Resampling Methods & bias-variance tradeoff

CS 530
Chapman

Spring 2021

1

Take 🏠 message for rest of course: Resampling

- What is the bias-variance tradeoff?
- Subset selection methods
- Data validation techniques
  - Why do we need data validation?
  - Putting it in the context of the bias-variance tradeoff
- Bootstrapping techniques

2

TABLE OF CONTENTS

3

EXIT SURVEY

4

Multicollinearity

- What is multicollinearity?
  - When 2 or more variables in a dataset are highly correlated—typically resulting in R^2>0.75 or so
- Why is it a problem?
  - In multiple linear regression,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \epsilon$$

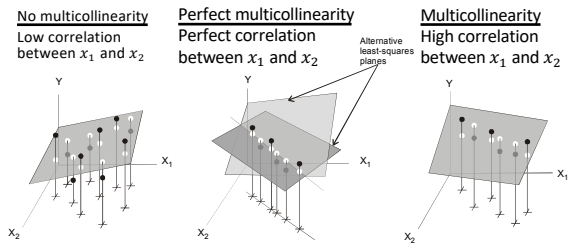
Each β_i represents the average effect on y caused by a unit increase in x_i, **assuming all other x terms—$x_1, \ldots, x_{i-1}, x_{i+1}, x_p$—stay constant**. multicollinearity violates this assumption because, if x_i and x_j are multicollinear, β_i would not change while β_j remains constant.

  - Practically, it means that such x_i and x_j would often both have large p values, making model pruning difficult.
  - Also, β_i & β_j values would be highly dependent on the specific training set

5

Multicollinearity

- Also, β_i & β_j values would be highly dependent on the specific training set we draw



6

BIAS-VARIANCE TRADEOFF

7

Bias vs. Variance

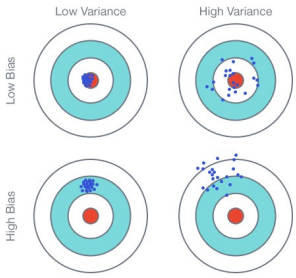


Image Source

8

Bias vs. Variance

Bias describes how far our estimates are from the true values.

Variance describes the spread of our estimates.

Ideally, we would like a model with low bias and low variance in the coefficient estimates $\{\hat{\beta}_j\}$.

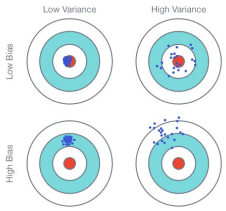


Image Source

9

Bias vs. Variance

Importantly, for linear regression:

A model with high bias will have coefficient estimates $\{\hat{\beta}_j\}$ that are, on average, very different from the true coefficients $\{\beta_j\}$.

A model with high variance will have coefficient estimates that are very sensitive to small changes in the training data.

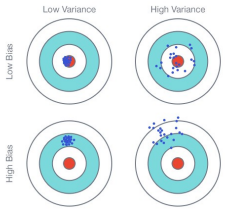


Image Source

10

Bias vs. Variance

Assume true model $y = f(x) + \varepsilon, (\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon))$.

We estimate f with $\hat{f}$ (say, using linear regression). Then the square prediction error at point x is

$$Err(x) = E\left[\left(y - \hat{f}(x)\right)^2\right].$$

This can be decomposed into

$$Err(x) = \underbrace{\left(E\left[\hat{f}(x)\right] - f(x)\right)^2}_{\text{Bias}^2} + \underbrace{E\left[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2\right]}_{\text{Variance}} + \underbrace{\sigma_\varepsilon{}^2}_{\text{Irreducible error}}.$$

11

Bias vs. Variance

$$E[Err(x)] = \underbrace{\left(E\left[\hat{f}(x)\right] - f(x)\right)^2}_{\text{Bias}^2} + \underbrace{E\left[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2\right]}_{\text{Variance}} + \underbrace{\sigma_\varepsilon{}^2}_{\text{Irreducible error}}.$$

- $\hat{f}(x)$ is the statistical model derived from the training set
- $f(x)$ is the true model from which the data was created (typically unknow for real data)
- σ_ε is the irreducible error for which the model cannot account ($y = f(x) + \varepsilon$)

12

Bias vs. Variance

$$MSE = \underbrace{\left(E[\hat{f}(x)] - f(x)\right)^2}_{\text{Bias}} + \underbrace{E\left[(\hat{f}(x) - E[\hat{f}(x)])^2\right]}_{\text{Variance}} + \underbrace{\sigma_\varepsilon{}^2}_{\text{Irreducible error}}.$$

Bias: How well the trained model, $\hat{f}$, can approximate the real model, f

Variance: How much variability does the trained model, $\hat{f}$, have? How dependent is it on the specific training set?

