

# ECE 532 - Lecture 16 - regularization examples

We have been looking at solving regularized problems:

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|_2^2 + \lambda r(x)$$

where  $r(x)$  is chosen to be: (other choices exist also!)

$$r(x) = \begin{cases} \|x\|_1 & : \text{lasso, promotes sparsity} \\ \|x\|_2^2 & : \text{tikhonov/ridge regression, small norm solution.} \\ \|x\|_\infty & : \text{equalized/quantized solution.} \end{cases}$$

## ★ Important to note:

→ although we have a formula for the  $L_2$  case, i.e.

$$\hat{x} = A^T b \text{ when } \lambda \rightarrow 0 \quad \text{and} \quad \hat{x} = V_1 \Sigma_1 (\Sigma_1^2 + \lambda I)^{-1} U_1^T b$$

in general, there is no formula for the  $L_1$  and  $L_\infty$  cases!

Instead, we must use iterative methods to find the solution.

Note: iterative methods are often used to solve  $L_2$  problems even though we have a formula because iterative methods are fast! (we will discuss iterative methods soon).

→ although  $L_2$  problem always has a unique solution, the same is not true for  $L_1$  and  $L_\infty$  cases.

$$\text{example: } \underset{x}{\text{min.}} \quad \|[1:1]x - 1\|^2 + \lambda \|x\|,$$

2

Example 1: predicting breast cancer from gene markers.

- we have  $m$  patients,  $m \approx 200$ .
  - each has been screened and tested. For each patient, we know the activity levels of  $n \approx 8000$  genes, and we also know whether they have breast cancer or not.

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0.3 & -0.21 & \cdots \\ 0.6 & -0.1 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \underbrace{\quad}_{\text{activity level for each of } n \text{ genes}} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

we'd like a design for  $x$  that predicts disease state.  
i.e. we want to make a linear classifier.

problem: in previous examples, e.g. iris classification, we had a large number of samples (flowers) and a small number of characteristics (sepal length, petal width, ...).  
but this time,  $Ax = b$  has exact (and  $\infty$  many) solutions!

Important note :

Matlab's  $A \setminus b$  is not the same as  $\text{pinv}(A) * b$  when  $A$  is not full column rank!

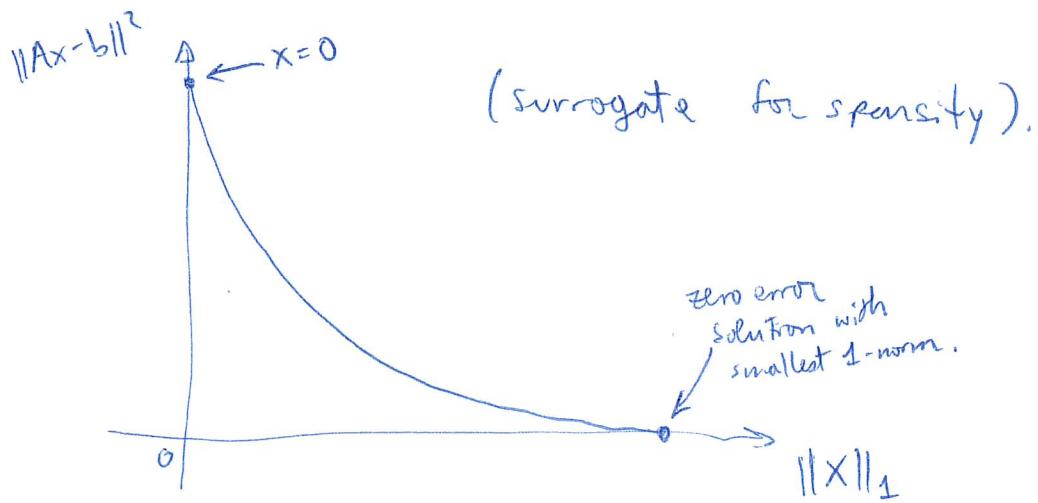
(3)

### Ex 1, cont'd

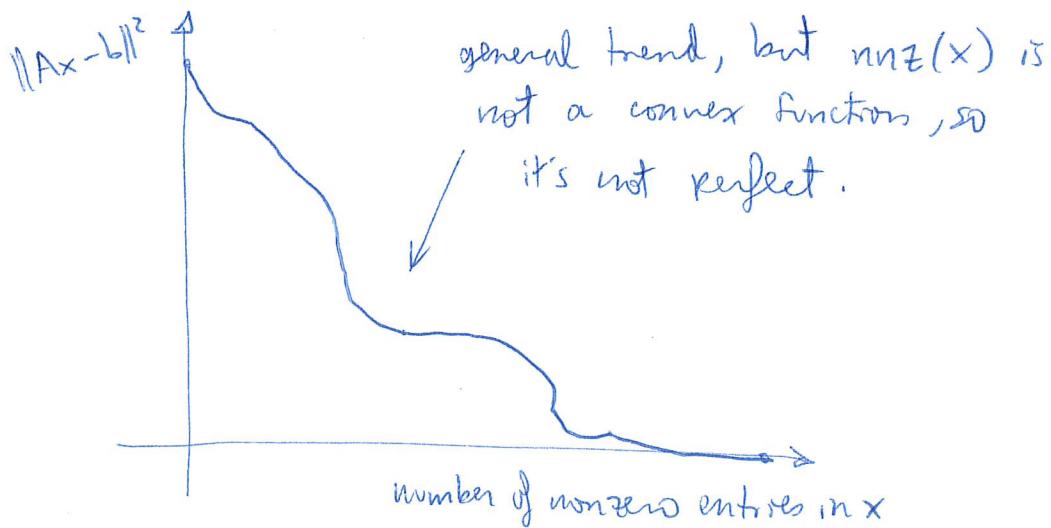
regularization can help us choose a solution. In this case, we expect a relatively small number of genes to be involved, so it makes sense to look for a sparse  $x$ .

