Day 12 - Ridge Regression and Model Validation

Agenda:

- Ridge Regression
- Bayesian Statistics
- Maximum A Posteriori Estimation vs Maximum Likelihood Estimatio
- Ridge Regression from a Bayesian Perspective
- Model Validation

Ridge Regression

Idea: penalize predictors that have large values of unknown parameters

Ridge formulation for least squares:

Given data $\{(x_i, y_i)\}_{i=1,\dots,n}$ w/ $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$

where $y = X\theta + \varepsilon$ w/ $\varepsilon \in \mathbb{R}^n$ has $N(0, \sigma^2)$ entries

Estimate θ by solving — ridge regression problem

$$\min_{\theta} \; \|y - X\theta\|^2 + \lambda \|\theta\|^2$$

$\lambda \|\theta\|^2$: ℓ_2 penalization / ℓ_2 regularization / weight decay

Solution is given by

$$\hat{\theta}_{ridge} = (X^t X + \lambda I_{d\times d})^{-1} X^t y$$

w/ $I_{d\times d}$ = $d \times d$ Identity matrix $= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

λ trades off between bias & variance

Small λ = low bias, high variance

large λ = high bias, low variance

Bayesian Statistics

We have been doing parameter estimation without any prior information about our parameters

Example:

Frequentist Perspective

Flip a coin 4 times.

You get H each time.

What is your estimate of bias of the coin?

$\hat{P}_{heads} = 1$

by maximum likelihood estimation

Bayesian Perspective

There are 2 urns.
One with fair coins, one with double tailed coins.
Choose a coin from a random urn

} prior information

Flip a coin 4 times.

You get H each time.

What is your estimate of bias of the coin?

$\hat{P}_{heads} = 0.5$

Example: You flip a coin once. You get H.

What is MLE estimate of P_{heads}

$\hat{P}_{heads} = 1$

There are 2 urns.
Urn 1 has a coin w/ $P(H) = 0.75$
Urn 2 — — — — — $= 0.25$

You flip a coin from a random urn once. You get H.

What is your estimate of $P(H)$ of the coin you drew?

Bayes' Theorem

$$P(\text{urn } 1 \mid H) = \frac{P(H \mid \text{urn } 1)\,P(\text{urn } 1)}{P(H)}$$

$$= \frac{0.75 \cdot 0.5}{0.75 \cdot 0.5 + 0.25 \cdot 0.5}$$

$$= 0.75$$

$$P(\text{urn } 2 \mid H) = 0.25.$$

It is most likely that the coins bias is $\boxed{0.75.}$

Bayes Theorem

Let X, Y have a joint distribution

$$P(X \mid Y) = \frac{P(Y \mid X)\,P(X)}{P(Y)}$$

works for discrete or continuous r.v.s

in this case P is likelihood

Maximum Likelihood Estimation vs Maximum A Posteriori Estimation

Given $S = \{ y_i \}_{i=1 \dots n}$

Given parametric model of $Y \sim f(y; \theta)$

pdf of Y for given θ

MLE:

$$\underset{\theta}{\operatorname{argmax}} \; P(S \mid \theta)$$

$$= \underset{\theta}{\operatorname{argmax}} \; \underbrace{\prod_{i=1}^{n} f(y_i; \theta)}_{\text{likelihood of data}}$$

"find paramaters that make data as likely as possible"

MAP:

Further given a prior distribution $Y \sim P(y)$

pdf of Y

$$\underset{\theta}{\operatorname{arg\,max}} \; P(\theta \mid S)$$

$$= \underset{\theta}{\operatorname{argmax}} \; \frac{P(S \mid \theta) \, P(\theta)}{P(S)}$$

$$= \underset{\theta}{\operatorname{argmax}} \; P(S \mid \theta) \, P(\theta)$$

$$= \underset{\theta}{\operatorname{arg\,max}} \; \log P(S \mid \theta) + \log P(\theta)$$

