Ch 5.1.1-2: Leave One Out Cross-validation

Lecture 9 - CMSE 381

Michigan State University
::
Dept of Computational Mathematics, Science & Engineering

February 7, 2024

Covered in this lecture

- LOO CV
- Outliers
- Leverage statistic
- k-fold CV

Section 1

Validation set

What's the problem?

- How well is my ML method doing? *Model Assessment*
- Which method is best for our data?
- How many features should I use? Which ones? *Model selection*
- What is the uncertainty in the learned parameters?

First, Cross Validation. Then Bootstrap.

Training Error vs Testing Error

Training Error

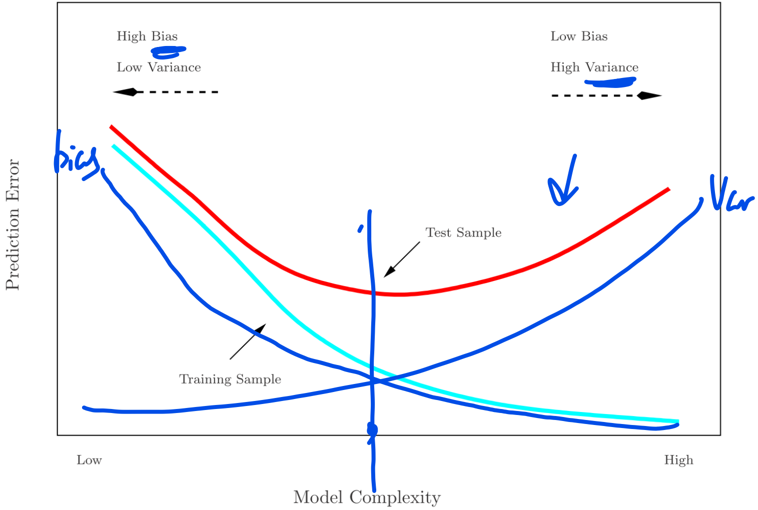
Easily calculated by applying stats learning method to the observations used in its training

- Massive under estimate
- overfitting!

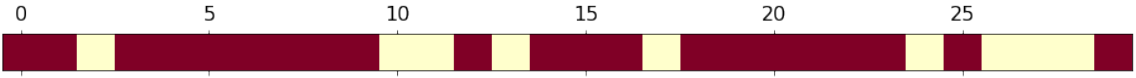
Testing Error

Avg error that results from using a statistical learning method to predict the response on a new observation, one that was not used in training the method.

Model tradeoffs



Validation set approach



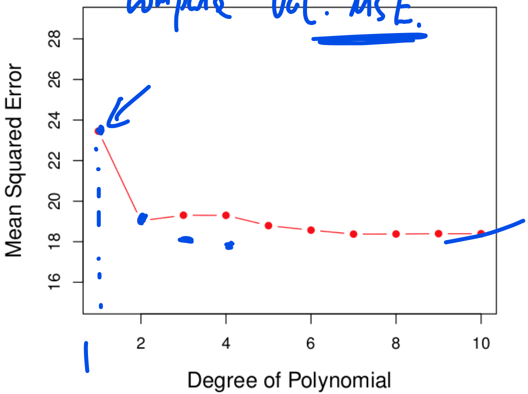
- Divide randomly into two parts:
  - Training set
  - Validation/Hold-out/Testing set
- Fit model on training set
- Use fitted model to predict response for observations in the test set
- Evaluate quality (e.g. MSE)

- This example has 30 data points
- Keep out 30% = 9 for testing in yellow

Example with the auto data

Train with 70% of training data

compute val. MSE.

