COMS 4721: Machine Learning for Data Science
Lecture 4, 1/26/2017

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

Columbia University

Regression with/without regularization

pairs n observations

data response

Given:

A data set $(x_1, y_1), \ldots, (x_n, y_n)$, where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. We standardize such that each dimension of x is zero mean unit variance, and y is zero mean.

such that we work without the bias. (This impacts the results but not the math.)

Model:

We define a model of the form

$$y \approx f(x; w).$$

[very generic]

co-variate vector ↑ ↖ model parameter

We particularly focus on the case where $f(x; w) = x^T w$. [x evaluated at x, w pair is just the dot-product.]

function

Learning:

We can learn the model by minimizing the objective (aka, "loss") function

sum of square errors

$$\mathcal{L} = \sum_{i=1}^{n}(y_i - x_i^T w)^2 + \lambda w^T w \quad \Leftrightarrow \quad \mathcal{L} = \|y - Xw\|^2 + \lambda \|w\|^2$$

penalty of squared magnitude of w x regularization parameters.

We've focused on $\lambda = 0$ (least squares) and $\lambda > 0$ (ridge regression).

l2 norm regularisation

BIAS-VARIANCE TRADE-OFF

BIAS-VARIANCE FOR LINEAR REGRESSION

zero mean gaussian noise

w → regression coefficient

X → [each co-variate vector]

having covariance $\sigma^2 I$

N-dim vector

mean

regression coefficient vector.

We can go further and hypothesize a *generative* model $y \sim N(Xw, \sigma^2 I)$ and some true (but unknown) underlying value for the parameter vector w.

multi-variate Gaussian r.v.

[we believe that there is some true vector w generating our data. We hypothesize this but don't know what it is]

- We saw how the least squares solution, $w_{\text{LS}} = (X^T X)^{-1} X^T y$, is unbiased but potentially has high variance: (by minimizing squared errors)

  ① [meaning that the expected value of our least squared sol^n under this hypothesis is equal to the true vector w.]

  $$\mathbb{E}[w_{\text{LS}}] = w, \quad \text{Var}[w_{\text{LS}}] = \sigma^2 (X^T X)^{-1}.$$

  ② However, the variance of our least square sol^n can be v. large depending on whether this inverse matrix has v. large values.

- By contrast, the ridge regression solution is $w_{\text{RR}} = (\lambda I + X^T X)^{-1} X^T y$. Using the same procedure as for least squares, we can show that

  the expected ridge regression sol^n under this model hypothesis $N(Xw, \sigma^2 I)$

  $$\mathbb{E}[w_{\text{RR}}] = (\lambda I + X^T X)^{-1} X^T X w, \quad \text{Var}[w_{\text{RR}}] = \sigma^2 Z (X^T X)^{-1} Z^T,$$

  how?

  where $Z = (I + \lambda (X^T X)^{-1})^{-1}$.

Covariate : a variable that is possibly predictive of the outcome under the study.

But we know that our expected ridge solution relates to the true vector W by premultiplying by this matrix.

Letting $\lambda \to 0$,

expected, variance revert to least square values.

*unbiased expected sol^n is the truth.

The expectation and covariance of w_{LS} and w_{RR} gives insight into how well we can hope to learn w in the case where our model assumption is correct.
(meaning where $y \sim N(Xw, \sigma^2 I)$ is the correct model assumption.)

- Least squares solution: unbiased,* but potentially high variance
- Ridge regression solution: biased, but lower variance than LS

*don't expect it to be truth

So which is preferable?

Ultimately, we really care about how well our solution for w generalizes to new data. Let (x_0, y_0) be future data for which we have x_0, but not y_0.

- Least squares predicts $y_0 = x_0^T w_{\text{LS}}$ for new data
- Ridge regression predicts $y_0 = x_0^T w_{\text{RR}}$

This what we care about how well are we going to do with future data.

