Delta Method, Bootstrap, and Cross Validation

C.Conlon

February 25, 2020

Applied Econometrics II

Bootstrap and Delta Method

- We know how to construct confidence intervals for parameter estimates: $\hat{\theta}_k \pm 1.96 SE(\hat{\theta}_k)$
- Often we are asked to construct standard errors or confidence intervals around model outputs that are not just parameter estimates: ie: $g(x_i, \hat{\theta})$.
- Sometimes we can't even write $g(x_i, \theta)$ as an explicit function of θ ie: $\Psi(g(x_i, \theta), \theta) = 0$.
- Two options:
  1. Delta Method
  2. Bootstrap

Delta Method

Delta method works by considering a Taylor Expansion of $g(x_i, \theta)$.

$$g(z) \approx g(z_0) + g'(z_0)(z - z_0) + o(||z - z_0||)$$

Assume that θ_n is asymptotically normally distributed so that:

$$\sqrt{n}(\theta_n - \theta_0) \sim N(0, \Sigma)$$

(How do we get this: OLS? GMM? MLE?).
Then we have that

$$\sqrt{n}(g(\theta_n) - g(\theta_0)) \sim N(0, D(\theta)'\Sigma D(\theta))$$

Where $D(\theta) = \frac{\partial g(x_i,\theta)}{\partial \theta}$ is the Jacobian of g with respect to theta evaluated at θ.
We need g to be continuously differentiable around the center of our expansion θ.

Delta Method: Examples

Start with something simple: $Y = \overline{X}_1 \cdot \overline{X}_2$ with $(X_{1i}, X_{2i}) \sim IID$. We know the CLT applies so that:

$$\sqrt{n}\begin{pmatrix} \overline{X}_1 - \mu_1 \\ \overline{X}_2 - \mu_2 \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma\right]$$

The Jacobian is just $D(\theta) = \begin{pmatrix} \frac{\partial g(\theta)}{\partial \theta_1} \\ \frac{\partial g(\theta)}{\partial \theta_2} \end{pmatrix} = \begin{pmatrix} s_2 \\ s_1 \end{pmatrix}$

So,

$$V(Y) = D(\theta)'\Sigma D(\theta) = \begin{pmatrix} \mu_2 & \mu_1 \end{pmatrix} \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} \begin{pmatrix} \mu_2 \\ \mu_1 \end{pmatrix}$$

$$\sqrt{n}(\overline{X}_1\overline{X}_2 - \mu_1\mu_2) \sim N(0, \mu_2^2\sigma_{11}^2 + 2\mu_1\mu_2\sigma_{12} + \mu_1^2\sigma_{22}^2)$$

Delta Method: Examples

Think about a simple logit:

$$P(Y_i = 1|X_i) = \frac{\exp^{\beta_0+\beta_1 X_i}}{1+\exp^{\beta_0+\beta_1 X_i}} \quad P(Y_i = 0|X_i) = \frac{1}{1+\exp^{\beta_0+\beta_1 X_i}}$$

Remember the "trick" to use GLM (log-odds):

$$\log P(Y_i = 1|X_i) - \log P(Y_i = 0|X_i) = \beta_0 + \beta_1 X_i$$

- Suppose that we have estimated $\hat{\beta_0}, \hat{\beta_1}$ via GLM/MLE but we want to know the confidence interval for the probability: $P(Y_i = 1|X_i, \hat{\theta})$
- The derivatives are a little bit tricky, but the idea is the same.
- This is what STATA should be doing when you type: `mfx, compute`

Delta Method: Other Examples

Often we have a regression like:

$$logY_i = \beta_0 + \beta_1 X_i + \gamma Income_i + \epsilon_i$$

And we are interested in β_1/γ so that we have β_i in units of “dollars”. Again Delta Method Works fine here.

Delta Method: Some Failures

But we need to be careful. Suppose that $\theta \approx 0$ and

- $g(x) = |X|$
- $g(x) = 1/X$
- $g(x) = \sqrt{X}$

These situations can arise in practice when we have weak instruments or other problems.

