

# **MATH466/MATH766**

## **Math of machine learning**

### **01/22 Lecture 4 HW1 and Sparsity in Machine Learning**

References:

- Ch2, 3 of High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications by John Wright and Yi Ma
- Ch3.4 of The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani and Jerome Friedman
- Ch6.3 of Linear Algebra and its Applications (Fourth Edition) by Gilbert Strang

Todays contents:

- regularization
- $\ell_1$  norm and sparsity
- sparse signal modeling

Important concepts:

- regularization
- sparsity

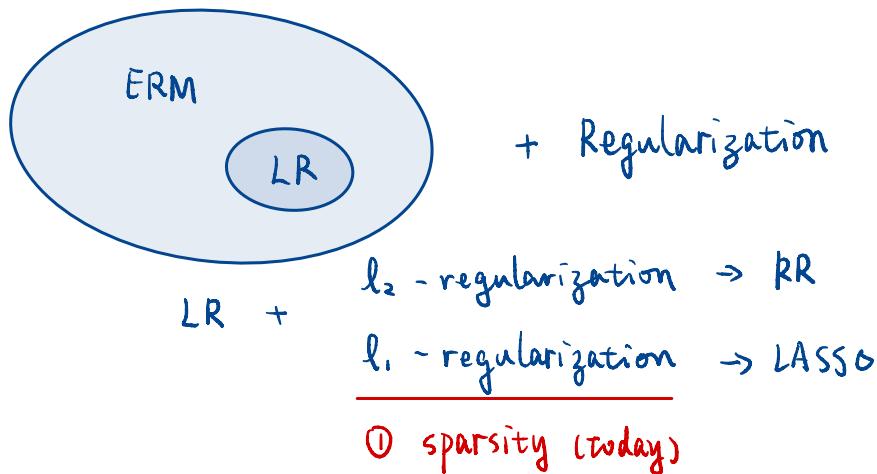
Recommend reading:

- Ch1 of High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications by John Wright and Yi Ma
- Ch3.4 of The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani and Jerome Friedman
- Ch3.3, 3.6, 3.8 of The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani and Jerome Friedman

Review.

Lec 2

Regression :



③ linear model  $\rightarrow$  nonlinear model (HW 1-2)  
parametric  $\rightarrow$  non-parametric

Lec 3

Classification : (linear)  
parameterize p + MLE  $\rightarrow$  logistic regression

② binary  $\rightarrow$  multiclass (HW 1-1)

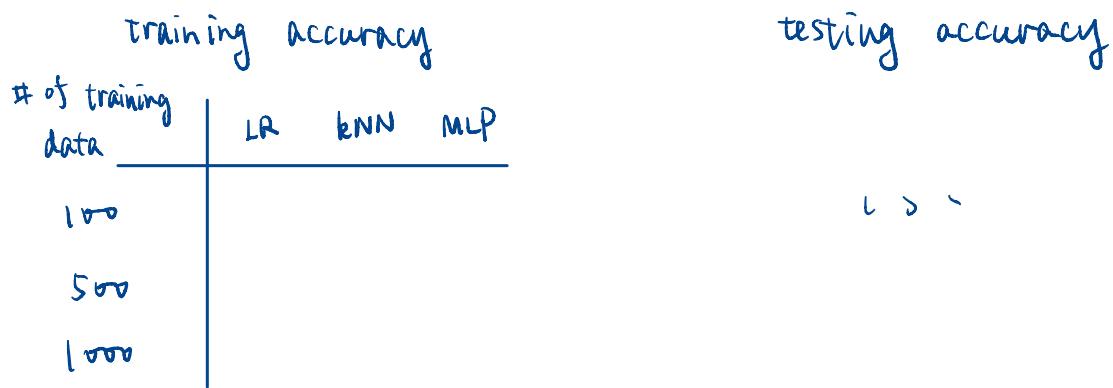
parametric ↑ maximize marginal distance  $\rightarrow$  SVM

non-parametric model - kNN

④ compare parametric v.s. non-parametric (HW 1-4)

HW1 prob 1. 2. go-through

Prob 3 linearly separable data / two-ring data



- 4 tables, ① compare performance of 3 methods on the same dataset w/ different # of training samples  
② compare the same method on different dataset  
③ compare different methods on different dataset

# Inverse Problem and Regularization

An oversimplified example: given a mapping  $\mathcal{L}: A \mapsto B = \mathcal{L}(A)$

forward problem: given  $A$ , find  $\mathcal{L}(A)$

e.g. 1.  $A$  is some function  $f$

$$B = \mathcal{L}(A) = \{(x_1, f(x_1)), (x_2, f(x_2)), \dots, (x_n, f(x_n))\}$$

e.g. 2.  $A$  is a clean image

$\mathcal{L}$  is some blurring / degenerating process

inverse problem: observe  $B$  (partial, noisy)

want to find  $A$  such that  $\mathcal{L}(A) = B$

e.g. 1.  $B$  training data set, supervised learning

e.g. 2.  $B$  is the blurred / degenerated image

want to find the clean image

Inverse problems are usually harder than forward problem.

