STATS305A - Lecture 17

John Duchi
Scribed by Michael Howes

11/18/21

Contents

1 The bootstrap

Today we will discuss two new techniques, the bootstrap and M-estimators.

1.1 Motivation

Suppose we have an estimate $\widehat{\beta}$ from some procedure. How can we get a handle on the sampling variability of $\widehat{\beta}$? Two approaches are:

- Make modelling assumptions such as $y = X\beta + \varepsilon$ where $\varepsilon \stackrel{\text{iid}}{\sim} (0, \sigma^2)$.
- Use the variability in the sample $(X_i, Y_i)_{i=1}^n \stackrel{\text{iid}}{\sim} P$.

The bootstrap is an example of the second approach. The bootstrap is due to our own Brad Efron (1979). The idea is if we define the *emprical distribution* of a sample $(x_i, y_i)_{i=1}^n$ to be:

$$\widehat{P} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{(x_i, y_i)}.$$

Then the idea of the bootstrap is that sampling from $\widehat{P}$ should approximate sampling from the true distribution P. There are many ways to implement this idea.

1.2 Resampling bootstrap

The resampling bootstrap is the most commonly used form of the bootstrap. It is often what people mean when they refer to the bootstrap. The procedure is as follow. First pick a large number B which will be the number of bootstrap resamples. For $b = 1, 2, \ldots, B$, do the following:

- Draw a sample of size n (X^{*b}, Y^{*b}) from $\widehat{P}$ where n is the size of the original sample. This is done by drawing indices $i_1, i_2, \ldots, i_n$ from $[n] = \{1, \ldots, n\}$ uniformly at random with replacement and then setting $X^{*b} = (X_{i_1}, \ldots, X_{i_n})$ and $Y^{*b} = (Y_{i_1}, \ldots, Y_{i_n})$.
- Then calculate $\widehat{\beta}^{*b} = \widehat{\beta}(X^{*b}, Y^{*b})$ from the bootstrap resample.

We can then use $(\widehat{\beta}^{*b})_{b=1}^{B}$ tp estimate the sampling distribution of $\widehat{\beta}$ when we draw $(X, Y) \overset{\text{iid}}{\sim} P$. There are two forms of error in this approximation:

- There is error from the Monte-Carlo sampling. This can be reduced by increasing B which should also be taken as large as computationally possible.
- There is also error because $\widehat{P} \neq P$. This error cannot be eliminated.

1.3 Theory of the bootstrap

How can one mathematically describe the "success" of the bootstrap? Suppose we have a parameter of interest $\theta \in \mathbb{R}$ and we are interested in an estimator $\widehat{\theta}$. For example we may have $\widehat{\theta} = c^T\widehat{\beta}$ or $\widehat{\theta} = \widehat{\beta}_j$ or $\widehat{\theta} = \|\widehat{\beta}\|_2^2$. Define

$$J_n(t, P) = \mathbb{P}(\sqrt{n}(\widehat{\theta}_n - \theta^*) \leq t),$$

where $\widehat{\theta}_n = \widehat{\theta}(X_1, \ldots, X_n; Y_1, \ldots, Y_n)$ is an estimate calculated on an i.i.d. sample drawn from P and θ^* is the population parameter. The function J_n is the (normalized) CDF of the sampling distribution of $\widehat{\theta}_n$ about θ^*. We can similarly define

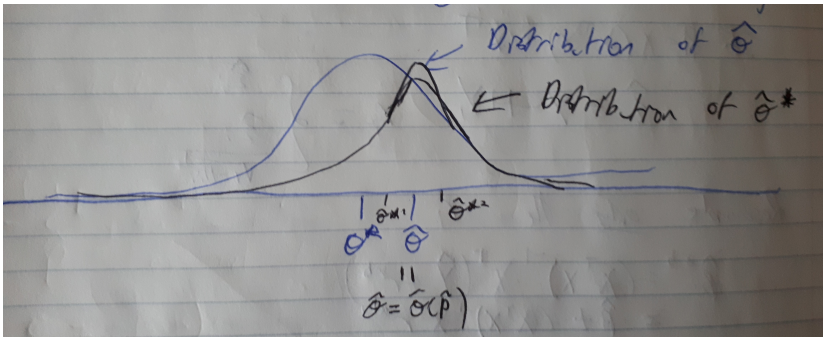
$$J_n(t, \widehat{P}) = \mathbb{P}(\sqrt{n}(\widehat{\theta}_n^* - \widehat{\theta}_n) \leq t),$$

where $\widehat{\theta}_n^* = \widehat{\theta}(X^*, Y^*)$ where (X^*, Y^*) is an i.i.d. sample of size n from $\widehat{P}$ (that is (X^*, Y^*) is a bootstrap sample). Note that for the distribution $\widehat{P}$, $\widehat{\theta}_n$ is the population parameter. This is why we are now centring at $\widehat{\theta}_n$. We call $J_n(t, \widehat{P})$ the bootstrap CDF of the statistic $\widehat{\theta}_n$.