Bootstrap

- Bootstrap takes a different approach.
  - Instead of estimating $\hat{\theta}$ and then using a first-order Taylor Approximation...
  - What if we directly tried to construct the sampling distribution of $\hat{\theta}$?
- Our data $(X_1, \ldots, X_n) \sim P$ are drawn from some measure P
  - We can form a nonparametric estimate $\hat{P}$ by just assuming that each X_i has weight $\frac{1}{n}$.
  - We can then simulate a new sample $X^* = (X_1^*, \ldots X_n^*) \sim \hat{P}$.
    - Easy: we take our data and construct n observations by sampling with replacement
  - Compute whatever statistic of X^*, $S(X^*)$ we would like.
    - Could be the OLS coefficients $\beta_1^*, \ldots, \beta_k^*$.
    - Or some function β_1^*/β_2^*.
    - Or something really complicated: estimate parameters of a game $\hat{\theta}^*$ and now find Nash Equilibrium of the game $S(X^*, \hat{\theta}^*)$ changes.
  - Do this B times and calculate at $Var(S_b)$ or $CI(S_1, \ldots, S_b)$.

Bootstrap: Bias Correction

The main idea is that $\hat{\theta}^{1*}, \ldots, \hat{\theta}^{B*}$ approximates the sampling distribution of $\hat{\theta}$. There are lots of things we can do now:

- We already saw how to calculate $Var(\hat{\theta}^{1*}, \ldots, \hat{\theta}^{B*})$.

$$\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}^{*}_{(b)} - \overline{\theta^{*}})^2$$

- Calculate $E(\hat{\theta}^{*}_{(1)}, \ldots, \hat{\theta}^{*}_{(B)}) = \overline{\theta^{*}} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{*}_{(b)}$.

Bootstrap: Bias Correction

- We can use the estimated bias to bias correct our estimates

$$
\begin{aligned}
Bias(\hat{\theta}) &= E[\hat{\theta}] - \theta \\
Bias_{bs}(\hat{\theta}) &= \overline{\theta^*} - \hat{\theta}
\end{aligned}
$$

Recall $\theta = E[\hat{\theta}] - Bias[\hat{\theta}]$:

$$
\hat{\theta} - Bias_{bs}(\hat{\theta}) = \hat{\theta} - (\overline{\theta^*} - \hat{\theta}) = 2\hat{\theta} - \overline{\theta^*}
$$

- Correcting bias isn't for free - variance tradeoff!
- Linear models are (hopefully) unbiased, but most nonlinear models are consistent but biased.

Bootstrap: Confidence Intervals

There are actually three ways to construct bootstrap CI's:

1. Obvious way: sort $\hat{\theta}^*$ then take $CI : [\hat{\theta}^*_{\alpha/2}, \hat{\theta}^*_{1-\alpha/2}]$.
2. Asymptotic Normal: $CI : \hat{\theta} \pm 1.96\sqrt{V(\hat{\theta}^*)}$. (CLT).
3. Better Way: let $W = \hat{\theta} - \theta$. If we knew the distribution of W then: $Pr(w_{1-\alpha/2} \leq W \leq w_{\alpha/2})$:

$$CI : [\hat{\theta} - w_{1-\alpha/2}, \hat{\theta} - w_{\alpha/2}]$$

We can estimate with $W^* = \hat{\theta}^* - \hat{\theta}$.

$$CI : [\hat{\theta} - w^*_{1-\alpha/2}, \hat{\theta} - w^*_{\alpha/2}] = [2\hat{\theta} - \theta^*_{1-\alpha/2}, 2\hat{\theta} - \theta^*_{\alpha/2}]$$

Why is this preferred? Bias Correction!

Bootstrap: Why do people like it?

