

ECE 532 - lecture 18 - proximal iterations

①

We saw the Landweber iteration for least squares:

$$x_{k+1} = x_k - \tau A^T (Ax_k - y)$$

let's see how we can modify it to work with regularized LS.

$$\min_x \|Ax - y\|^2 + \lambda R(x)$$

*Essential idea : what if $A = I$ and the regularizer can be "separated": $R(x) = \sum_{i=1}^n r_i(x_i)$. Then the optimization problem becomes to minimize:

$$\|x - y\|^2 + \lambda R(x) = \sum_{i=1}^n \left\{ (x_i - y_i)^2 + \lambda r_i(x_i) \right\},$$

so we can optimize over x by separately optimizing over each x_i .

Note: the regularizers of interest to us are separable for L_1, L_2 .

$$\begin{aligned} \|x\|^2 &= x_1^2 + x_2^2 + \dots + x_n^2 \\ \|x\|_1 &= |x_1| + |x_2| + \dots + |x_n| \end{aligned} \quad \left. \begin{array}{l} \text{separable.} \end{array} \right.$$

$\|x\|_\infty$ is not separable

But $A \neq I$ so we also have to deal with that issue.. We will find a separable approximation:

Define $L(x) = \|Ax - y\|^2 + R(x)$,

(2)

$$\begin{aligned}
 L(x) &= \|Ax - y\|^2 + \lambda R(x) \\
 &= \|Ax_k - y + Ax - Ax_k\|^2 + \lambda R(x) \\
 &= \underbrace{\|Ax_k - y\|^2}_{C \text{ (constant)}} + 2(Ax_k - y)^T A(x - x_k) + \underbrace{\|A(x - x_k)\|^2}_{\leq \|A\|^2 \|x - x_k\|^2} + \lambda R(x).
 \end{aligned}$$

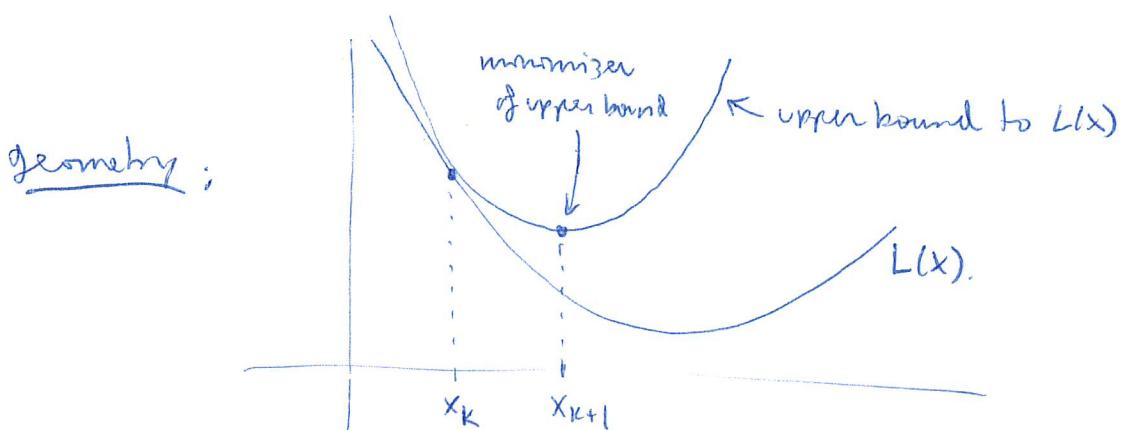
Suppose we choose $0 < \tau < \frac{1}{\|A\|^2}$. Then

$$L(x) \leq C + 2(Ax_k - y)^T A(x - x_k) + \tau^{-1} \|x - x_k\|^2 + \lambda R(x).$$

Note: This upper bound touches $L(x)$ at $x = x_k$ (they are equal).

Let's choose x to minimize this upper bound, and this will be our next iterate. i.e.

$$\begin{aligned}
 x_{k+1} &= \arg \min_x \left\{ 2(Ax_k - y)^T A(x - x_k) + \tau^{-1} \|x - x_k\|^2 + \lambda R(x) \right\} \\
 &= \arg \min_x \left\{ 2\tau(Ax_k - y)^T A(x - x_k) + \|x - x_k\|^2 + \lambda \tau R(x) \right\}.
 \end{aligned}$$



$$x_{k+1} = \underset{x}{\operatorname{argmin}} \left\{ 2\tau(Ax_k - y)^T A(x - x_k) + \|x - x_k\|^2 + \tau\lambda R(x) \right\}, \quad (3)$$

define $z_k = x_k - \tau A^T(Ax_k - y)$. (the Landweber iterate).

