COMS 4721: Machine Learning for Data Science

Lecture 6, 2/2/2017

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 "fat" (i.e., $d \gg n$). This is called an "underdetermined" problem.

d is now potentially much greater than n.

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

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

w is very long, much longer than y
So we have an infinite number of potential solutions
There are infinite number of vectors w that can solve this type of a problem when the number of unknowns is greater than the number of equations.

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 ℓ_2 REGRESSION

(not necessarily an algorithm want to use, but will introduce 2 different techniques very useful in machine learning.)

ONE SOLUTION (LEAST NORM)

how?

One possible solution to the underdetermined problem is

LS $sol^n = (XX^T)^{-1} X^T y$

to see why that's a sol^n, simply multiply the left

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

side by X.

we have kind of flipped that sol^n

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

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

the new solution we found is also a solution to the problem.

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

In fact, there are an infinite number of possible δ, because $d > n$. (the no. of deltas we can pick)

why?

We can show that w_{ln} is the solution with smallest ℓ_2 norm. We will use the proof of this fact as an excuse to introduce two general concepts.

So our goal is to show that among all possible solutions, the least-square has the smallest L_2 norm → why do we want to show this? Is the L_2 norm sol^n preferred over others?

 use inequalities toward the end of showing something that we have to prove.

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

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

defined on prev. slide

[vector has to satisfy our linear problem.]

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

new sol^n. least norm sol^n.

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

y - y = 0 *

The vector constructed from the difference btw any sol^n and the least norm sol^n is orthogonal to the least norm solution.

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

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

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

*** 0

expanding by treating this as 1 vector & as II^nd vector.

select $w = w_{ln}$ then we satisfy our minimum

What is in the columnspace & what is in the nullspace?

* The previous vector δ in the null space of X that we discussed is now equal to the difference between our new solution, which can any new solⁿ & the least norm solⁿ.

didn't understand, is δ a variable? Why would we get the same vector in nullspace if we just randomly another solⁿ from it.

Will the vector after subtraction always be in the null space.

** X times w is equal to y, by the definition of our choice of y.
And X times $w_{LS} = y$ (proof on prev. slide)

What does this mean?

*** The squared norm of any solⁿ that we choose is equal to the two terms. It's equal to the squared norm of our least norm solⁿ + squared norm of the difference between our least norm solⁿ and any other solution we choose.

- The sum of those 2 numbers is greater than the squared least norm solⁿ. Therefore, what we've proven that any solⁿ that we choose has got to be greater than the least norm solution, and it's all hinged on the fact that any solⁿ that we choose has a dot product that results in 0.

What does this conclusion mean?

TOOLS: LAGRANGE MULTIPLIERS (How we can actually derive the least-norm solⁿ?)

& using analysis.

Instead of starting from the solution, start from the problem, vector of Lagrange multipliers

$$w_{\text{ln}} = \arg\min_{w} \; w^T w \quad \text{subject to} \quad Xw = y.$$

create an objective function by adding Lagrange multipliers.

- Introduce Lagrange multipliers: $\mathcal{L}(w,\eta) = w^T w + \eta^T (Xw - y)$. *

Goal:
- Minimize $\mathcal{L}$ over w maximize over η. If $Xw \neq y$, we can get $\mathcal{L} = +\infty$. **
- The optimal conditions are

By maximizing over η & minimizing over w, we must pick a w such that $Xw = y$

Gradient of an objective over w is equal to 0.

$$① \; \nabla_w \mathcal{L} = 2w + X^T \eta = 0, \quad ② \; \nabla_\eta \mathcal{L} = Xw - y = 0.$$

Gradient over eta is equal to zero.

We have everything necessary to find the solution:

1. From first condition: $w = -X^T \eta / 2$
2. Plug into second condition: $\eta = -2(XX^T)^{-1} y$
3. Plug this back into #1: $w_{\text{ln}} = X^T (XX^T)^{-1} y$

3 conditions from two equations?

What about other KKT conditions?

* ① 0, if the vector w satisfies the equality.
② non-zero, if doesn't satisfy " " (constraint)

** ① If w does not satisfy this equality, then we pick a vector η such that our maximum is ∞.
② whereas if we pick w, that does satisfy this equality, then this term must be equal to 0.

