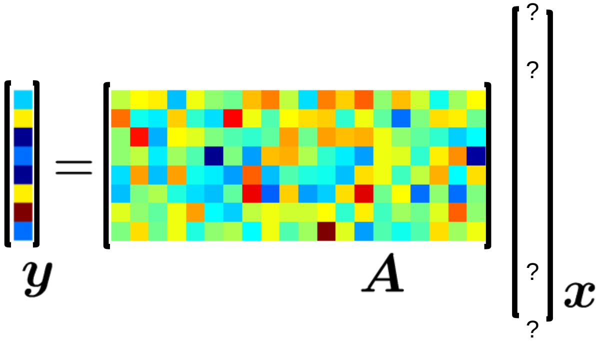
Lecture III: Algorithms

John Wright
Electrical Engineering
Columbia University

Two convex optimization problems

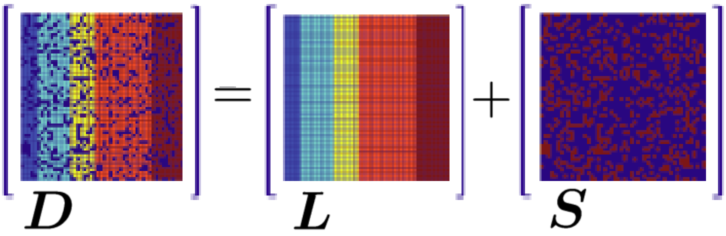
ℓ^1 **minimization** seeks a **sparse solution** to an **underdetermined** linear system of equations:

$$\min \ \|\boldsymbol{x}\|_1 \ \text{ s.t. } \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$$



Robust PCA expresses an input data matrix as a sum of a **low-rank** matrix $\boldsymbol{L}$ and a **sparse** matrix $\boldsymbol{S}$.

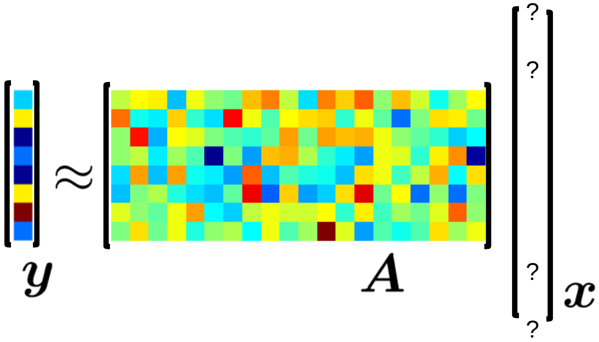
$$\min \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \ \text{ s.t. } \ \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$



Two noise-aware variants

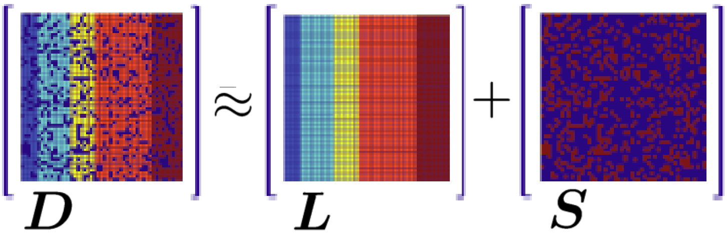
Basis pursuit denoising seeks a **sparse *near*-solution** to an **underdetermined** linear system:

$$\min \ \|\boldsymbol{x}\|_1 \ + \ \frac{\lambda}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$



Noise-aware Robust PCA *approximates* an input data matrix as a sum of a **low-rank** matrix $\boldsymbol{L}$ and a **sparse** matrix $\boldsymbol{S}$.

$$\min \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \frac{\gamma}{2}\|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$$



Many possible applications …

CHRYSLER SETS STOCK SPLIT, HIGHER DIVIDEND

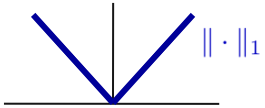
Chrysler Corp said its board declared a three-for-two stock split in the form of a 50 pct stock dividend and raised the quarterly dividend by seven pct.

The company said the dividend was rai
35 cts on a pre-split basis, equal to a 25
basis.
Chrysler said the stock dividend is pa
record March 23 while the cash dividend is
of record March 23. It said cash will be pai
With the split, Chrysler said 13.2 mln sh
in its stock repurchase program that began
now has a target of 56.3 mln shares with t
Chrysler said in a statement the actio
standing performance over the past few y
about the company's future."

… *if* we can solve these core optimization problems
accurately, efficiently, *and* ***scalably.***

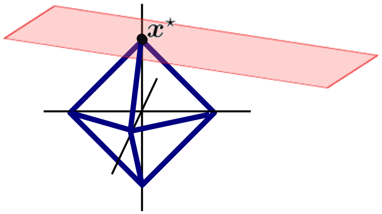
Key challenges: nonsmoothness and scale

Nonsmoothness: structure-inducing regularizers such as $\|\cdot\|_1,\ \|\cdot\|_*$ are **not differentiable**:



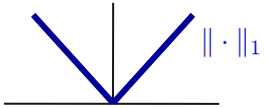
Great for structure recovery …

… challenging for optimization.

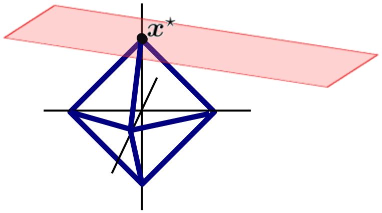


Key challenges: nonsmoothness and scale

Nonsmoothness: structure-inducing regularizers such as $\|\cdot\|_1, \ \|\cdot\|_*$ are **not differentiable**:



Great for structure recovery …
… challenging for optimization.