13

Bias vs. Variance

Assume true model $y = f(x) + \varepsilon, (\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon))$.

We estimate f with $\hat{f}$ (say, using linear regression). Then the square prediction error at point x is

$$Err(x) = E\left[\left(y - \hat{f}(x)\right)^2\right].$$

This can be decomposed into

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[(\hat{f}(x) - E[\hat{f}(x)])^2\right] + \sigma_\varepsilon{}^2.$$

or

$$\text{MSE} = \left[\text{Bias}\left(\hat{f}(x_0)\right)\right]^2 + \text{Var}\left(\hat{f}(x_0)\right) + \text{Var}(\varepsilon)$$

14

Bias vs. Variance

Linear regression gives us coefficient estimates $\{\hat{\beta}_j\}$ that are unbiased (Gauss-Markov Theorem). Having unbiased coefficient estimates means that if we apply linear regression to several samples of data, the average of those coefficient estimates should approach the true coefficients $\{\beta_j\}$.

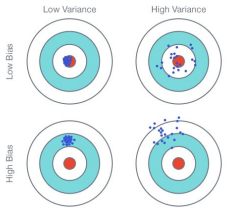


Image Source

15

Bias vs. Variance

Under the strong assumptions mentioned before, linear regression guarantees a model with zero bias in the coefficient estimates $\{\hat{\beta}_j\}$. But it **does not guarantee small variance** of the coefficient estimates.
Multicollinearity (for example) may lead to high variance in coefficient estimates.

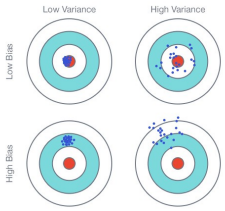


Image Source

16

Bias vs. Variance

How do we deal with the potential variance in multiple linear regression?

1. Subset selection: simplify our model using fewer variables,)
2. Regularization: shrink the coefficient estimates (ridge, LASSO, or elastic net).

Generally:
But why not simply and directly minimize the variance? Because of the bias-variance tradeoff.

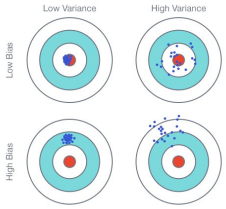
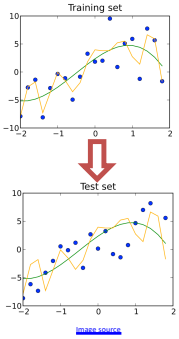


Image Source

17

Learning differs from pure optimization

In pure, classical optimization, we want to optimize (minimize) a function over some known data. This is called the training data in machine-learning. Hence, we want to minimize the training error.



In machine learning, we want to minimize a function over an unseen dataset (the test set), given only a dataset **with "similar" statistics** (the training set). So, we want to minimize the test error (a.k.a. generalization error, validation error). This is a different, more difficult, problem.

Image source

18

Learning differs from pure optimization

So, when optimizing, we want to capture the main statistical features of the training set—the "gist" of the training set—without being affected by fine details that distinguish the training set from the test set.

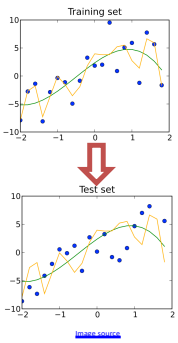


Image source

19

Training vs. Test Errors

The dots on the top panel were created from the black curve, f, with added noise. We estimate f with a linear-regression line ($\hat{f}_1$, in orange), a low-order smoothing spline ($\hat{f}_2$, in blue), and a higher-order smoothing spline ($\hat{f}_3$, in green).

In real life, we do not have f, only the black dots. But we are trying to approximate f.

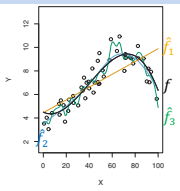


Image Source

20

Training vs. Test Errors

The dots on the top panel were created from the black curve, f, with added noise. We estimate f with a linear-regression line ($\hat{f}_1$, in orange), a low-order smoothing spline ($\hat{f}_2$, in blue), and a higher-order smoothing spline ($\hat{f}_3$, in green).

