Sparse Optimization
Lecture: Dual Methods, Part I

Instructor: Wotao Yin

July 2013

online discussions on piazza.com

Those who complete this lecture will know

- dual (sub)gradient iteration
- augmented ℓ_1 iteration (linearzed Bregman iteration)
- dual smoothing minimization
- augmented Lagrangian iteration
- Bregman iteration and addback iteration

Review

Last two lectures

- studied explicit and implicit (proximal) gradient updates
- derived the Lagrange dual problem
- overviewed the following dual methods
  1. dual (sub)gradient method (a.k.a. Uzawa's method)
  2. dual proximal method (a.k.a., augmented Lagrangian method (ALM))
  3. operator splitting methods applied to the dual of

$$\min_{\mathbf{x},\mathbf{z}} \{f(\mathbf{x}) + g(\mathbf{z}) : \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}\}$$

The operator splitting methods studied includes

- forward-backward splitting
- Peaceman-Rachford splitting
- Douglas-Rachford splitting (giving rise to ADM or ADMM)

This lecture study these dual methods in more details and present their applications to sparse optimization models.

About sparse optimization

During the lecture, keep in mind that sparse optimization models typically have one or more *nonsmooth* yet *simple* part(s) and one or more *smooth* parts with *dense* data.

Common preferences:

- to the nonsmooth and simple part, proximal operation is preferred over subgradient descent
- if smoothing is applied, exact smoothing is preferred over inexact smoothing
- to the smooth part with dense data, simple (gradient) operator is preferred over more complicated operators
- when applying divide-and-conquer, fewer divisions and simpler subproblems are preferred

In general, the dual methods appear to be more versatile than the primal-only methods (e.g., (sub)gradient and prox-linear methods)

Dual (sub)gradient ascent

Primal problem

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}.$$

Lagrangian relaxation:

$$\mathcal{L}(\mathbf{x}; \mathbf{y}) = f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$$

Lagrangian dual problem

$$\min_{\mathbf{y}} \ d(\mathbf{y}) \quad \text{or} \quad \max_{\mathbf{y}} -d(\mathbf{y})$$

If g is *differentiable*, you can apply

$$\mathbf{y}^{k+1} \leftarrow \mathbf{y}^k - c^k \nabla d(\mathbf{y}^k)$$

otherwise, apply

$$\mathbf{y}^{k+1} \leftarrow \mathbf{y}^k - c^k \mathbf{g}, \quad \text{where } \mathbf{g} \in \partial d(\mathbf{y}^k).$$

Dual (sub)gradient ascent

Derive ∇d or ∂d

- by hand, or
- use $\mathcal{L}(\mathbf{x};\mathbf{y}^k)$: compute $\mathbf{x}^k \leftarrow \min_{\mathbf{x}} \mathcal{L}(\mathbf{x};\mathbf{y}^k)$, then $\mathbf{b} - \mathbf{A}\bar{\mathbf{x}} \in \partial d(\mathbf{y}^k)$.

Iteration:

$$\mathbf{x}^{k+1} \leftarrow \min_{\mathbf{x}} \mathcal{L}(\mathbf{x};\mathbf{y}^k),$$
$$\mathbf{y}^{k+1} \leftarrow \mathbf{y}^k + c^k(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}).$$

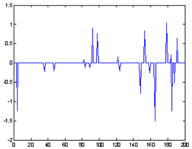
Application: augmented ℓ_1 minimization, a.k.a. linearized Bregman.

Augmented ℓ_1 minimization

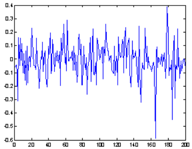
Augment ℓ_1 by ℓ_2^2:

$$\text{(L1+LS)} \quad \min \|\mathbf{x}\|_1 + \frac{1}{2\alpha}\|\mathbf{x}\|_2^2 \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}$$

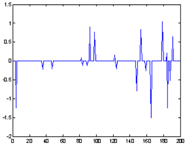
- primal objective becomes strongly convex (but still non-differentiable)
- hence, its dual is unconstrained and differentiable
- a sufficiently large but finite α leads the exact ℓ_1 solution
- related to: linearized Bregman algorithm, the elastic net model
- a test with Gaussian $\mathbf{A}$ and sparse $\mathbf{x}$ with Gaussian entries:



$\min\{\|\mathbf{x}\|_1 : \mathbf{A}\mathbf{x} = \mathbf{b}\}$



$\min\{\|\mathbf{x}\|_2^2 : \mathbf{A}\mathbf{x} = \mathbf{b}\}$



$\min\{\|\mathbf{x}\|_1 + \frac{1}{25}\|\mathbf{x}\|_2^2 : \mathbf{A}\mathbf{x} = \mathbf{b}\}$

Exactly the same as ℓ_1 solution

Lagrangian dual of (L1+LS)

Theorem (*Convex Analysis*, Rockafellar [1970])

If a convex program has a strictly convex objective, it has a unique solution and its Lagrangian dual program is differentiable.