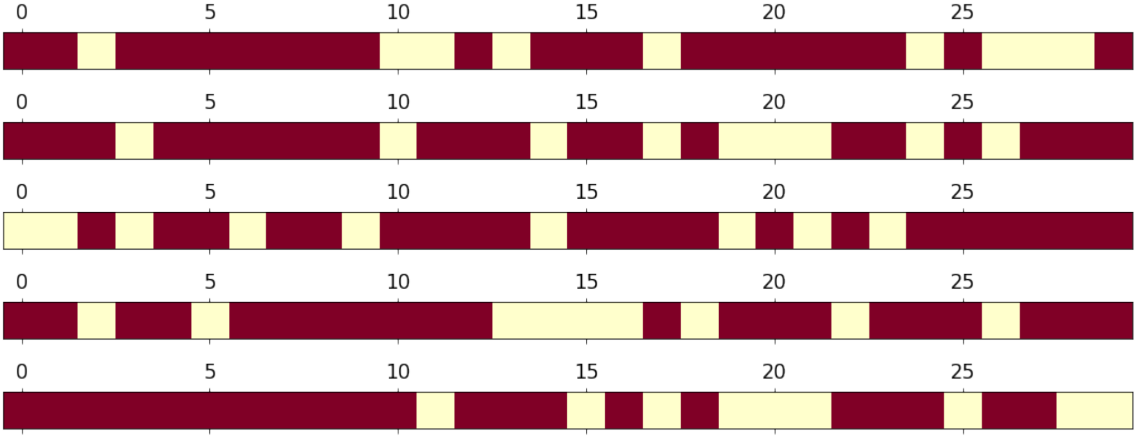


Predicting mpg using horsepower:

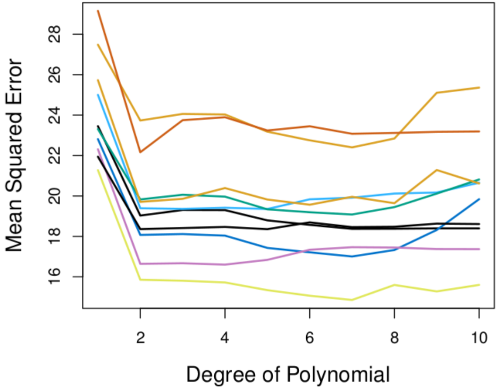
$$\texttt{mpg} = \beta_0 + \beta_1 \mathbf{hp} + \beta_2 \mathbf{hp}^2 + \cdots + \beta_p \mathbf{hp}^p$$

- Shows the TEST error
- Can conclude that linear isn't enough to model the data

Rinse and repeat



Again example with `auto` data

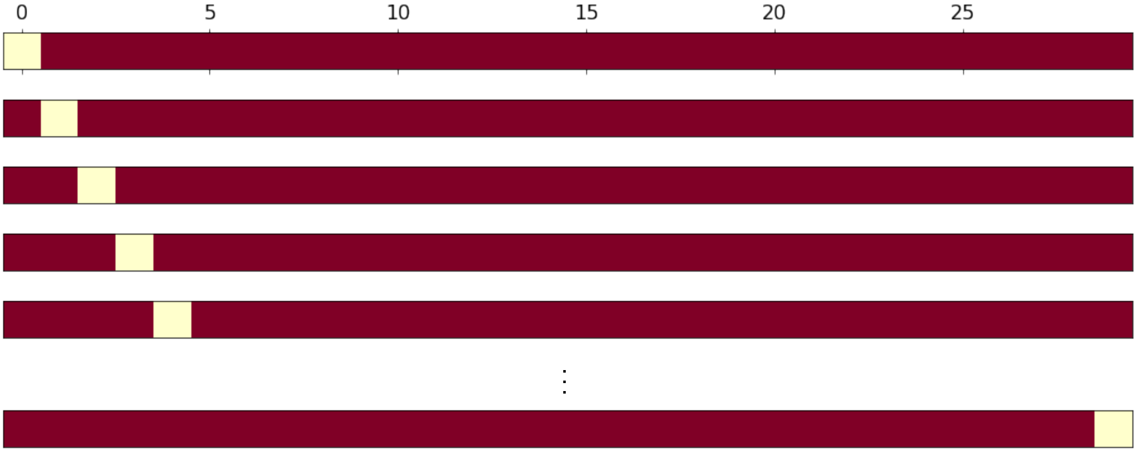


- Repeat the procedure with different splits
- Highly variable results, no concensus about the error
- Further complicated: Machine learning methods do worse when trained on fewer observations, so validation set error rate tends to overestimate test error rate when fitted on the whole data set.