BIAS-VARIANCE FOR LINEAR REGRESSION

So if this the model assumption that we're going to make, then the performance that we use on the new data to measure their quality of sol^n should also take into consideration

In keeping with the square error measure of performance, we could calculate the the expected squared error of our prediction: squared error.

given all of the previous data we've seen

explain.

$$\mathbb{E}\left[(y_0 - x_0^T\hat{w})^2|X, x_0\right] = \int_{\mathbb{R}}\int_{\mathbb{R}^n}(y_0 - x_0^T\hat{w})^2 p(y|X, w)p(y_0|x_0, w)\,dy\,dy_0.$$

squared error on prediction.

integrating out here is the vector Y? ?

integrating out all response

- The estimate $\hat{w}$ is either w_{LS} or w_{RR}.
- The distributions on y, y_0 are Gaussian with the true (but unknown) w.
- We condition on knowing $x_0, x_1, \ldots, x_n$.

variables both what we have seen in the past & what we're going to see in future.

responses to test set acc. to distribution which I hypothesize.

In words this is saying:

but I don't know what it is.

- Imagine I know X, x_0 and assume some true underlying w.
- I generate $y \sim N(Xw, \sigma^2 I)$ and approximate w with $\hat{w} = w_{\text{LS}}$ or w_{RR}.
- I then predict $y_0 \sim N(x_0^T w, \sigma^2)$ using $y_0 \approx x_0^T\hat{w}$.

But I don't get see to y and I don't get to see y_0

What is the expected squared error of my prediction?

how can calculate $\hat{w}$ without y!

*

We integrate over both the distribution of Y given X and W.

So for a particular W, we calculate the distribution on Y.

** distribution of new response given true but unknown vector W.

$\hat{W}$ → depended on vector y and matrix X.

So when we're integrating out Y that integral is going to impact how y functions in this vector $\hat{w}$.

didn't understand, would y part come out?

AIM: we're going to see how this expected squared error of our prediction changes as a function of the data we have, the covariats, X_0 & x_0 and also some true but unknown underlying regression coefficient vector

don't get this) how can we ever measure the underlying coefficient.

BIAS-VARIANCE FOR LINEAR REGRESSION

Calculations imply we're conditioning on x & x_0

We can calculate this as follows (assume conditioning on x_0 and X),

$$\mathbb{E}[(y_0 - x_0^T \hat{w})^2] = \mathbb{E}[y_0^2] - 2\mathbb{E}[y_0]x_0^T\mathbb{E}[\hat{w}] + x_0^T\mathbb{E}[\hat{w}\hat{w}^T]x_0$$

linearity of expectation.

We can say that the expectation given X and x_0 of y_0 & $\hat{w}$ ($\mathbb{E}(y_0\hat{w})$) is equal to the product to expectation of $\mathbb{E}(y_0 \times \hat{w})$

- Since y_0 and $\hat{w}$ are independent, $\mathbb{E}[y_0\hat{w}] = \mathbb{E}[y_0]\mathbb{E}[\hat{w}]$.
- Remember: * $\mathbb{E}[\hat{w}\hat{w}^T] = \text{Var}[\hat{w}] + \mathbb{E}[\hat{w}]\mathbb{E}[\hat{w}]^T$ how?

  we again assume y_0 is a $N(X w, \sigma)$ ← $\mathbb{E}[y_0^2] = \sigma^2 + (x_0^T w)^2$ why? variance squared? why?

We're assuming that we have the data x_0 and x, and we're conditioning on ground truth w. We've made an independence assumption that the responses are independent of each other. So if you tell me what x_1 is & what w is, then y_1 is independent of y_2 given x_2 and w.

ain't we finding w?

why given w?

Why not a iid assumption?
Is the model considered not generative?

is this at the time of prediction?

* The expectation of outer product of a vector is equals to the variance of that vector times the outer product of the expectation of that vector.

We can calculate this as follows (assume conditioning on x_0 and X),

$$\mathbb{E}[(y_0 - x_0^T \hat{w})^2] = \mathbb{E}[y_0^2] - 2\mathbb{E}[y_0] x_0^T \mathbb{E}[\hat{w}] + x_0^T \mathbb{E}[\hat{w}\hat{w}^T] x_0$$

- Since y_0 and $\hat{w}$ are independent, $\mathbb{E}[y_0 \hat{w}] = \mathbb{E}[y_0]\mathbb{E}[\hat{w}]$.
- Remember:

$$\begin{aligned} \mathbb{E}[\hat{w}\hat{w}^T] &= \operatorname{Var}[\hat{w}] + \mathbb{E}[\hat{w}]\mathbb{E}[\hat{w}]^T \\ \mathbb{E}[y_0^2] &= \sigma^2 + (x_0^T w)^2 \end{aligned}$$

replaced y_0 with its mean because the gaussian assumption of noise y_0 is separated