Lagrangian: (separable in $\mathbf{x}$ for fixed $\mathbf{y}$)

$$\mathcal{L}(\mathbf{x};\mathbf{y}) = \|\mathbf{x}\|_1 + \frac{1}{2\alpha}\|\mathbf{x}\|_2^2 - \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$$

Lagrange dual problem:

$$\min_{\mathbf{y}} d(\mathbf{y}) = -\mathbf{b}^\top\mathbf{y} + \frac{\alpha}{2}\|\mathbf{A}^\top\mathbf{y} - \mathrm{Proj}_{[-1,1]^n}(\mathbf{A}^\top\mathbf{y})\|_2^2$$

note: $\mathrm{shrink}(x,\gamma) = \max\{|x|-\gamma, 0\}\mathrm{sign}(x) = x - \mathrm{Proj}_{[-\gamma,\gamma]}(x)$.

Objective gradient:

$$\nabla d(\mathbf{y}) = -\mathbf{b} + \alpha\mathbf{A}\,\mathrm{shrink}(\mathbf{A}^\top\mathbf{y})$$

Dual gradient iteration:

$$\mathbf{x}^{k+1} = \alpha\,\mathrm{shrink}(\mathbf{A}^\top\mathbf{y}^k),$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k + c^k(\mathbf{b} - \mathbf{A}\mathbf{x}^{k+1}).$$

How to choose α

- **Exact smoothing**: $\exists$ a finite α^0 so that all $\alpha > \alpha^0$ lead to ℓ_1 solution
- In practice, $\alpha = 10\|\mathbf{x}_{\rm sol}\|_\infty$ suffices[1], with recovery guarantees under RIP, NSP, and other conditions.
- Although $\alpha > \alpha^0$ lead to the same and unique primal solution $\mathbf{x}^*$, the dual solution set $\mathcal{Y}^*$ is a *multi-set* and it depends on α.
- Dynamically adjusting α may *not* be a good idea for the dual algorithms.

[1] Lai and Yin [2012]

Exact regularization

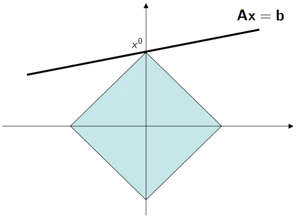
Theorem (Friedlander and Tseng [2007], Yin [2010])

There exists a finite $\alpha^0 > 0$ *such that whenever* $\alpha > \alpha^0$, *the solution to*

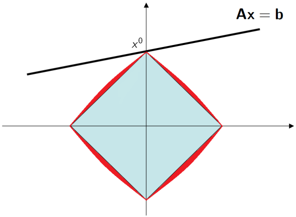
$$\text{(L1+LS)} \qquad \min \|\mathbf{x}\|_1 + \frac{1}{2\alpha}\|\mathbf{x}\|_2^2 \quad \textit{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}$$

is also a solution to

$$\text{(L1)} \qquad \min \|\mathbf{x}\|_1 \quad \textit{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}.$$



L1



L1+LS

L1+LS in compressive sensing

If in some scenario, L1 gives exact or stable recovery provided that

$$\#\text{measurements } m \geq C \cdot F(\text{signal dim } n, \text{ signal sparsity } k).$$

Then, adding $\frac{1}{2\alpha}\|\mathbf{x}\|_2^2$, the condition becomes

$$\#\text{measurements } m \geq (C + O(\frac{1}{2\alpha})) \cdot F(\text{signal dim } n, \text{ signal sparsity } k).$$

Theorem (exact recovery, Lai and Yin [2012])

Under the assumptions

1. $\mathbf{x}^0$ *is* k*-sparse, and* $\mathbf{A}$ *satisfies RIP with* $\delta_{2k} \leq 0.4404$*, and*
2. $\alpha \geq 10\|\mathbf{x}^0\|_\infty$,

(L1+LS) *uniquely recovers* $\mathbf{x}^0$.

The bound on δ_{2k} is tighter than that for ℓ_1, and it depends on the bound on α.

Stable recovery

For approximately sparse signals and/or noisy measurements, consider:

$$\min_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 + \frac{1}{2\alpha}\|\mathbf{x}\|_2^2 : \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \leq \sigma \right\} \tag{1}$$

Theorem (stable recovery, Lai and Yin [2012])

Let $\mathbf{x}^0$ be an arbitrary vector, $\mathcal{S} = \{$largest k entries of $\mathbf{x}^0\}$, and $\mathcal{Z} = \mathcal{S}^C$. Let $\mathbf{b} := A\mathbf{x}^0 + \mathbf{n}$, where $\mathbf{n}$ is arbitrary noisy. If $\mathbf{A}$ satisfies RIP with $\delta_{2k} \leq 0.3814$ and $\alpha \geq 10\|\mathbf{x}^0\|_\infty$, then the solution $\mathbf{x}^*$ of (1) with $\sigma = \|\mathbf{n}\|_2$ satisfies

$$\|\mathbf{x}^* - \mathbf{x}^0\|_2 \leq \bar{C}_1 \cdot \|\mathbf{n}\|_2 + \bar{C}_2 \cdot \|\mathbf{x}^0_{\mathcal{Z}}\|_1/\sqrt{k},$$

where $\bar{C}_1$, and $\bar{C}_2$ are constants depending on δ_{2k}.