Section 2

Leave-One-Out Cross-Validation (LOOCV)

The idea



The idea in mathy words

- Remove (x_1, y_1) for testing.
- Train the model on $n-1$ points: $\{(x_2, y_2), \cdots, (x_n, y_n)\}$
- Calculate $\mathrm{MSE}_1 = (y_1 - \hat{y}_1)^2$

- Remove (x_2, y_2) for testing.
- Train the model on $n-1$ points: $\{(x_1, y_1), (x_3, y_3), \cdots, (x_n, y_n)\}$
- Calculate $\mathrm{MSE}_2 = (y_2 - \hat{y}_2)^2$

- Rinse and repeat

Return the score:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{MSE}_i$$

LOOCV Pros and Cons

Advantages:

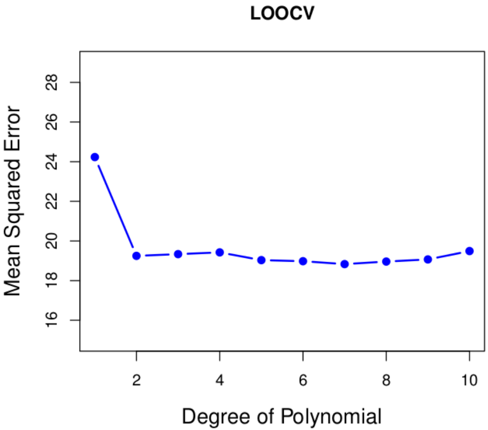
- Using almost all data every time
- Stable results: no randomness so always the same answer on a fixed data set
- Evaluation is performed with more training data

Disadvantages:

- Computationally expensive; fit the model n times
- Doesn't shake up the data enough: estimates from each fold are highly correlated so their averages have high variance.

Next goal: Speed up the LOOCV, but to do that, we need to get some terminology first

Again example with auto data



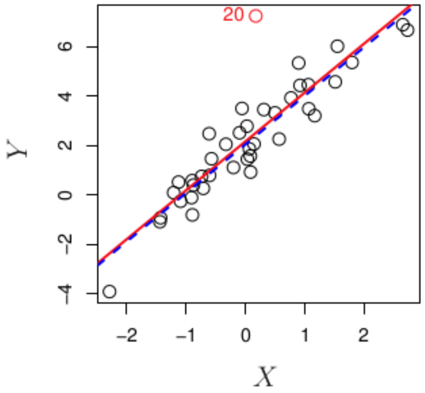
- There is no randomness. Same every time.

Section 3

The one time you can cheat (by not computing every model fit)

Outliers

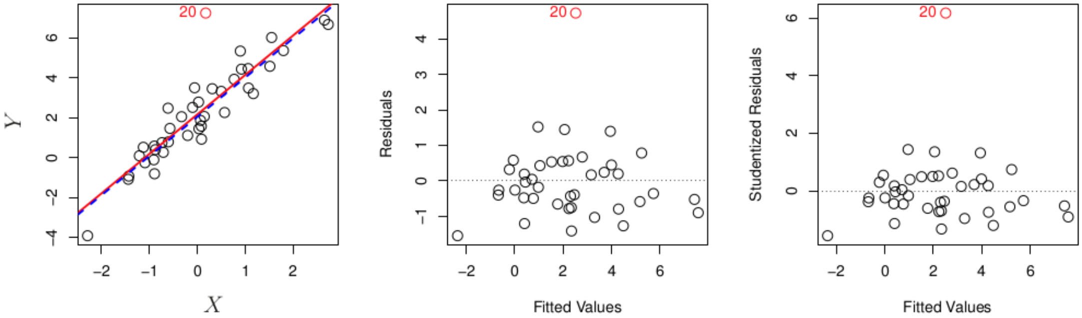
An *outlier* is a point for which y_i is far from the value predicted by the model. $|y_i - \hat{y}_i|$ *is large*
From Ch 3.3, stuff skipped earlier