In the bottom panel, we see the MSE versus flexibility (loosely degrees of freedom of estimation). In grey we see the error over the training set, which keeps decreasing with flexibility. But we care only about the error on the test set, in red, which has a u-shape. It first decreases with flexibility, then increases back.

The dashed black line at MSE=1 in the bottom panel marks the minimum error possible for any estimator of f.

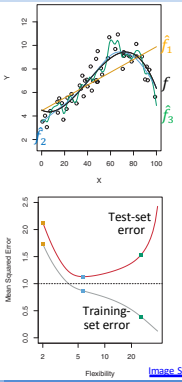


Image Source

21

Training vs. Test Errors

The colored squares on the red & grey curves in the bottom panel correspond to the MSE for the curves with corresponding colors on the top.

$\hat{f}_1$, underfits the data, its not a strong enough model to capture the variance in the data. It thus has a large test error (and training error)—*high bias*.

$\hat{f}_3$, overfits the training data, capturing some of the noise added on f together with f. So, while its training error is smaller than $\hat{f}_2$'s, its test error is larger—*high variance*.

This u-shape can be show to be a tradeoff between 2 competing factors: *bias* & *variance*.

$\hat{f}_2$, is a good fit for f. Its test error is close to minimum—neither underfits nor overfits.

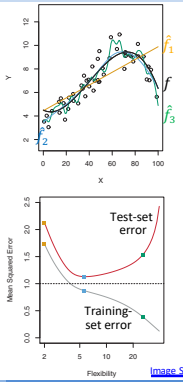
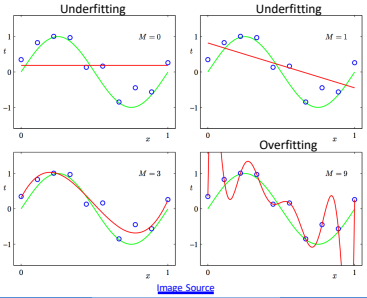


Image Source

22

Underfitting and Overfitting

- Data points
- Real model generating data points, *never known to us*
- Model we fit to data



Training error = 0, but model explains too much random noise in data

Image Source

23

Mean Squared Error

To explain the bias-variance tradeoff, we first need to define the Mean Squared Error (MSE) of a model:

$$\text{MSE} = \frac{1}{n} RSS = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The MSE is to the RSS what the mean is to the sum. It is a measure of the mean difference between the model & data points.

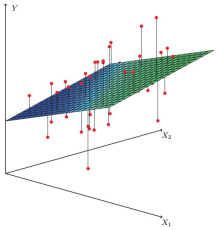


Image Source

24

Bias-Variance Tradeoff

What is the bias-variance tradeoff? It can be shown that, for the test set:

$$\text{MSE} = \text{Var}\left(\hat{f}(x_0)\right) + \left[\text{Bias}\left(\hat{f}(x_0)\right)\right]^2 + \text{Var}(\epsilon) \geq 0$$

For the other terms in the equation:

$\text{Var}\left(\hat{f}(x_0)\right)$ ← The variance of our coefficient estimates.

$\left[\text{Bias}\left(\hat{f}(x_0)\right)\right]^2$ ← The bias (squared) of our coefficient estimates.

$\text{Var}(\epsilon)$ ← The variance of the error term.

The variance of the error term, $\text{Var}(\epsilon)$, is based only on the data, not on our model. (It is the dotted line in our example above—the noise added to the black curve). But we can modify our model to either increase variance & decrease bias or to increase bias & decrease variance.

25

Bias vs. Variance

Variance: how much our estimate, $\hat{f}$, would change if we used a different training set. Ideally $\hat{f}$ would not change much between draws from the population, hence between the training set and test set. Variance is a measure of the "spread" of $\hat{f}$. The green curve highly depends on the draw—high variance— while the blue and even more the orange ones not that much—low variance.

Bias: error in approximating complex data, f, with a simpler model, $\hat{f}$. The orange square has the highest bias. A line is too simple to explain non-linear f. And the green square has the lowest bias.

Bias-variance tradeoff: typically the more complex the model, the lower the bias, but the higher the variance.

