

COMS 4721: Machine Learning for Data Science

Lecture 4, 1/26/2017

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute
Columbia University

Given:

A data set $(x_1, y_1), \dots, (x_n, y_n)$, 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.

Model:

We define a model of the form

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

We particularly focus on the case where $f(x; w) = x^T w$.

Learning:

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

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

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

BIAS-VARIANCE TRADE-OFF

NMC - Unbiased Estimator

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 .

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

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

$$\mathbb{E}(w_{LS}) = w_{LS}$$

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

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

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

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.

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

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 $y_0 = x_0^T w_{LS}$
- ▶ Ridge regression predicts $y_0 = x_0^T w_{RR}$

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

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

- ▶ 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, \dots, x_n .

In words this is saying:

- ▶ 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 $y_0 \approx x_0^T \hat{w}$.

What is the expected squared error of my prediction?

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: $\mathbb{E}[\hat{w}\hat{w}^T] = \text{Var}[\hat{w}] + \mathbb{E}[\hat{w}]\mathbb{E}[\hat{w}]^T$

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

$$y_0^2 = (x_0^T w)^2 + 2(x_0^T w)\epsilon + \epsilon^2$$

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

BIAS-VARIANCE FOR LINEAR REGRESSION

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: $\mathbb{E}[\hat{w}\hat{w}^T] = \text{Var}[\hat{w}] + \mathbb{E}[\hat{w}]\mathbb{E}[\hat{w}]^T$

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

Plugging these values in:

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

We have shown that if

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

then

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

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. ✓

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

This idea is more general:

- Imagine we have a model: $y = f(x; w) + \epsilon$, $\mathbb{E}(\epsilon) = 0$, $\text{Var}(\epsilon) = \sigma^2$
- 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):

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

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?)

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 K th group.
3. Do this K times, holding out each group once.
4. Evaluate performance using the cumulative set of predictions.

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!

1	2	3	4	5
Train	Train	Validation	Train	Train

BAYES RULE

Motivation

We've discussed the ridge regression objective function

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

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.

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 for moving from one probability to another

1. $P(A, B) = P(A|B)P(B) = P(B|A)P(A)$ ✓
2. $P(A) = \sum_b P(A, B = b)$ ✓
3. $P(B) = \sum_a P(A = a, B)$ ✓ = $P(B|A=a)P(A=a)$

Using these three equalities, we automatically can say

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

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

This is known as “Bayes rule.”

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.

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

$$P(B|A) = P(A|B) P(B) / P(A)$$

posterior
likelihood
prior
marginal

The diagram shows the formula $P(B|A) = P(A|B) P(B) / P(A)$ with each term highlighted in yellow. Below the terms are curly braces and labels: $P(B|A)$ is labeled 'posterior', $P(A|B)$ is labeled 'likelihood', $P(B)$ is labeled 'prior', and $P(A)$ is labeled 'marginal'. A large, hand-drawn oval encircles the entire formula and its labels.

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.

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, ✓

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

In this equation,

- ▶ $p(X|\theta)$ 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 $p(\theta|X)$.

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 numbers (x_1, \dots, x_n) .

Assume the flips are independent, meaning

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

$$x_i = 0 \text{ or } 1$$

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

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

What is the posterior distribution of π given x_1, \dots, x_n ?

$$p(\pi | x_1, x_2, \dots, x_n)$$

From Bayes rule,

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

There is a trick that is often useful:

- ▶ The denominator only normalizes the numerator, doesn't depend on π .
- ▶ We can write $p(\pi|x) \propto p(x|\pi)p(\pi)$. (“ \propto ” \rightarrow “proportional to”)
- ▶ Multiply the two and see if we recognize anything:

$$\begin{aligned} p(\pi|x_1, \dots, x_n) &\propto \left[\prod_{i=1}^n \pi^{x_i} (1 - \pi)^{1-x_i} \right] \left[\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \pi^{a-1} (1 - \pi)^{b-1} \right] \\ &\propto \pi^{\sum_{i=1}^n x_i + a - 1} (1 - \pi)^{\sum_{i=1}^n (1-x_i) + b - 1} \end{aligned}$$

doesn't have π

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

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,

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

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

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

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

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

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

Ridge regression has prior condition on w , using the penalty

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

$$\begin{aligned}
 w_{\text{MAP}} &= \arg \max_w \ln p(w|y, X) \\
 &= \arg \max_w \ln \frac{p(y|w, X)p(w)}{p(y|X)} \\
 &= \arg \max_w \ln p(y|w, X) + \ln p(w) - \ln p(y|X)
 \end{aligned}$$

Bayes's

$p(\theta/X) = \frac{p(X|\theta)p(\theta)}{p(X)}$
 posterior = likelihood = prior

- ▶ Contrast this with ML, which only focuses on the likelihood.
- ▶ The normalizing constant term $\ln p(y|X)$ doesn't involve w . Therefore, we can maximize the first two terms alone.
- ▶ In many models we don't know $\ln p(y|X)$, so this fact is useful. ✓

MAP using our defined prior gives:

$$\begin{aligned} w_{\text{MAP}} &= \arg \max_w \ln p(y|w, X) + \ln p(w) \\ &= \arg \max_w -\frac{1}{2\sigma^2} (y - Xw)^T (y - Xw) - \frac{\lambda}{2} w^T w + \text{const.} \end{aligned}$$

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

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

- ▶ The solution is $w_{\text{MAP}} = (\lambda \sigma^2 I + X^T X)^{-1} X^T y$.
- ▶ 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.**

[We redefined original λ to $\lambda \sigma^2$]

from $w \sim p(w)$

→ we used Maximum Likelihood (ML) Estimation for linear Regression, whereas we used MAP Estimation for Ridge Regression cause it has a prior (i.e. w has a distribution, i.e. w is bounded by distribution.)