Cross Validation, Bootstrap and Jackknife

Three essential tools

Regression with a Saturated Model

- Imagine a regression with 100 data points and 100 X variables (or 99 and a constant).
- No colinearity.
- What is the residual variance? R^2?
- Residual variance is 0, so $R^2=1$.
- *This is true even if the X variables are all random noise.*

Two solutions

- Penalize R^2 for adding extra predictors (adjusted R^2)
- Set aside some data to fit the model, some to test.
  - *Cross Validation*

More complicated models

- $E[Y] = f(X \mid \theta)$
- $f()$ is a black box (e.g., neural network, support vector machine, Bayes net)
- Find set of parameters $\tilde{\theta}$ that minimize

$$\sum (Y_i - \tilde{Y}_i)^2 \quad \text{or}$$

$$\sum |Y_i - \tilde{Y}_i| \quad \text{where}$$

$$\tilde{Y}_i = f(X_i|\tilde{\theta})$$

- Model will always fit better on training data than out of training sample

Cross Validation

- Split data into *training* and *test* sets
  - Often 10% of data
- Fit model (pick parameters) using training data.
- Evaluate fit using test data
- Really useful for comparing models
- Sometimes use *n-fold* cross validation
  - Divide data into *n* pieces
  - Use *n-1* for training and the last piece for testing
  - Repeat using each of the *n* pieces as the test piece

Cross-Validation in R

- Use `test.samp <- sample.int(nrow(X), p*nrow(X))` to generate index of test data set.
- `X.test <- X[test.samp,]`
- `X.train <- X[-test.samp,]`
- `fit <- lm(…, data=X.test)`
- `predict(fit, newdata=X.test)`

Akeike Information Criteria (AIC)

- Generalization of the idea of adjusted R^2
- *Deviance* (negative 2 times log likelihood) is penalized for number of parameters in model.

$$AIC = 2k - 2\log\left(\mathcal{L}(Y|X, \tilde{\theta})\right)$$

- Can compare non-nested models. Low values of AIC are best
- Generic function in R, `AIC(fit)`

3 methods for getting standard errors

If standard errors are difficult to calculate analytically, they can be calculated by looping over subsamples.

- Bootstrap standard errors
- Jackknife standard errors

Jackknife

- Maurice Quenouille (1949, 1956).
  John Tukey (1958)
- Tukey gave it the name *jackknife* because it was like a Boy Scout's "rough and ready" tool
- Used in NAEP (jackknife weights are supplied)

http://www.wisegeek.com/what-is-a-jackknife.htm

Key Idea

- Standard error is standard deviation of statistic over many possible samples
- We can make *n* samples of a close size by simply leaving out each data point in turn.

Tricks for doing this in R

- Use `data[-i, ]` to remove one row at a time (within loop)
- Pre-allocate storage for results (saves time).
- Make sure that all the calculations (including missing data imputation) are inside the loop.

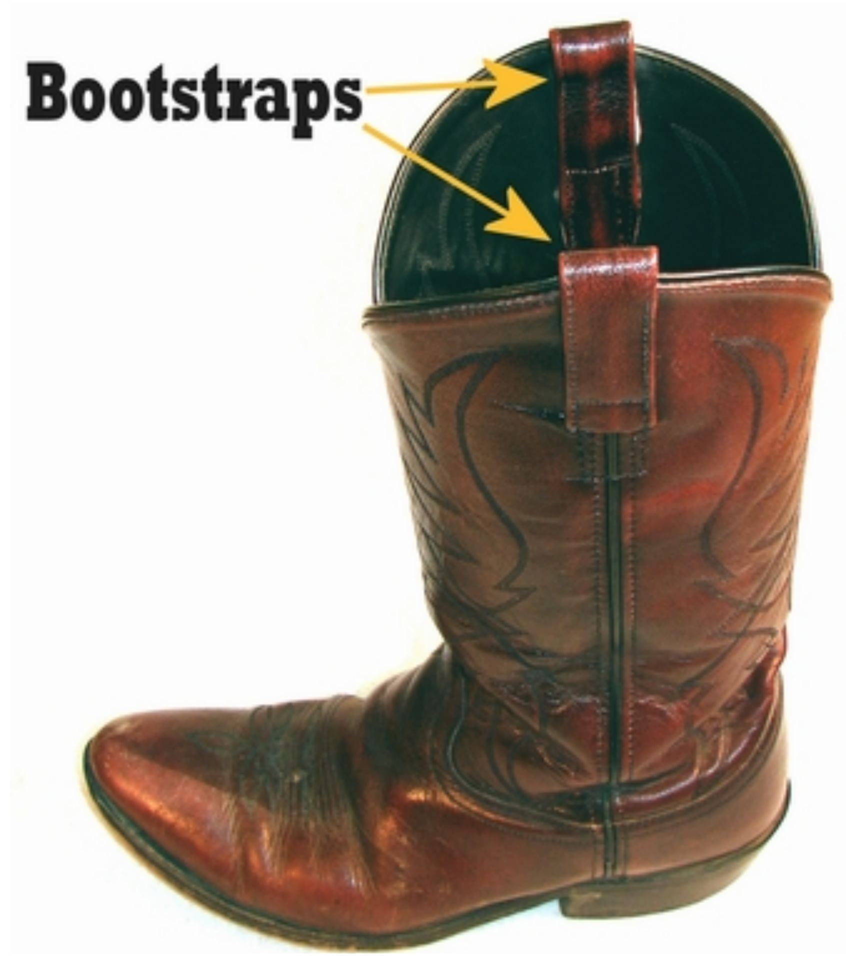
Generic R code

- est(X) be estimator
- imp(X) be imputation function

```
X.est <- est(imp(X))
n<-nrow(X)
jest <- matrix(NA,n,length(X.est))
for (i in 1:nrow(X)){
   X.imp <- imp(X[-i,])
   jest[i,] <- est(X.imp)
}
X.jse <- sqrt((n-1)^2/n*diag(var(jest)))
X.jbias <- n*X.est -(n-1)*colMeans(jest))
```

Bootstrap

- Name comes from 19th C expression "pull oneself over a fence by one's bootstrap"
- Efron, B. *(1979). "Bootstrap methods: Another look at the jackknife". The Annals of Statistics* ***7*** *(1): 1–26. doi: 10.1214/aos/1176344552.*



http://www.lemen.com/imageBootstrap1.html

Tricks for doing this in R

- Use `sample.int(nrow(X),nrow(X), replace=TRUE)` to get each boostrap sample.
- Use `X[samp,]` to get the sample.
- Pre-allocate storage for results (saves time).
- Make sure that all the calculations (including missing data imputation) are inside the loop.

Generic R code

- est(X) be estimator
- imp(X) be imputation function

```
X.est <- est(imp(X))
n<-nrow(X)
B <- 100 ## Number of bootstrap samples
best <- matrix(NA,B,length(X.est))
for (i in 1:B){
   samp <- sample.int(n,n,replace=TRUE)
   X.imp <- imp(X[samp,])
   best[i,] <- est(X.imp)
}
X.bse <- sqrt(diag(var(best)))
X.Bbias <- X.est - colMeans(best)
```