So we can use the lasso and adjust  $\lambda$  and find solutions.

for each  $\lambda$ , solve  $\min \|Ax - b\|^2 + \lambda \|x\|_1$ ,  
 (again, we'll see how to do this later!) and plot the trade-off curve:



If we plot actual sparsity:



(4)

## Example 2 : Hovercraft path planning (a control problem)

position at time  $t$ .

desired final position.

at  $t=0$ , craft is at  $x=0$  (not moving).at  $t=40$ , we want  $x=10$  (again not moving).we can apply thruster force at  $t=0, 1, 2, \dots$ ★ Define  $\{x_0, x_1, \dots, x_{t-1}\}$  = position at time  $t$ . $\{v_0, v_1, \dots, v_{t-1}\}$  = velocity at time  $t$ . $\{u_0, u_1, \dots, u_{t-1}\}$  = thrust at time  $t$ .

★ Suppose equations of motion are

$$\begin{cases} \frac{d}{dt}x(t) = v(t) \\ \frac{d}{dt}v(t) = u(t) \end{cases} \quad \text{"double integrator".}$$

★ Discretized equations: ( $\dot{x}(t) \approx x(t) - x(t-1)$ ),

$$\underbrace{\begin{bmatrix} v_{t+1} \\ x_{t+1} \end{bmatrix}}_{z_{t+1}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} v_t \\ x_t \end{bmatrix}}_{z_t} + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_B u_t.$$

this is a linear dynamical system! (ref HW #1),

(5)

initial condition:  $\mathbf{z}_0 = \begin{bmatrix} v_0 \\ x_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

terminal condition:  $\mathbf{z}_{40} = \begin{bmatrix} v_{40} \\ x_{40} \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$

variables:  $u_0, u_1, \dots, u_{39}$ .

$$U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{39} \end{bmatrix}$$

constraint ...  $\mathbf{z}_1 = A\mathbf{z}_0 + Bu_0$

$$\mathbf{z}_2 = A\mathbf{z}_1 + Bu_1 = A^2\mathbf{z}_0 + ABu_0 + Bu_1$$

$\vdots$

$$\mathbf{z}_{40} = \underbrace{A^{40}\mathbf{z}_0}_{b. \text{ zero.}} + \underbrace{A^{39}Bu_0 + A^{38}Bu_1 + \dots + ABu_{38} + Bu_{39}}_{[A^{39}B \dots B] \underbrace{\begin{bmatrix} u_0 \\ \vdots \\ u_{39} \end{bmatrix}}_{P.}}$$

$$[A^{39}B \dots B] \begin{bmatrix} u_0 \\ \vdots \\ u_{39} \end{bmatrix}$$

problem: Find  $u$  such that  $Pu = b$ .

$$\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 2 \times 40 & 40 \times 1 & 2 \times 1 \end{array}$$

highly underdetermined!

how should we choose  $u$ ?

- minimize  $\|u\|_2$ : higher thrust is increasingly more costly. Maybe appropriate if engine is less efficient at higher thrust. (more fuel required). Or cost is proportional to "input energy"
- minimize  $\|u\|_1$ : cost is proportional to thrust. ~~More cost~~ also promotes sparcity.
- minimize  $\|u\|_\infty$ : cost is proportional to maximum thrust required. (i.e. engine is free to operate, you pay more for bigger engine!)

(6)

problem:  $\begin{bmatrix} \text{minimize } \|u\|_\alpha \\ \text{subject to } Pu = b. \end{bmatrix}$

first approach: minimize  $\underset{u}{\|Pu - b\|_2^2} + \lambda \|u\|_\alpha$

(regularize) and take the limit of small  $\lambda > 0$ .

second approach: if  $Pu = b$ , then  $u = \underbrace{P^T b}_{\text{particular solution}} + \underbrace{V_2 w}_{\text{general element of } N(P)}$

(parametrize) then, minimize  $\underset{w}{\|P^T b + V_2 w\|_\alpha}$

and  $\hat{u} = P^T b + V_2 \hat{w}$ .

[show demo!]

