Choosing the Tuning Parameter

1 Motivation

Implementing either the hard or soft constraint bound estimator requires a tuning parameter (ϵ or κ, respectively). The Monte Carlo simulations confirm that performance of the estimator varies substantially with the tuning parameter. When we know the DGP, as in a Monte Carlo simulation, we also know the true bounds, and so we can try different values of the tuning parameter to find one that optimizes a measure of performance, such as MSE. Of course, in practice we don't know the DGP. So we would like to find some sort of data-driven way to choose the tuning parameter.

2 Problem statement

Let's focus on the lower bound, and either the soft or hard constraint, with a fixed norm. All of the following will work the same way regardless of whether it's the hard or soft constraint, or what norm we choose. Let's first just look at the soft constraint. Let $\hat{t}(\kappa; \boldsymbol{w})$ denote the estimate of the lower bound when the tuning parameter is κ and the observed realization of the data vector $\boldsymbol{W} \equiv \{W_i\}_{i=1}^n$ is $\boldsymbol{w} \equiv \{w_i\}_{i=1}^n$.[1] Sometimes I will let $\hat{T}(\kappa) \equiv \hat{t}(\kappa; \boldsymbol{W})$ notationally.

Our goal is to choose κ to optimize some a measure of performance for $\hat{T}(\kappa)$ as an estimator of $t^\star$. The "right" measure is itself not an obvious question here, but to start with something simple, let's focus on the mean-squared error:[2]

$$\text{MSE}(\kappa) \equiv \mathbb{E}\left[\left(\hat{T}(\kappa) - t^\star\right)^2\right] \equiv \mathbb{E}\left[\left(\hat{t}(\kappa; \boldsymbol{W}) - t^\star\right)^2\right], \tag{1}$$

[1] In the mixed logit model, we had $w_i \equiv (y_i, x_i)$.

[2] For example, we might want to penalize over-estimates of lower bounds more than under-estimates, since an under-estimate leads to bounds that are conservative, but not incorrect, whereas an over-estimate leads to incorrect bounds. This is sort of an unexplored issue, as far as I know, but maybe something we can return to.

where $t^\star$ is the population lower bound and the expectation is over $\boldsymbol{W}$. We want to find a κ that minimizes MSE(κ). The problem is we don't know the population distribution of $\boldsymbol{W}$, which means that we don't know $\mathbb{E}$ or $t^\star$. We can make this explicit by calling the unknown population distribution F, and then writing

$$\text{MSE}(\kappa) \equiv \mathbb{E}_F \left[\left(\hat{t}(\kappa, \boldsymbol{W}) - t^\star(F) \right)^2 \right], \tag{2}$$

where $\mathbb{E}_F$ means take the expectation (of $\hat{t}(\kappa, \boldsymbol{W})$ in this case) when each W_i is distributed i.i.d. according to F, and $t^\star(F)$ is the population lower bound when W is distributed according to F.

3 Bootstrap

The rough idea of the bootstrap is to estimate F, replace MSE(κ) with the estimated F, and then compute it by drawing from the estimated F. There are two main ways we might estimate F in the mixed logit model. I will postpone that specific discussion until the next section, and first just assume that we have some estimator $\hat{F}$ of F.

Given $\hat{F}$, here's how the bootstrap choice of tuning parameter would work:

1. Replace $t^\star(F)$ by $t^\star(\hat{F})$. This stays fixed through the following steps.

2. Draw a sample $\boldsymbol{w}_s \equiv \{w_{i,s}\}_{i=1}^n$ from $\hat{F}$, where each $w_{i,s}$ is drawn independently of $w_{j,s}$ for $i \neq j$. (That is, it's an i.i.d. sample, just like we are assuming for $\{W_i\}_{i=1}^n$)

3. Compute

$$\left(\hat{t}(\kappa, \boldsymbol{w}_s) - t^\star(\hat{F}) \right)^2 \tag{3}$$

   for "many" values of κ, i.e. on some grid that we hopefully can make reasonable fine in practice.

4. Repeat $s = 1, \ldots, S$ times, where S is a large number, but not necessarily too large. The larger S is, the better the approximation, but we want to know how well things are going to work in practice, when a practitioner might take S smaller for practical reasons. So I would say take $S = 500$, although we could go lower too (say $S = 250$).

5. Now we have an estimate of $\text{MSE}(\kappa)$:

$$\widehat{\text{MSE}}(\kappa) \equiv \frac{1}{S}\sum_{s=1}^{S}\left(\hat{t}(\kappa, \boldsymbol{w}_s) - t^\star(\hat{F})\right)^2 \tag{4}$$

6. Let $\hat{\kappa}$ denote the value of κ that minimizes $\widehat{\text{MSE}}(\kappa)$ over the grid of κ.

7. Go back and compute $\hat{T}(\hat{\kappa})$ as before using only the original sample. This is now our estimate of the lower-bound using the bootstrap selection of the tuning parameter.