Scale … typical problems involve $\mathbf{10^4 - 10^6}$ **unknowns**, or more.

$$\text{Time} = (\#\text{iterations for an } \varepsilon\text{-accurate soln.}) \times (\text{time per iteration})$$

Classical **interior point methods** (e.g., SeDuMi, SDPT3): great convergence rate (linear or better), but $\Omega(\#\text{unknowns}^3)$ cost per iteration. *High accuracy for small problems.*

First-order (gradient-like) algorithms: slower (sublinear) convergence rate, but very cheap iterations. *Moderate accuracy even for large problems.*

Why care? Practical impact of algorithm choice

Time required to solve a 1,000 x 1,000 matrix recovery problem:

Algorithm	Accuracy	Rank	$\|E\|_0$	# iterations	time (sec)
IT	5.99e-006	50	101,268	8,550	**119,370.3**
DUAL	8.65e-006	50	100,024	822	1,855.4
APG	5.85e-006	50	100,347	134	1,468.9
APG$_P$	5.91e-006	50	100,347	134	82.7
EALM$_P$	2.07e-007	50	100,014	34	37.5
IALM$_P$	3.83e-007	50	99,996	23	**11.8**

Four orders of magnitude improvement, just by choosing the right algorithm to solve the convex program.

This is the difference between theory that will have impact "someday" and practical computational techniques that can be applied right now…

This lecture: Three key techniques

In this hour lecture, we will focus on **three recurring ideas** that allow us to address the challenges of nonsmoothness and scale:

Proximal gradient methods: coping with *nonsmoothness*

Optimal first-order methods: *accelerating convergence*

Augmented Lagrangian methods: handling *constraints*

For more depth / breadth, please see the references at the end of these slides or Lieven Vandenberghe's lectures this afternoon!

Why worry about nonsmoothness?

The best uniform **rate of convergence** for **first-order methods*** for minimizing $f \in \mathcal{F}$ depends very strongly on smoothness:

Function class $\mathcal{F}$	Minimax suboptimality $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$
smooth f convex, differentiable $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
nonsmooth f convex $\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\| \leq M\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CM\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{\sqrt{k}} = \Theta\left(\frac{1}{\sqrt{k}}\right)$

* *Such as gradient descent. See e.g., Nesterov, "Introductory Lectures on Convex Optimization"*

Why worry about nonsmoothness?

The best uniform **rate of convergence** for **first-order methods*** for minimizing $f \in \mathcal{F}$ depends very strongly on smoothness:

Function class $\mathcal{F}$	Minimax suboptimality $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$
smooth f convex, differentiable $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \le L\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
nonsmooth f convex $\lvert f(\boldsymbol{x}) - f(\boldsymbol{x}')\rvert \le M\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CM\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{\sqrt{k}} = \Theta\left(\frac{1}{\sqrt{k}}\right)$

For $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \le \varepsilon$, need $k = O(\varepsilon^{-2})$ iter. for worst **nonsmooth** f

Can we exploit special structure of $\|\cdot\|_1, \ \|\cdot\|_*$ to get accuracy comparable to gradient descent (for smooth functions) ?

What does gradient descent do, anyway?

Consider $\min \ f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

What does gradient descent do, anyway?

Consider $\min \; f(\boldsymbol{x}),$ with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$\hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) \quad \doteq \quad f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2$$

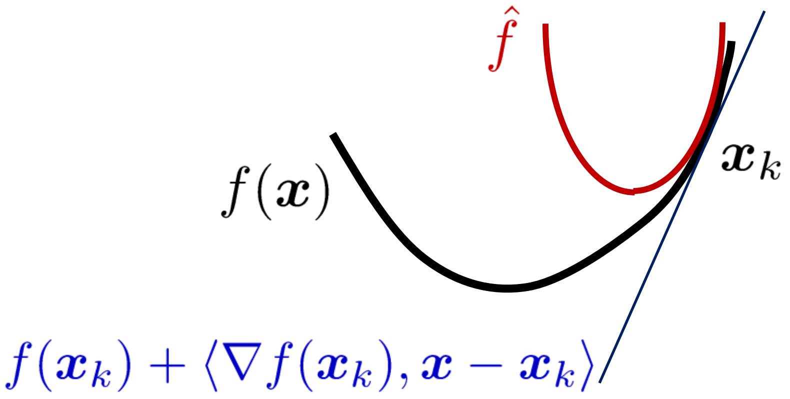
What does gradient descent do, anyway?

Consider $\min \ f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$\hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) \doteq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2$$



$\hat{f}$
$\boldsymbol{x}_k$
$f(\boldsymbol{x})$
$f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle$

What does gradient descent do, anyway?

Consider $\min \; f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$
\begin{aligned}
\hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) \quad &\doteq \quad f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2 \\
&= \quad \frac{L}{2}\left\|\boldsymbol{x} - \left(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right)\right\|_2^2 + \underline{\varphi(\boldsymbol{x}_k).}
\end{aligned}
$$

Doesn't depend on $\boldsymbol{x}$

What does gradient descent do, anyway?

Consider $\min \; f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$\begin{aligned} \hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) &\doteq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2 \\ &= \frac{L}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2 + \varphi(\boldsymbol{x}_k). \end{aligned}$$

Key observation: $\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{x}_k)$.