Implementation

- Since dual is C^1 and unconstrained, various first-order techniques apply
  - accelerated gradient descent[2]
  - Barzilai-Borwein step size[3] / (non-monotone) line search[4]
  - Quasi Newton method (with some cautions since it is not C^2)
- Fits the dual-decomposition framework, easy to parallelize (later lecture)
- Results generalize to ℓ_1-like functions.
- Matlab codes and demos at
  `www.caam.rice.edu/~optimization/linearized_bregman/`

[2] Nesterov [1983]

[3] Barzilai and Borwein [1988]

[4] Zhang and Hager [2004]

Review: convex conjugate

Recall convex conjugate (the Legendre transform):

$$f^*(\mathbf{y}) = \sup_{\mathbf{x} \in \mathrm{dom} f} \{\mathbf{y}^T \mathbf{x} - f(\mathbf{x})\}$$

- f^* is convex since it is point-wise maximum of linear functions
- if f is proper, closed, convex, then $(f^*)^* = f$, i.e.,

$$f(\mathbf{x}) = \sup_{\mathbf{y} \in \mathrm{dom} f^*} \{\mathbf{y}^T \mathbf{x} - f^*(\mathbf{y})\}.$$

Examples:

- $f(\mathbf{x}) = \iota_{\mathcal{C}}(\mathbf{x})$, indicator function, and $f^*(\mathbf{y}) = \sup_{\mathbf{x} \in \mathcal{C}} \mathbf{y}^T \mathbf{x}$, support function
- $f(\mathbf{x}) = \iota_{\{-1 \le \mathbf{x} \le 1\}}$ and $f^*(\mathbf{y}) = \|\mathbf{y}\|_1$
- $f(\mathbf{x}) = \iota_{\{\|\mathbf{x}\|_2 \le 1\}}$ and $f^*(\mathbf{y}) = \|\mathbf{y}\|_2$
- lots of smooth examples ......

Review: convex conjugate

One can introduce an alternative representation via convex conjugacy

$$f(\mathbf{x}) = \sup_{\mathbf{y} \in \mathrm{dom} f^*} \{\mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{b}) - h^*(\mathbf{y})\} = h(\mathbf{A}\mathbf{x} + \mathbf{b}).$$

Example:

- let $\mathcal{C} = \{\mathbf{y} = [\mathbf{y}_1; \mathbf{y}_2] : \mathbf{y}_1 + \mathbf{y}_2 = 1,\ \mathbf{y}_1, \mathbf{y}_2 \geq 0\}$ and

$$\mathbf{A} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

  Since $\|\mathbf{x}\|_1 = \sup_{\mathbf{y}}\{(\mathbf{y}_1 - \mathbf{y}_2)^T\mathbf{x} - \iota_{\mathcal{C}}(\mathbf{y})\} = \sup_{\mathbf{y}}\{\mathbf{y}^T\mathbf{A}\mathbf{x} - \iota_{\mathcal{C}}(\mathbf{y})\}$, we have

$$\|\mathbf{x}\|_1 = \iota_{\mathcal{C}}^*(\mathbf{A}\mathbf{x})$$

  where $\iota_{\mathcal{C}}^*([\mathbf{x}_1; \mathbf{x}_2]) = \max\{\mathbf{x}_1, \mathbf{x}_2\}$ entry-wise.

Dual smoothing

Idea: strongly convexify h^* $\Longrightarrow$ f becomes differentiable and has Lipschitz ∇f

Represent f using h^*:

$$f(\mathbf{x}) = \sup_{\mathbf{y} \in \mathrm{dom} h^*} \{\mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{b}) - h^*(\mathbf{y})\}$$

Strongly convexify h^* by adding strongly convex function d:

$$\hat{h}^*(\mathbf{y}) = h^*(\mathbf{y}) + \mu d(\mathbf{y})$$

Obtain a differentiable approximation:

$$f_\mu(\mathbf{x}) = \sup_{\mathbf{y} \in \mathrm{dom} h^*} \{\mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{b}) - \hat{h}^*(\mathbf{y})\}$$

$f_\mu(\mathbf{x})$ is differentiable since $h^*(\mathbf{y}) + \mu d(\mathbf{y})$ is strongly convex.