While calculating or approximating $J_n(t, P)$ maybe be very hard, we can always get a handle on $J_n(t, \widehat{P})$ by drawing more bootstrap samples. Under some regularity conditions with have that with probability 1,

$$\sup_{t \in \mathbb{R}} \left| J_n(t, P) - J_n(t, \widehat{P}) \right| \xrightarrow{n} 0.$$

So for large n, $J_n(t, \widehat{P})$ is close to $J_n(t, P)$. It can shown that under some assumptions the rate of convergence is $\sqrt{\frac{\log n}{n}}$. We thus have a picture that looks like the one below. The bootstrap density of $\widehat{\theta}^*$ is close to the true density of $\widehat{\theta}$.



We can use the bootstrap to define $1-\alpha$ confidence intervals for parameters $\theta \in \mathbb{R}$. One option is

$$CI_\alpha = \left[\text{Quantile}_{\alpha/2}(\widehat{\theta}^{*b}), \text{Quantile}_{1-\alpha/2}(\widehat{\theta}^{*b})\right].$$

Another option is

$$CI'_\alpha = \widehat{\theta} \pm z_{1-\frac{\alpha}{2}} \sqrt{\text{Var}(\widehat{\theta}^{*b}|\widehat{P})},$$

where $z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quartile of the standard normal. One can show that

$$\mathbb{P}(\theta^* \in CI_\alpha), \mathbb{P}(\theta^* \in CI'_\alpha) \xrightarrow{n} 1-\alpha.$$

1.4 The bootstrap in linear models

One variant of the bootstrap is called the *parametric bootstrap* which can be used if one has faith in their model. The procedure is as follows. In the linear model $y = X\beta + \varepsilon$,

- Fit model $\widehat{\beta}$ based on original sample.
- Resample from the distribution $\widehat{\beta}$ defines in the linear model ie $y^* = X\widehat{\beta} + \varepsilon^*$ but only resample the residuals.
- To resample the residuals, let $H = X(X^TX)^{-1}X^T$, $\widehat{\varepsilon} = Y - HY$ and $r_i = \widehat{\varepsilon}_i$ or $r_i = \frac{\widehat{\varepsilon}_i}{\sqrt{1-H_{ii}}}$ (both are used in practice).
- Generate bootstrap resamples by resampling *only* r_i to get r_i^{*b}. Our bootstrap resample of $\widehat{\beta}$ is then
$$\widehat{\beta}^{*b} = (X^TX)^{-1}X^T(y + r^{*b}) = \widehat{\beta} + (X^TX)^{-1}X^Tr^{*b}.$$

We can then use the bootstrap resamples $\widehat{\beta}^{*b}$ as described above. Some pros of this method are:

- It works well when we have a fixed design matrix X.
- It is more efficient provided the model is true.

A major drawback is:

- It really relies on the model assumptions. It especially relies on the noise being homeskedastic. Problems occur if $\mathbb{E}[\varepsilon_i^2] = \sigma_i^2 \neq \sigma_j^2 = \mathbb{E}[\varepsilon_j^2]$.

Thus people often use the *non-parametric bootstrap* for the linear model which does not have the above drawback. Here we resample from $\widehat{P}$ to get bootstrap resamples (X^{*b}, y^{*b}) and then we calculate

$$\widehat{\beta}^{*b} = ((X^{*b})^TX^{*b})^{-1}(X^{*b})^Ty^{*b}.$$

This doesn't rely on the model being true and it allows from non homeskedastic noise. There is a drawback that for small sample sizes, the matrix X^{*b} may be rank deficient but for reasonably large n, this does not happen often.

