

# COMS 4721: Machine Learning

## Lecture 6, 2/7/2019

Prof. John Paisley

Department of Electrical Engineering  
& Data Science Institute  
Columbia University

# UNDERDETERMINED LINEAR EQUATIONS

We now consider the regression problem  $y = Xw$  where  $X \in \mathbb{R}^{n \times d}$  is “wide” (i.e.,  $d \gg n$ ). This is called an underdetermined problem.

- ▶ There are more dimensions than observations.
- ▶  $w$  now has an infinite number of solutions satisfying  $y = Xw$ .

$$\begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} & X & \end{bmatrix} \begin{bmatrix} w \end{bmatrix}$$

These sorts of high-dimensional problems often come up:

- ▶ In gene analysis there are 1000's of genes but only 100's of subjects.
- ▶ Images can have millions of pixels.
- ▶ Even polynomial regression can quickly lead to this scenario.

# MINIMUM $\ell_2$ REGRESSION

# ONE SOLUTION (LEAST NORM)

One possible solution to the underdetermined problem is

$$w_{\text{ln}} = X^T(XX^T)^{-1}y \quad \Rightarrow \quad Xw_{\text{ln}} = XX^T(XX^T)^{-1}y = y.$$

We can construct another solution by adding to  $w_{\text{ln}}$  a vector  $\delta \in \mathbb{R}^d$  that is in the *null space*  $\mathcal{N}$  of  $X$ :

$$\delta \in \mathcal{N}(X) \quad \Rightarrow \quad X\delta = 0 \text{ and } \delta \neq 0$$

and so  $X(w_{\text{ln}} + \delta) = Xw_{\text{ln}} + X\delta = y + 0$ .

In fact, there are an infinite number of possible  $\delta$ , because  $d > n$ .

We can show that  $w_{\text{ln}}$  is the solution with smallest  $\ell_2$  norm. We will use the proof of this fact as an excuse to introduce two general concepts.

# TOOLS: ANALYSIS

We can use *analysis* to prove that  $w_{\text{ln}}$  satisfies the optimization problem

$$w_{\text{ln}} = \arg \min_w \|w\|^2 \quad \text{subject to} \quad Xw = y.$$

(Think of mathematical analysis as the use of inequalities to prove things.)

*Proof:* Let  $w$  be another solution to  $Xw = y$ , and so  $X(w - w_{\text{ln}}) = 0$ . Also,

$$\begin{aligned}(w - w_{\text{ln}})^T w_{\text{ln}} &= (w - w_{\text{ln}})^T X^T (XX^T)^{-1} y \\ &= \underbrace{(X(w - w_{\text{ln}}))^T}_{= 0} (XX^T)^{-1} y = 0\end{aligned}$$

As a result, the vector  $w - w_{\text{ln}}$  is *orthogonal* to  $w_{\text{ln}}$ . It follows that

$$\|w\|^2 = \|w - w_{\text{ln}} + w_{\text{ln}}\|^2 = \|w - w_{\text{ln}}\|^2 + \|w_{\text{ln}}\|^2 + 2 \underbrace{(w - w_{\text{ln}})^T w_{\text{ln}}}_{= 0} > \|w_{\text{ln}}\|^2$$

# TOOLS: LAGRANGE MULTIPLIERS

Instead of starting from the solution, start from the problem,

$$w_{\text{in}} = \arg \min_w w^T w \quad \text{subject to} \quad Xw = y.$$

- ▶ Introduce Lagrange multipliers:  $\mathcal{L}(w, \eta) = w^T w + \eta^T (Xw - y)$ .
- ▶ Maximize over  $\eta$ , minimize over  $w$ . If  $Xw \neq y$ , we can get  $\mathcal{L} = +\infty$ , so  $\eta$  effectively *forces* this equality.
- ▶ The optimality conditions are

$$\nabla_w \mathcal{L} = 2w + X^T \eta = 0, \quad \nabla_\eta \mathcal{L} = Xw - y = 0.$$

We have everything necessary to find the solution:

1. From first condition we know:  $w = -X^T \eta / 2$
2. Plug #1 into second condition to find:  $\eta = -2(XX^T)^{-1}y$
3. Plug #2 back into #1 to find the solution:  $w_{\text{in}} = X^T(XX^T)^{-1}y$

# SPARSE $\ell_1$ REGRESSION

# LS AND RR IN HIGH DIMENSIONS

## LS and RR not suited for high-dimensional data

- ▶ Modern problems can have many dimensions/features/predictors
- ▶ Only a few of these may be important or relevant for predicting  $y$
- ▶ Therefore, we often need some form of “feature selection”
- ▶ Some drawbacks of LS and RR in high dimensions are:
  - ▶ They weight all dimensions without favoring subsets of dimensions
  - ▶ The unknown “important” dimensions are mixed in with irrelevant ones
  - ▶ They generalize poorly to new data, weights may not be interpretable
  - ▶ LS solution not unique when  $d > n$ , meaning no unique predictions



# REGRESSION WITH PENALTIES

## Penalty terms

Recall: General  $\ell_2$ -penalized regression is of the form

$$\mathcal{L} = \sum_{i=1}^n (y_i - f(x_i; w))^2 + \lambda \|w\|^2$$

We've referred to the term  $\|w\|^2$  as a *penalty term* and used  $f(x_i; w) = x_i^T w$ .

