PLSC 40601

Week 2: Sample-splitting, bagging, honesty.

Molly Offer-Westort

Department of Political Science,
University of Chicago

Spring 2023

Housekeeping

- ?

Sample splitting

What is our goal in fitting a model?

- Given some data $(Y_1, X_1), \ldots, (Y_N, X_N)$, we fit a model, $\hat{f}(X)$.

- Suppose our goal is prediction for the next observation.

- Given X_{N+1}, we want to minimize

$$L\left(Y_{N+1}, \hat{f}(X_{N+1})\right) = \left(Y_{N+1} - \hat{f}(X_{N+1})\right)^2$$

- We may be interested not just in how our method performs on one specific observation, but how it performs in expectation

$$\mathrm{Err} = \mathrm{E}\left[L(Y, \hat{f}(X))\right]$$

- What should the expectation be taken over? Can/should we hold the data we used for fitting the model fixed?

What is our goal in fitting a model?

- Difference between conditional error and expected test error
  - Conditional test error:

$$\mathrm{Err}_{\mathcal{T}} = \mathrm{E}\left[L(Y, \hat{f}(X)) \middle| \mathcal{T}\right]$$

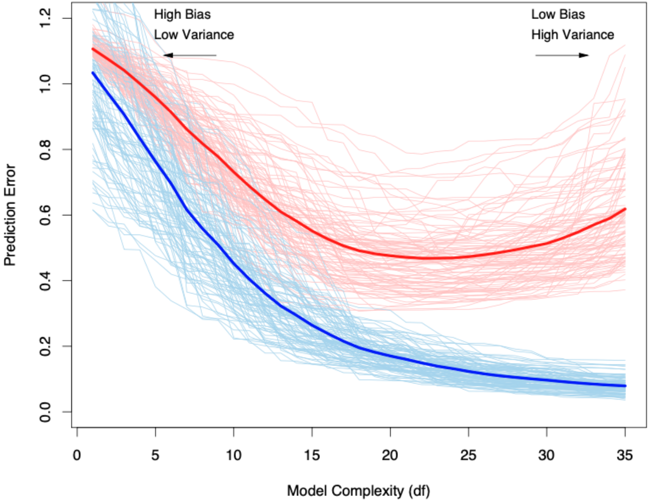
    Training set $\mathcal{T}$ is fixed.

  - Expected test error:

$$\mathrm{Err} = \mathrm{E}\left[L(Y, \hat{f}(X))\right] = E\left[\mathrm{E}\left[L(Y, \hat{f}(X)) | \mathcal{T}\right]\right]$$

- We may be interested in $\mathrm{Err}_{\mathcal{T}}$, in practice most estimating methods will give us estimates of Err.

What is our goal in fitting a model?



Hastie et al. (2009)

Blue is in-sample error, red is out-of-sample error.

What is our goal in fitting a model?

- We may want to use expected test error to **select among models**, or versions of models

- And, once we have selected a version of a model, we may want to **assess** how a selected model performs.

What is our goal in fitting a model?

- We can’t measure expected test error directly.

What is our goal in fitting a model?

- A procedure that allows us to estimate it:
  - Split data into three parts

  - Fit models to the training set.
  - Estimate prediction error of models in validation set.
  - Select model with minimum error in validation set.
  - Then get generalization error of just that model on test set.

- Why do we need to estimate the prediction error of the selected model *again*? Winner's curse.

Cross-validation.

Cross-validation.

- We can potentially get more out of our data by cross-validating.

Version 1	Training	Validation
Version 2	Validation	Training

$$\widehat{\mathrm{Err}}_{CV} = \sum_{i=1}^{N} L\left(y_i, \hat{f}^{-k(i)}(x_i)\right)$$

$\hat{f}^{-k(i)}$ are the fits from the folds k that do not contain i.

K-fold cross validation.

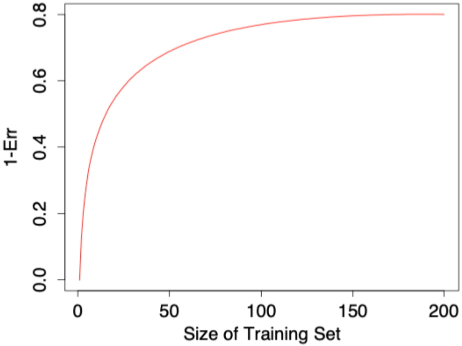
Version 1	Training	Training	Training	Training	Validation
Version 2	Training	Training	Training	Validation	Training
Version 3	Training	Training	Validation	Training	Training
Version 4	Training	Validation	Training	Training	Training
Version 5	Validation	Training	Training	Training	Training

$$\widehat{\mathrm{Err}}_{CV} = \sum_{i=1}^{N} L\left(y_i, \hat{f}^{-k(i)}(x_i)\right)$$

$\hat{f}^{-k(i)}$ are the fits from the folds k that do not contain i.

Cross-validion.