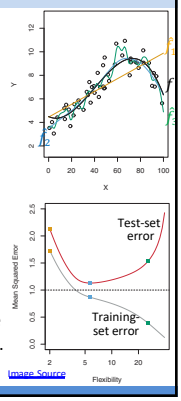


Image Source

26

Bias vs. Variance

Increasing flexibility decreases bias. But, at some point, bias plateaus. Variance, in contrast, rises at an ever-increasing rate with flexibility.

This bias-variance tradeoff gives us the MSE's u-shape.

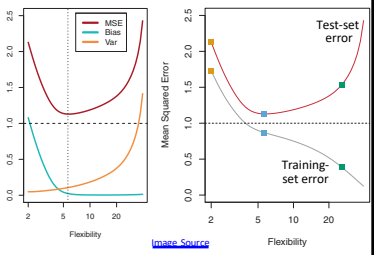


Image Source

$$\text{MSE} = \left[\text{Bias}\left(\hat{f}(x_0)\right)\right]^2 + \text{Var}\left(\hat{f}(x_0)\right) + \text{Var}(\varepsilon)$$

27

Bias vs. Variance

The overall picture is thus:

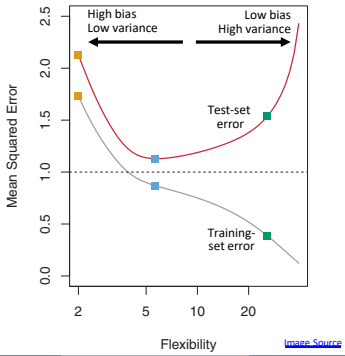
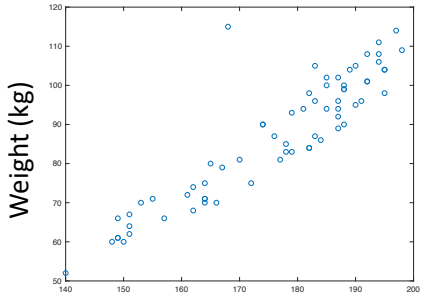


Image Source

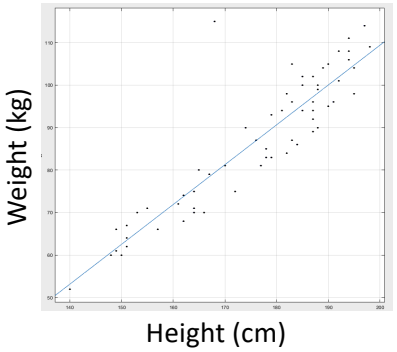
28

More on the irreducible error



29

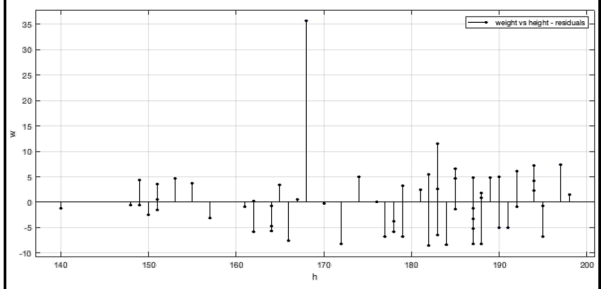
More on the irreducible error



30

More on the irreducible error

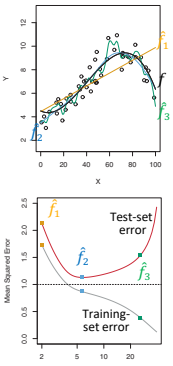
Residuals plot: variance in weight not due to height



31

Irreducible error & overfitting

- Underfitting: $\hat{f}_1$ too weak to capture structure of f
  - $\left(E[\hat{f}(x)] - f(x)\right)^2$ (bias) is large
- Overfitting: $\hat{f}_3$ captures irreducible noise on top of f
  - $E\left[\left(\hat{f}(x)^2 - E[\hat{f}(x)]\right)^2\right]$ (variance) is large



32

Bias-variance tradeoff mathematically

Given a training set

$$\{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\} \in \mathbb{R}^p \times \mathbb{R},$$

we seek

$$h_n : \mathbb{R}^p \to \mathbb{R}$$

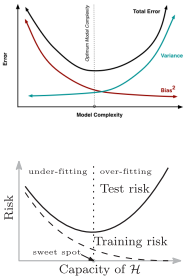
that predicts y for some $\vec{x} \notin \{\vec{x}_1, \ldots, \vec{x}_n\}$. h_n is chosen from some function class $\mathcal{H}$.