Residuals

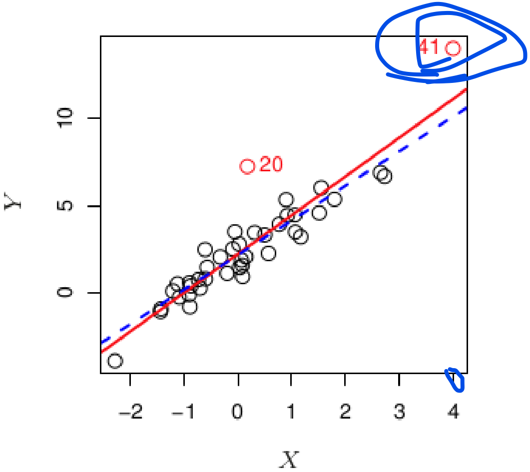
- Red line: Regression with all data
- Blue dashed: Regression after removing outlier 20
- Even though this is an outlier, didn't have much effect in this case.

Residuals



Residuals are $|y_i - \hat{y}_i|$. Problem with no clear way to decide which is far enough to constitute outlier. Studentized version: divide the residual by its std error. Common choice is ≥ 3 is outlier.

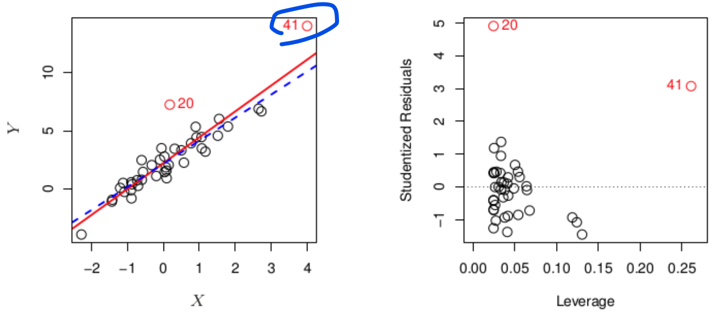
High Leverage



Observations with *high leverage* have an unusual value for x_i.

- Same data as previous but with extra data point 41
- Easy to find in $p = 1$ case, as at left, since just look for big x value.
- Red line: fit all data
- Blue dashed line: fit after removing high leverage point 41
- Point out big movement in linear regression line

Leverage statistic



Version for $p = 1$

$$h_i = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{j=1}^{n}(x_j - \overline{x})^2}$$

Generalizes to higher p, but not defined here. Compy can do it.

Leverage statistic properties

$$h_i = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{j=1}^{n}(x_j - \overline{x})^2}$$

- Increases with distance of x_i from $\overline{x}$.
- Always between $1/n$ and 1
- Average leverage for all observations is always $(p+1)/n$ (previous page this is $2/41 = 0.049$)
- If leverage stat is way bigger than $(p+1)/n$, then likely high leverage.
- 41 on previous page is both high leverage and outlier, so bad all around
- 20 on previous page had little effect on linear regression because low leverage even though it's an outlier

Speeding up LOOCV

Warning: This only works for least squares linear or polynomial regression.

fit the model on $\{(x_i, y_i)\}_{i=1}^{n}$

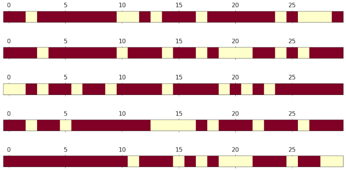
$\hat{y}_i = \hat{f}(x_i)$

$$\frac{1}{n}\sum_{i=1}^{n} \mathrm{MSE}_i = CV_{(n)} = \boxed{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2}$$

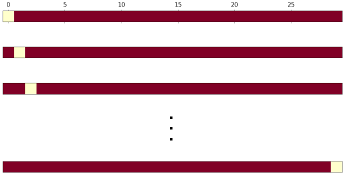
- Left side is original LOOCV computation
- Right side uses $\hat{y}_i$ fit on the whole data set
- Yay, only one model fitting!
- Looks similar to regular MSE