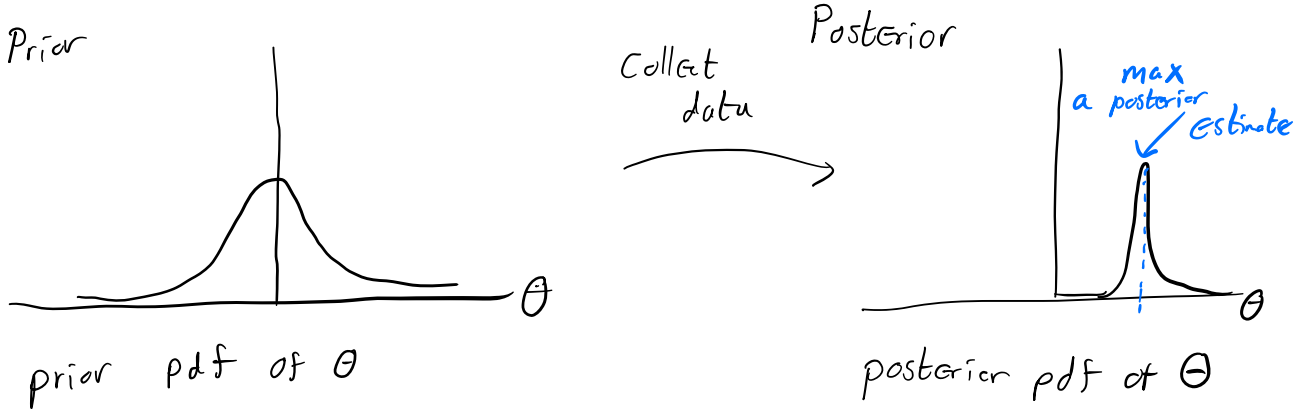
term from MLE

regularization by the prior

"find most likely parameters given the data"

Caveat: you need a prior

Visualization: Your uncertainty over θ changes after you collect data.



prior pdf of θ | posterior pdf of θ

MAP is the special case of MLE for an uninformative prior

MLE

$$\underset{\theta}{\text{argmax}}\ P(S|\theta)$$

MAP

$$\arg\max_{\theta} P(\theta|S)$$

$$= \arg\max_{\theta} \log P(S|\theta) + \log P(\theta)$$

These are the equivalent if $\log P(\theta)$ is constant in θ. (no value of θ is more or less likely than any other value of θ) Eg uniform distribution

Example:

Suppose $\theta \sim N(0, \gamma^2)$ for known γ, σ.

$X \sim N(\theta, \sigma^2)$

You have a dataset $S = \{X_1\}$ all from some parameter θ.

What is MAP estimate of θ?

$$\hat{\theta}_{map} = \arg\max_\theta \log P(S|\theta) + \log P(\theta)$$

$$= \arg\max_\theta \log \frac{1}{\sqrt{2\pi}\sigma} e^{-(X_1-\theta)^2/2\sigma^2} + \log \frac{1}{\sqrt{2\pi}\gamma} e^{-\theta^2/2\gamma^2}$$

$$= \arg\max_\theta -\frac{(X_1-\theta)^2}{2\sigma^2} - \frac{\theta^2}{2\gamma^2}$$

$$= \arg\min_\theta (X_1-\theta)^2 + \left(\frac{\sigma^2}{\gamma^2}\right)\theta^2$$

Solution given by $-2(X_1-\theta) + 2\frac{\sigma^2}{\gamma^2}\theta = 0$

$$X_1 - \theta = \frac{\sigma^2}{\gamma^2}\theta \Rightarrow \boxed{\hat{\theta}_{MAP} = \frac{X_1}{1+\sigma^2/\gamma^2}}$$

Note: MLE estimate is

$$\hat{\theta}_{MLE} = \arg\max_\theta \log P(S|\theta) = \arg\min_\theta (X_1-\theta)^2$$

$$= X_1$$

$$\boxed{\hat{\theta}_{MLE} = X_1}$$

Ridge Regression from a Bayesian Perspective

Estimate θ by solving ← ridge regression problem

$$\min_{\theta} \; \|y - X\theta\|^2 + \lambda \|\theta\|^2$$

λ: ℓ_2 penalization / ℓ_2 regularization / weight decay

Viewing this problem from a Bayesian perspective, we see it as MAP estimation w/ a Bayesian Prior

Suppose $\theta \sim N(0, \gamma^2 I_d) \in \mathbb{R}^d$

$y \sim N(X^t\theta, \sigma^2 I_n) \in \mathbb{R}^n$, $X \in \mathbb{R}^d$

MAP estimate given by

$$\begin{aligned}
&\arg\max_{\theta} \log P(S|\theta) + \log P(\theta) \\
&= \arg\max_{\theta} \cancel{\log\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n} - \frac{\|y - X^t\theta\|^2}{2\sigma^2} + \cancel{\log\left(\frac{1}{\sqrt{2\pi}\gamma}\right)^d} - \frac{\|\theta\|^2}{2\gamma^2} \\
&= \arg\min_{\theta} \frac{\|y - X^t\theta\|^2}{2\sigma^2} + \frac{\|\theta\|^2}{2\gamma^2} \\
&= \arg\min_{\theta} \|y - X^t\theta\|^2 + \frac{\sigma^2}{\gamma^2}\|\theta\|^2 \qquad \frac{\sigma^2}{\gamma^2} = \lambda
\end{aligned}$$

So, ridge regression is MAP estimation under a Gaussian Prior puts a bias toward small values of θ (higher likelihoods in prior)

Model Validation

Suppose you have multiple trained predictors you are choosing between. How do you select the best one?