SPARSE ℓ_1 REGRESSION

(very useful and often done.)

restore noise as far as predicting y is concerned

Usually not suited for high-dimensional data

- Modern problems: Many dimensions/features/predictors
- Only a few of these may be important or relevant for predicting y
- Therefore, we need some form of "feature selection"
- Least squares and ridge regression: are not useful in this respect because
  - Treat all dimensions equally without favoring subsets of dimensions
  - The relevant dimensions are averaged with irrelevant ones
  - Problems: Poor generalization to new data, interpretability of results

Why?

And so we're going to discuss a method now for doing linear regression where we also try to find subsets of the dimensions in x that are going to be useful for predicting y.

Penalty terms

Recall: General ridge regression is of the form

For ridge regression we use dot product as our function

sum of squared errors — penalty on the squared norm of w.

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

↳ in a way our goodness of fit term

What are the other functions?

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

how well does our model fit the data

Penalized fitting

The general structure of the optimization problem is

$$\text{total cost} = \text{goodness-of-fit term} + \text{penalty term}$$ on the model parameters

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

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

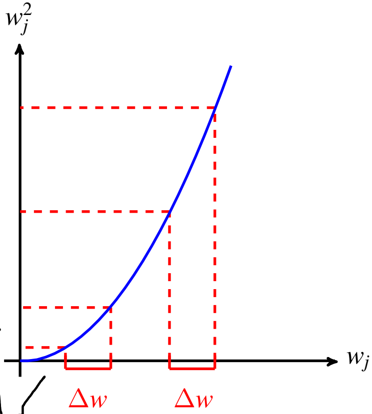
The squared norm penalty doesn't treat all values of w equally.

Start at a certain point, then we want to see how much do we reduce the objective func. by subtracting some Δw from that. If we reduce w_j by Δw, the total reduction in objective function depends on starting point

Intuitions

- Quadratic penalty: Reduction in cost depends on $|w_j|$.
- Suppose we reduce w_j by Δw. The effect on $\mathcal{L}$ depends on the starting point of w_j.
- Consequence: We should favor vectors w whose entries are of similar size, preferably small.

Why similar?



So the squared penalty on the vector w is going to 1st prefer to reduce the dimensions of w that have larger magnitudes, before trying to reduce the dimensions that have smaller magnitudes

Starting from a smaller value, that reduction in the magnitude of w reduces our penalty term by much less.

Conclusion:

l_2 norm is not necessarily going to be something that can give us sparsity ⇒

Why?

Because it going to try to make all of the values in the vector w equal in size. It's going to encourage solutions that have values that are equal in size. Because when a dimension of w becomes much larger in magnitude, the squared penalty squares that magnitude and suddenly that dimension is penalized quite a bit.

Whereas if all the values are roughly equal in size then they all contribute about the same amount to the penalty, the squared norm penalty.

The goal is to find sparse solutions. The squared norm penalty on w is not going to give us that.

SPARSITY (What does it mean to find sparse solutions?)

*We want to make the entries of w that have non-zero values as small as possible. while still predicting well

why small for non-zero) What is the gain in that.

Setting

- Regression problem with n data points $x \in \mathbb{R}^d$, $d \gg n$.
- 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?

essentially we want to make values in $w = 0$.

- Each entry of w corresponds to a dimension of the data x.
- If $w_k = 0$, the prediction is

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

the k^{th} dimension of x is not going to contribute anything to our prediction of y if we use the dot production as our function.

so the prediction does not depend on the kth dimension.

- * Feature selection: Find a 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: trying to find dimensions of w of our regression coefficient vector that should be equal to 0 and which one should not.

*

Penalty goal

Find a penalty term which encourages sparse solutions.

Quadratic penalty vs sparsity

- Suppose w_k is large, all other w_j are very small but non-zero
- Sparsity: Penalty should keep w_k, 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 small value.

Overall, a quadratic penalty favors many small, but non-zero values.

why?

Solution

Sparsity can be achieved using *linear* penalty terms.

* Squared norm is not going to give us sparse sol^{n}s.
Because it going to penalize large values more than small values. And so once the value of a dimension in w becomes very close to 0, the penalty is very little. And so we aren't going to necessarily encourage that to go zero with the squared norm penalty.
⇒ So it turns out that we can achieve this sparsity by using a linear term instead.

Sparse regression

LASSO: Least Absolute Shrinkage and Selection Operator

from ridge regression.