At each iteration, the gradient descent minimizes a (separable) quadratic approximation to the objective function, formed at $\boldsymbol{x}_k$.

What does gradient descent do, anyway?

Consider $\min\ f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$\begin{aligned} \hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) \quad &\doteq \quad f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \tfrac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2 \\ &= \quad \tfrac{L}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2 + \varphi(\boldsymbol{x}_k). \end{aligned}$$

Key observation: $\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{x}_k).$

At each iteration, the gradient descent minimizes a (separable) quadratic approximation to the objective function, formed at $\boldsymbol{x}_k$.

Rate for gradient descent: $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \ \leq \ \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k} \ = \ O\left(\frac{1}{k}\right)$

Borrowing the approximation idea…

$$\min \ \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1$$

Borrowing the approximation idea…

$$\min \ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1$$

smooth ⌣ *nonsmooth* ∨

Borrowing the approximation idea…

$$\min \ \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1 \qquad \equiv \qquad \min \quad f(\boldsymbol{x}) \ + \ g(\boldsymbol{x})$$

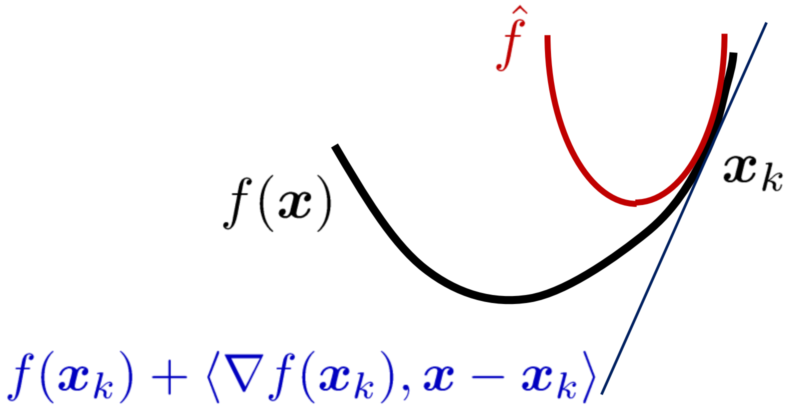
smooth *nonsmooth*

Borrowing the approximation idea…

$$\min \ \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1 \qquad \equiv \qquad \min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} \ + \ \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Just **approximate the smooth part:**

$$\hat{F}(\boldsymbol{x},\boldsymbol{x}_k) \quad \doteq \quad f(\boldsymbol{x}_k)+\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}-\boldsymbol{x}_k\rangle+\frac{L}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2+g(\boldsymbol{x})$$


$\hat{f}$
$f(\boldsymbol{x})$
$\boldsymbol{x}_k$
$f(\boldsymbol{x}_k)+\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}-\boldsymbol{x}_k\rangle$

Borrowing the approximation idea…

$$\min\ \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1 \qquad \equiv \qquad \min \quad \underset{\textit{smooth}}{f(\boldsymbol{x})}\ +\ \underset{\textit{nonsmooth}}{g(\boldsymbol{x})}$$

Just **approximate the smooth part:**

$$\begin{aligned} \hat{F}(\boldsymbol{x},\boldsymbol{x}_k) &\doteq f(\boldsymbol{x}_k)+\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}-\boldsymbol{x}_k\rangle+\tfrac{L}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2+g(\boldsymbol{x}) \\ &= \tfrac{L}{2}\|\boldsymbol{x}-(\boldsymbol{x}_k-\tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2+g(\boldsymbol{x})+\varphi(\boldsymbol{x}_k). \end{aligned}$$

Borrowing the approximation idea…

$$\min\ \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1 \qquad \equiv \qquad \min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} \ + \ \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Just **approximate the smooth part:**

$$\begin{aligned}\hat{F}(\boldsymbol{x},\boldsymbol{x}_k) &\doteq f(\boldsymbol{x}_k)+\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}-\boldsymbol{x}_k\rangle+\tfrac{L}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2+g(\boldsymbol{x})\\ &= \tfrac{L}{2}\|\boldsymbol{x}-(\boldsymbol{x}_k-\tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2+g(\boldsymbol{x})+\varphi(\boldsymbol{x}_k).\end{aligned}$$

… and then **minimize to get the next iterate:**

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x},\boldsymbol{x}_k)\\ &= \arg\min_{\boldsymbol{x}}\ \tfrac{L}{2}\|\boldsymbol{x}-(\boldsymbol{x}_k-\tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2+g(\boldsymbol{x}).\end{aligned}$$

This is called a **proximal gradient algorithm**.

Proximal gradient algorithm

$\min\ f(\boldsymbol{x}) + g(\boldsymbol{x})$, with f convex differentiable, ∇f L-Lipschitz.

Proximal Gradient:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \tfrac{L}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2 + g(\boldsymbol{x})$$

Converges at the **same rate as gradient descent**:

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}^*) \ \le\ \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k} \ =\ O\left(\tfrac{1}{k}\right)$$

Efficient whenever we can easily solve the **proximal problem**

$$\mathrm{prox}_{\mu g}(\boldsymbol{z}) \ =\ \arg\min_{\boldsymbol{x}}\ \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

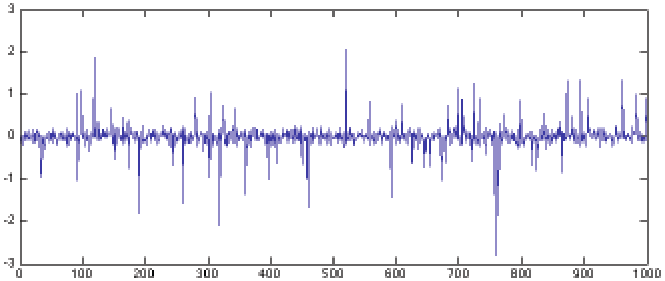
i.e., minimize g plus a separable quadratic.

