1.88e-01	655	13	1.06e-03
200/10	15665	4	1.05e-02	12180	96	2.45e-01	812	12	1.02e-03
500/10	49471	5	1.21e-02	10900	203	5.91e-01	1132	16	7.63e-04

Unknown M				continuation			APG + continuation		
n/r	p	p/d_r	μ	iter	#sv	error	iter	#sv	error
100/10	5666	3	8.21e-03	429	32	1.06e-03	74	10	1.46e-04
200/10	15665	4	1.05e-02	278	49	4.38e-04	73	10	1.02e-04
500/10	49471	5	1.21e-02	484	125	5.50e-04	72	10	8.06e-05

...the importance of acceleration!

Conditional Gradient

Also known as "Frank-Wolfe" after the authors who devised it in the 1950s. Later analysis by Dunn (around 1990). Suddenly a topic of enormous renewed interest; see for example (Jaggi, 2013).

$$\min_{x \in \Omega} f(x),$$

where f is a convex function and Ω is a closed, bounded, convex set.

Start at $x_0 \in \Omega$. At iteration k:

$$v_k := \arg\min_{v \in \Omega} v^T \nabla f(x_k);$$

$$x_{k+1} := (1 - \alpha_k)\, x_k + \alpha_k\, v_k, \quad \alpha_k = \frac{2}{k+2}.$$

- Potentially useful when it is easy to minimize a linear function over the *original* constraint set Ω;
- Admits an elementary convergence theory: $1/k$ sublinear rate.
- Same convergence theory holds if we use a line search for α_k.

Conditional Gradient for Atomic-Norm Constraints

Conditional Gradient is particularly useful for optimization over atomic-norm constraints.

$$\min f(x) \text{ s.t. } \|x\|_{\mathcal{A}} \le \tau.$$

Reminder: Given the set of atoms $\mathcal{A}$ (possibly infinite) we have

$$\|x\|_{\mathcal{A}} := \inf \left\{ \sum_{a \in \mathcal{A}} c_a \, : \, x = \sum_{a \in \mathcal{A}} c_a a, \; c_a \ge 0 \right\}.$$

The search direction v_k is $\tau \bar{a}_k$, where

$$\bar{a}_k := \arg\min_{a \in \mathcal{A}} \langle a, \nabla f(x_k) \rangle.$$

That is, we seek the atom that lines up best with the negative gradient direction $-\nabla f(x_k)$.

Generating Atoms

We can think of each step as the "addition of a new atom to the basis." Note that x_k is expressed in terms of $\{\bar{a}_0, \bar{a}_1, \dots, \bar{a}_k\}$.

If few iterations are needed to find a solution of acceptable accuracy, then we have an approximate solution that's represented in terms of few atoms, that is, sparse or compactly represented.

For many atomic sets $\mathcal{A}$ of interest, the new atom can be found cheaply.

Example: For the constraint $\|x\|_1 \le \tau$, the atoms are $\{\pm e_i \, : \, i = 1, 2, \dots, n\}$. if i_k is the index at which $|[\nabla f(x_k)]_i|$ attains its maximum, we have

$$\bar{a}_k = -\text{sign}([\nabla f(x_k)]_{i_k}) \, e_{i_k}$$

Example: For the constraint $\|x\|_\infty \le \tau$, the atoms are the 2^n vectors with entries ± 1. We have

$$[\bar{a}_k]_i = -\text{sign}[\nabla f(x_k)]_i, \quad i = 1, 2, \dots, n.$$

More Examples

Example: Nuclear Norm. For the constraint $\|X\|_* \le \tau$, for which the atoms are the rank-one matrices, we have $\bar{A}_k = u_k v_k^T$, where u_k and v_k are the first columns of the matrices U_k and V_k obtained from the SVD $\nabla f(X_k) = U_k \Sigma_k V_k^T$.