TL;DR

Validation set



LOO-CV



LOO-CV Score

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{MSE}_i$$

Cheap trick for regression

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

Section 4

k-fold CV

- $\eta_1, \dots, \eta_n$ are i.i.d $N(0,1)$

$$\mathrm{Var}\left(\frac{\eta_1 + \dots \eta_n}{n}\right) = \frac{\mathrm{Var}(\eta_i)}{n} = \frac{1}{n}$$

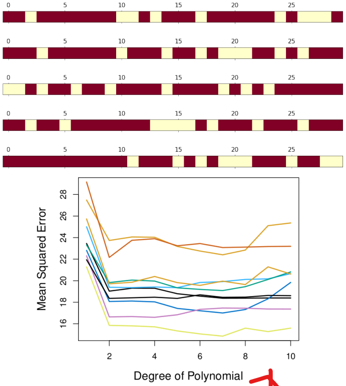
- $\eta_1 = \eta_2 = \dots \eta_n$, $\eta_i \sim N(0,1)$

$$\mathrm{Var}\left(\frac{\eta_1 + \dots \eta_n}{n}\right) = \mathrm{Var}(\eta_1) = 1$$

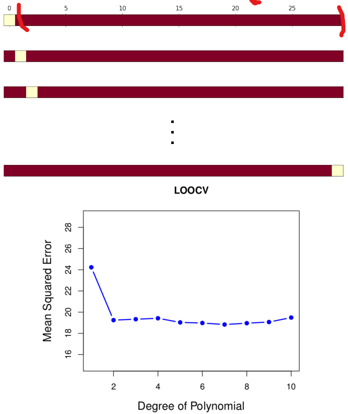
Approximations of Test Error

bias ↓ ↓ Var

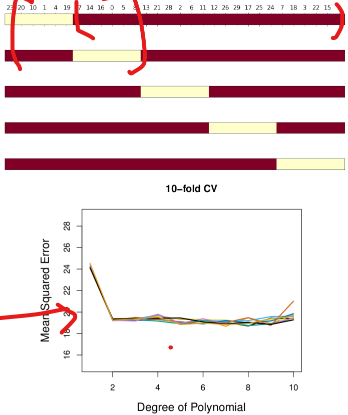
Validation Set



LOOCV

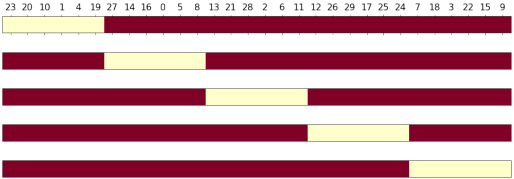


K-fold CV



Definition of k-fold CV

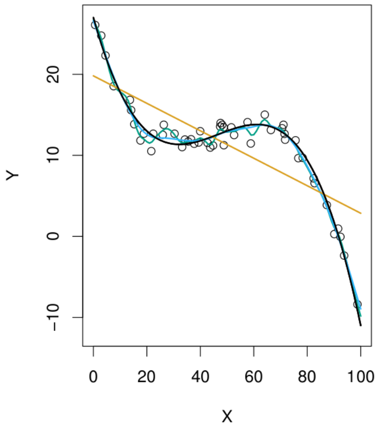
- Randomly split data into k-groups (folds)
- Approximately equal sized. For the sake of notation, say each set has ℓ points
- Remove ith fold U_i and reserve for testing.
- Train the model on remaining points
- Calculate $\mathrm{MSE}_i = \frac{1}{\ell} \sum_{(x_j, y_j) \in U_i} (y_j - \hat{y}_j)^2$
- Rinse and repeat