## Penalized fitting

The general structure of the optimization problem is

$$\text{total cost} = \text{goodness-of-fit term} + \text{penalty term}$$

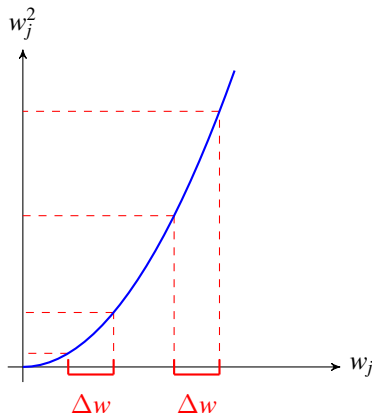
- ▶ Goodness-of-fit measures how well our model  $f$  approximates the data.
- ▶ Penalty term makes the solutions we don't want more “expensive”.

What kind of solutions does the choice  $\|w\|^2$  favor or discourage?

# QUADRATIC PENALTIES

## Intuitions

- ▶ Quadratic penalty: Reduction in cost depends on  $|w_j|$ .
- ▶ Suppose we reduce  $w_j$  by  $\Delta w$ . The effect on  $\mathcal{L}$  depends on the starting point of  $w_j$ .
- ▶ We penalize larger values much more than smaller ones.
- ▶ Consequence: We will favor vectors  $w$  whose entries are of similar size, preferably small.



## Our setting

- ▶ Regression problem with  $n$  data points  $x \in \mathbb{R}^d$ ,  $d \gg n$ .
- ▶ Model data using a linear function,  $y \approx f(x; w) = x^T w$ .
- ▶ Goal: Select a small subset of the  $d$  dimensions and switch off the rest. This is sometimes referred to as “feature selection”.

## What does it mean to “switch off” a dimension?

- ▶ Each entry of  $w$  corresponds to a dimension of the data  $x$ .
- ▶ If  $w_k = 0$ , the prediction does not depend on the  $k$ th dimension,

$$f(x; w) = x^T w = w_1 x_1 + \cdots + 0 \cdot x_k + \cdots + w_d x_d,$$

- ▶ Feature selection: Find a vector  $w$  that (1) predicts well, and (2) has only a small number of non-zero entries.
- ▶ A  $w$  for which most dimensions  $= 0$  is called a *sparse* solution.

# SPARSITY AND PENALTIES

## Penalty goal

Find a penalty term which encourages sparse solutions.

## Quadratic penalty vs sparsity

- ▶ Suppose using least squares  $w_k$  is large, all other  $w_j$  are very small
- ▶ Sparsity: The penalty should keep  $w_k$  large, and push other  $w_j$  to zero
- ▶ Quadratic penalty: Will favor entries  $w_j$  which all have similar size, and so it will push  $w_k$  towards a smaller value.

## One solution

Sparsity can be achieved using a *linear* penalty term.

# LASSO

## Sparse regression

One penalty that encourages a sparse solution is known as the “LASSO.”

**LASSO:** Least Absolute Shrinkage and Selection Operator

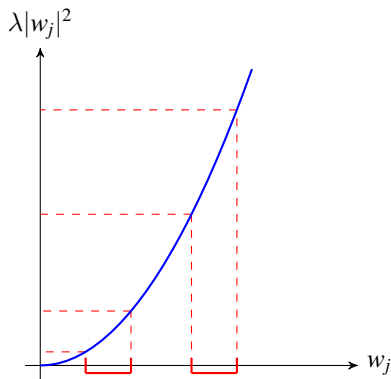
With the LASSO, we replace the  $\ell_2$  penalty with an  $\ell_1$  penalty:

$$w_{\text{lasso}} = \arg \min_w \|y - Xw\|_2^2 + \lambda \|w\|_1$$

where  $\|w\|_1 = \sum_{j=1}^d |w_j|$ . This is also called  $\ell_1$ -regularized regression.