With the LASSO, we replace the ℓ_2 penalty with an ℓ_1 penalty:

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

where

$$\|w\|_1 = \sum_{j=1}^{d} |w_j|.$$

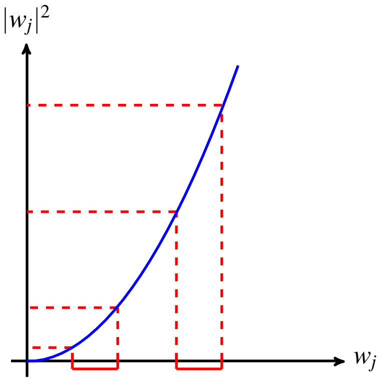
[Just the sum of absolute values of its entries]

This is also called ℓ_1-regularized regression.

Ridge regression think of as L_2-regularized regression

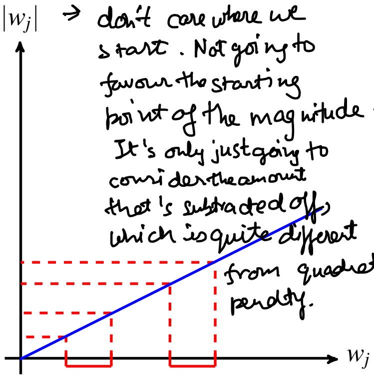
If we start at a certain value and subtract off a certain amount,it doesn't matter whether we start from a smaller value or from a higher value. Subtracting off the same amount is going to reduce the penalty by the same amount.

Quadratic penalty



Reducing a large value w_j achieves a larger cost reduction.

Linear penalty

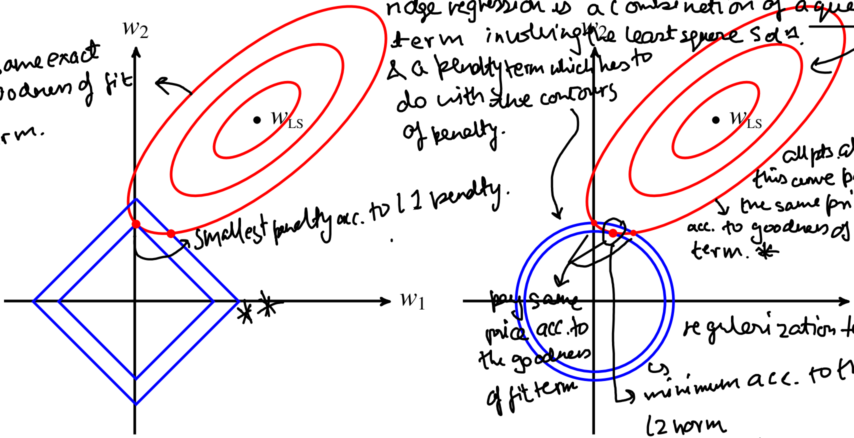


→ don't care where we start. Not going to favour the starting point of the magnitude. It's only just going to consider the amount that's subtracted off, which is quite different from quadratic penalty.

Cost reduction does not depend on the magnitude of w_j.

Ridge Regression vs LASSO [For visualisation, look at problem $d=2$]

we can represent the objective for l_2 problem where ridge regression is a combination of a quadratic term involving the least square sol^n. & a penalty term which has to do with the contours of penalty.



Same exact goodness of fit term.

smallest penalty acc. to l1 penalty.

**

all pts along this curve pay the same price acc. to goodness of fit term. *

pay same price acc. to the goodness of fit term

regularization term

minimum acc. to the l2 norm

***

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

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

* we can relate this term to the sum of squared errors term, as discussed in lec 3. So every value along this curve (ellipse) pays the same price acc. to the goodness of fit term.

** From this type of intuitive plot, we're going to get sparse solutions. So if we continue to shrink our sol^n acc. to a penalty (the diamond), with the constraint that the solution must follow along the ellipse. We're going to eventually pick a point that's zero along one of the axes.

Because we're going to eventually pick a point that's zero along one of the axes. Because we're going to hit one of the sharp points in the diamond and those fall along the zeros of the axes. → why?

why?

*** Whereas with the L2 penalty, we don't have any sharp points, we don't hit any those sharp points. And so we don't get a sparse sol^n.

don't understand the significance of sharp points?

