36-724 Spring 2006: Cross-Validation vs. Bootstrapping

Brian Junker

April 5a, 2006

- Quick Review of $K$-Fold Cross-Validation
- Simple Bootstrap Cross-Validation
- Leave-one-out Bootstrap Cross-Validation
- The .632 Bootstrap
Quick Review of $K$-Fold Cross-Validation

- Divide up the data into $K$ roughly-equal-sized parts.
- Let $\hat{f}(x)^{-k}$ be the fitted value (classification, prediction, etc.) for $x$ with the $k^{th}$ part of the data removed, and let $k(i)$ be the part of the data containing $x_i$.
- Then the $K$-fold cross-validation criterion is

$$CV = \frac{1}{N} \sum_{i=1}^{N} L(y, \hat{f}^{-k(i)}(x_i))$$

where $L(y, \hat{y})$ is some appropriate loss function [e.g. $L(y, \hat{y}) = (y - \hat{y})^2$, if we are interested in (E)MSE].

- Bias-variance tradeoff in estimating error with CV:
  - $K$ large: lower bias (large training sets), higher variance (training sets similar)
  - $K$ small: higher bias (small training sets), lower variance (training sets less similar)
Simple Bootstrap Cross-Validation

A simple bootstrap prediction error could be constructed as follows:

- Let the original data set be
  \[
  \mathcal{S} = \begin{cases}
  y_1 & x_{11} & \cdots & x_{1p} \\
  \vdots & \vdots & \ddots & \vdots \\
  y_N & x_{N1} & \cdots & x_{Np}
  \end{cases}
  \]

- Draw bootstrap samples \(\mathcal{S}_b, b = 1, \ldots, B\), where
  \[
  \mathcal{S}_b = \begin{cases}
  y^{*b}_1 & x^{*b}_{11} & \cdots & x^{*b}_{1p} \\
  \vdots & \vdots & \ddots & \vdots \\
  y^{*b}_N & x^{*b}_{N1} & \cdots & x^{*b}_{Np}
  \end{cases}
  \]

- From each bootstrap sample \(\mathcal{S}_b\) train our model \(\hat{f}^{*b}(x)\).
- Compute
  \[
  \hat{\text{Err}}_{\text{boot}} = \frac{1}{B} \frac{1}{N} \sum_{b=1}^{B} \sum_{i=1}^{N} L(y_i, \hat{f}^{*b}(x_i))
  \]
**Problem:** The “full data set” act like the test set (generates $y_i$’s), and the “bootstrap samples” act like training sets (generate $\hat{f}^*_{sb}(x_i)$’s).

- When $(y_i, x_i) \not\in S_b$, the term $\sum_{i=1}^{N} L(y_i, \hat{f}^*_{sb}(x_i))$ looks like cross-validation error;

- When $(y_i, x_i) \in S_b$, the term $\sum_{i=1}^{N} L(y_i, \hat{f}^*_{sb}(x_i))$ looks like training-set error.

Since $S_b$’s are created by sampling with replacement from $S$

$$P[(y_i, x_i) \in S_b] = 1 - (1 - \frac{1}{N})^N \approx 1 - e^{-1} \approx 0.632,$$

$\hat{\text{Err}}_{boot}$ can be considerably biased downward.
Leave-one-out Bootstrap Cross-Validation

A bootstrap error estimate that tries to fix the problem is the "leave-one-out" bootstrap,

\[ \hat{\text{Err}}^{(1)} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} L(y_i, \hat{f}_b^*(x_i)) \]

where \( C^{-i} = \{b : (y_i, x_i) \notin S_b\} \). Note that

- The average number of distinct elements in the \( S_b \)'s retained in \( \hat{\text{Err}}^{(1)} \) is about 0.632 \( \cdot N \)

- So, \( \hat{\text{Err}}^{(1)} \) tends to have low-variance/high-bias for estimating \( \text{Err} = E[L(Y, \hat{f}(X))] \) like 2-fold cross-validation.
The .632 Bootstrap

A compromise bootstrap error estimate is

\[ \hat{\text{Err}}^{(0.632)} = (0.368) \cdot \text{err} + (0.632) \cdot \hat{\text{Err}}^{(1)} \]

- HTF observe that
  - Derivation is complicated but basically it tries to reduce the bias of \( \hat{\text{Err}}^{(1)} \) by pulling it toward the training-set error \( \text{err} \).
  - \( \hat{\text{Err}}^{(0.632)} \) works well in light (under-) fitting situations, but can break down with overfit.
  - \( \hat{\text{Err}}^{(0.632)} \) can be improved by adjusting the coefficients 0.368 and 0.632 for the “no-information” error rate obtained by training on a data sets in which all possible combinations \((y_i, x_i')\) are considered.
Here is a comparison of these various prediction error estimates...