One difficulty: non-unique solution

Regularization is a common strategy in inverse problem

Idea / Goal: include prior knowledge / belief / preference  
to make the solution unique

Outcome: solution of certain property

e.g. image denoising. Rudin - Osher - Fatemi (ROF) model  
(or total variation denoising)  
edge-preserving

"sparsity" is a common belief  $\rightarrow$  e.g. matrix completion (Netflix problem)  
an important property people pursue in many applications  
 $\rightarrow$  e.g. LASSO, compressed sensing

## Sparsity

For  $w \in \mathbb{R}^d$ ,  $\|w\|_0 := \sum_{j=1}^d 1_{\{w_j \neq 0\}} = \# \text{ of non-zero entries in } w$

sparsity wants small  $\|w\|_0$  ( $\|\cdot\|_0$  is called "l<sub>0</sub>-norm" but is not a norm)

$$\text{LASSO : } \min_w \frac{1}{2} \|Xw - y\|^2 + \lambda \|w\|_1 \quad (\ast_1)$$

$$\text{where } \|w\|_1 := \sum_{j=1}^d |w_j|$$

Q: Why not using l<sub>0</sub>-norm as a regularization?

What's the benefit of l<sub>1</sub>-regularization?

Q: Does l<sub>1</sub>-regularization encourages sparsity? Why?

Answer Today!

Q: How to solve  $(\ast_1)$ ?

## Explanation 1

Consider  $d=2$

$$\text{Assume } \hat{w} = \underset{w}{\operatorname{argmin}} \frac{1}{2} \|Xw - y\|^2$$

$$\text{contour of } R(w) = \frac{1}{2} \|Xw - y\|^2$$

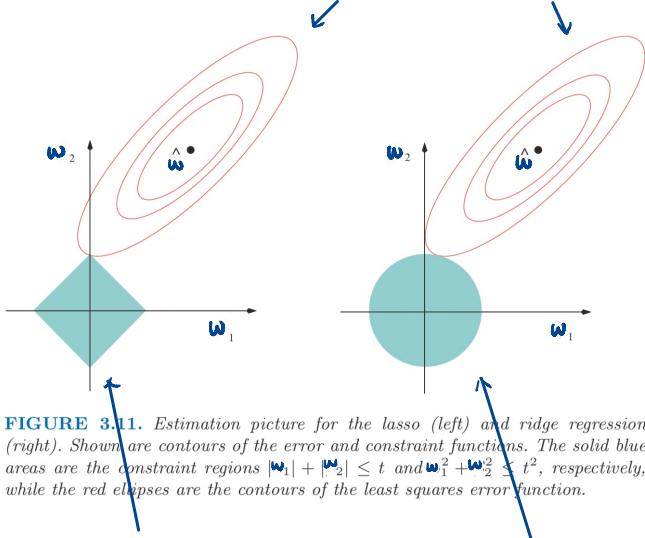


FIGURE 3.11. Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions  $|w_1| + |w_2| \leq t$  and  $w_1^2 + w_2^2 \leq t^2$ , respectively, while the red ellipses are the contours of the least squares error function.

grow the contour of

$R$  and  $\|w\|_1$

$\|w\|_2$

at the same "speed"

until they meet

$\rightarrow$  solution under  $\frac{l_1}{l_2}$  regularization

## Explanation 2

Consider  $x = (1, \varepsilon)$ ,  $\varepsilon > 0$ ,  $\varepsilon \ll 1$

$$\text{then } \|x\|_1 = 1 + \varepsilon, \quad \|x\|_2^2 = 1 + \varepsilon^2$$

If we want to reduce  $\|x\|_1$  or  $\|x\|_2^2$ , reducing which entry is more "efficient"?

	$(1-\delta, \varepsilon)$	$(1, \varepsilon-\delta)$	(assume $\delta < \varepsilon$ , $\delta < 1$ )
$\ x\ _1$	$1 - \delta + \varepsilon$	$1 + \varepsilon - \delta$	
$\ x\ _2^2$	$1 + \varepsilon^2$	$1 + (\varepsilon - \delta)^2$	

To obtain the same amount of diminish in  $\|x\|_2^2$ ,

reducing the "large" entry is more "efficient".