Prox. operators for structure-inducing norms

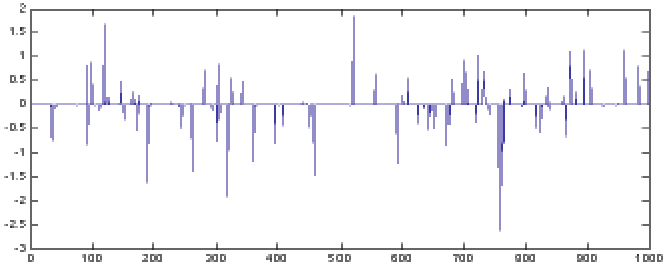
$$\operatorname{prox}_{\mu g}(\boldsymbol{z}) \;=\; \arg\min_{\boldsymbol{x}} \;\tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

For $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$, $\operatorname{prox}_{\mu g}(\boldsymbol{z})$ is given by **soft thresholding** the elements of $\boldsymbol{z}$: $\mathcal{S}_\mu(z) = \operatorname{sign}(z)\max\{|z|-\mu, 0\}.$

This operator shrinks all of the elements of $\boldsymbol{z}$ towards zero:



$\boldsymbol{z}$



$\mathcal{S}_\mu(\boldsymbol{z})$

It can be computed in linear time (very efficient).

Prox. operators for structure-inducing norms

$$\mathrm{prox}_{\mu g}(\boldsymbol{z}) \;=\; \arg\min_{\boldsymbol{x}} \; \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

For $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$, $\mathrm{prox}_{\mu g}(\boldsymbol{z})$ is given by **soft thresholding** the elements of $\boldsymbol{z}$: $\mathcal{S}_\mu(z) = \mathrm{sign}(z)\max\{|z| - \mu, 0\}.$

For $g(\boldsymbol{X}) = \|\boldsymbol{X}\|_*$, $\mathrm{prox}_{\mu g}(\boldsymbol{Z})$ is given by **soft thresholding** the **singular values** of $\boldsymbol{Z}$: for $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$,

$$\mathrm{prox}_{\mu g}(\boldsymbol{Z}) \;=\; \boldsymbol{U}\mathcal{S}_\mu[\boldsymbol{\Sigma}]\boldsymbol{V}^*.$$

Again efficient (same cost as a singular value decomposition).

Similar expressions exist for other structure inducing norms.

Summing up: proximal gradient

$$\min \; f(\boldsymbol{x}) + g(\boldsymbol{x}), \quad \text{with } f \text{ convex differentiable, } \nabla f \; L\text{-Lipschitz.}$$

Proximal Gradient:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \tfrac{L}{2} \|\boldsymbol{x} - (\boldsymbol{x}_k - \tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2 + g(\boldsymbol{x})$$

Converges at the **same rate as gradient descent**:

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}^*) \;\leq\; \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k} \;=\; O\left(\tfrac{1}{k}\right)$$

Efficient whenever we can easily solve the **proximal problem**

$$\operatorname{prox}_{\mu g}(\boldsymbol{z}) \;=\; \arg\min_{\boldsymbol{x}} \; \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

This is the case for many structure-inducing norms.

What have we accomplished so far?

Function class $\mathcal{F}$	Suboptimality $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$
smooth f convex, differentiable $\lVert \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}') \rVert \le L \lVert \boldsymbol{x} - \boldsymbol{x}' \rVert$	$\frac{CL\lVert \boldsymbol{x}_0 - \boldsymbol{x}^* \rVert^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
smooth + structured nonsmooth: $F = f + g$ + f, g convex, $\lVert \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}') \rVert \le L \lVert \boldsymbol{x} - \boldsymbol{x}' \rVert$	$\frac{CL\lVert \boldsymbol{x}_0 - \boldsymbol{x}^* \rVert^2}{k} = O\left(\frac{1}{k}\right)$
nonsmooth f convex $\lvert f(\boldsymbol{x}) - f(\boldsymbol{x}') \rvert \le M \lVert \boldsymbol{x} - \boldsymbol{x}' \rVert$	$\frac{CM\lVert \boldsymbol{x}_0 - \boldsymbol{x}^* \rVert}{\sqrt{k}} = \Theta\left(\frac{1}{\sqrt{k}}\right)$

Still a gap between convergence rate of proximal gradient, $O(1/k)$ and the optimal $O(1/k^2)$ rate for smooth f.

Can we close this gap?

Why is the gradient method suboptimal?

For smooth f, gradient descent is also suboptimal…
intuitively, for badly conditioned functions it may "chatter":

Gradient descent

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)$$

Why is the gradient method suboptimal?