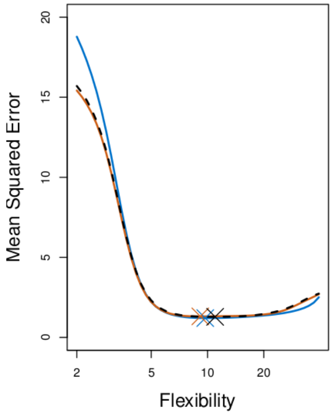


Return

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} \mathrm{MSE}_i$$

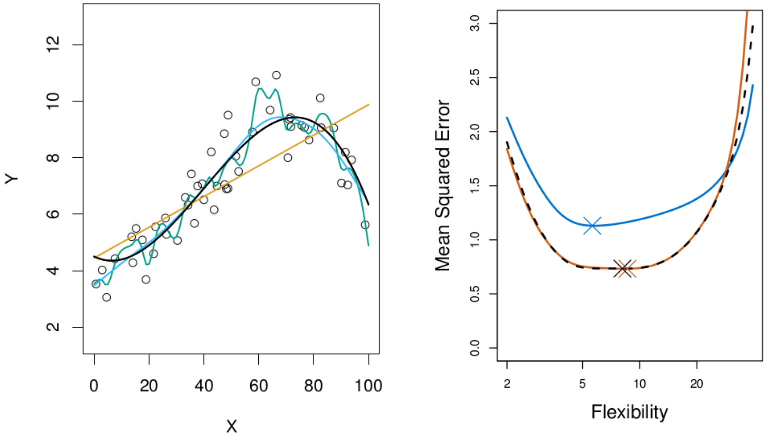
Comparison with simulated data: Ex 3





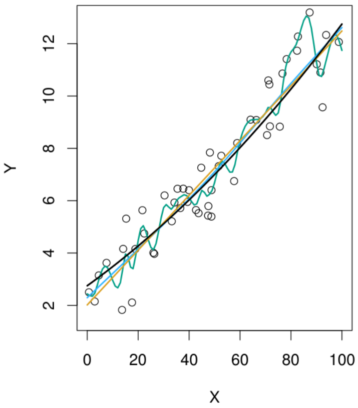
- Blue: true test MSE is shown in blue
- Black dashed: the LOOCV estimate
- Orange: 10-fold CV
- The crosses indicate the minimum of each of the MSE curves.
- Nearly the same

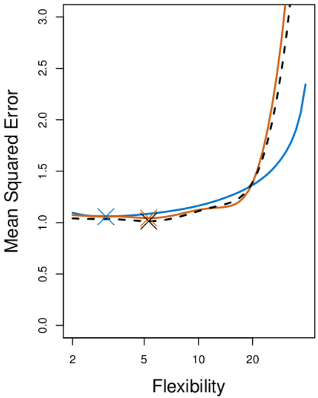
Comparison with simulated data: Ex 1



- Blue: true test MSE is shown in blue
- Black dashed: the LOOCV estimate
- Orange: 10-fold CV
- The crosses indicate the minimum of each of the MSE curves.
- **Underestmates the true test MSE**

Comparison with simulated data: Ex 2





Close for lower flexibility, then overestimates at higher ranges

Takeaways from the examples

- In all plots, LOOCV and 10-fold CV are similar
- Sometimes really only care about the minimum point in the test MSE because want to get the right degree of freedom for a model. All examples have approximately same x-coord for that

Bias-Variance Tradeoff: Bias

$$E(y_0 - \hat{f}(x_0))^2 = \mathrm{Var}(\hat{f}(x_0)) + \left[\mathrm{Bias}(\hat{f}(x_0))\right]^2 + \mathrm{Var}(\varepsilon)$$

- 1.... Validation set overestimates test error b/c used small subset of data
- 3.... k-fold gives medium level of bias b/c training set has approximately $(k-1)n/k$ observations
- 2....LOOCV gives approximately unbiased estimate since uses almost all data every time

- soooooooooo LOOCV better than k-fold just for bias

Bias-Variance Tradeoff: Variance

$$E(y_0 - \hat{f}(x_0))^2 = \mathrm{Var}(\hat{f}(x_0)) + \left[\mathrm{Bias}(\hat{f}(x_0))\right]^2 + \mathrm{Var}(\varepsilon)$$

- LOOCV higher variance:
  - avg the outputs of n fitted models, but all on almost identical observations
  - high correlation with each other
- k-fold
  - k fitte models somewhat less correlated with each other
  - mean of many highly correlated quantities has higher variance than those not corrleated

- Empirically, $k = 5$ or 10 has been shown to be a happy medium