Example: augmented ℓ_1

- primal problem $\min\{\|\mathbf{x}\|_1 : \mathbf{A}\mathbf{x} = \mathbf{b}\}$
- dual problem: $\max\{\mathbf{b}^T\mathbf{y} + \iota_{[-1,1]^n}(\mathbf{A}^T\mathbf{y})\}$
- $f(\mathbf{y}) = \iota_{[-1,1]^n}(\mathbf{y})$ is non-differentiable
- let $f^*(\mathbf{x}) = \|\mathbf{x}\|_1$ and represent $f(\mathbf{y}) = \sup_{\mathbf{x}}\{\mathbf{y}^T\mathbf{x} - f^*(\mathbf{x})\}$,
- add $\frac{\mu}{2}\|\mathbf{x}\|_2$ to $f^*(\mathbf{x})$ and obtain

$$f_\mu(\mathbf{y}) = \sup_{\mathbf{x}}\{\mathbf{y}^T\mathbf{x} - (\|\mathbf{x}\|_1 + \frac{\mu}{2}\|\mathbf{x}\|_2^2)\} = \frac{1}{2\mu}\|\mathbf{y} - \mathrm{Proj}_{[-1,1]^n}(\mathbf{y})\|_2^2$$

- $f_\mu(\mathbf{y})$ is differentiable; $\nabla f_\mu(\mathbf{y}) = \frac{1}{\mu}\,\mathrm{shrink}(\mathbf{y})$.
- On the other hand, we can also smooth $f^*(\mathbf{x}) = \|\mathbf{x}\|_1$ and obtain differentiable $f_\mu^*(\mathbf{x})$ by adding $d(\mathbf{y})$ to $f(\mathbf{y})$. (see the next slide ...)

Example: smoothed absolute value

- Recall
$$f^*(x) = |x| = \sup_y \{yx - \iota_{[-1,1]}(y)\}$$

Let $d(y) = y^2/2$

$$f_\mu^* = \sup_y \{yx - (\iota_{[-1,1]}(y) + \mu y^2/2)\} = \begin{cases} x^2/(2\mu), & |x| \le \mu, \\ |x| - \mu/2, & |x| > \mu, \end{cases}$$

which is the Huber function

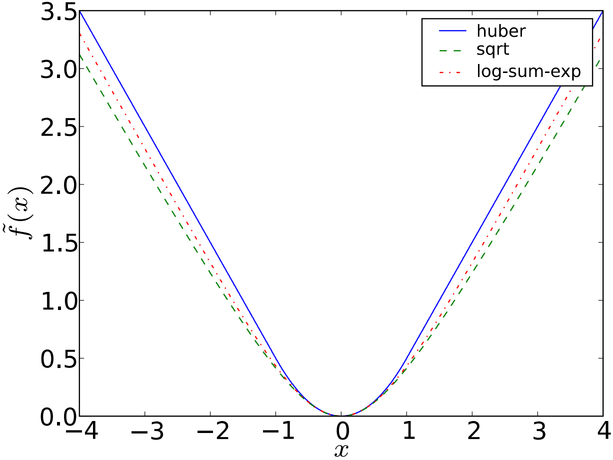
- let $d(y) = 1 - \sqrt{1 - y^2}$
$$f_\mu^* = \sup_y \{yx - (\iota_{[-1,1]}(y) - \mu\sqrt{1-y^2})\} - \mu = \sqrt{x^2 + \mu^2} - \mu.$$

- Recall
$$|x| = \sup_{\mathbf{y}} \{(y_1 - y_2)x - \iota_{\mathcal{C}}(y)\}$$

for $\mathcal{C} = \{\mathbf{y} : y_1 + y_2 = 1,\ y_1, y_2 \ge 0\}$. Let $d(y) = y_1 \log y_1 + y_2 \log y_2 + \log 2$

$$f_\mu^*(x) = \sup_{\mathbf{y}} \{(y_1 - y_2)x - (\iota_{\mathcal{C}}(y) + \mu d(y))\} = \mu \log \frac{e^{x/\mu} + e^{-x/\mu}}{2}.$$

Compare three smoothed functions



Courtesy of L. Vandenberghe

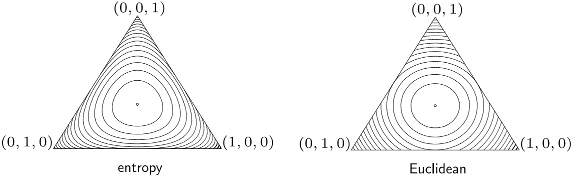
Example: smoothed maximum eigenvalue

Let $\mathcal{C} = \{\mathbf{Y} \in \mathcal{S}^n : \mathrm{tr}\mathbf{Y} = 1, \mathbf{Y} \succeq 0\}$. Let $\mathbf{X} \in \mathcal{S}^n$

$$f(\mathbf{X}) = \lambda_{\max}(\mathbf{X}) = \sup_{\mathbf{Y}} \{\mathbf{Y} \bullet \mathbf{X} - \iota_{\mathcal{C}}(\mathbf{Y})\}$$

Negative entropy of $\{\lambda_i(\mathbf{Y})\}$:

$$d(\mathbf{Y}) = \sum_{i=1}^{n} \lambda_i(\mathbf{Y}) \log \lambda_i(\mathbf{Y}) + \log n$$



(Courtesy of L. Vandenberghe)

Smoothed function

$$f_\mu(\mathbf{X}) = \sup_{\mathbf{Y}} \{\mathbf{Y} \bullet \mathbf{X} - (\iota_{\mathcal{C}}(\mathbf{Y}) + \mu d(\mathbf{Y}))\} = \mu \log \left(\sum_{i=1}^{n} e^{\lambda_i(\mathbf{X})/\mu} \right) - \mu \log n$$

Application: smoothed minimization[5]

Instead of solving

$$\min f(\mathbf{x}),$$

solve

$$\min f_\mu(\mathbf{x}) = \sup_{\mathbf{y}\in\mathrm{dom}h^*} \{\mathbf{y}^T(\mathbf{A}\mathbf{x}+\mathbf{b}) - [h^*(\mathbf{y}) + \mu d(\mathbf{y})]\}$$

by gradient descent, with acceleration, line search, etc......