For smooth f, gradient descent is also suboptimal…
intuitively, for badly conditioned functions it may “chatter”:

Gradient descent

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)$$

The *heavy ball method* treats the iterate as a point mass with momentum, and hence, a tendency to continue moving in direction $\boldsymbol{x}_k - \boldsymbol{x}_{k-1}$:

Heavy ball

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k) + \beta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

Nesterov's optimal method

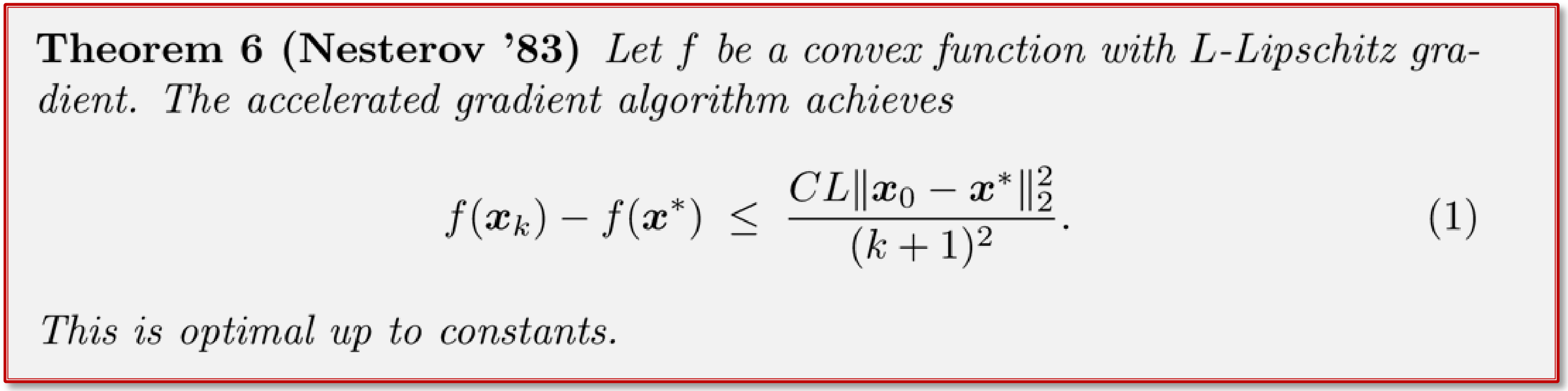
Shares some intuition with heavy ball, but not identical.

Heavy ball : $$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k) + \beta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

Nesterov : $$\boldsymbol{y}_k = \boldsymbol{x}_k + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$
$$\boldsymbol{x}_{k+1} = \boldsymbol{y}_k - \alpha \nabla f(\boldsymbol{y}_k)$$

with a very special choice of β_k to ensure the optimal rate:

$$\beta_k = \tfrac{t_k - 1}{t_{k+1}} \qquad t_{k+1} = \tfrac{1+\sqrt{1+4t_k^2}}{2} \qquad \alpha = 1/L$$



Theorem 6 (Nesterov '83) *Let f be a convex function with L-Lipschitz gradient. The accelerated gradient algorithm achieves*

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \;\leq\; \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2}{(k+1)^2}. \tag{1}$$

This is optimal up to constants.

What about smooth + nonsmooth?

$$\min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} \; + \; \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \quad \doteq \quad f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \tfrac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

What about smooth + nonsmooth?

$$\min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} \; + \; \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \quad \doteq \quad f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

Again, **replace the gradient step** with minimization of the upper bound:

$$\boldsymbol{x}_{k+1} \quad = \quad \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x}, \boldsymbol{y}_k)$$

What about smooth + nonsmooth?

$$\min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} \; + \; \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \quad \doteq \quad f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \tfrac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

Again, **replace the gradient step** with minimization of the upper bound:

$$\begin{aligned} \boldsymbol{x}_{k+1} \quad &= \quad \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \\ &= \quad \arg\min_{\boldsymbol{x}} \tfrac{1}{L} \|\boldsymbol{x} - (\boldsymbol{y}_k - \tfrac{1}{L} \nabla f(\boldsymbol{y}_k))\|^2 + g(\boldsymbol{x}) \end{aligned}$$

What about smooth + nonsmooth?

$$\min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} \; + \; \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \quad \doteq \quad f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

Again, **replace the gradient step** with minimization of the upper bound:

$$\begin{aligned} \boldsymbol{x}_{k+1} \quad &= \quad \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \\ &= \quad \arg\min_{\boldsymbol{x}} \frac{1}{L}\|\boldsymbol{x} - (\boldsymbol{y}_k - \frac{1}{L}\nabla f(\boldsymbol{y}_k))\|^2 + g(\boldsymbol{x}) \\ &= \quad \operatorname{prox}_{L^{-1}g}(\boldsymbol{y}_k - \frac{1}{L}\nabla f(\boldsymbol{y}_k)). \end{aligned}$$

What about smooth + nonsmooth?

$$\min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} \; + \; \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \quad \doteq \quad f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \tfrac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

Again, **replace the gradient step** with minimization of the upper bound:

$$\begin{aligned} \boldsymbol{x}_{k+1} \quad &= \quad \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \\ &= \quad \arg\min_{\boldsymbol{x}} \tfrac{1}{L} \|\boldsymbol{x} - (\boldsymbol{y}_k - \tfrac{1}{L} \nabla f(\boldsymbol{y}_k))\|^2 + g(\boldsymbol{x}) \\ &= \quad \operatorname{prox}_{L^{-1} g}(\boldsymbol{y}_k - \tfrac{1}{L} \nabla f(\boldsymbol{y}_k)). \end{aligned}$$

Making the **same special choice** $\boldsymbol{y}_k = \boldsymbol{x}_k + \beta_k (\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$, we obtain an ***accelerated* proximal gradient** algorithm.

Accelerated proximal gradient algorithm

$$\min \; f(\boldsymbol{x}) + g(\boldsymbol{x}),$$ with f convex, differentiable, ∇f L-Lipschitz.