33

Bias-variance tradeoff mathematically

The "bias-variance tradeoff" suggests that we must balance underfitting and overfitting.

We must control the capacity of function class $\mathcal{H}$. If it is too small (high bias), $h_n \in \mathcal{H}$ will likely underfit. If it is too large (high variance), $h_n \in \mathcal{H}$ will likely overfit. So, we must find find the "sweet spot" to best generalize to the test set.



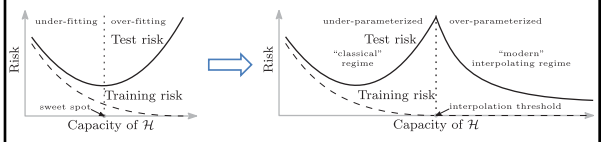
34

Bias-variance tradeoff discussion

- The deconstruction of the MSE works

$$\text{MSE} = \text{Var}\left(\hat{f}(x_0)\right) + \left[\text{Bias}\left(\hat{f}(x_0)\right)\right]^2 + \text{Var}(\epsilon) \geq 0$$

- However, it does not necessarily entail a tradeoff
- There are claims that for deep neural-networks



35

SUBSET SELECTION

36

Subset Selection

- For linear models we are guaranteed unbiased estimators (Gauss-Markov Theorem). But how do we reduce the variance?
- The simplest way to reduce variance in a multiple linear regression models is to *use fewer variables*.
- Subset selection: find better (or best) multiple linear regression model for our data by considering subsets of the total set of available variables.
- How do we pick which variables to keep in our model? We seek highest adjusted R^2, lowest AIC or BIC, etc.

37

Subset Selection

Best Subset Selection example:

Suppose we are trying to predict compatibility of subjects to a government job based on height, GPA, and number of friends on Facebook. Our 3 explanatory variables (Height, GPA, Friends) to choose from, gives $2^3 = 8$ possible models to consider.

Subsets
Empty—No Variables
Height
GPA
Friends
Height, GPA
Height, Friends
GPA, Friends
Height, GPA, Friends

38

Subset Selection

Best Subset Selection example:

We then choose the best model out of all possible subsets by taking the model with, say, the lowest AIC value. In this case, we would choose the model containing {Height, Friends}.

(Needless to say, this example is not supported by any empirical data.)

Subsets	AIC
Empty—No Variables	248
Height	150
GPA	170
Friends	164
Height, GPA	57
Height, Friends	42
GPA, Friends	49
Height, GPA, Friends	51

39

Subset Selection

The downside of Best Subset Selection:

It is only feasible when the number of variables is small. What if we have 50 variables with

$$2^{50} = 1{,}125{,}899{,}906{,}842{,}624$$

possible subsets to consider? Best subset-selection method is often very computationally expensive to infeasible.

40

Subset Selection

Alternatives to Best Subset Selection:

How can we select subsets for many variables?

- Forward Stepwise Selection
  (Add variables one-at-a-time.)
- Backward Stepwise Selection
  (Remove variables one-at-a-time.)

41

Subset Selection

Forward Stepwise Selection:

- Start with a model that contains no variables.
- Consider all models with one variable. Pick the best model from this set using R^2, AIC, or BIC. Note it and its score in a list.
- Once we've picked a model with one variable, consider all models with an additional variable. Pick the best model from this set using R^2, AIC, or BIC. Add it and its score to our list.
- Continue adding variables and appending to the list until we reach the saturated model.
- Pick the model with the best R^2, AIC, or BIC from our list.

42

Subset Selection

Backward Stepwise Selection:

- Start with a model that contains all variables.
- Consider all models with one variable removed. Pick the best model using R^2, AIC, or BIC. Note it and its score in a list.
- Once we've picked a model with one variable removed, consider all models with an additional variable removed. Pick the best model from this set using R^2, AIC, or BIC. Add it and its score to our list.
- Repeat with removing variables and appending to the list until we reach the empty model.
- Pick the model with the best R^2, AIC, or BIC from our list.