1.5 Weighted bootstrap

This is sometimes called Bayesian bootstrap although it isn't really Bayesian at all. We first fix a distribution W which is supported on $[0, \infty)$ and satisfies $\mathbb{E}[W] = 1$ and $\text{Var}[W] < \infty$. Often people use the exponetial distribution with scale 1.

At every bootstrap iteration b, we sample weights $w_1^{*b}, \dots, w_n^{*b} \overset{\text{iid}}{\sim} W$. We then define a distribution based on these weights and our observed sample $(x_i, y_i)_{i=1}^n$,

$$\widehat{P}^{*b} = \frac{\sum_{i=1}^n w_i^{*b} \mathbf{1}_{(x_i, y_i)}}{\sum_{i=1}^n}.$$

We then define

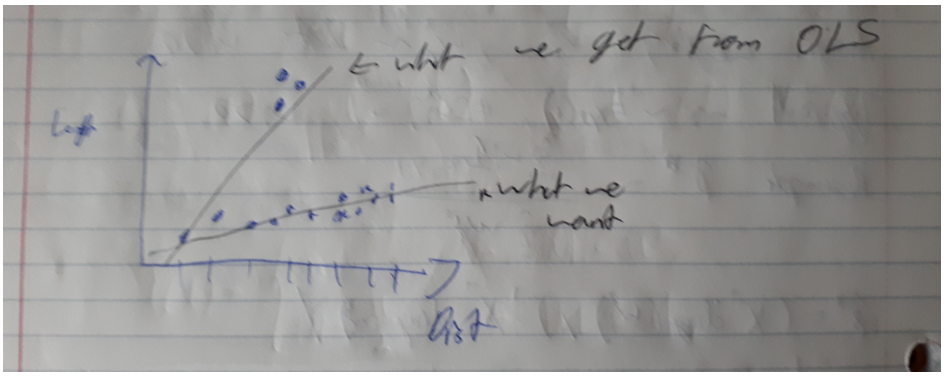
$$\begin{aligned}
\widehat{\beta}^{*b} &= \underset{c \in \mathbb{R}^d}{\text{argmin}}\, \mathbb{E}_{\widehat{P}^{*b}} \left[\left\| y - X^T c \right\|_2^2 \right] \\
&= \underset{c \in \mathbb{R}^d}{\text{argmin}} \frac{\sum_{i=1}^n w_i^{*b} (y_i - x_i^T c)^2}{\sum_{i=1}^n w_i^{*b}} \\
&= (X^T \operatorname{diag}(w_i^{*b}) X)^{-1} X^T \operatorname{diag}(w_i^{*b}) y.
\end{aligned}$$

There are similar asymptotic results for the weighted bootstrap as there are for the resampling bootstrap.

2 M-estimators and robust regression

Suppose we want to fit a regression $y_i = x_i^T \beta + \varepsilon_i$ but maybe we no longer have $\varepsilon_i \sim \mathsf{N}(0, \sigma^2)$. Maybe there are some outlying y_i's that are much larger than we'd expect for normally distributed errors.

For a histrical example, imagine trying to predict phone call length from distance to family. We might expect that the further apart two people are, the less often they would catch up and so their phone calls would be longer. The resulting would look something like this:



There is a problem because there are some people who just really likedtalking on the phone for a long time. These data points really mess up the linear model. Some solutions are:

- Remove outliers. This can be hard as it is not always clear what make something an outlier.
- Choose an alternative to squared error to use when fitting (Huber, 70s-90s).

We will discuss the second approach.

Definition 1. A *loss function* is a function $l : \mathbb{R} \to \mathbb{R}^+$. We will sometimes require that $l(0) = 0$.

Definition 2. Given data $(x_i, y_i)_{i=1}^n$ and a loss function l, the *M-estimator* for predicting Y from X is the linear estimator $\widehat{f}(x) = x^T \widehat{\beta}$ where

$$\widehat{\beta} = \underset{b}{\text{argmin}} \frac{1}{n} \sum_{i=1}^n l(y_i - x_i^T b).$$

The rough recipe is to choose a loss to capture properties we want the predictor to have. For example if $Y = f(X) + \varepsilon$, then we might want

$$\operatorname*{argmin}_t \mathbb{E}[l(Y - t)|X = x] = f(x).$$

For example for when l is squared error we get the conditional mean above and when l is absolute error we get the conditional median. Some examples of loss functions are:

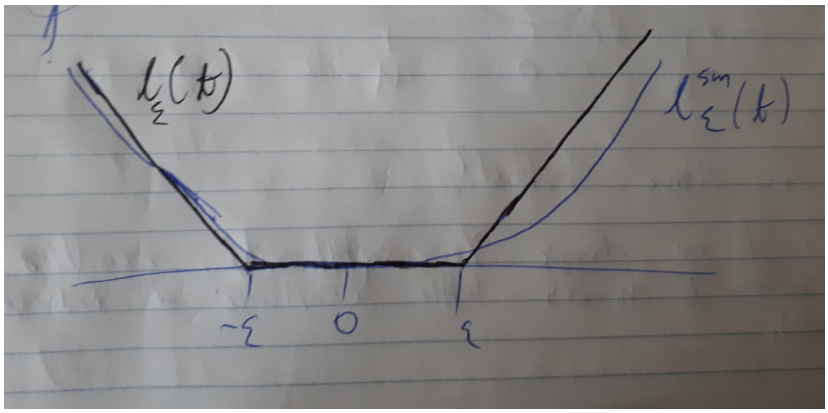
- *Absolute error loss*: $l(t) = |t|$. As stated above this has the property that

$$\operatorname*{argmin}_t \mathbb{E}[|Y - t|] = \text{median}(Y).$$

- *ε-insensitive loss*: If we don't care about if the predicted value is off by less than ε we can use the loss

$$l^{\varepsilon}(t) = (t - \varepsilon)_+ + (-t - \varepsilon)_+.$$

  This loss looks like the black function



- *Smoothed ε-insensitive loss* is a smooth version of ε-insensitive loss. It is defined by:

$$l_{\varepsilon}^{sm}(t) = \log(1 + \exp(t - \varepsilon)) + \log(1 + \exp(-t - \varepsilon)).$$

  This is the blue function in the picture above. We also have rescaled versions of the smoothed ε-insensitive loss. For $a > 0$ define,

$$l_{\varepsilon,a}^{sm}(t) = a \cdot l_{\varepsilon}^{sm}\left(\frac{t}{a}\right),$$

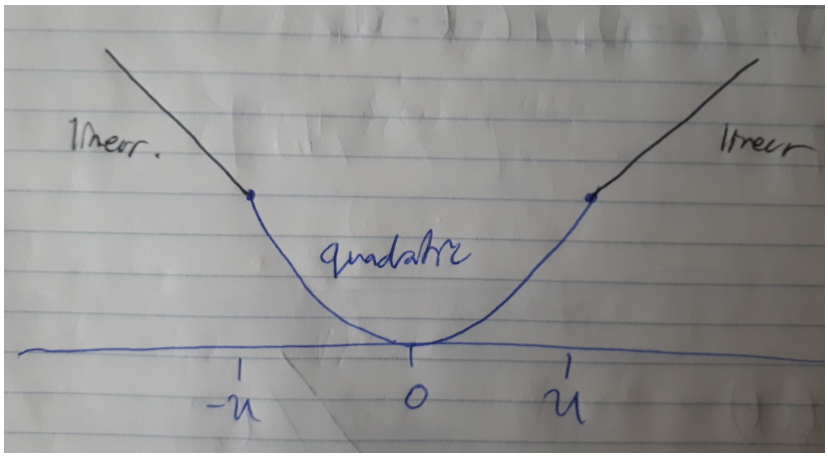
  which satisfies

$$\lim_{a \searrow 0} l_{\varepsilon,a}^{sm}(t) = l_{\varepsilon}(t).$$

- *Huber loss*: for $u > 0$, define

$$l(t) = \begin{cases} \frac{1}{2u}t^2 & \text{if } |t| \le u, \\ |t| - \frac{u}{2} & \text{if } |t| \ge u. \end{cases}$$

  Huber loss looks like this:



Why are these M-estimators more "robust"? The rough idea is that if we change a single y_i arbitrarily, then the resulting estimator $\widehat{\beta}$ changes little. Thus the influence of individual observations is small.

Some loss functions give estimators with this property, others do not. For example, we have seen that under squared error, points with high leverage can have a large influence on the model.

It turns out that two things are key for an M-estimator to be robust:

- The features X_i are bounded.
- The loss function is (1)-Lipschitz. That is for all $t, s \in \mathbb{R}$, $|l(t) - l(s)| \leq t - s$ or equivialently $|l'(t)| \leq 1$ for all $t \in \mathbb{R}$.

This will be discussed more later and will be explored on upcoming homework.