$\ell_2$ -norm square "discourage" sparsity.

## Explanation 3

Let  $B_p := \{x : \|x\|_p \leq 1\}$

$$\text{In } d\text{-dimension, } \text{vol}(B_\infty) = 2^d$$

$$\text{vol}(B_2) = \begin{cases} \pi^{(d-1)/2} / \left(\frac{d}{2}\right) \left(\frac{d-2}{2}\right) \cdots \frac{1}{2} & d \text{ is odd} \\ \frac{\pi^{d/2}}{(d/2)!} & d \text{ is even} \end{cases}$$

$$\text{vol}(B_1) = 2^d / d!$$

$$d=2, \quad \text{vol}(B_1) = 0.5 \times \text{vol}(B_\infty) \approx 0.63 \times \text{vol}(B_2)$$

$$d=1000, \quad \text{vol}(B_1) \approx 10^{-2568} \times \text{vol}(B_\infty) \approx 10^{-1485} \times \text{vol}(B_2)$$

Note.

$\ell_1$ -norm regularization does **NOT** always give the sparsest solution

The sparsity of the solution depends on the contour of  $\|Xw-y\|_2^2$ .

(or other fidelity loss we may see in the future)

## Sparse signal modeling

$$\begin{matrix} \in \mathbb{R}^m \\ y \\ \hline \text{observation} \end{matrix} = \begin{matrix} \in \mathbb{R}^{m \times n} \\ D \\ \hline \text{unknown} \end{matrix} \begin{matrix} \in \mathbb{R}^n \\ x \\ \hline \end{matrix}$$

solve  $x$  from  $y$  : underestimating system

need additional information / knowledge

e.g. biological vision system

$y \sim \text{visual sensor}$

$D = [d_1, d_2, \dots, d_n]$  ~ dictionary of elementary patterns

$x \sim \text{sparse coefficients}$

e.g. Compressed sensing

sampling

$x \sim$  signal. want to use less measurement to recover  $x$

Dx : encoding / sampling

$y \rightarrow x$  : decoding

Before compressed sensing : Nyquist - Shannon sampling theorem

# of samples depends on signal frequency

can be expensive if signal contains both low-freq  
and high-freq components.

compressed sensing : around 2004 .

if  $x$  is sparse, fewer samples are needed for perfectly recovering  $x$

Problem formulation 1:  $\ell_0$ -minimization

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{D}\mathbf{x} = \mathbf{y}$$

How to solve: exhaustive search

For  $k = 0, 1, \dots, n$

For each  $I \subseteq \{1, 2, \dots, n\}$  of size  $k$

If the system  $\mathbf{A}_I \mathbf{z} = \mathbf{y}$  has a solution  $\mathbf{z}$

Set  $\mathbf{x}_I = \mathbf{z}$ ,  $\mathbf{x}_{I^c} = 0$

return  $\mathbf{x}$

Issue: worst case complexity  $n^{\|\mathbf{x}_0\|_0}$  ( $\mathbf{x}_0$  ground truth)

Problem formulation 2:  $\ell_1$ -minimization (relaxation)

Problem formulation 3: Noisy observations or Approximate Sparsity

Basis Pursuit:

LASSO

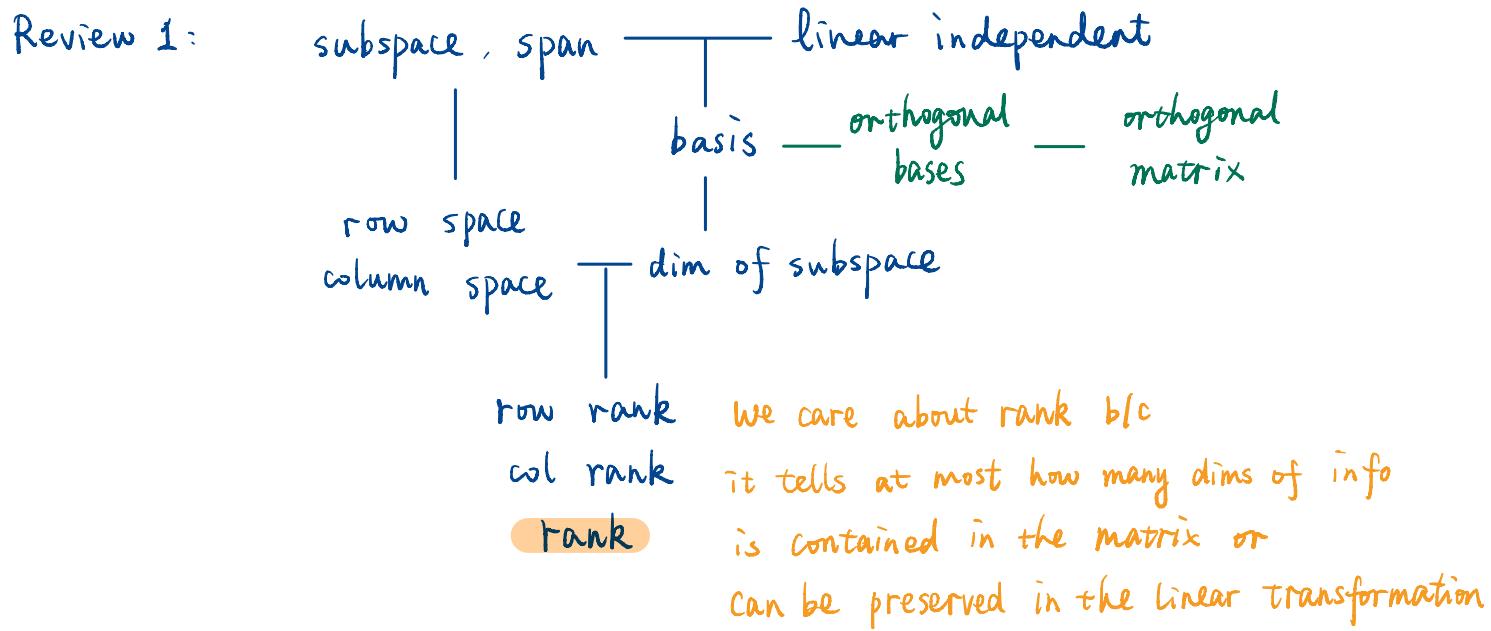
Thm. Suppose  $D_{ij} \sim \text{iid } N(0, \frac{1}{m})$ ,  $\mathbf{y} = \mathbf{D}\mathbf{x}_0 + \mathbf{z}$

where  $\mathbf{x}_0$  is  $k$ -sparse,  $\mathbf{z}_i \sim \text{iid } N(0, \frac{\sigma^2}{m})$

Then solving the LASSO with  $\lambda = C \cdot 2\sigma \sqrt{\frac{\log n}{m}}$  for a large enough  $C$

gives  $\|\tilde{\mathbf{x}} - \mathbf{x}_0\|_2 \leq C' \sigma \sqrt{\frac{k \log n}{m}}$  with high probability (w.h.p.)

Low-rank : "sparsity" of matrices



Example: Netflix problem, low-rank matrix completion

	user 1	user 2	...	
movie 1	9.8	?		observing *'s want to fill in ?'s and make recommendations
movie 2	7.6	5.4		
:		*		
	?	3.2	*	

Assumption: the matrix has low-rank (comparing to # of rows/cols)

need some better characterizations of rank for computational / theoretical studies.

Review 2: eigen-decomp of real symmetric matrices  $A \in \mathbb{R}^{n \times n}$

$$A = U \Lambda U^T, \quad U \in \mathbb{R}^{n \times n} \text{ orthogonal}$$

$$\Lambda \in \mathbb{R}^{n \times n} \text{ diagonal}$$

$$\text{rank}(A) = \# \text{ of nonzero entries of } \text{diag}(\Lambda)$$

How about general real matrices?

## Singular Value Decomposition (SVD)

Any  $A \in \mathbb{R}^{m \times n}$  can be factored into

$$A = U \Sigma V^T$$

$U \in \mathbb{R}^{m \times m}$     $\Sigma \in \mathbb{R}^{m \times n}$     $V \in \mathbb{R}^{n \times n}$   
 orthogonal   diagonal   orthogonal

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)} \geq 0$$

$$\text{rank}(A) = \# \text{ of nonzero entries of } \text{diag}(\Sigma)$$

understanding SVD (take  $m > n$  as an example)

$$\text{Let } U = [u_1, u_2, \dots, u_m], V = [v_1, v_2, \dots, v_n], \Sigma =$$

$$\text{then } AV = U\Sigma$$

$$\begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ & & & 0 \end{bmatrix}$$

Analogue: rank of matrices  $\sim \| \cdot \|_0$  of vectors

?  $\sim \| \cdot \|_1$  of vectors