Plugging these values in:

$$\begin{aligned} \mathbb{E}[(y_0 - x_0^T \hat{w})^2] &= \sigma^2 + (x_0^T w)^2 - 2(x_0^T w)(x_0^T \mathbb{E}[\hat{w}]) + (x_0^T \mathbb{E}[\hat{w}])^2 + x_0^T \operatorname{Var}[\hat{w}] x_0 \\ &= \sigma^2 + x_0^T (w - \mathbb{E}[\hat{w}])(w - \mathbb{E}[\hat{w}])^T x_0 + x_0^T \operatorname{Var}[\hat{w}] x_0 \end{aligned}$$

quadratic form

why ground truth w given here? NOW!

And now we just need to i/p the variance of either least squares / ridge regression soln. And the expected value of " / " depending whether $\hat{w}$ is " or " .

BIAS-VARIANCE FOR LINEAR REGRESSION

We have shown that if

1. $y \sim N(Xw, \sigma^2 I)$ and $y_0 \sim N(x_0^T w, \sigma^2)$, and
2. we approximate w with $\hat{w}$ according to some algorithm,

then

how our expected sol^n differs from true sol^n ① for LS this term is 0 ② for RR is non-zero, it is a biased approximation

$$\mathbb{E}[(y_0 - x_0^T \hat{w})^2 | X, x_0] = \underbrace{\sigma^2}_{noise} + \underbrace{x_0^T (w - \mathbb{E}[\hat{w}])(w - \mathbb{E}[\hat{w}])^T x_0}_{squared\ bias} + \underbrace{x_0^T \mathrm{Var}[\hat{w}] x_0}_{variance}$$

didn't understand how tradeoff changes here.

We see that the *generalization error* is a combination of three factors:

1. Measurement noise – we can't control this given the model.
2. Model bias – how close to the solution we expect to be on average.
3. Model variance – how sensitive our solution is to the data.

These 2 terms correspond to our uncertainty about $\hat{w}$ and its relationship to the true underlying w.

We saw how we can find $\mathbb{E}[\hat{w}]$ and $\mathrm{Var}[\hat{w}]$ for the LS and RR solutions.

How well we expect to do on new data given the algorithm that we used to approximate w with old data, is going to take into account these three terms.

① Simply the noise which we can never get rid of.
It's the noise of the sensor.

② The second term is how close we expect our sol^n to be to the truth. How biased is our solution.

③ The third term how much variance is there in our solution.

So a sol^n with low bias but very high variance, so no bias & very high variance which least square sol^n can have. We have II^nd term equal to 0 but III^rd term might be massive.

Whereas, another sol^n like the ridge regression sol^n. Which has some bias but potentially much smaller variance, can trade off some non-negative (positive) value here (II^nd term) for very big reduction of the value here (III^rd term) from the LS sol^n.

So that's the bias variance tradeoff.

So we can, again, calculate the variance.
So we can compare the variance of the ridge regression
solution to the least-square solution.
However, for the ridge regression solution, we don't know what w is.
And so even though we can calculate this term,
this entire term for least squares because w cancels here,
for ridge regression, we still have w in this term.
And so the bias really depends on what the value of w is.
So we can make some statements, potentially about values for
w that works out very well, or values of w where this term blows up.
And also, the relationship of our solution to the new observation
that we're trying to make a prediction for is factored in by using
x-naught here.

BIAS-VARIANCE TRADE-OFF

This idea is more general:

- Imagine we have a model: $y = f(x; w) + \epsilon,\ \mathbb{E}(\epsilon) = 0,\ \mathrm{Var}(\epsilon) = \sigma^2$
- We approximate f by minimizing a loss function: $\hat{f} = \arg\min_f \mathcal{L}_f$.
- We apply $\hat{f}$ to new data, $y_0 \approx \hat{f}(x_0) \equiv \hat{f}_0$.