Gradient is given by:

$$\nabla f_\mu(\mathbf{x}) = \mathbf{A}^T\bar{\mathbf{y}}, \quad \text{where } \bar{\mathbf{y}} = \operatorname*{arg\,max}_{\mathbf{y}\in\mathrm{dom}h^*}\{\mathbf{y}^T(\mathbf{A}\mathbf{x}+\mathbf{b}) - [h^*(\mathbf{y}) + \mu d(\mathbf{y})]\}.$$

If $d(\mathbf{y})$ is strongly convex with modulus $\nu > 0$, then

- $h^*(\mathbf{y}) + \mu d(\mathbf{y})$ is strongly convex with modulus at least $\mu\nu$
- $\nabla f_\mu(\mathbf{x})$ is Lipschitz continuous with constant no more than $\|\mathbf{A}\|^2/\mu\nu$.

Error control by bounding $|f_\mu(\mathbf{x}) - f(\mathbf{x})|$ or $\|\mathbf{x}_\mu^* - \mathbf{x}^*\|$.

---

[5] Nesterov [2005]

Augmented Lagrangian (a.k.a. method of multipliers)

Augment $\mathcal{L}(\mathbf{x};\mathbf{y}^k) = f(\mathbf{x}) - (\mathbf{y}^k)^T(\mathbf{A}\mathbf{x}-\mathbf{b})$ by adding $\frac{c}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$.

Augmented Lagrangian:

$$\mathcal{L}_A(\mathbf{x};\mathbf{y}^k) = f(\mathbf{x}) - (\mathbf{y}^k)^T(\mathbf{A}\mathbf{x}-\mathbf{b}) + \frac{c}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$$

Iteration:

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} \mathcal{L}_A(\mathbf{x};\mathbf{y}^k) \\ \mathbf{y}^{k+1} &= \mathbf{y}^k + c(\mathbf{b} - \mathbf{A}\mathbf{x}^{k+1}) \end{aligned}$$

from $k=0$ and $\mathbf{y}^0 = \mathbf{0}$. $c > 0$ can change.

The objective of the first step is convex in $\mathbf{x}$, if $f(\cdot)$ is convex, and linear in $\mathbf{y}$.

Equivalent to the dual implicit (proximal) iteration.

Augmented Lagrangian (a.k.a. method of multipliers)

Recall KKT conditions (omitting the complementarity part):

$$\begin{aligned} \text{(primal feasibility)} \quad & \mathbf{A}\mathbf{x}^* = \mathbf{b} \\ \text{(dual feasibility)} \quad & 0 \in \partial f(\mathbf{x}^*) - \mathbf{A}^T \mathbf{y}^* \end{aligned}$$

Compare the 2nd condition with the optimality condition of ALM subproblem

$$0 \in \partial f(\mathbf{x}^{k+1}) - \mathbf{A}^T(\mathbf{y}^k + c(\mathbf{b} - \mathbf{A}\mathbf{x}^{k+1})) = \partial f(\mathbf{x}^{k+1}) - \mathbf{A}^T \mathbf{y}^{k+1}$$

Conclusion: dual feasibility is maintained for $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})$ for all k.

Also, it "works toward" primal feasibility:

$$-(\mathbf{y}^k)^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \frac{c}{2}\|\mathbf{A} - \mathbf{b}\|_2^2 = \frac{c}{2}\langle \mathbf{A}\mathbf{x} - \mathbf{b}, \sum_{i=1}^{k}(\mathbf{A}\mathbf{x}^i - \mathbf{b}) + (\mathbf{A}\mathbf{x} - \mathbf{b})\rangle$$

It keeps adding penalty to the violation of $\mathbf{A}\mathbf{x} = \mathbf{b}$. In the limit, $\mathbf{A}\mathbf{x}^* = \mathbf{b}$ holds (for polyhedral $f(\cdot)$ in finitely many steps).