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \left\{ 2(x_k - z_k)(x - x_k) + \|x - x_k\|^2 + \tau\lambda R(x) \right\}.$$

$$= \underset{x}{\operatorname{argmin}} \left\{ \| (x - x_k) + (x_k - z_k) \|^2 - \underbrace{\| x_k - z_k \|^2}_{\text{const.}} + \tau\lambda R(x) \right\}$$

$$= \underset{x}{\operatorname{argmin}} \left\{ \|x - z_k\|^2 + \tau\lambda R(x) \right\}. \quad \begin{bmatrix} \text{this is called the} \\ \text{proximal operator of } R \end{bmatrix}$$

If $R(x) = 0$ or $\lambda = 0$, we recover $x_{k+1} = z_k$ (Landweber!).

but otherwise, we have a new formula. But behold... it is now a separable optimization problem!

$$\Rightarrow (x_{k+1})_i = \underset{x}{\operatorname{argmin}} \left\{ (x - (z_k)_i)^2 + \tau\lambda r_i(x) \right\}.$$

so if we have an efficient way of solving the scalar problem $\underset{x}{\operatorname{argmin}} \left\{ (x - z)^2 + \lambda r_i(x) \right\}$ then this is no more

difficult than computing z_k , the Landweber iterate.

This variant is called the proximal point algorithm

(4)

L_2 -regularization

we must solve: $\underset{x}{\operatorname{argmin}} \left\{ (x-z)^2 + \tau \lambda x^2 \right\}.$

$$\frac{d}{dx} = 0 \Rightarrow 2(x-z) + 2\tau \lambda x = 0$$

$$\Rightarrow x = \frac{1}{1+\tau \lambda} z.$$

$$\text{so } (x_{k+1})_i = \frac{1}{1+\tau \lambda} (z_k)_i. \Rightarrow x_{k+1} = \frac{1}{1+\tau \lambda} z_k.$$

i.e.
$$x_{k+1} = \frac{1}{1+\tau \lambda} (x_k - \tau A^T(Ax_k - y)).$$

just a scaled version of Landweber.

Note: if we apply Landweber directly to the LS problem, i.e.

$$\text{use: } \|Ax-y\|^2 + \lambda \|x\|^2 = \left\| \begin{bmatrix} A \\ \sqrt{\lambda} I \end{bmatrix} x - \begin{bmatrix} y \\ 0 \end{bmatrix} \right\|^2$$

then; we obtain $x_{k+1} = x_k - \tau \left((A^T A + \lambda I) x_k - A^T y \right)$

$$\text{or } x_{k+1} = (x_k - \tau A^T(Ax_k - y)) - \tau \lambda x_k.$$

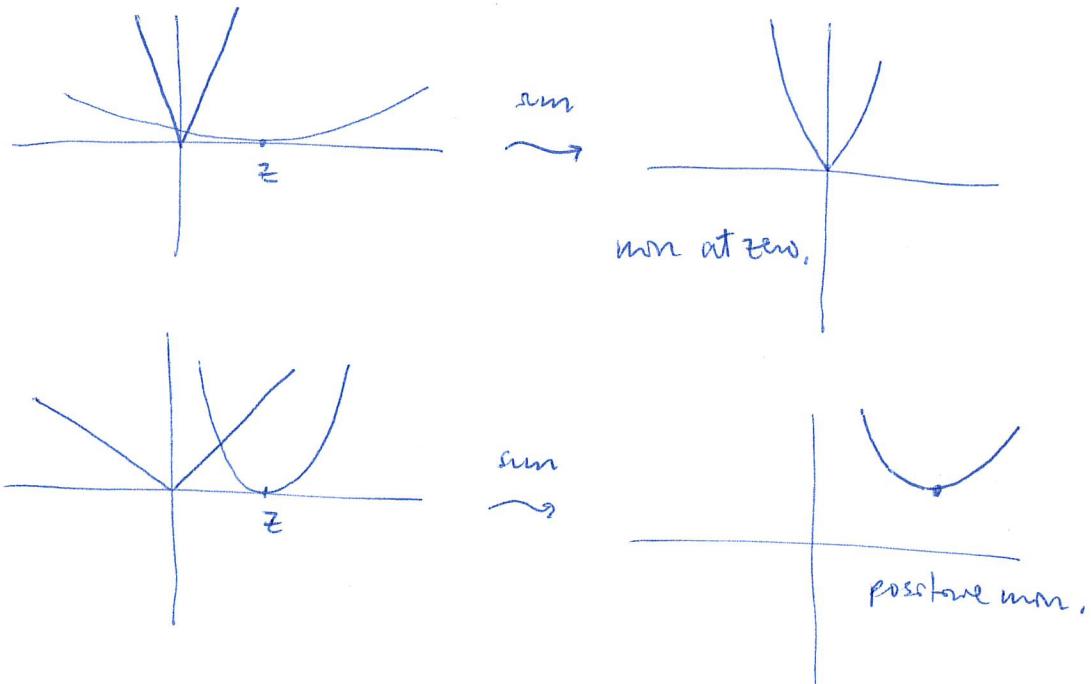
This is not the same as doing the procedure we described above.