Then integrating everything out (y, X, y_0, x_0):

$$\begin{aligned}
\mathbb{E}[(y_0 - \hat{f}_0)^2] &= \mathbb{E}[y_0^2] - 2\mathbb{E}[y_0 \hat{f}_0] + \mathbb{E}[\hat{f}_0^2] \\
&= \sigma^2 + f_0^2 - 2f_0 \mathbb{E}[\hat{f}_0] + \mathbb{E}[\hat{f}_0]^2 + \mathrm{Var}[\hat{f}_0] \\
&= \underbrace{\sigma^2}_{noise} + \underbrace{(f_0 - \mathbb{E}[\hat{f}_0])^2}_{squared\ bias} + \underbrace{\mathrm{Var}[\hat{f}_0]}_{variance}
\end{aligned}$$

This is interesting in principle, but is deliberately vague (What is f?) and usually can't be calculated (What is the distribution on the data?)

Cross-validation

An easier way to evaluate the model is to use cross-validation.

The procedure for K-fold cross-validation is very simple:

1. Randomly split the data into K roughly equal groups.
2. Learn the model on $K-1$ groups and predict the held-out Kth group.
3. Do this K times, holding out each group once.
4. Evaluate performance using the cumulative set of predictions.

gives us a sense of on how our model will perform on unseen data

For the case of the regularization parameter λ, the above sequence can be run for several values with the best-performing value of λ chosen.

The data you test the model on should never be used to train the model!

just the sum of performances on each held-out group

1	2	3	4	5
Train	Train	Validation	Train	Train

BAYES RULE

very useful for quantifying our uncertainty in model parameters.

→ discuss it in the context of linear regression problems so far.

Motiving through ridge regression.

isn't this just a constraint, why are you including it in the measure of generalizability.

a measure of how well the vector w can predict the data that we have.

Motivation

We've discussed the ridge regression objective function

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

like a prior belief on what w should be. So in a sense it constraints w. It imposes a prior belief on the idea that the mag. of w shouldn't be too large.

The regularization term $\lambda w^T w$ was imposed to penalize values in w that are large. This reduced potential high-variance predictions from least squares.

In a sense, we are imposing a "prior belief" about what values of w we consider to be good.

Question: Is there a mathematical way to formalize this?

Answer: Using probability we can frame this via Bayes rule.

Review: Probability statements

Imagine we have two events, A and B, that may or may not be related, e.g.,

- A = “It is raining”
- B = “The ground is wet”

We can talk about probabilities of these events,

- $P(A)$ = Probability it is raining
- $P(B)$ = Probability the ground is wet

We can also talk about their *conditional* probabilities,

- $P(A|B)$ = Probability it is raining *given* that the ground is wet
- $P(B|A)$ = Probability the ground is wet *given* that it is raining

We can also talk about their *joint* probabilities,

- $P(A,B)$ = Probability it is raining *and* the ground is wet

There are simple rules from moving from one probability to another

1. $P(A, B) = P(A|B)P(B) = P(B|A)P(A)$
2. $P(A) = \sum_b P(A, B = b)$ → Marginal probability. (When you add that extra word marginal you're implying there's some additional event that you're integrating or summing out.
3. $P(B) = \sum_a P(A = a, B)$

↳ sum/integrated over all possible values of A.

Using these three equalities, we automatically can say

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{\sum_a P(B|A = a)P(A = a)}$$

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{P(A|B)P(B)}{\sum_b P(A|B = b)P(B = b)}$$

is marginal probability of joint probability of a & b where we integrated/summed over a.

This is known as "Bayes rule." → allows us to quantify what we don't know using prob. distributions.

BAYES RULE

P(B|A) → Posterior belief of unknown b, given the observation a.
P(A|B) → Likelihood of seeing what we saw given a certain setting for b.

Bayes rule lets us quantify what we don't know. Imagine we want to say something about the probability of B given that A happened. likelihood of observing what we saw given a certain setting for b.

Bayes rule says that the probability of B after knowing A is:

prob. after obtaining some data A.

$$\underbrace{P(B|A)}_{\textit{posterior}} = \underbrace{P(A|B)}_{\textit{likelihood}}\,\underbrace{P(B)}_{\textit{prior}} \;/\; \underbrace{P(A)}_{\textit{marginal}}$$

a probability of b that we have to define apriori.

Notice that with this perspective, these probabilities take on new meanings.

That is, $P(B|A)$ and $P(A|B)$ are both "conditional probabilities," but they have different significance.

When we think about what we do and don't know. These manipulations on previous slide which are just simply conditional & joint and marginal probabilities take on new meanings.