Augmented Lagrangian (a.k.a. Method of Multipliers)

Compared to dual (sub)gradient ascent

Pros:

- It converges for nonsmooth and extended-value f (thanks to the proximal term)

Cons:

- If f is nice and dual ascent works, it may be slower than dual ascent since the subproblem is more difficult
- The term $\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ in the $\mathbf{x}$-subproblem couples different blocks of $\mathbf{x}$ (unless $\mathbf{A}$ has a block-diagonal structure)

Application/alternative derivation: Bregman iterative regularization[6]

[6] Osher, Burger, Goldfarb, Xu, and Yin [2005], Yin, Osher, Goldfarb, and Darbon [2008]

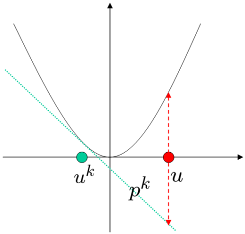
Bregman Distance

Definition: let $r(\mathbf{x})$ be a convex function

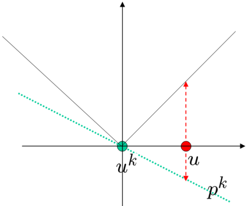
$$D_r(\mathbf{x},\mathbf{y};\mathbf{p}) = r(\mathbf{x}) - r(\mathbf{y}) - \langle \mathbf{p}, \mathbf{x}-\mathbf{y}\rangle, \quad \text{where } \mathbf{p} \in \partial r(\mathbf{y})$$

Not a distance but has a flavor of distance.

Examples: $D_{\ell_2^2}(u,u^k;p^k)$ versus $D_{\ell_1}(u,u^k;p^k)$



differentiable case



non-differentiable case

Bregman iterative regularization

Iteration

$$\begin{aligned}\mathbf{x}^{k+1} &= \arg\min D_r(\mathbf{x}, \mathbf{x}^k; \mathbf{p}^k) + g(\mathbf{x}),\\ \mathbf{p}^{k+1} &= \mathbf{p}^k - \nabla g(\mathbf{x}^{k+1}),\end{aligned}$$

starting $k = 0$ and $(\mathbf{x}^0, \mathbf{p}^0) = (\mathbf{0}, \mathbf{0})$. The update of $\mathbf{p}$ follows from

$$0 \in \partial r(\mathbf{x}^{k+1}) - \mathbf{p}^k + \nabla g(\mathbf{x}^{k+1}),$$

so in the next iteration, $D_r(\mathbf{x}, \mathbf{x}^{k+1}; \mathbf{p}^{k+1})$ is well defined.

Bregman iteration is related, or equivalent, to

1. Proximal point iteration
2. Residual addback iteration
3. Augmented Lagrangian iteration

Bregman and Proximal Point

If $r(\mathbf{x}) = \frac{c}{2}\|\mathbf{x}\|_2^2$, Bregman method reduces to the classical proximal point method

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} g(\mathbf{x}) + \frac{c}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2.$$

Hence, Bregman iteration with function r is r-proximal algorithm for $\min g(\mathbf{x})$

Traditional, $r = \ell_2^2$ or other smooth functions is used as the proximal function. Few uses non-differentiable convex functions like ℓ_1 to generate the proximal term because ℓ_1 is not stable!

But using ℓ_1 proximal function has interesting properties.

Bregman convergence

If $g(\mathbf{x}) = p(\mathbf{A}\mathbf{x} - \mathbf{b})$ is a strictly convex function that penalizes $\mathbf{A}\mathbf{x} = \mathbf{b}$, which is feasible, then the iteration

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} D_r(\mathbf{x}, \mathbf{x}^k; \mathbf{p}^k) + g(\mathbf{x})$$

converges to a solution of

$$\min r(\mathbf{x}), \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}.$$

Recall, the augmented Lagrangian algorithm also has a similar property.

Next we restrict our analysis to

$$g(\mathbf{x}) = \frac{c}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Residual addback iteration

If $g(\mathbf{x}) = \frac{c}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, we can derive the *equivalent* iteration

$$\begin{aligned}
\mathbf{b}^{k+1} &= \mathbf{b} + (\mathbf{b}^k - \mathbf{A}\mathbf{x}^k),\\
\mathbf{x}^{k+1} &= \arg\min r(\mathbf{x}) + \frac{c}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}^{k+1}\|_2^2.
\end{aligned}$$

Interpretation:

- every iteration, the residual $\mathbf{b} - \mathbf{A}\mathbf{x}^k$ is added back to $\mathbf{b}^k$;
- every subproblem is identical but with different data.

Bregman = Residual addback

Equivalence the two forms is given by $\mathbf{p}^k = -c\mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b}^k)$.

Proof by induction. Assume both iterations have the same $\mathbf{x}^k$ so far

($c = \delta$ below)

$$\begin{aligned}
\mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} r(\mathbf{x}) - \langle \mathbf{p}^k, \mathbf{x} \rangle + \frac{\delta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\
&= \arg\min_{\mathbf{x}} r(\mathbf{x}) + \frac{\delta}{2}\|\mathbf{A}\mathbf{x} - \big(\mathbf{b} + (\mathbf{b}^k - \mathbf{A}\mathbf{x}^k)\big)\|_2^2 \\
&= \arg\min_{\mathbf{x}} r(\mathbf{x}) + \frac{\delta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}^{k+1}\|_2^2,
\end{aligned}$$

$$\begin{aligned}
\mathbf{p}^{k+1} &= \mathbf{p}^k - \delta\mathbf{A}^T(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \\
&= -\delta\mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b}^k) - \delta\mathbf{A}^T(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \\
&= -\delta\mathbf{A}^T\left(\mathbf{A}\mathbf{x}^{k+1} - \big(\mathbf{b} + (\mathbf{b}^k - \mathbf{A}\mathbf{x}^k)\big)\right) \\
&= -\delta\mathbf{A}^T(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}^{k+1}).
\end{aligned}$$

Bregman = Residual addback = Augmented Lagrangian

Assume $f(\mathbf{x}) = r(\mathbf{x})$ and $g(\mathbf{x}) = \frac{c}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$.