43

Subset Selection

Comparison of subset selection methods:

- Best subset selection requires $O(2^p)$ models.
- Forward and backward stepwise selection each require $O(p^2)$ models, making them computationally much cheaper.
- Best subset selection will find the multiple linear regression model with the best fit—on the training set.
- Forward & backward stepwise selection do not guarantee the model with absolute best fit; they do not consider all possible models. They will land on a local minimum in the space of models.

44

VALIDATION METHODS

45

Motivation for Validation

We discussed a few methods for creating a model to fit a data set:

- Ordinary Least Squares Regression:
  - Simple Linear Regression
  - Multiple Linear Regression

But how do we know, say, which combination of parameters will work best on the test set? How do we know we are not overfitting the training set?

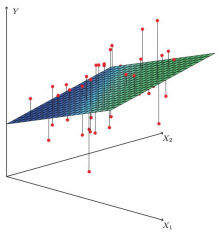


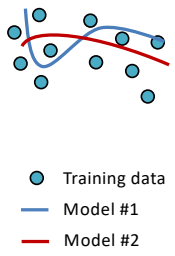
Image Source

46

Motivation for Validation

In real life, we are not given the underlying generating function, f. But we want to find a good approximation, $\hat{f}$, that would work well on the test set.

1. How do we find the best model on the test set when we only have the training set?
2. How can we approximate the accuracy (goodness of fit) on the test set when we only have the training set?



47

What Are Resampling Methods?

We want to build model that will work well on unseen dataset—the test set. But we are limited by a given, finite dataset—the training set. We can better approximate the unknown test set by taking "different looks" at our given training set by subsampling it.

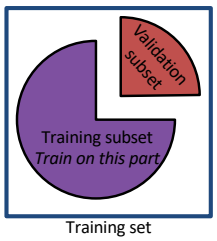
Image Source

48

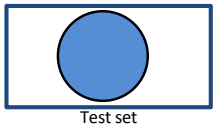
Validation

How do we construct a model that will work well on a dataset we have never seen? We use validation techniques.

We assume: the unseen test set has the same statistics as our training set (at least locally). Then we train the model using only a subset of the training set. And we test it on a disjoint subset of that training set, not used for model training. Model performance on this disjoint validation subset is taken as an indication of its future performance on the actual test set.



Training set


Test set

49

Validation

Validation Methods:

1. Validation-Set Approach: Split the data set into 2 disjoint subsets: a training set and a test set.
2. K-fold Cross-Validation: Split the data into *K* equal subsets. Then create *K* models, each trained on all but the *K*'th subset (the training set) and test it on the *K*'th subset (the test set).
3. LOOCV (Leave-One-Out Cross-Validation): Create *n* models—one for each data point. For each model, train on all but the *n*'th sample and test on the *n*'th sample. (Same as *n*-fold cross-validation.)

50

I. Validation-Set Approach

The validation-set approach is simplest and most straightforward way to use cross-validation:

1. Randomly assign data samples into training set and a test set (also named validation or hold-out set). 50-50, 60-40, or 70-30% splits are common.
2. Train the model on the training set. Test it on the test set.

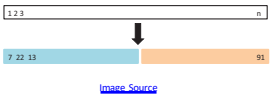


Image Source

51

I. Validation-Set Approach

Example:

- We look at errors on a test versus the time spent studying for that test (in minutes)
- There appears to be a downward, plateauing trend
- Note: A 50-50, 60-40, 70-30 validation-set split means we have only 50, 60, or 70% of the data on which to train our model and only on 50, 40, or 30% of the data to test.

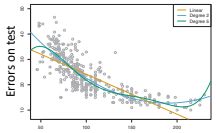


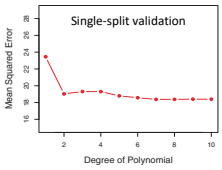
Image Source

52

I. Validation-Set Approach

Drawbacks:

- Test-subset depends on the luck of the data split, resulting in highly variable estimates of test-set error
- Extreme statistics are more likely from smaller training sets. Also, the less data we use to train the model, the less likely it will be well trained:
  - The test error that the validation set estimates tends to overestimate the true test error—i.e., the validation-set test error is too high
  - The method suffers from high bias because only a part of the training set is used to train the model



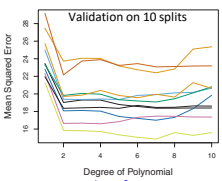


Image Source

53

II. *K*-fold Cross-Validation

We can expand the validation-set idea and mitigate its drawbacks using *K*-fold cross-validation:

1. Randomly split dataset into *K* equal-sized subsets, or folds
2. Treat each fold as validation set (train on all but *K*'th fold and test on *K*'th fold only)

- The overall error is then the mean error over all *K* models.
- Most common are 5- or 10-fold cross-validation

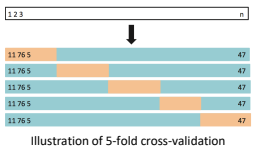


Illustration of 5-fold cross-validation

Image Source

54

II. *K*-fold Cross-Validation

Advantages of *K*-fold cross-validation:

- 80 to 90% of data used for training (in 5- to 10-fold cross validation resp.)
  - Less bias than for validation-set
- Less overestimation of the true test error rate
  - Because a larger fraction of the training set trains the model
- Averaging over all folds yields more consistent results than the validation-set approach

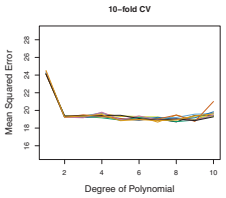
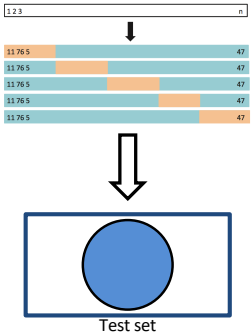


Image Source

55

II. *K*-fold Cross-Validation

Image Source



You run K models (one per fold). How do you determine the model to use on the test set?

You use K-fold to find the best type of model (e.g., you might be deciding between several different machine-learning algorithms). Then you find the parameters to use on the test set by running the model once on the entire training set.

56

III. Leave-One-Out Cross-Validation (LOOCV)

Leave-One-Out Cross-Validation (LOOCV) is a special case of *K*-fold cross-validation, where the number of folds (k) equals the number of points in the data set (n). Hence, for each fold, we create a model that trains on all but 1 sample (the "left-out" sample) and is tested on the left-out sample. Overall n models are created. Error rate is

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

(or proportion of erroneously classified samples of n samples)

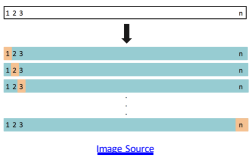


Image Source

59

III. Leave-One-Out Cross-Validation (LOOCV)

Advantages of LOOCV:

- All but 1 sample (almost entire training set) used to train model
  - Suffers from least bias among the 3 validation methods
  - Gives consistent results (randomness not part of technique)
- Least overestimates the test-error rate among the 3 validation methods
- Yields most consistent results

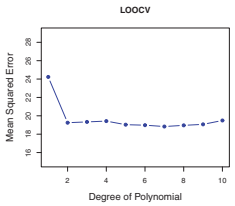


Image Source

60

III. Leave-One-Out Cross-Validation (LOOCV)

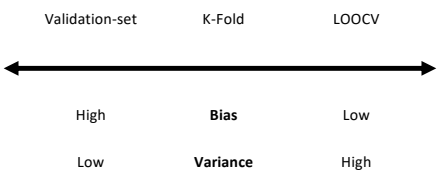
Drawbacks of LOOCV:

- Most computationally expensive: includes construction of n models
- LOOCV suffers from high variance
  - Each pair of folds are highly correlated ($n - 2$ samples in common)
  - So, each pair of models trained on almost identical subsets of samples (assuming $n \gg 2$)
  - Hence n model outputs highly positively correlated
  - The more correlated the data the more variable its mean

61

Validation and the Bias-Variance Tradeoff

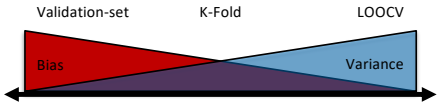
The decision which validation method to use can be aided by the bias-variance tradeoff



63

Validation and the Bias-Variance Tradeoff

The decision which validation method to use can be aided by the bias-variance tradeoff



Empirically, 5- or 10-fold cross validation typically works best, suffering from neither excessive bias nor excessive variance. It is thus most often used.

64

Validation: Limitations

Like everything else in machine-learning, validation should not be used blindly. Its limitations should be borne in mind.