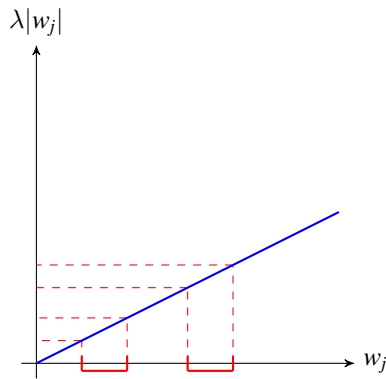
# QUADRATIC PENALTIES

## Quadratic penalty



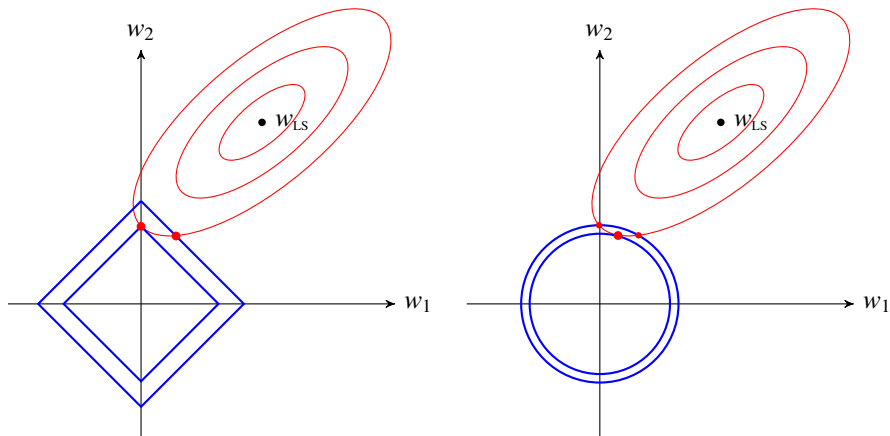
Reducing a large value  $w_j$  achieves a larger cost reduction.

## Linear penalty



Cost reduction does not depend on the magnitude of  $w_j$ .

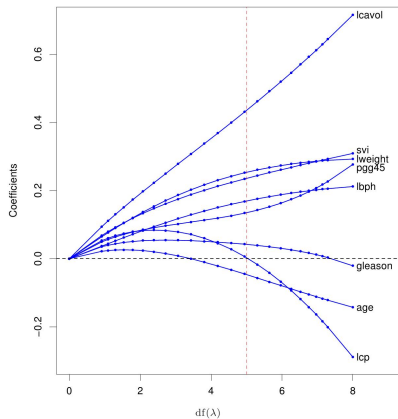
# RIDGE REGRESSION VS LASSO



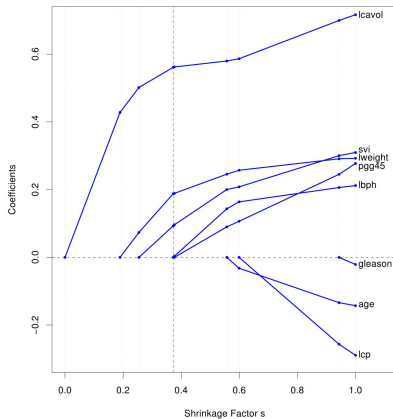
This figure applies to  $d < n$ , but gives intuition for  $d \gg n$ .

- ▶ Red: Contours of  $(w - w_{LS})^T (X^T X) (w - w_{LS})$  (see Lecture 3)
- ▶ Blue: (left) Contours of  $\|w\|_1$ , and (right) contours of  $\|w\|_2^2$

# COEFFICIENT PROFILES: RR vs LASSO



(a)  $\|w\|_2$  penalty



(b)  $\|w\|_1$  penalty



# $\ell_p$ REGRESSION

## $\ell_p$ -norms

These norm-penalties can be extended to all norms:

$$\|w\|_p = \left( \sum_{j=1}^d |w_j|^p \right)^{\frac{1}{p}} \quad \text{for } 0 < p \leq \infty$$

## $\ell_p$ -regression

The  $\ell_p$ -regularized linear regression problem is<sup>1</sup>

$$w_{\ell_p} := \arg \min_w \|y - Xw\|_2^2 + \lambda \|w\|_p^p$$

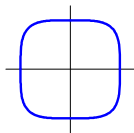
We have seen:

- ▶  $\ell_1$ -regression = LASSO
- ▶  $\ell_2$ -regression = ridge regression

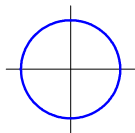
---

<sup>1</sup>In the cases  $p = \infty$  and  $p = 0$ ,  $\|w\|_p$  is used instead of  $\|w\|_p^p$ .