- Econometricians like the bootstrap because under certain conditions it is higher order efficient for the confidence interval construction (but not the standard errors).
  - Intuition: because it is non-parametric it is able to deal with more than just the first term in the Taylor Expansion (actually an Edgeworth Expansion).
  - Higher-order asymptotic theory is best left for real econometricians!
- Practitioner's like the bootstrap because it is easy.
  - If you can estimate your model once in a reasonable amount of time, then you can construct confidence intervals for most parameters and model predictions.

Bootstrap: When Does It Fail?

- Bootstrap isn't magic. If you are constructing standard errors for something that isn't asymptotically normal, don't expect it to work!
- The Bootstrap exploits the notion that your sample is IID (by sampling with replacement). If IID does not hold, the bootstrap may fail (but we can sometimes fix it!).
- Bootstrap depends on asymptotic theory. In small samples weird things can happen. We need $\hat{P}$ to be a good approximation to the true P (nothing missing).

Bootstrap: Variants

The bootstrap I have presented is sometimes known as the nonparametric bootstrap and is the most common one.

Parametric Bootstrap ex: if $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ then we can estimate $(\hat{\beta}_0, \hat{\beta}_1)$ via OLS.
Now we can generate a bootstrap sample by drawing an x_i at random with replacement $\hat{\beta}_0 + \hat{\beta}_1$ and then drawing independently from the distribution of estimated residuals $\hat{\epsilon}_i$.

Wild Bootstrap Similar to parametric bootstrap but we rescale ϵ_i to allow for heteroskedasticity

Block Bootstrap For correlated data (e.g.: time series). Blocks can be overlapping or not.

Bootstrap vs Delta Method

- Delta Method works best when working out Jacobian $D(\theta)$ is easy and statistic is well approximated with a linear function (not too curvy).
- I would almost always advise Bootstrap unless:
  - Delta method is trivial e.g.: β_1/β_2 in linear regression.
  - Computing model takes many days so that 10,000 repetitions would be impossible.
- Worst case scenario: rent time on Amazon EC2!
  - I "bought" over $1,000 of standard errors recently.
- But neither is magic and both can fail!

Cross Validation

Cross Validation appears superficially similar to bootstrap but asks a different question.

- Bootstrap tries to construct an empirical analogue to the sampling distribution of $\hat{\theta}$.
- CV tries to measure what the expected out of sample (OOS or EPE) prediction error of a new never seen before dataset.
- The main consideration is to prevent overfitting.
    - In sample fit is always going to be maximized by the most complicated model.
    - OOS fit might be a different story.
    - 1-NN might do really well in-sample, but with a new sample might perform badly.

Sample Splitting/Holdout Method and CV

Cross Validation is actually a more complicated version of sample splitting that is one of the organizing principles in machine learning literature.

Training Set This is where you estimate parameter values.

Validation Set This is where you choose a model- a bandwidth h or tuning parameter λ by computing the error.

Test Set You are only allowed to look at this after you have chosen a model. Only Test Once: compute the error again on fresh data.

- Conventional approach is to allocate 50-80% to training and 10-20% to Validation and Test.
- Sometimes we don't have enough data to do this reliably.

Sample Splitting/Holdout Method

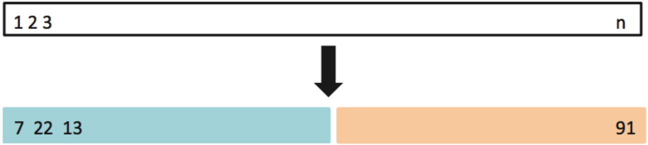

FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

Challenge with Sample Splitting

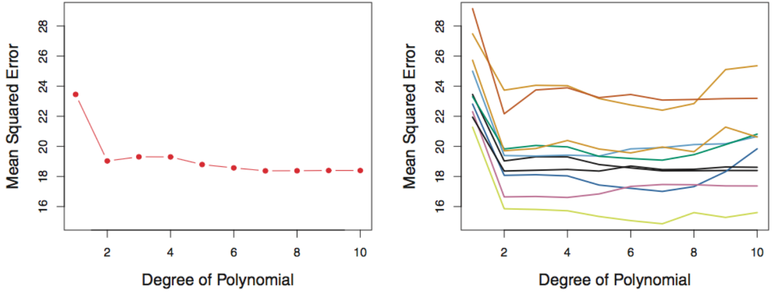

FIGURE 5.2. *The validation set approach was used on the* **Auto** *data set in order to estimate the test error that results from predicting* **mpg** *using polynomial functions of* **horsepower**. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