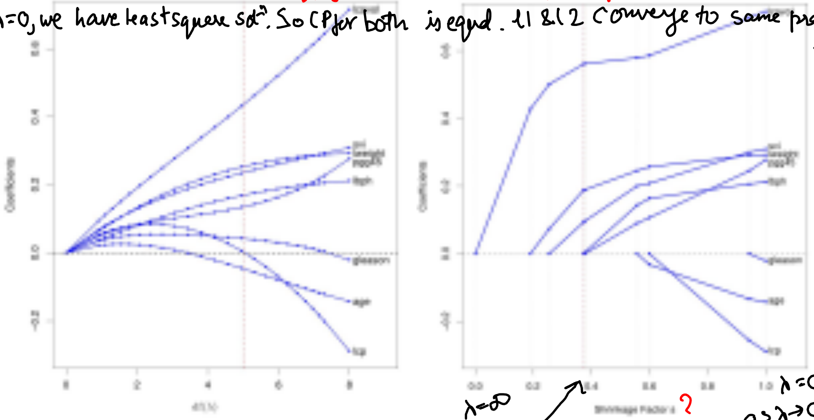
I can't visualize the example being applicable in a general scenario.

COEFFICIENT PROFILES: RR VS LASSO

CP for RR is non-sparse. The only points at which the profile crosses 0 is by coincidence.

No meaning to it)
Why would coefficient change from +ve to -ve with lambda variation?
Why doesn't it happen for L1?

When $\lambda=0$, we have least square sol^n. So CP for both is equal. l1 & l2 converge to same problem. *



$\lambda=\infty$

?

$\lambda=0$
as $\lambda \to 0$
$s \to 1$

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

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

And so this is the sparsity that we get from l_1. At a certain value of λ, For ex: we have 3 dimensions that are non-zero and the remaining are all equal to zero.

(Because ∞ times l_1 penalty is going to be 0 if.

as $\lambda \to \infty$
$S \to 0$

* As we increase lambda for l1 problem, what we see is that certain dimensions fall to zero and then stay at zero.

It's quite different from the l_2 problem where again, all dimensions are being shrunk to 0 as $\lambda \uparrow$. They're all being shrunk to 0 but they never hit zero.

Except by passing through coincidentally, or λ is ∞.

?

ℓ_p REGRESSION (we can generalise this to all norms, l_p norms.)

ℓ_p-norms (just a penalization on regression coefficients.)

These norm-penalties can be extended to all norms:

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

ℓ_p-regression

The ℓ_p-regularized linear regression problem is

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

We have seen:

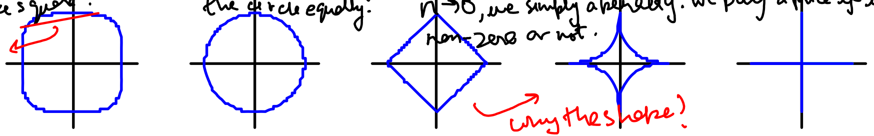
- ℓ_1-regression = LASSO $(p=1)$
- ℓ_2-regression = ridge regression $(p=2)$

ℓ_p PENALIZATION TERMS level sets of the penalty.

when p gets larger penalties look more like square.

When p=2, then we would penalize all vectors along the circle equally.

As p shrinks from 1 to 0, we then get this sort of an inverted penalty where it then starts shrinking to the point where in infinite limit as n→0, we simply a penalty. We pay a price if it's non-zero or not.



$p = 4$ $p = 2$ $p = 1$ $p = 0.5$ $p = 0.1$

how to show this with math?

p	Behavior of $\|\| . \|\|_p$
$p = \infty$	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)
$p \to 0$	Simply records whether an entry is non-zero, i.e. $\|\|w\|\|_0 = \sum_j \mathbb{I}\{w_j \neq 0\}$

lost the convexity of a set of points under a certain norm.

What are the ramifications of not being convex?

why isn't p=2 also convex?

Solution of ℓ_p problem

why?

ℓ_2 aka ridge regression. Has a closed form solution

ℓ_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 true solution can be found *exactly* using iterative algorithms

why not closed form?

(not convex) ($p < 1$) — We can only find an approximate solution (i.e., the best in its "neighborhood") using iterative algorithms. why? (Any case where we can't use them?)

only local optimal  p<1  p≥1

Three techniques formulated as optimization problems

what if n<d?

when not?

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

we didn't discuss an algorithm for this, but we did discuss how we can find the global solution to this problem using an iterative algorithm derived from convex optimisation.

which one? iterative?