To see how well this works, we again use a Monte Carlo simulation. Now on *each* Monte Carlo replication, we perform *all* of the steps above, starting with estimating $\hat{F}$ from a draw of $\boldsymbol{w} \equiv \{w_i\}_{i=1}^n$. So if, for example, there are $M = 1000$ Monte Carlo replications, and $S = 500$, and the grid of κ has $K = 100$ points, then we are talking about a simulation that requires computing $\hat{T}(\kappa)$ $1000 \times 500 \times 100 = 5 \times 10^7$ or 50 million times.

4 Two choices of $\hat{F}$ in the mixed logit model

4.1 Nonparametric bootstrap

The most common way to estimate F is with the empirical distribution. Given $\boldsymbol{w} \equiv \{w_i\}_{i=1}^n$, the empirical distribution is defined as

$$\hat{F}(w) \equiv \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}[w_i \leq w]. \tag{5}$$

Drawing a sample $\boldsymbol{w}_s \equiv \{w_{i,s}\}_{i=1}^n$ from $\hat{F}$ means drawing each $w_{i,s}$ "with replacement" from $\{w_i\}_{i=1}^n$ with equal probability n^{-1} and independently of all other $w_{j,s}$. This is called the nonparametric bootstrap.

The nonparametric bootstrap might be a bit awkward for the current application because it's not really clear how to define $t^\star(\hat{F})$. If, as is usually the case, the empirical bounds are empty with $\hat{F}$—that is $\hat{Q}^\star \neq 0$—then $t^\star(\hat{F})$ is not defined. This was the motivation for estimating $t^\star$ to begin with. Instead, let's replace $t^\star(\hat{F})$ by $\hat{T}(+\infty)$ and see how that works.[3]

[3]Remember that the soft constraint approach with $\kappa = +\infty$ is the same as the hard constraint approach with $\epsilon = 0$.

4.2 Parametric bootstrap

As before, let $\hat{Q}$ be the criterion function. Consider the MinQ problem, and now let $\hat{\theta}^\star$ be some solution to it, so that $\hat{Q}(\hat{\theta}^\star) = \hat{Q}^\star$.[4] In the mixed logit model, θ entirely determines the conditional distribution of Y given X, via

$$\mathbb{P}[Y = 1|X = x] = \sum_{r=1}^{R} \theta_r \lambda(b_r, x). \tag{6}$$

So we can view $\hat{\theta}^\star$ as providing us with an estimator of $\mathbb{P}[Y = 1|X = x]$ for all x—that is, of the distribution of Y conditional on X. We also need an estimate of the marginal distribution of X. For this we can use the empirical density of X from our sample:

$$\hat{f}(x) \equiv \sum_{i=1}^{n} \mathbb{1}[X_i = x]. \tag{7}$$

Now we have an estimator of F:

$$\hat{F}(1, x) \equiv \left(\sum_{r=1}^{R} \hat{\theta}_r^\star \lambda(b_r, x) \right) \hat{f}(x). \tag{8}$$

This is called the parametric bootstrap because, unlike the nonparametric bootstrap, it's using the model to estimate F. Drawing from $\hat{F}$ can be done in two steps:

1. Draw $x_{i,s}$ according to $\hat{f}(x)$—same as in the nonparametric bootstrap, but now only for $x_{i,s}$.

2. Draw $y_{i,s}$ to be 1 with probability $\sum_{r=1}^{R} \hat{\theta}_r^\star \lambda(b_r, x_{i,s})$.

One attraction of the parametric bootstrap here is that $t^\star(\hat{F})$ is now well-defined. Namely:

$$t^\star(\hat{F}) \equiv \min_{\theta \in \Theta} \; t(\theta) \quad \text{s.t.} \quad \sum_{r=1}^{R} \theta_r \lambda(b_r, x) = \sum_{r=1}^{R} \hat{\theta}_r^\star \lambda(b_r, x) \quad \text{for all } x. \tag{9}$$

This problem is always going to be feasible, since $\hat{\theta}^\star \in \Theta$ satisfies all of the constraints by construction.

[4] The solution isn't necessarily unique, but that won't matter for the following—any solution should be fine.

5 Iteration

In the nonparametric bootstrap, we replaced $t^\star(\hat{F})$ by $\hat{T}(+\infty)$. Let's call $\hat{\kappa}_0 \equiv +\infty$, and then call $\hat{\kappa}_1 \equiv \hat{\kappa}$. Then we could think about iterating the procedure be repeating it with $t^\star(\hat{F})$ replaced by $\hat{T}(\hat{\kappa}_1)$, and then letting $\hat{\kappa}_2 \equiv \hat{\kappa}$ be the new $\hat{\kappa}$ that arises. Repeat to get a sequence $\hat{\kappa}_1, \hat{\kappa}_2, \ldots$. Does the sequence look like it is converging? What happens to the sequence $\hat{T}(\hat{\kappa}_1), \hat{T}(\hat{\kappa}_2), \ldots$?

Computing the sequence should not add too much computational burden I think. One just needs to save the $\hat{t}(\kappa, \boldsymbol{w}_s)$ for all $s = 1, \ldots, S$ and all κ in the grid. Then each iteration of κ just requires changing the centering of $\widehat{\mathrm{MSE}}(\kappa)$ and then finding the minimum (across the finite κ grid), both of which are computationally fast operations.