

MATH466/MATH766

Math of machine learning

01/22-01/27 Lecture 4-5 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
- ℓ_1 norm and sparsity
- sparse signal modeling

Important concepts:

- regularization
- sparsity

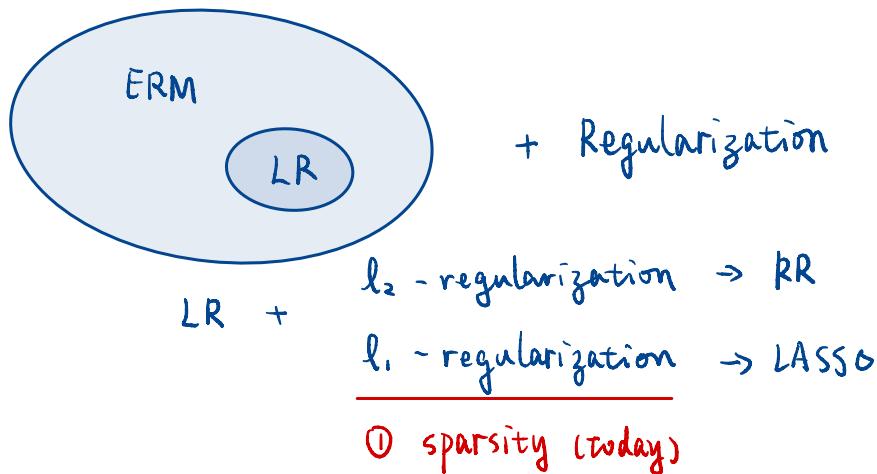
Recommend reading:

- 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



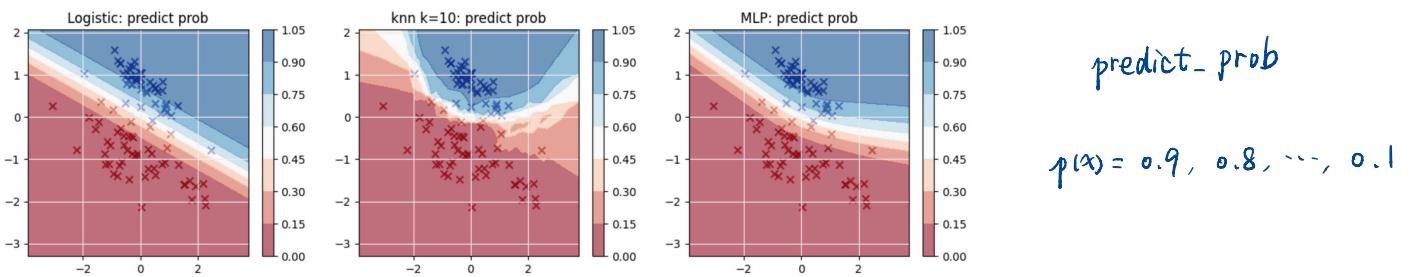
4 tables. ① compare performance of 3 methods on the same dataset

w/ different # of training samples

② compare the same method on different dataset

③ compare different methods on different dataset

about decision boundary plot



about warning on maximum iteration

iterative algorithm.

$$\min_{\theta} L(\theta)$$

e.g. For $k=1, 2, \dots, K$

if $\|\nabla \mathcal{L}(\theta^k)\| > \text{tolerance}$

$$\theta^{k+1} = \theta^k - \tau \nabla \mathcal{L}(\theta^k)$$

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 "l₀-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 l₀-norm as a regularization?

What's the benefit of l₁-regularization?

Q: Does l₁-regularization encourages sparsity? Why?

Q: How to solve (\ast_1) ?