Bayes rule with continuous variables

Bayes rule generalizes to continuous-valued random variables as follows. However, instead of *probabilities* we work with *densities*.

- Let θ be a continuous-valued model parameter.
- Let X be data we possess. Then by Bayes rule,

likelihood of data given the model variable.

Prior probability of model variable.

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta} = \frac{p(X|\theta)p(\theta)}{p(X)}$$

integral of numerator over all possible values of the model variable, which is just marginal prob. of the data.

Posterior probability of the model parameter/variable given the data we've observed.

In this equation,

So in this case, using Bayes rule for model learning we possess the knowledge of everything necessary in principle.

- $p(X|\theta)$ is the likelihood, known from the model definition.
- $p(\theta)$ is a prior distribution that we define.
- Given these two, we can (in principle) calculate $p(\theta|X)$.

So for example, the probability of the data given the model variable, which is the likelihood is known from the model definition. So we define some generative distribution on our data, given some model with its unknown variables.

↑ $p(X|\theta)$

$p(\theta) \rightarrow$ The prior probability on the model variables is also something that we're going to define.

$p(\theta|X) \rightarrow$ So what we want is the posterior probability of all model variables, and we can write that in terms of things that we know. And then the only question is whether we can calculate this integral $(p(X|\theta)p(\theta))$ in the denominator in order to give a closed form analytic expression for the posterior probability.

EXAMPLE: COIN BIAS (example) We are trying to learn the bias of the coin.

What it means to be independent: $X_1 \dots X_n$ are independent of each other if the likelihood of all data is just equal to the product of likelihood of each observation.

We have a coin with bias π towards "heads". (Encode: heads = 1, tails = 0)

We flip the coin many times and get a sequence of n independently numbers $(x_1, \dots, x_n)$. Assume the flips are independent, meaning

Bernoulli random variable

$$p(x_1, \dots, x_n|\pi) = \prod_{i=1}^{n} p(x_i|\pi) = \prod_{i=1}^{n} \pi^{x_i}(1-\pi)^{1-x_i}.$$

We choose a prior for π which we define to be a beta distribution,

a, b > 0 (strictly positive)

$$p(\pi) = Beta(\pi|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\pi^{a-1}(1-\pi)^{b-1}.$$

What is the posterior distribution of π given $x_1, \dots, x_n$?
(the sequence that we've observed)

EXAMPLE: COIN BIAS

From Bayes rule,

likelihood of a particular sequence given the coin bias times the prior on the coin bias.

$$p(\pi|x_1,\ldots,x_n) = \frac{p(x_1,\ldots,x_n|\pi)p(\pi)}{\int_0^1 p(x_1,\ldots,x_n|\pi)p(\pi)d\pi}.$$

integral over all possible values π can take.

There is a trick that is often useful:

- The denominator only normalizes the numerator, doesn't depend on π. (so it is a function of π that integrates to 1) completely removed by integrating it out
- We can write $p(\pi|x) \propto p(x|\pi)p(\pi)$. ("$\propto$" $\rightarrow$ "proportional to")

  posterior $\propto$ likelihood prior
- Multiply the two and see if we recognize anything:

$$p(\pi|x_1,\ldots,x_n) \propto \underbrace{\left[\prod_{i=1}^n \pi^{x_i}(1-\pi)^{1-x_i}\right]}_{\text{likelihood}} \underbrace{\left[\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\pi^{a-1}(1-\pi)^{b-1}\right]}_{\text{prior}}$$

$$\propto \pi^{\sum_{i=1}^n x_i + a - 1}(1-\pi)^{\sum_{i=1}^n (1-x_i) + b - 1}$$

since we are working with proportionality this term doesn't involve π at all. So when we normalize the product of these 2 terms, this term is going to appear in both numerator & denominator, it's gonna cancel out.

We recognize this as $p(\pi|x_1,\ldots,x_n) = Beta(\sum_{i=1}^n x_i + a, \sum_{i=1}^n (1-x_i) + b)$

So if we're only interested in giving the posterior as a function that's proportial. Meaning the posterior is equal to this function times some no. such that it integrates to 1, we don't have to worry about any term like that multiplied onto function of pi, that doesn't involve pi.

Do we recognize any distributions that are proportial to this function.