(5)

L_1 - regularization

we must solve: $\underset{x}{\operatorname{argmin}} \left\{ (x-z)^2 + \lambda |x| \right\}.$

several cases:



if $x > 0$, $(x-z)^2 + \lambda |x| = (x-z)^2 + \lambda x$ (minimized at $z - \frac{\lambda}{2}$).

if $x < 0$ $(x-z)^2 + \lambda |x| = (x-z)^2 - \lambda x$ (minimized at $z + \frac{\lambda}{2}$).

$$\text{Therefore, } x_{\text{opt}} = \begin{cases} z + \frac{\lambda}{2} & \text{if } z < -\frac{\lambda}{2} \\ 0 & \text{if } -\frac{\lambda}{2} < z < \frac{\lambda}{2} \\ z - \frac{\lambda}{2} & \text{if } z > \frac{\lambda}{2}. \end{cases}$$

more compact: $x_{\text{opt}} = \operatorname{sign}(z) \left(|z| - \frac{\lambda}{2} \right)_+$

where $(\cdot)_+$ is the "soft-thresholding" function

$$(x)_+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} = \max(x, 0)$$

(6)

 L_1 - cont'd

$$(x_{k+1})_i = \arg \min_x \left\{ (x - (z_k)_i)^2 + \tau \lambda |x| \right\}.$$

$$= \text{sign}((z_k)_i) \left(|(z_k)_i| - \frac{\tau \lambda}{2} \right)_+.$$

So in general,

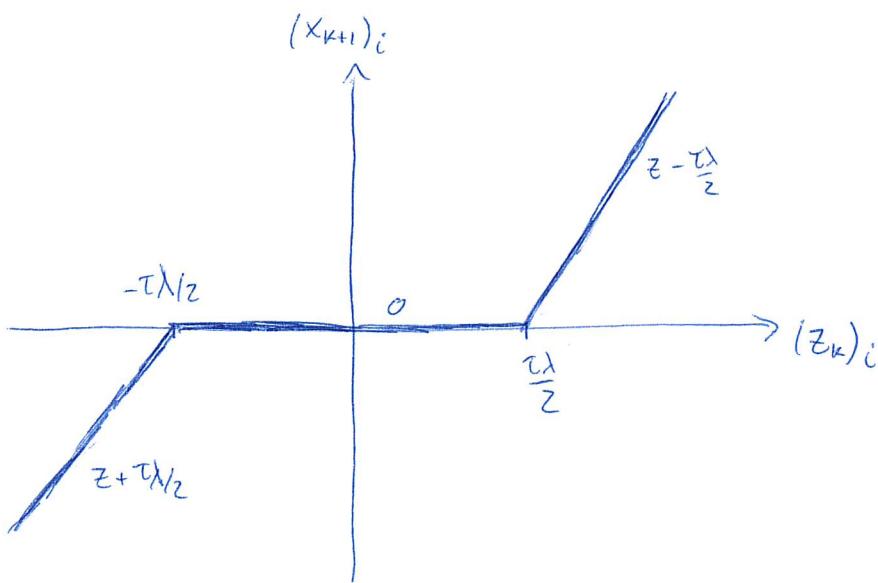
$$x_{k+1} = \text{sgn}(z_k) \circ \left(|z_k| - \frac{\tau \lambda}{2} \right)_+,$$

↑ elementwise multiplication.

so the algorithm is:

1. compute landweber $z_k = x_k - \tau A^T(Ax_k - y)$,
2. soft-threshold $x_{k+1} = \text{sgn}(z_k) \circ \left(|z_k| - \frac{\tau \lambda}{2} \right)_+$,
3. repeat.

This is called "iterative soft-thresholding algorithm" (ISTA).



★ ISTA iterates
are themselves sparse!

This is a big bonus,
because partially converged
solutions still have the sparsity
structure we seek.