Example: sum-of-ℓ_2. For the constraint

$$\sum_{i=1}^{m} \|x_{[i]}\|_2 \le \tau,$$

the atoms are the vectors a that contain all zeros except for a vector $u_{[i]}$ with unit 2-norm in the $[i]$ block position. (Infinitely many.) The atom $\bar{a}_k$ contains nonzero components in the block i_k for which $\|[\nabla f(x_k)]_{[i]}\|$ is maximized, and the nonzero part is

$$u_{[i]} = -[\nabla f(x_k)]_{[i_k]} / \|[\nabla f(x_k)]_{[i_k]}\|.$$

Other Enhancements

Reoptimizing. Instead of fixing the contribution α_k from each atom at the time it joins the basis, we can periodically and approximately reoptimize over the current basis.

- This is a finite dimension optimization problem over the (nonnegative) coefficients of the basis atoms.
- It need only be solved approximately.
- If any coefficient is reduced to zero, it can be dropped from the basis.

Dropping Atoms. Sparsity of the solution can be improved by dropping atoms from the basis, if doing so does not degrade the value of f too much (see (Rao et al., 2013)).

In the important least-squares case, the effect of dropping can be evaluated efficiently.

References I

Akaike, H. (1959). On a successive transformation of probability distribution and its application to the analysis fo the optimum gradient method. *Annals of the Institute of Statistics and Mathematics of Tokyo*, 11:1–17.

Barzilai, J. and Borwein, J. (1988). Two point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

Bioucas-Dias, J. and Figueiredo, M. (2007). A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16:2992–3004.

Brucker, P. (1984). An O(n) algorithm for quadratic knapsack problems. *Operations Research Letters*, 3:163–166.

Candès, E. and Romberg, J. (2005). ℓ_1-MAGIC: Recovery of sparse signals via convex programming. Technical report, California Institute of Technology.

Combettes, P. and Pesquet, J.-C. (2011). Signal recovery by proximal forward-backward splitting. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer.

Combettes, P. and Wajs, V. (2006). Proximal splitting methods in signal processing. *Multiscale Modeling and Simulation*, 4:1168–1200.

References II

Ferris, M. C. and Munson, T. S. (2002). Interior-point methods for massive support vector machines. *SIAM Journal on Optimization*, 13(3):783–804.

Figueiredo, M., Nowak, R., and Wright, S. (2007). Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing*, 1:586–598.

Fountoulakis, K., Gondzio, J., and Zhlobich, P. (2012). Matrix-free interior point method for compressed sensing problems. Technical Report, University of Edinburgh.

Gertz, E. M. and Wright, S. J. (2003). Object-oriented software for quadratic programming. *ACM Transations on Mathematical Software*, 29:58–81.

Gondzio, J. and Fountoulakis, K. (2013). Second-order methods for l1-regularization. Talk at *Optimization and Big Data* Workshop, Edinburgh.

Hale, E., Yin, W., and Zhang, Y. (2008). Fixed-point continuation for l1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19:1107–1130.

Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. Technical Report, École Polytechnique, France.

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. (2011). Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12:2297–2334.

Lorenz, D. (2011). Constructing test instances for basis pursuit denoising. arXiv.org/abs/1103.2897.

References III

Maculan, N. and de Paula, G. G. (1989). A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$. *Operations Research Letters*, 8:219–222.

Moreau, J. (1962). Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A Math*, 255:2897–2899.

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Doklady*, 27:372–376.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York.

Rao, N., Shah, P., Wright, S. J., and Nowak, R. (2013). A greedy forward-backward algorithm for atomic-norm-constrained optimization. In *Proceedings of ICASSP*.

Toh, K.-C. and Yun, S. (2010). An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6:615–640.

Wright, S., Nowak, R., and Figueiredo, M. (2009). Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57:2479–2493.

Xiao, L. and Zhang, T. (2012). A proximal-gradient homotopy method for the sparse least-squares problem. *SIAM Journal on Optimization*. (to appear; available at http://arxiv.org/abs/1203.3002).