→ Beta

And so here we have a posterior probability distribution on the bias of the coin takes into account the prior and the data and gives us a measure of the uncertainity of what π is as represented by this function.

(we can now relate ridge regression to a probability distribution)

this is something called

MAXIMUM A POSTERIORI

Least squares and maximum likelihood

When we modeled data pairs (x_i, y_i) with a linear model, $y_i \approx x_i^T w$, we saw that the least squares solution, (the value of w that minimizes the sum of squared errors.)

$$w_{\text{LS}} = \arg\min_{w} (y - Xw)^T (y - Xw),$$

was equivalent to the maximum likelihood solution when $y \sim N(Xw, \sigma^2 I)$..

So we can view the LS sol^{n}. probabilistically as being a max. likelihood sol^{n} for this model on the data y.

The question now is whether a similar probabilistic connection can be made for the ridge regression problem.

First, make an assumption of a prior model for the vector w.

y is generated from a multi-variate Gaussian

Ridge regression and Bayesian modeling

The likelihood model is $y \sim N(Xw, \sigma^2 I)$. What about a prior for w?

Let us assume that the prior for w is Gaussian, $w \sim N(0, \lambda^{-1} I)$. Then

d-dimensional multi-variate Gaussian

density function that corresponds to this prob. distribution

$$p(w) = \left(\frac{\lambda}{2\pi}\right)^{\frac{d}{2}} e^{-\frac{\lambda}{2} w^T w}.$$

We can now try to find a w that satisfies both the data likelihood, and our prior conditions about w.

(belief) vector

MAXIMUM A POSERIORI ESTIMATION

[log is monotonic, so same value max. Concept as before.]

Maximum *a poseriori* (MAP) estimation seeks the most probable value w under the posterior:

posterior distribution of w given y & X.

So this is what we want to ultimately maximize over w.

$$\begin{aligned} w_{\text{MAP}} &= \arg\max_w \ \ln \overbrace{p(w|y,X)}^{} \\ &= \arg\max_w \ \ln \frac{p(y|w,X)p(w)}{p(y|X)} \quad \text{Bayes rule} \\ &= \arg\max_w \ \underbrace{\ln p(y|w,X)}_{\text{likelihood}} + \underbrace{\ln p(w)}_{\text{prior}} - \ln \cancel{p(y|X)} \end{aligned}$$

product to sum

normalizing constant, evidence term, marginal likelihood of y given X

- Contrast this with ML, which only focuses on the likelihood.

We have added 2 terms. ① log of prior ② Additional term that doesn't actually involve w.

- The normalizing constant term $\ln p(y|X)$ doesn't involve w. Therefore, we can maximize the first two terms alone.

We are interested in the location of w that maximizes (argmax) & not max

- In many models we don't know $\ln p(y|X)$, so this fact is useful.

(in many models we can't calculate this integral in closed form.)

MAP FOR LINEAR REGRESSION

MAP using our defined prior gives:

since we consumed a Gaussian likelihood.

likelihood prior.

$$w_{\text{MAP}} = \arg\max_w \; \ln p(y|w,X) + \ln p(w)$$

$$= \arg\max_w \; -\frac{1}{2\sigma^2}(y - Xw)^T(y - Xw) - \frac{\lambda}{2}w^Tw + \text{const.}$$

removed constants that don't impact the solⁿ.

Calling this objective $\mathcal{L}$, then as before we find w such that

Maximize over find gradient of $\mathcal{L}$

$$\nabla_w \mathcal{L} = \frac{1}{\sigma^2}X^Ty - \frac{1}{\sigma^2}X^TXw - \lambda w = 0$$

why not y^TX)

- The solution is $w_{\text{MAP}} = (\lambda\sigma^2 I + X^TX)^{-1}X^Ty$.
- Notice that $w_{\text{MAP}} = w_{\text{RR}}$ (modulo a switch from λ to $\lambda\sigma^2$)
- RR maximizes the posterior, while LS maximizes the likelihood.

Just like least squares solⁿ maximizes the likelihood. So it corresponds to max. likelihood solⁿ. Ridge regression maximises the posterior under 0 mean Gaussian prior assumption on w. And so the ridge regression solution corresponds to the map solⁿ. $w_{LS} = w_{ML}$ $w_{RR} = w_{MAP}$

Simply redefined original λ to $\lambda\sigma^2$