Eg Ridge regression

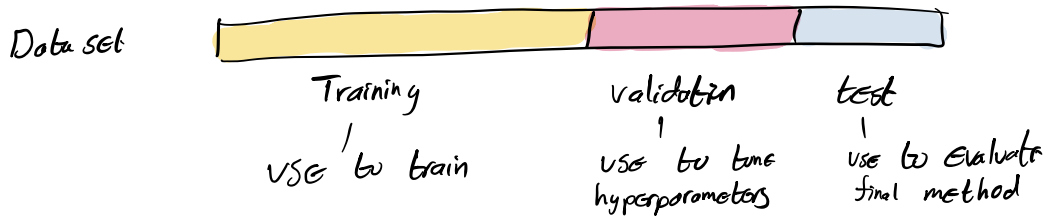
parameter

$$\min_{\theta} \|y - X\theta\|^2 + \lambda \|\theta\|^2$$

Which value of λ should you choose?

λ: hyperparameter

Ideally, use validation data



$$\min_{\lambda} \frac{1}{m} \sum_{i=1}^{m} \ell\left(y_i^{val}, f_{\lambda}(x_i)\right)$$

f_{λ}: predictor w/ hyperparameter λ

Challenges: Need data for validation,
Need independent data for test

data is often expensive

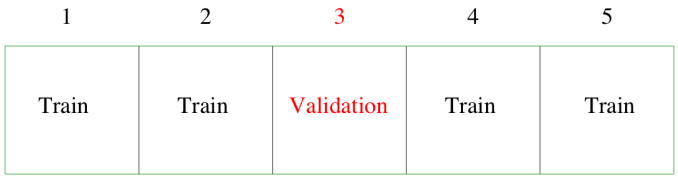
K-fold Cross Validation

Chap 7 of Hastie

7.10.1 K-Fold Cross-Validation

Ideally, if we had enough data, we would set aside a validation set and use it to assess the performance of our prediction model. Since data are often scarce, this is usually not possible. To finesse the problem, K-fold cross-validation uses part of the available data to fit the model, and a different part to test it. We split the data into K roughly equal-sized parts; for example, when $K = 5$, the scenario looks like this:

"folds"



For the kth part (third above), we fit the model to the other $K-1$ parts of the data, and calculate the prediction error of the fitted model when predicting the kth part of the data. We do this for $k = 1, 2, \ldots, K$ and combine the K estimates of prediction error.

Here are more details. Let $\kappa : \{1, \ldots, N\} \mapsto \{1, \ldots, K\}$ be an indexing function that indicates the partition to which observation i is allocated by the randomization. Denote by $\hat{f}^{-k}(x)$ the fitted function, computed with the kth part of the data removed. Then the cross-validation estimate of prediction error is

$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i)). \tag{7.48}$$

avg over each datapoint

train on all data not in the fold containing i^{th} data point

Equivalently: average over K folds of the average loss on each fold when trained on all other folds

Typical choices of K are 5 or 10 (see below). The case $K = N$ is known as *leave-one-out* cross-validation. In this case $\kappa(i) = i$, and for the ith observation the fit is computed using all the data except the ith.

Given a set of models $f(x, \alpha)$ indexed by a tuning parameter α, denote by $\hat{f}^{-k}(x, \alpha)$ the αth model fit with the kth part of the data removed. Then for this set of models we define

$$\mathrm{CV}(\hat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i, \alpha)). \tag{7.49}$$

The function $\mathrm{CV}(\hat{f}, \alpha)$ provides an estimate of the test error curve, and we find the tuning parameter $\hat{\alpha}$ that minimizes it. Our final chosen model is $f(x, \hat{\alpha})$, which we then fit to all the data.

Hyper parameter tuning

How to choose # of folds?

If $K=N$,
- have to solve N problems
  $\Rightarrow$ expensive.
- low bias, high variance

If K too small, not enough data used for training model.

Compromise $k = 5$ or 10

7.10.2 The Wrong and Right Way to Do Cross-validation

Consider a classification problem with a large number of predictors, as may arise, for example, in genomic or proteomic applications. A typical strategy for analysis might be as follows:

1. Screen the predictors: find a subset of "good" predictors that show fairly strong (univariate) correlation with the class labels
2. Using just this subset of predictors, build a multivariate classifier.
3. Use cross-validation to estimate the unknown tuning parameters and to estimate the prediction error of the final model.

Is this a correct application of cross-validation?

Here is the correct way to carry out cross-validation in this example:

1. Divide the samples into K cross-validation folds (groups) at random.
2. For each fold $k = 1, 2, \ldots, K$
    (a) Find a subset of "good" predictors that show fairly strong (univariate) correlation with the class labels, using all of the samples except those in fold k.
    (b) Using just this subset of predictors, build a multivariate classifier, using all of the samples except those in fold k.
    (c) Use the classifier to predict the class labels for the samples in fold k.

The error estimates from step 2(c) are then accumulated over all K folds, to produce the cross-validation estimate of prediction error. The lower panel

In general, with a multistep modeling procedure, cross-validation must be applied to the entire sequence of modeling steps. In particular, samples must be "left out" before any selection or filtering steps are applied. There is one qualification: initial *unsupervised* screening steps can be done before samples are left out. For example, we could select the 1000 predictors with highest variance across all 50 samples, before starting cross-validation. Since this filtering does not involve the class labels, it does not give the predictors an unfair advantage.