Cross Validation

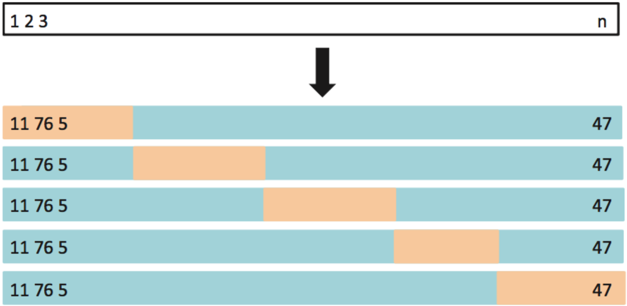

FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

k-fold Cross Validation

- Break the dataset into k equally sized "folds" (at random).
- Withhold $i = 1$ fold
  - Estimate the model parameters $\hat{\theta}^{(-i)}$ on the remaining $k - 1$ folds
  - Predict $\hat{y}^{(-i)}$ using $\hat{\theta}^{(-i)}$ estimates for the ith fold (withheld data).
  - Compute $MSE_i = \frac{1}{k \cdot N} \sum_j (y_j^{(-i)} - \hat{y}_j^{(-i)})^2$.
  - Repeat for $i = 1, \ldots, k$.
- Construct $\widehat{MSE}_{k,CV} = \frac{1}{k} \sum_i MSE_i$

Leave One Out Cross Validation (LOOCV)

Same as k-fold but with $k = N$.

- Withhold a single observation i
- Estimate $\hat{\theta}_{(-i)}$.
- Predict $\hat{y}_i$ using $\hat{\theta}^{(-i)}$ estimates
- Compute $MSE_i = \frac{1}{N}\sum_j (y_i - \hat{y}_i(\hat{\theta}^{(-i)}))^2$.

Note: this requires estimating the model N times which can be costly.

Cross Validation

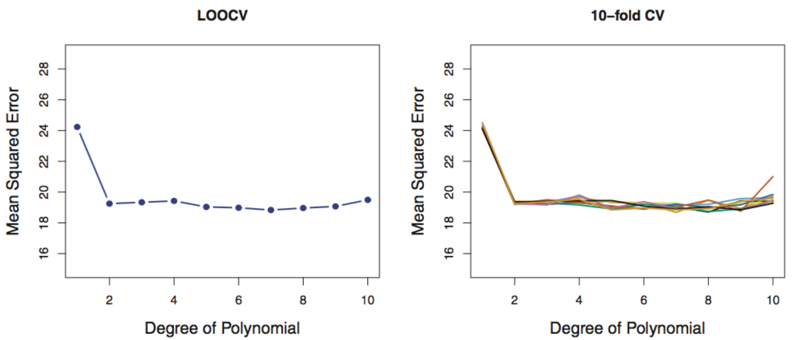

FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: *10-fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.*

Cross Validation

- Main advantage of cross validation is that we use all of the data in both estimation and in validation.
  - For our purposes validation is mostly about choosing the right bandwidth or tuning parameter.
- We have much lower variance in our estimate of the OOS mean squared error.
  - Hopefully our bandwidth choice doesn't depend on randomness of splitting sample.

Test Data

- In Statistics/Machine learning there is a tradition to withhold 10% of the data as Test Data.
- This is completely new data that was not used in the CV procedure.
- The idea is to report the results using this test data because it most accurately simulates true OOS performance.
- We don't do much of this in economics.
  (Should we do more?)