Addback iteration:

$$\mathbf{b}^{k+1} = \mathbf{b}^k + (\mathbf{b} - \mathbf{A}\mathbf{x}^k) = \cdots = \mathbf{b}^0 + \sum_{i=0}^{k}(\mathbf{b} - \mathbf{A}\mathbf{x}^i).$$

Augmented Lagrangian iteration:

$$\mathbf{y}^{k+1} = \mathbf{y}^k + c(\mathbf{b} - \mathbf{A}\mathbf{x}^{k+1}) = \cdots = \mathbf{y}^0 + c\sum_{i=0}^{k+1}(\mathbf{b} - \mathbf{A}\mathbf{x}^i).$$

Bregman iteration:

$$\mathbf{p}^{k+1} = \mathbf{p}^k + c\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^{k+1}) = \cdots = \mathbf{p}^0 + c\sum_{i=0}^{k+1}\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^i).$$

Their equivalence is established by

$$\mathbf{y}^k = c\mathbf{b}^{k+1} \quad \text{and} \quad \mathbf{p}^k = \mathbf{A}^T\mathbf{y}^k, \quad k = 0, 1, \ldots$$

and initial values $\mathbf{x}^0 = 0$, $\mathbf{b}^0 = 0$, $\mathbf{p}^0 = 0$, $\mathbf{y}^0 = 0$.

Residual addback in a regularization perspective

Adding the residual $\mathbf{b} - \mathbf{A}\mathbf{x}^k$ back to $\mathbf{b}^k$ is somewhat counter intuitive. In the regularized least-squares problem

$$\min_{\mathbf{x}} r(\mathbf{x}) + \frac{c}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

the residual $\mathbf{b} - \mathbf{A}\mathbf{x}^k$ contains both unwanted error and wanted features.

The question is how to extract the features out of $\mathbf{b} - \mathbf{A}\mathbf{x}^k$.

An intuitive approach is to solve

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} r'(\mathbf{y}) + \frac{c'}{2}\|\mathbf{A}\mathbf{y} - (\mathbf{b} - \mathbf{A}\mathbf{x}^k)\|_2$$

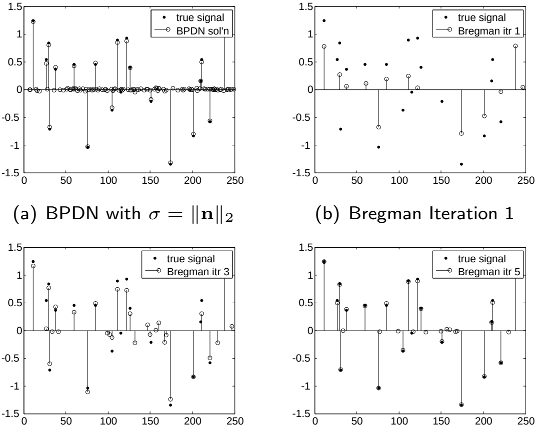
and then let

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \mathbf{y}^*.$$

However, the addback iteration keeps the same r and adds residuals back to $\mathbf{b}^k$. Surprisingly, this gives good *denoising* results.

Good denoising effect

Compare to addback (Bregman) to BPDN: $\min\{\|\mathbf{x}\|_1 : \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \le \sigma\}$ where $\mathbf{b} = \mathbf{A}\mathbf{x}^o + \mathbf{n}$.



(a) BPDN with $\sigma = \|\mathbf{n}\|_2$

(b) Bregman Iteration 1

(c) Bregman Iteration 3

(d) Bregman Iteration 5

Good denoising effect

From this example,

- given noisy observations and starting being over regularized, some *intermediate* solutions have better fitting and less noise than

$$\mathbf{x}_\sigma^* = \arg\min\{\|\mathbf{x}\|_1 : \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \leq \sigma\},$$

  in fact, no matter how σ is chosen.

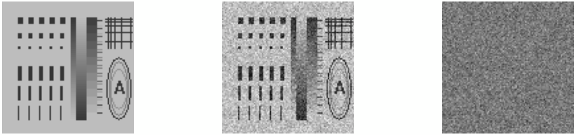
- the addback intermediate solutions are *not* on the path of $\mathbf{x}_\sigma^* = \arg\min\{\|\mathbf{x}\|_1 : \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \leq \sigma\}$ by varying $\sigma > 0$.
- Recall if the add iteration is continued, it will converge to the solution of

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1, \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}.$$

Example of total variation denoising

Problem: $\mathbf{u}$ a 2D image, $\mathbf{b}$ noisy image. Noise is Gaussian.