l₁-regularization is better on pursuing sparsity solution comparing to l₂

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

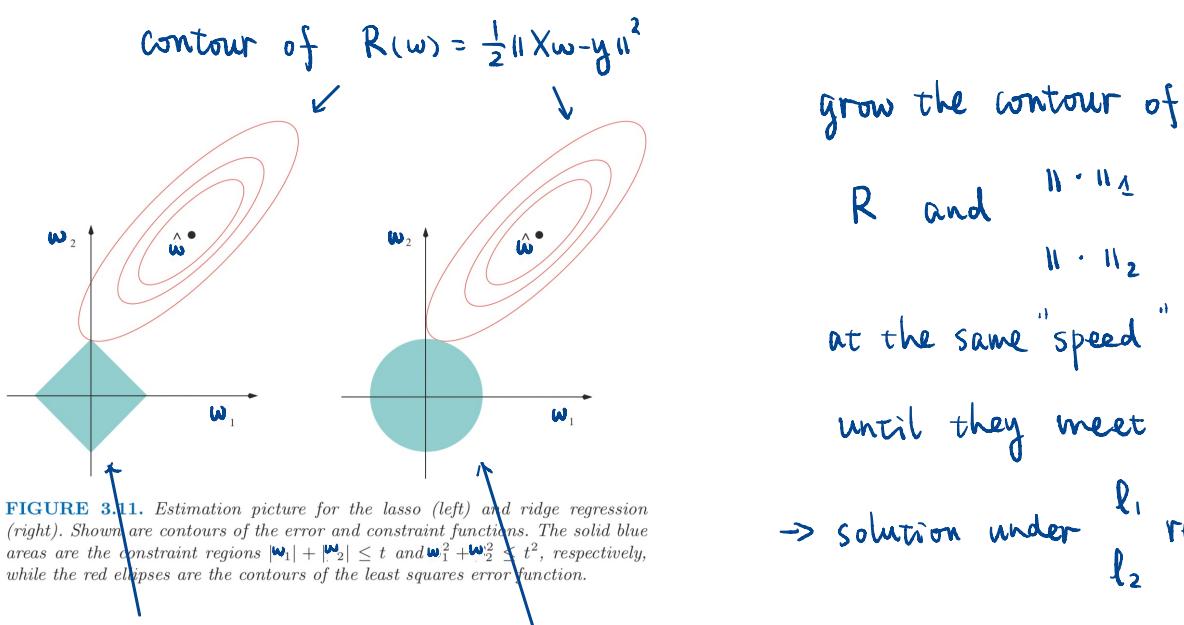


FIGURE 3.11. Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|w_1| + |w_2| \leq t$ and $w_1^2 + w_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the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

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

l_2 -norm contour grows equally fast in all directions (isotropic)

l_1 -norm ——— faster on axes (anisotropic)

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

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

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

ℓ_2 -norm square "discourage" sparsity.

Explanation 3

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

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

$$\text{vol}(B_2) = \begin{cases} \pi^{(d-1)/2} / (\frac{d}{2})(\frac{d-2}{2})(\frac{d-4}{2}) \dots \frac{1}{2} \\ \text{d is odd} \\ \text{d is even} \end{cases}$$

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

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

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

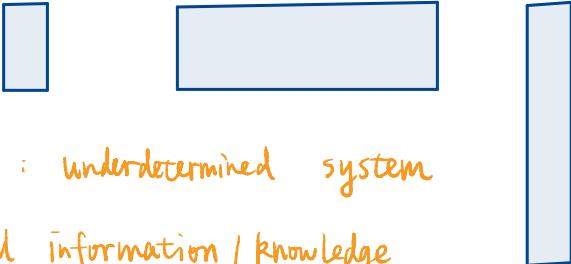
Note.

ℓ_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)

Case study: Sparse signal modeling

$$\underbrace{\underline{y} \in \mathbb{R}^m}_{\text{observation}} = \underbrace{D \in \mathbb{R}^{m \times n}}_{\text{dictionary}} \underbrace{\underline{x} \in \mathbb{R}^n}_{\text{unknown}}$$


$m \ll n$

solve x from y : underdetermined system

need additional information / knowledge

e.g. biological vision system

y ~ visual sensor

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

x ~ sparse coefficients

e.g. Compressed sensing

sampling

x ~ 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: l_0 -minimization

$$\min_x \|x\|_0 \quad \text{s.t.} \quad Dx = y$$

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 $A_I z = y$ has a solution z

Set $x_I = z$, $x_{I^c} = 0$

return x

Issue: worst case complexity $n^{\|x_0\|_0}$ (x_0 ground truth)

Problem formulation 2: l_1 -minimization (relaxation)

$$\min_x \|x\|_1 \quad \text{s.t.} \quad Dx = y$$

Problem formulation 3: Noisy observations or Approximate Sparsity

Basis Pursuit: $\min_x \|x\|_1 \quad \text{s.t.} \quad \|y - Dx\|_2 \leq \varepsilon$