- How do we pick K?
- $K = N$? Low bias, possibly high variance (our prediction sets are very similar).
- $K = 5$? Lower variance, possibly higher bias. How much does the prediction change as we change the size of the data set?



Hastie et al. (2009)

Cross-validion.

- Rule of thumb is often 5 or 10.

Bootstrapping.

Bootstrapping

- Another approach, typically used to estimate the variability of an estimate over random samples, is bootstrapping.
- If we knew the CDF of our population, we would know exactly how to sample from the distribution to determine the sampling variation of our estimate.
- While we do not, we can *suppose* that the empirical CDF produced by the data that we observe is identical to the population CDF.
- We can then just resample with replacement from our observed data, and see how much our estimates vary across resamples.

Bootstrapping for variability of the estimate due to random sampling

The bootstrapping procedure is:

- For b in $1 \ldots B$:
  1. Take a sample of size N *with replacement* from the observed data
  2. Apply the estimating procedure on the bootstrap sample.

- Calculate the standard deviation of the estimate over these many bootstrap estimates.

Bootstrapping

- How can we translate this method to estimate test error?

Bootstrapping for test error

The bootstrapping procedure is:

- For b in $1 \dots B$:
  1. Take a sample of size N *with replacement* from the observed data
  2. Apply the fitting procedure on the bootstrap sample to produce $\hat{f}^b$.

- For each unit i, find the error from all of the bootstrap samples that do *not* contain i

$$\widehat{\mathrm{Err}}_{Boot} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} L(y_i, \hat{f}^b(x_i))$$

C^{-i} is the set of bootstrap samples that do not contain i, $|C^{-i}|$ is the size of this set.

Bootstrapping

- How does the bootstrapping procedure perform?
- We still have the problem of too-small sample size; on average, we only have $0.632 \times N$ unique observations in each boostrap sample.
- This means the bootstrap error *over* estimates the test error.
- Solution:

$$\widehat{\mathrm{Err}}_{Boot} = 0.368\overline{\mathrm{err}} + 0.632\widehat{\mathrm{Err}}_{Boot}$$

  where $\overline{\mathrm{err}}$ is the training error.
- This works...OK.
- Some alternatives in Hastie et al. (2009).

Bagging.

Bagging

- We can combine estimates across samples to get smoother, or better estimators.

- **B**ootstrap **agg**regat**ing**.

Bootstrap estimation for test error

- For b in $1 \dots B$:
  1. Take a sample of size N *with replacement* from the observed data.
  2. Apply the fitting procedure on the bootstrap sample to produce $\hat{f}^b$.

- The bagging estimate is

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{i=1}^{N} \hat{f}^b(x)$$

Bagging

- This is less interesting for something like a linear model, where $\hat{f}_{bag}(x) \to \hat{f}(x)$ as $B \to \infty$, since all of our observations are equally weighted in the sample, we'll reproduce the same thing, or possibly a worse version of it.

- This is more interesting with something "ragged" like regression trees, where different trees can give us different branching behavior that we can smooth over.

- If we're using a classifier, each model can get a "vote" for each x, and the class with the most votes wins.

- Or we can use averages of classifiers to produce probabilities, rather than just class predictions.

Bagging

- How many bootstrap replicates?
- 25? 50? See how your results change with more replicates.

Honesty.

Honesty

- Returning to (causal) inference…we might like to use these methods to get valid inference, potentially on causal targets.

An honest tree algorithm

1. Split the sample into two folds.
2. Use the first fold to learn splits of the tree.
3. Estimate response within leaves using the second fold.

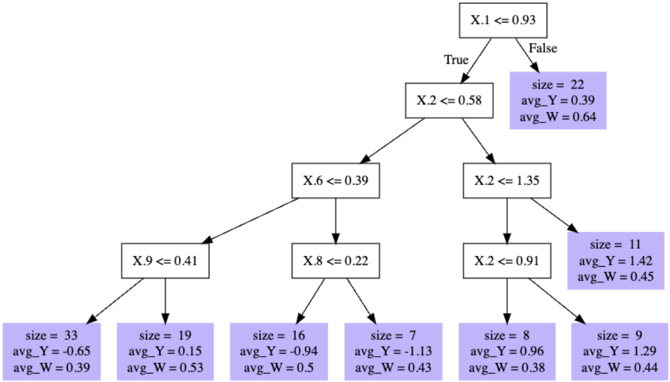
- This can result in some leaves being empty. Prune them?

- This procedure reduces bias relative to those proposed by Breiman (2001).

An honest tree algorithm

```
> library(grf)
> set.seed(60637)
> n <- 500
> p <- 10
> X <- matrix(rnorm(n * p), n, p)
> W <- rbinom(n, 1, 0.5)
> Y <- pmax(X[, 1], 0) * W + X[, 2] +
+   pmin(X[, 3], 0) + rnorm(n)
> c.forest <- causal_forest(X, Y, W)
```

An honest tree



References I

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.