(a) original (b) noisy f, SNR=7.4 (c) noise+128, δ=40



Apply the addback iteration with $\mathbf{A} = I$ and

$$r(\mathbf{u}) = \mathrm{TV}(\mathbf{u}) = \|\nabla \mathbf{u}\|_1.$$

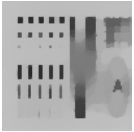
The subproblem has the form

$$\min_{\mathbf{u}} \mathrm{TV}(\mathbf{u}) + \frac{\delta}{2}\|\mathbf{u} - \mathbf{f}^k\|_2^2,$$

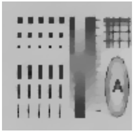
where initial $\mathbf{f}^0$ is a *noisy* observation of $\mathbf{u}_{\text{original}}$.

(d) u_1: 1st step, λ=0.002

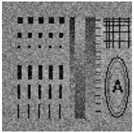
(e) u_2: 2nd step



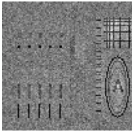
(f) u_3: 3rd step



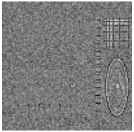
(g) $f-u_1+128$



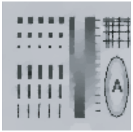
(h) $f-u_2+128$



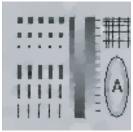
(i) $f-u_3+128$



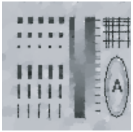
(j) u_4: 4th step



(k) u_5: 5th step



(l) u_6: 6th step

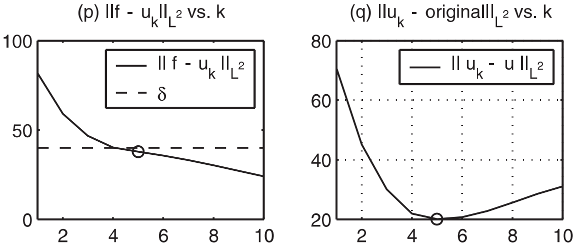


(m) $f - u_4+128$

(n) $f - u_5+128$

(o) $f - u_6+128$

When to stop the addback iteration?



The 2nd curve shows that optimal stopping iteration is 5.

The 1st curve shows that residual just gets across noise level.

Solution: stop when $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|_2 \approx$ noise level. Some theoretical results exist[7]

[7] Osher, Burger, Goldfarb, Xu, and Yin [2005]

Numerical stability

The addback/augmented-Lagrangian iterations are *more stable* the Bregman iteration, though they are same same on paper.

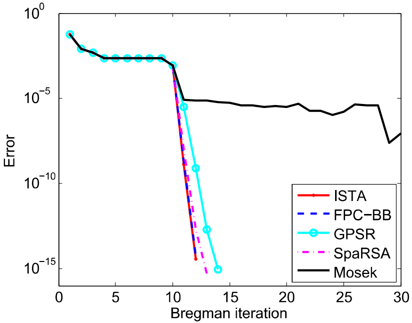
Two reasons:

- $\mathbf{p}^k$ in the Bregman iteration may gradually lose the property of $\in \mathcal{R}(\mathbf{A}^T)$ due to *accumulated round-off errors*; the other two iterations *explicitly* multiply $\mathbf{A}^T$ and are thus more stable
- The addback iteration enjoy errors forgetting and error cancellation when r is polyhedral.

Numerical stability for ℓ_1 minimization

ℓ_1-based errors forgetting simulation:

- use five different subproblem solvers for ℓ_1 Bregman iterations
- for each subproblem, stop solver at accuracy 10^{-6}
- track and plot $\frac{\|\mathbf{x}^k - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2}$ vs iteration k



Errors made in subproblems get cancelled iteratively. See Yin and Osher [2012].

Summary

- Dual gradient method, after smoothing
- Exact smoothing for ℓ_1
- Smooth a function by adding a strongly convex function to its convex conjugate
- Augmented Lagrangian, Bregman, and residual addback iterations; their equivalence
- Better denoising result of residual addback iteration
- Numerical stability, error forgetting and cancellation of residual addback iteration; numerical instability of Bregman iteration

(Incomplete) References:

R Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

M.-J. Lai and W. Yin. Augmented ℓ_1 and nuclear-norm models with a globally linearly convergent algorithm. *Submitted to SIAM Journal on Imaging Sciences*, 2012.

M.P. Friedlander and P. Tseng. Exact regularization of convex programs. *SIAM Journal on Optimization*, 18(4):1326–1350, 2007.

W. Yin. Analysis and generalizations of the linearized Bregman method. *SIAM Journal on Imaging Sciences*, 3(4):856–877, 2010.

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

J. Barzilai and J.M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

Hongchao Zhang and William W Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal on Optimization*, 14(4): 1043–1056, 2004.

Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *SIAM Journal on Multiscale Modeling and Simulation*, 4(2):460–489, 2005.

W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for l1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008.

W. Yin and S. Osher. Error forgetting of Bregman iteration. *Journal of Scientific Computing*, 54(2):684–698, 2012.