LASSO $\min_x \lambda \|x\|_1 + \frac{1}{2} \|y - Dx\|_2^2$

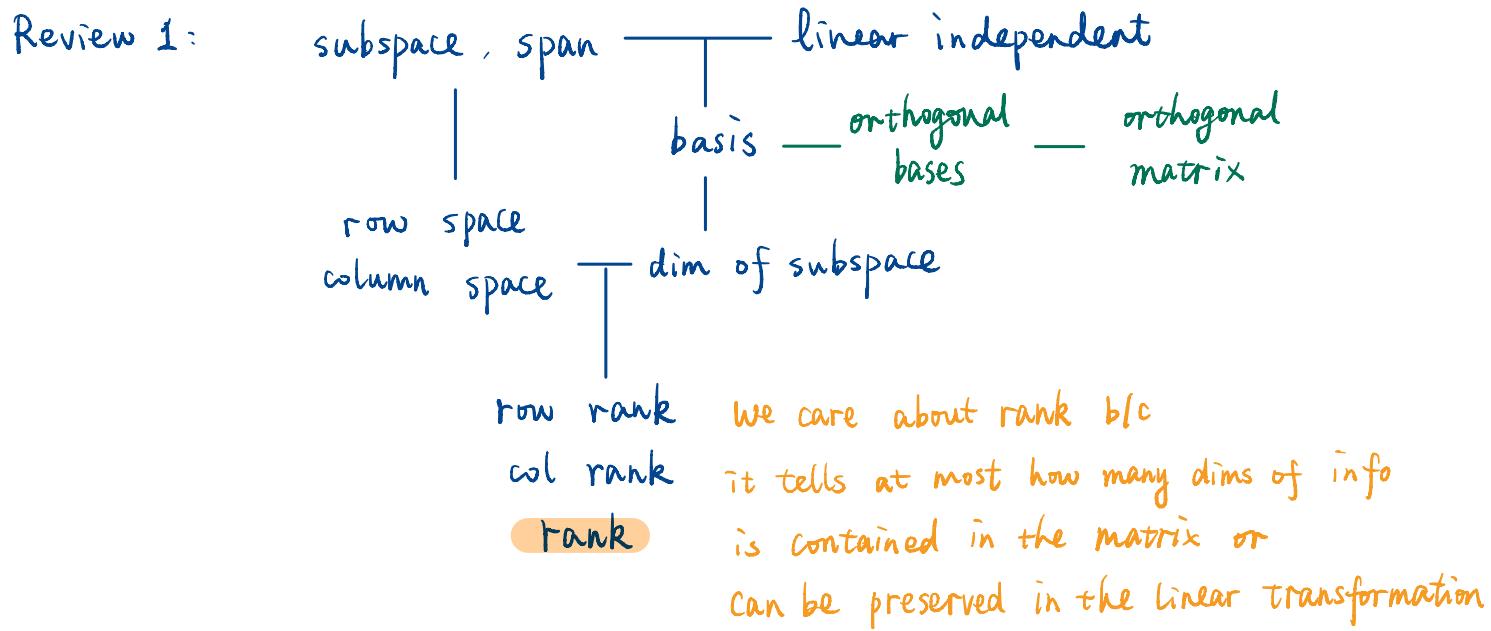
Thm. Suppose $D_{ij} \sim \text{iid } N(0, \frac{1}{m})$, $y = Dx_0 + z$

where x_0 is k -sparse, $z_i \sim \text{iid } N(0, \frac{\sigma^2}{m})$

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

gives $\|\tilde{x} - x_0\|_2 \leq C' \sigma \sqrt{\frac{k \log n}{m}}$ 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

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

$m > n$

$$\Sigma$$

$m < n$

$$\Sigma$$

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

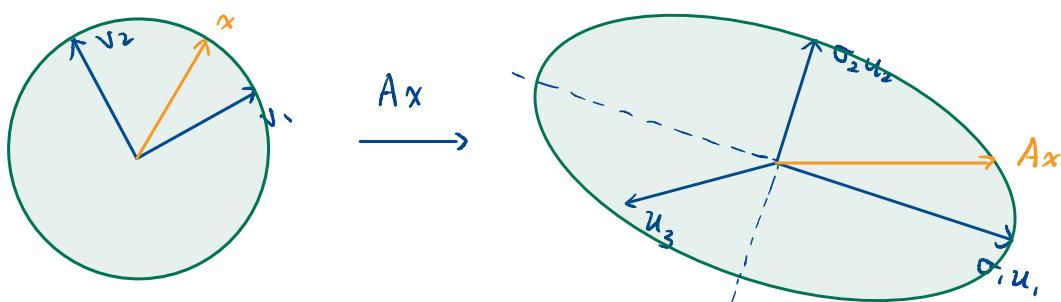
understanding SVD (take $m > n$ as an example)

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

then $AV = U\Sigma$

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

$$[Av_1, Av_2, \dots, Av_n] = [\sigma_1 u_1, \sigma_2 u_2, \dots, \sigma_n u_n]$$



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

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

$$\text{rank}(A) = \sum_{i=1}^{\min(m,n)} \mathbf{1}_{\{\sigma_i > 0\}}$$

$$\text{nuclear norm } \|A\|_* = \sum_{i=1}^{\min(m,n)} |\sigma_i|$$