Lecture 11: Cross validation

Reading: Chapter 5

Lester Mackey
October 14, 2015

(Slide credits: Sergio Bacallado)

Announcements

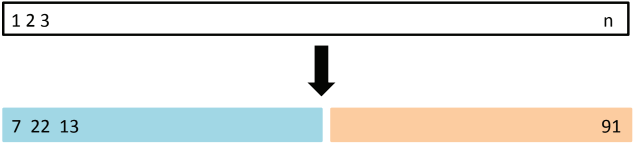
- Homework 4 is online.
- Two practice midterm exams are online. Will discuss midterm details on Friday.

Validation set approach

Goal: Estimate the test error for a supervised learning method.

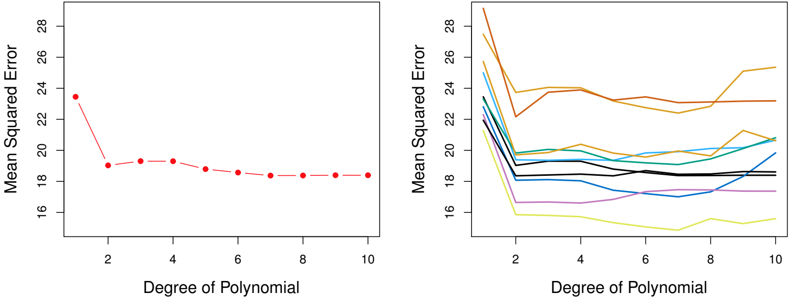
Strategy:

- Split the data in two parts.
- Train the method in the first part.
- Compute the error on the second part.



Validation set approach

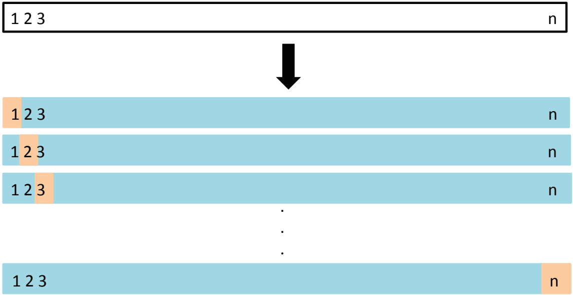
Polynomial regression to estimate `mpg` from `horsepower` in the Auto data.



- Estimates can vary considerably with different train-validation splits.
- Only subset of points used to evaluate model.

Leave one out cross-validation

- For every $i = 1, \ldots, n$:
  - train the model on every point except i,
  - compute the test error on the held out point.
- Average the test errors.



Leave one out cross-validation

- For every $i = 1, \ldots, n$:
  - train the model on every point except i,
  - compute the test error on the held out point.
- Average the test errors.

$$\mathsf{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i^{(-i)})^2$$

Prediction for the i sample without using the ith sample.

Leave one out cross-validation

- For every $i = 1, \ldots, n$:
  - train the model on every point except i,
  - compute the test error on the held out point.
- Average the test errors.

$$\mathsf{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(y_i \neq \hat{y}_i^{(-i)})$$

... for a classification problem.

Leave one out cross-validation

Computing $\text{CV}_{(n)}$ can be computationally expensive, since it involves fitting the model n times.

- A single linear regression fit takes $O(p^3 + np^2)$ time

For linear regression, there is a shortcut:

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2$$

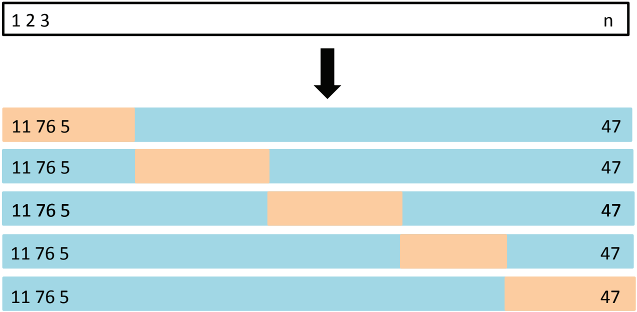
where

$$h_i = \frac{\partial \hat{y}_i}{\partial y_i} = (\underbrace{\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T}_{\text{Hat matrix } \mathbf{H}})_{i,i}$$

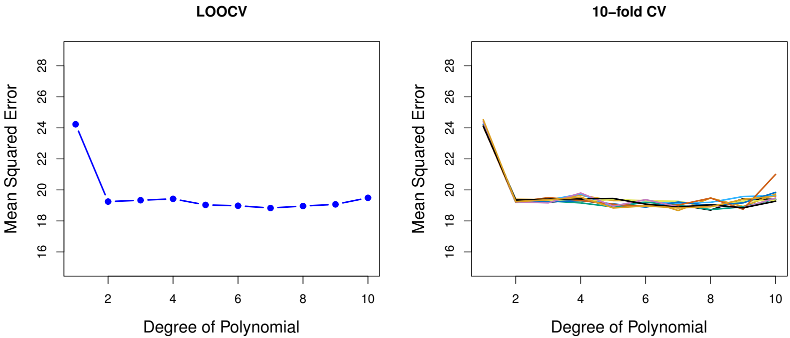
is the leverage statistic.

k-fold cross-validation

- Split the data into k subsets or *folds*.
- For every $i = 1, \dots, k$:
  - train the model on every fold except the ith fold,
  - compute the test error on the ith fold.
- Average the test errors.