Accelerated Proximal Gradient:

$$\text{Repeat} \left\{ \begin{aligned} \boldsymbol{y}_k &= \boldsymbol{x}_k + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \\ \boldsymbol{x}_{k+1} &= \operatorname{prox}_{L^{-1}g}(\boldsymbol{y}_k - \tfrac{1}{L}\nabla f(\boldsymbol{y}_k)) \end{aligned} \right.$$

with $\beta_k = \frac{t_k - 1}{t_{k+1}}$ and $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$.

Converges at the **same rate as Nesterov's optimal gradient method**:

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}^*) \;\leq\; \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{(k+1)^2} \;=\; O\left(\frac{1}{k^2}\right)$$

Again, efficient whenever we can easily solve the **proximal problem**

$$\operatorname{prox}_{\mu g}(\boldsymbol{z}) \;=\; \arg\min_{\boldsymbol{x}} \; \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

What have we accomplished so far?

Function class $\mathcal{F}$	Suboptimality $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$
smooth f convex, differentiable $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
smooth + structured nonsmooth: $F = f + g$ f, g convex, $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
nonsmooth f convex $\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\| \leq M\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CM\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{\sqrt{k}} = \Theta\left(\frac{1}{\sqrt{k}}\right)$

For composite functions $F = f + g$, with f smooth,
***if g has an efficient proximal operator**, we achieve*
the same (optimal) rate as if F was smooth.

What about constraints?

Consider the **equality constrained** problem

$$\min \ \|\boldsymbol{x}\|_1 \ \text{ s.t. } \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad (*)$$

Continuation: solve a sequence of unconstrained problems of form

$$\min \ \|\boldsymbol{x}\|_1 + \tfrac{\mu}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2,$$

with $\mu \nearrow \infty$. Solutions converge to the solution to $(*)$.

Big downside: **conditioning**. For $f(\boldsymbol{x}) = \frac{\mu}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$, the gradient is L-Lipschitz, with $L = \mu\|\boldsymbol{A}^*\boldsymbol{A}\|$. As $\mu \nearrow \infty$, the unconstrained problems get harder and harder to solve.

Is there a better-structured way to enforce equality constraints?

The method of multipliers

$$\min \; F(\boldsymbol{x}) \;\; \text{s.t.} \;\; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad\qquad (*)$$

The **Lagrangian** is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \;=\; F(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle$$

The method of multipliers

$$\min \; F(\boldsymbol{x}) \;\; \text{s.t.} \;\; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad\qquad (*)$$

The **augmented Lagrangian** is

$$\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) \; = \; F(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle + \underline{\underline{\tfrac{\rho}{2} \| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \|_2^2}}.$$

Extra penalty term

The method of multipliers

$$\min \; F(\boldsymbol{x}) \;\; \text{s.t.} \;\; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad (*)$$

The **augmented Lagrangian** is

$$\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) \;=\; F(\boldsymbol{x}) \;+\; \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle \;+\; \tfrac{\rho}{2} \| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \|_2^2.$$

The **method of multipliers** solves $(*)$ by seeking a saddle point of $\mathcal{L}_\rho$:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}_k) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y}). \end{aligned}$$

The method of multipliers

$$\min\ F(\boldsymbol{x})\ \text{ s.t. }\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad (*)$$

The **augmented Lagrangian** is

$$\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) \ = \ F(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle + \tfrac{\rho}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

The **method of multipliers** solves $(*)$ by seeking a saddle point of $\mathcal{L}_\rho$:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}_k) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y}). \end{aligned}$$

Solves a **sequence of unconstrained problems**: $\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}_k)$

Penalty parameter $\rho > 0$ can be constant (**avoids ill-conditioning**) , or increasing for (faster convergence).

Summing up: Method of multipliers

Solves, e.g., $\min\ F(\boldsymbol{x})$ s.t. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$, with F convex, lsc.

Method of multipliers (augmented Lagrangian)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}_k) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y}). \end{aligned}$$

Classical method [Hestenes ‘69, Powell ‘69], see also [Bertsekas ‘82].

Avoids conditioning problems with the continuation / penalty method.

Under very general conditions $\boldsymbol{\lambda}_k$ converges to a dual optimal point,

$$\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{y}\| \to 0, \text{ and } F(\boldsymbol{x}_k) \to \inf\{\, F(\boldsymbol{x}) \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \,\}\,.$$

[Rockafellar ‘73, Eckstein ‘12] .

What have we accomplished so far?

Consider the robust PCA problem

$$\min \; \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$

Augmented Lagrangian

$$\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$$

The **method of multipliers** is

$$\begin{aligned} (\boldsymbol{L}_{k+1}, \boldsymbol{S}_{k+1}) &= \arg\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}_k, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2 \\ \boldsymbol{\Lambda}_{k+1} &= \boldsymbol{\Lambda}_k + \rho(\boldsymbol{L}_k + \boldsymbol{S}_k - \boldsymbol{D}) \end{aligned}$$

Each iteration is a large nonsmooth optimization problem…

Is there special structure we can exploit to simplify the iterations?