- Validation sets should be randomly chosen and representative of the total population to avoid biases
- Validation assumed stationary statistics—i.e., that the training and test sets are drawn from the same distribution. If not:
  - Is there a relation between the training and test sets that we can model? Can we transform the training and test sets to get more stationary statistics?
  - Are the data locally stationary? Can we correct to get more stationary data?
- If model is missing some relevant variables, our model might work well on training set (e.g., due to luck or overfitting) but not on test set

69

BOOTSTRAPPING

70

Bootstrapping

A method similar to validation and very useful in machine-learning is bootstrapping. It allows us to

1. find confidence intervals for coefficient estimates,
2. test whether our results are better than expected by chance

and many more.

The name comes from "pulling oneself up by one's bootstraps", because we use our sample data to create more samples.

Baron Munchausen pulls himself and his horse out of a swamp by his pigtails

Image Source

71

Bootstrapping coefficient estimates

Example: There exists a recording of spikes from a patient with implanted depth electrodes that took years to obtain. There is an hour (3600 s) of data, with the number of spikes per second for each second. However, we can store only 100 s (2.8%) of the data at a time:
20 2 18 4 4 10 3 3 30 …

Using only those 100 samples from the population of 3600, can we compute the interval where the overall mean of the population is likely to be?

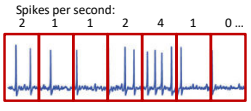


Image Source

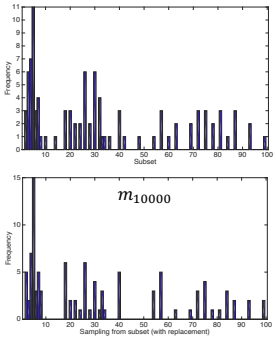
72

Bootstrapping coefficient estimates

We sample 100 samples from our subset—with replacement—and calculate the mean over that sample. We repeat this, say, 10,000 times, and sort those means in ascending order:

$$m_1, m_2, \ldots, m_{10000}.$$

The mean of those means is 32.42, and that is our best estimate for the population mean. But how good of an estimate is it?



73

Bootstrapping chance level

- You want to use a health dataset to predict death based on patient demographics and health history
- Your dataset includes 101,374 patients, 1,234 of them died
- Your prediction model's accurately is 99%. Is your model good?

75

Bootstrapping chance level

- You want to use a health dataset to predict death based on patient demographics and health history
- Your dataset includes 101,374 patients, 1,234 of them died
- Your prediction model's accurately is 99%. Is your model good?
- It is not, because just predicting that the person will not die would result in $\frac{(101374-1234)}{101374} = 98.8\%$

76

Bootstrapping chance level

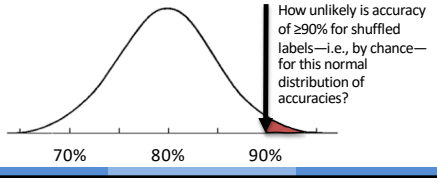
- To know that

1. Randomly permute the labels on the training set
2. Train model on randomly permuted training set and test on test set
3. Repeat above (say) 10,000 times
4. Is your accuracy above (say) 97.5 percentile of randomly permuted models?
   - Yes: say your model works
   - No: back to the drawing board

77

Bootstrap shuffling

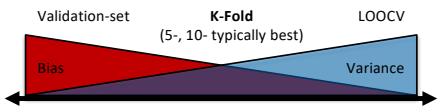
- We have a dataset where one class is 80% and the other 20%. Our algorithm achieves 90% accuracy. Is it much more than we would expect by chance?
- We permute (shuffle) the labels, train & validate, and compute our algorithm's accuracy. We do this, say, 1000 times, and get a distribution of accuracies.



78

Conclusion: Resampling

- Validation Methods
  - Validation-Set Approach
  - *K*-Fold Cross-Validation
  - Leave-One-Out Cross-Validation (LOOCV)
- Limitations of Validation
  - Assumes good model with stationary statistics
- The Bootstrap method allows us to
  - Estimate population statistics from a sample
  - Estimate chance level on model prediction



80

Group exercise

In breakout rooms, we will now work on a google doc

81

Take 🏠 message for rest of course: Resampling

- Validation methods (validation-set, cross-validation, leave-1-out cross-validation) are a systematic way to estimate generalization error from training set
  - Most people use 5- or 10-fold cross-validation
- Bootstrapping methods allow us to:
  - Estimate population statistics from (small) sample without assumptions on the population's distribution
  - Estimate chance level on model prediction

82