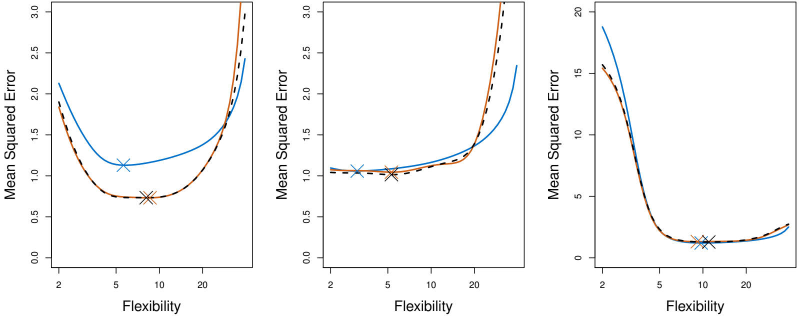
LOOCV vs. k-fold cross-validation



- k-fold CV depends on the chosen split.
- In k-fold CV, we train the model on less data than what is available. This introduces **bias** into the estimates of test error.
- In LOOCV, the training samples highly resemble each other. This increases the **variance** of the test error estimate.

Choosing an optimal model

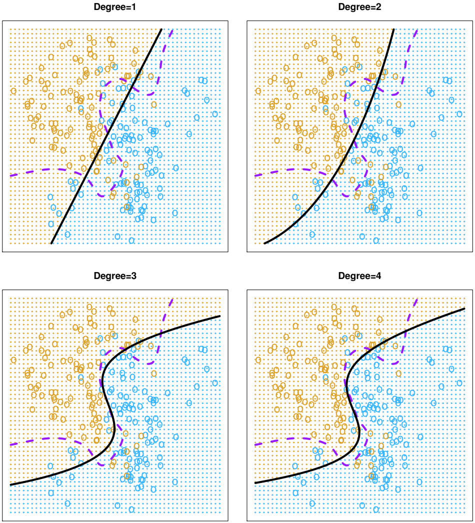
—— True test error, - - - LOOCV, —— 10-fold CV



Even when the CV error estimates differ significantly from the true test error, the model with minimum cross validation error often has relatively small test error.

Choosing an optimal model

In a classification problem, things look similar.



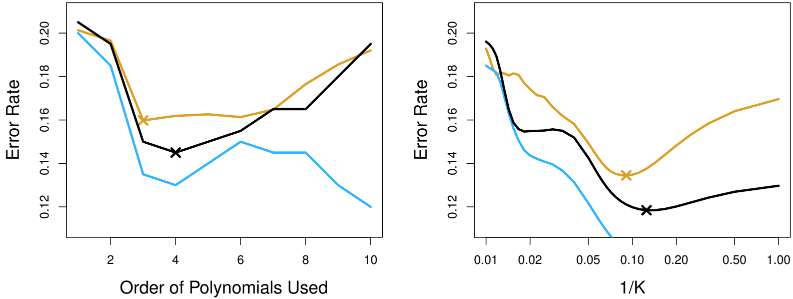
- - - Bayes boundary

—— Logistic regression augmented with polynomial predictors of increasing degree.

Choosing an optimal model

In a classification problem, things look similar.

—— Test error, —— 10-fold CV, —— Training error



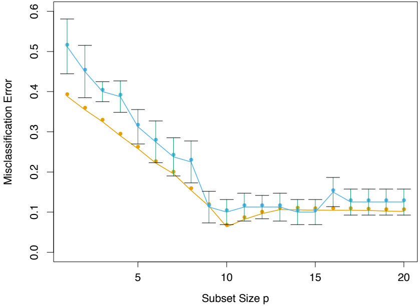
Question: Why do we plot $1/K$ and not K?

Side Note: In left plot, training error rate sometimes increases!

- Logistic regression does not directly minimize 0-1 error rate (it maximizes likelihood instead)

The one standard error rule

Forward stepwise selection



Blue: 10-fold cross validation
Yellow: True test error

- Curves minimized at $p = 10$.
- Models with $9 \leq p \leq 15$ have very similar CV error.
- The error bars represent ± 1 standard error

  $$SE = \frac{1}{k}\sqrt{\sum_{i=1}^{k}(err_i - CV)^2},$$

  around the CV estimate

  $$CV = \frac{1}{k}\sum_{i=1}^{k} err_i$$

  for $err_i =$ mean error in fold i
- **One SE rule:** Choose the simplest model with CV error no more than one SE above lowest CV error.

The wrong way to do cross validation

Reading: Section 7.10.2 of The Elements of Statistical Learning.

We want to classify 200 individuals according to whether they have cancer or not. We use logistic regression onto 1000 measurements of gene expression.

Proposed strategy:

- Using all the data, select the 20 most significant genes using z-tests.
- Estimate the test error of logistic regression with these 20 predictors via 10-fold cross validation.

The wrong way to do cross validation

To see how that works, let's use the following simulated data:

- Each gene expression measurement is standard normal and independent of all others.
- The response (cancer or not) is sampled from a fair coin flip independent of all "genes".

What should the misclassification rate be for any classification method using these predictors?

Roughly 50%.

The wrong way to do cross validation

We run this simulation, and obtain a CV error rate of 3%!

Why is this?

- Since we only have 200 individuals in total, among 1000 variables, at least some will be correlated with the response.
- We did variable selection using *all of the data*, so the variables we selected have some correlation with the response in every subset or fold in the cross validation.

The **right** way to do cross validation

- Divide the data into 10 folds.
- For $i = 1, \ldots, 10$:
  - Using every fold except i, perform the variable selection and fit the model with the selected variables.
  - Compute the error on fold i.
- Average the 10 test errors obtained.

In our simulation, this produces an error estimate of close to 50%.

Moral of the story: You can and often should validate your entire data processing & learning pipeline, even variable selection!