Special structure: Separable objectives

$$\boxed{\min \quad \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to S is easy:

$$\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \quad = \quad \arg\min_{\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$$

Special structure: Separable objectives

$$\boxed{\min \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L}+\boldsymbol{S}=\boldsymbol{D}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S},\boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\rangle + \frac{\rho}{2}\|\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{S}$ is easy:

$$\begin{aligned}
\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S},\boldsymbol{\Lambda}) \quad &= \quad \arg\min_{\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\rangle + \frac{\rho}{2}\|\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\|_F^2 \\
&= \quad \arg\min_{\boldsymbol{S}} \lambda\|\boldsymbol{S}\|_1 + \frac{\rho}{2}\|\boldsymbol{S}-(\boldsymbol{D}-\boldsymbol{L}-\tfrac{1}{\rho}\boldsymbol{\Lambda})\|_F^2 + \varphi(\boldsymbol{L},\boldsymbol{D},\boldsymbol{\Lambda})
\end{aligned}$$

Special structure: Separable objectives

$$\boxed{\min \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L}+\boldsymbol{S}=\boldsymbol{D}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S},\boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\rangle + \frac{\rho}{2}\|\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{S}$ is easy:

$$\begin{aligned}
\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S},\boldsymbol{\Lambda}) &= \arg\min_{\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\rangle + \frac{\rho}{2}\|\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\|_F^2 \\
&= \arg\min_{\boldsymbol{S}} \lambda\|\boldsymbol{S}\|_1 + \frac{\rho}{2}\|\boldsymbol{S}-(\boldsymbol{D}-\boldsymbol{L}-\frac{1}{\rho}\boldsymbol{\Lambda})\|_F^2 + \varphi(\boldsymbol{L},\boldsymbol{D},\boldsymbol{\Lambda}) \\
&= \operatorname{prox}_{\lambda\rho^{-1}\|\cdot\|_1}(\boldsymbol{D}-\boldsymbol{L}-\rho^{-1}\boldsymbol{\Lambda}).
\end{aligned}$$

Special structure: Separable objectives

$$\boxed{\min \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L}+\boldsymbol{S}=\boldsymbol{D}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S},\boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\rangle + \frac{\rho}{2}\|\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{S}$ is easy:

$$\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S},\boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\lambda\rho^{-1}\|\cdot\|_1}(\boldsymbol{D}-\boldsymbol{L}-\rho^{-1}\boldsymbol{\Lambda}).$$

Special structure: Separable objectives

$$\boxed{\min \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\rangle + \frac{\rho}{2}\|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{S}$ is easy:

$$\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\lambda\rho^{-1}\|\cdot\|_1}(\boldsymbol{D} - \boldsymbol{L} - \rho^{-1}\boldsymbol{\Lambda}).$$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{L}$ is also easy:

$$\arg\min_{\boldsymbol{L}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\rho^{-1}\|\cdot\|_*}(\boldsymbol{D} - \boldsymbol{S} - \rho^{-1}\boldsymbol{\Lambda}).$$

Special structure: Separable objectives

$$\boxed{\min \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L}+\boldsymbol{S}=\boldsymbol{D}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S},\boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\rangle + \frac{\rho}{2}\|\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{S}$ is easy:

$$\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S},\boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\lambda\rho^{-1}\|\cdot\|_1}(\boldsymbol{D}-\boldsymbol{L}-\rho^{-1}\boldsymbol{\Lambda}).$$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{L}$ is also easy:

$$\arg\min_{\boldsymbol{L}} \mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S},\boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\rho^{-1}\|\cdot\|_*}(\boldsymbol{D}-\boldsymbol{S}-\rho^{-1}\boldsymbol{\Lambda}).$$

Why not just alternate?

$$\begin{aligned}
\boldsymbol{L}_{k+1} &= \arg\min_{\boldsymbol{L}} \mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{S}_k,\boldsymbol{\Lambda}_k) &&= \operatorname{prox}_{\rho^{-1}\|\cdot\|_*}(\boldsymbol{D}-\boldsymbol{S}_k-\rho^{-1}\boldsymbol{\Lambda}_k).\\
\boldsymbol{S}_{k+1} &= \arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}_{k+1},\boldsymbol{S},\boldsymbol{\Lambda}_k) &&= \operatorname{prox}_{\lambda\rho^{-1}\|\cdot\|_1}(\boldsymbol{D}-\boldsymbol{L}_{k+1}-\rho^{-1}\boldsymbol{\Lambda}_k).\\
\boldsymbol{\Lambda}_{k+1} &= \boldsymbol{\Lambda}_k + \rho(\boldsymbol{L}_{k+1}+\boldsymbol{S}_{k+1}-\boldsymbol{D})
\end{aligned}$$

More generally: Alternating Directions MoM

$$\boxed{\min \; f(\boldsymbol{x}) + h(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \rangle + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y}\|_F^2$

Alternating Directions Method of Multipliers (ADMM)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{\lambda}_k) \\ \boldsymbol{z}_{k+1} &= \arg\min_{\boldsymbol{z}} \mathcal{L}_\rho(\boldsymbol{x}_{k+1}, \boldsymbol{z}, \boldsymbol{\lambda}_k) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_{k+1} - \boldsymbol{y}) \end{aligned}$$

Alternating Directions MoM

$$\boxed{\min \; f(\boldsymbol{x}) + h(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \rangle + \frac{\rho}{2} \| \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \|_F^2$

Alternating Directions Method of Multipliers (ADMM)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{\lambda}_k) \\ \boldsymbol{z}_{k+1} &= \arg\min_{\boldsymbol{z}} \mathcal{L}_\rho(\boldsymbol{x}_{k+1}, \boldsymbol{z}, \boldsymbol{\lambda}_k) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_{k+1} - \boldsymbol{y}) \end{aligned}$$

Convergence: if f, h closed, proper, convex functions, and $\mathcal{L}$ has a saddle point, then … $\boldsymbol{\lambda}_k$ converges to a dual optimal point, $\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{z}_k \to \boldsymbol{y}$ and $f(\boldsymbol{x}_k) + h(\boldsymbol{z}_k) \to \inf\{\, f(\boldsymbol{x}) + h(\boldsymbol{z}) \mid \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y} \,\}$.