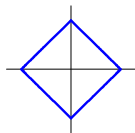
# $\ell_p$ PENALIZATION TERMS



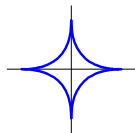
$$p = 4$$



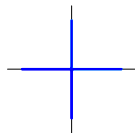
$$p = 2$$



$$p = 1$$



$$p = 0.5$$



$$p = 0.1$$

$p$	Behavior of $\  \cdot \ _p$
$p = \infty$	Norm measures largest absolute entry, $\ w\ _\infty = \max_j  w_j $
$p > 2$	Norm focuses on large entries
$p = 2$	Large entries are expensive; encourages similar-size entries
$p = 1$	Encourages sparsity
$p < 1$	Encourages sparsity as for $p = 1$ , but contour set is not convex (i.e., no “line of sight” between every two points inside the shape)
$p \rightarrow 0$	Simply records whether an entry is non-zero, i.e. $\ w\ _0 = \sum_j \mathbb{I}\{w_j \neq 0\}$

# COMPUTING THE SOLUTION FOR $\ell_p$

## Solution of $\ell_p$ problem

$\ell_2$  aka ridge regression. Has a closed form solution

$\ell_p$  ( $p \geq 1, p \neq 2$ ) — By “convex optimization.” We won’t discuss convex analysis in detail in this class, but two facts are important

- ▶ There are no “local optimal solutions” (i.e., local minimum of  $\mathcal{L}$ )
- ▶ The solution can be found using iterative algorithms

( $p < 1$ ) — We can only find an approximate solution (i.e., the best in its “neighborhood”) using iterative algorithms.

## Three techniques formulated as optimization problems

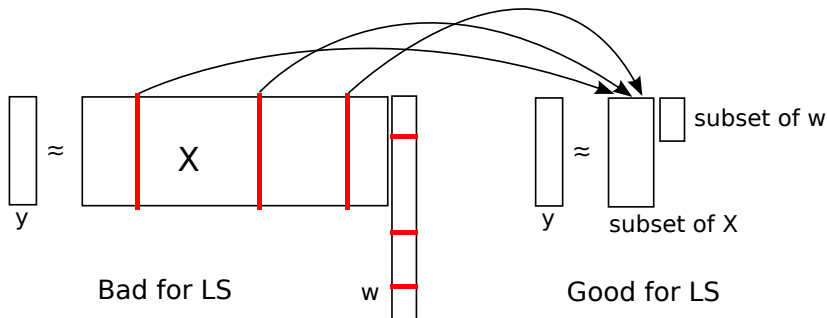
Method	Good-o-fit	penalty	Solution method
Least squares	$\ y - Xw\ _2^2$	none	Analytic solution exists if $X^T X$ invertible
Ridge regression	$\ y - Xw\ _2^2$	$\ w\ _2^2$	Analytic solution exists always
LASSO	$\ y - Xw\ _2^2$	$\ w\ _1$	Numerical optimization to find solution

# GREEDY SPARSE REGRESSION

# GREEDY SUBSET SELECTION

There are very many algorithms for finding sparse solutions. Some of them use greedy methods. Many approaches build on least squares. For example,

- ▶ Imagine we knew a good subset of  $k < n \ll d$  columns of  $X \in \mathbb{R}^{n \times d}$  for which the corresponding dimensions of  $w$  are  $\neq 0$ .
- ▶ We could pick out that subset of  $w$  and learn it using least squares.
- ▶ The question is how do we pick that subset?



# ORTHOGONAL MATCHING PURSUITS

OMP (also called *forward stepwise regression*) sequentially picks columns of  $X$  and allows the corresponding dimensions in  $w$  to  $\neq 0$ .

It has two-steps. Given the indexes of  $k$  columns selected from  $X$  (call it  $\mathcal{I}_k$ ):

1. Find the least squares solution and the approximation error (residual),

$$w_{\text{LS}}^{(k)} = (X_{\mathcal{I}_k}^T X_{\mathcal{I}_k})^{-1} X_{\mathcal{I}_k}^T y, \quad r^{(k)} = y - X_{\mathcal{I}_k} w_{\text{LS}}^{(k)}.$$

2. “Activate” the column of  $X$  that correlates the most with the error,

$$\text{Pick } j\text{th column of } X \text{ (call it } X_j), \text{ where } j = \arg \max_{j'} \frac{|X_{j'}^T r^{(k)}|}{\|X_{j'}\|_2 \|r^{(k)}\|_2}.$$

Comments:

- #1.  $X_{\mathcal{I}_k} w_{\text{LS}}^{(k)}$  gets us as close to  $y$  as possible using only the active columns.
- #2. The angle  $\theta$  between vectors  $a$  and  $b$  is found by  $a^T b = \|a\|_2 \|b\|_2 \cos \theta$ .  
Thus adding  $X_j$  gets us closest to  $y$  if it's the last column we can pick.