Convergence rate $O(1/k)$, in a certain sense [He+Yuan ‘11].

Linearized Alternating Directions MoM

$$\boxed{\min \; f(\boldsymbol{x}) + h(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \rangle + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y}\|_F^2$

ADMM:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{\lambda}_k) \\ &= \arg\min_{\boldsymbol{x}} \underline{f(\boldsymbol{x}) + \tfrac{\rho}{2} \|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \tfrac{1}{\rho}\boldsymbol{\lambda}_k\|_2^2} \end{aligned}$$

Complicated if $\boldsymbol{A}, \boldsymbol{B} \neq \boldsymbol{I}$

Linearized ADMM: just take a proximal gradient step…

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \tfrac{\rho}{2\tau} \|\boldsymbol{x} - (\boldsymbol{x}_k - \tau \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \tfrac{1}{\rho}\boldsymbol{\lambda}_k))\|_2^2 \\ &= \operatorname{prox}_{\frac{\tau}{\rho} f}(\boldsymbol{x}_k - \tau \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} - \tfrac{1}{\rho}\boldsymbol{\lambda}_k)) \end{aligned}$$

Much more efficient if f has a simple proximal operator.

Linearized Alternating Directions MoM

$$\boxed{\min \; f(\boldsymbol{x}) + h(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y}}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \rangle + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y}\|_F^2$

Linearized ADMM

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \operatorname{prox}_{\frac{\tau}{\rho} f}(\boldsymbol{x}_k - \tau \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \tfrac{1}{\rho}\boldsymbol{\lambda}_k)) \\
\boldsymbol{z}_{k+1} &= \operatorname{prox}_{\frac{\tau}{\rho} h}(\boldsymbol{z}_k - \tau \boldsymbol{B}^*(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \tfrac{1}{\rho}\boldsymbol{\lambda}_k)) \\
\boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_{k+1} - \boldsymbol{y})
\end{aligned}$$

See, e.g., [S. Ma 2012]. Convergent if $\tau < \min\{\|\boldsymbol{A}\|^2, \|\boldsymbol{B}\|^2\}$.

Handles problems with more than two terms, e.g., $\sum_i f_i(\boldsymbol{x}_i)$.

Now can take advantage of two types of special structure …
separability of the objective and *prox capability* of f, h.

Finally, what have we accomplished?

Time required to solve a 1,000 x 1,000 robust PCA problem:

Algorithm	Accuracy	Rank	$\|E\|_0$	# iterations	time (sec)
IT	5.99e-006	50	101,268	8,550	**119,370.3**
DUAL	8.65e-006	50	100,024	822	1,855.4
APG	5.85e-006	50	100,347	134	1,468.9
APG_P	5.91e-006	50	100,347	134	82.7
EALM_P	2.07e-007	50	100,014	34	37.5
IALM_P	3.83e-007	50	99,996	23	**11.8**

THIS LECTURE

Four orders of magnitude improvement, just by choosing the right algorithm to solve the convex program:

Proximal gradient ⇒ Accelerated proximal gradient ⇒ ALM ⇒ ADMoM

Recap and Conclusions

Key challenges of **nonsmoothness** and **scale** can be mitigated by using **special structure** in sparse and low-rank optimization problems:

Efficient proximity operators $\Rightarrow$ *proximal gradient methods*

Separable objectives $\Rightarrow$ *alternating directions methods*

Efficient **moderate-accuracy solutions** for **very large problems**.

Special tricks can further improve specific cases (factorization for low-rank)

Techniques in this literature apply quite broadly.

Extremely useful tools for creative problem formulation / solution.

Fundamental **theory** guiding engineering **practice**:

What are the basic principles and limitations?
What specific structure in my problem can allow me to do better?

To read more…

Problem complexity and lower bounds:

Nesterov – Introductory Lectures on Convex Optimization: A Basic Course 2004
Nemirovsky – Problem Complexity and Method Efficiency in Convex Optimization

Proximal gradient methods:

Accelerated gradient methods:

Nesterov – A method of solving a convex programming problem with convergence rate O(1/k^2), 1983
Tseng – On Accelerated Proximal Gradient Methods for Convex-Concave Optimization, 2008
Beck+Teboulle – A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, 2009

Augmented Lagrangian:

Hestenes – Multiplier and gradient methods, 1969
Powell – A method for nonlinear constraints in minimization problems, 1969
Rockafellar – Augmented Lagrangians and the Proximal Point Algorithm in Convex Programming, 1973
Bertsekas – Constrained Optimization and Lagrange Multiplier Methods, 1982

Alternating directions:

Glowinski+Marocco – Sur l’approximation, par elements finis d’ordre un, et la resolution, par … 1975
Gabay+Mercier – A dual algorithm for the solution of nonlinear variational problems … 1976
Eckstein+Bertsekas – On the Douglas-Rachford splitting method and the proximal point … 1992
Boyd et. al. – Distributed optimization and statistical learning via the alternating directions … 2010
Eckstein – Augmented Lagrangian and Alternating Directions Methods for Convex Optimization 2012