DS 102 Data, Inference, and Decisions **Spring 2020**

Lecture 26: Bootstrap

Lecturer: Jacob Steinhardt

Today, we will discuss the bootstrap, a general method for estimating standard errors, confidence intervals, and other measures of uncertainty. Other widespread approaches to estimating uncertainty like chi-square or Student's t-tests require deriving specific, algebraic formulas for each setting. In some cases, such as the sample mean, deriving these closed-form expressions for the standard error is not too challenging. However, in more complicated cases, we may not be able to write down a simple formula for the uncertainty in our statistic(s) of interest. The bootstrap is a single, unified approach for approximating these measures of uncertainty using computer simulation.

26.1 Problem Setting

We will begin by understanding the basic problem setting in which we might choose to apply the bootstrap.

Suppose we have some dataset consisting of n elements, $x_1, \dots x_n$. We also have some statistic or estimator $\hat{\theta}$ of interest which is a function of our data: $\hat{\theta} = \hat{\theta}(x_1, \dots, x_n)$. Let θ^* denote the true population parameter. Often, our goal is to understand how close $\hat{\theta}$ is to θ^*—for example, by coming up with a confidence region S so that $\theta^* \in S$ with probability $1 - p$.

Example 26.1. We might choose to use the bootstrap in a variety of settings, including:

- **Estimating the mean of a 1D distribution.** In this case, $x_1, \dots, x_n \in \mathbb{R}$ and $\hat{\theta}(x_1, \dots, x_n) = \frac{1}{n}(x_1 + \cdots + x_n)$. We might ask: How close is this empirical mean to the true mean? Can we generate good confidence intervals here?
- **Linear Regression.** Here, our data is $(x_1, y_1), \dots, (x_n, y_n)$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. We are interested in fitting the weights of our model, so that $\hat{\theta}(x_1, y_1, \dots, x_n, y_n) = \arg\min_w \sum_{i=1}^n (y_i - w^\top x_i)^2$. We might ask: How close is $\hat{\theta}$ to the true parameters or the population limit?
- **Mixture models.** Suppose we believe our data $x_1, \dots, x_n$ comes from a mixture of Gaussians. If we assume there are two mixture components, we might ask "how well have we estimated both means?" Alternatively, we might try to estimate the number of mixture components from our data—thereby seeking to answer "how many mixture components are there?"

26.2 The Bootstrap Algorithm

In general, we think of our data $x_1, \ldots, x_n$ are independent samples from some population distribution p^*, and think about the population parameters as the infinite sample limit:

$$\theta^* = \lim_{n\to\infty} \hat{\theta}(x_1, \ldots, x_n);\ x_i \overset{\text{iid}}{\sim} p^*.$$

Then, conceptually, we can think about the noise in our estimate $\hat{\theta}$) in the following way: Suppose that in addition to computing $\hat{\theta}(x_1, \ldots, x_n)$ on our sample, we were provided with some independent re-sample from p^*, $x'_1, \ldots, x'_n$, on which we could compute $\hat{\theta}(x'_1, \ldots, x'_n)$. We could imagine repeating this for several different re-samples, yielding a collection of different values for our estimator $\hat{\theta}$. We can think of this collection of values as an independent sample from the distribution over $\hat{\theta}$ and use it to estimate the uncertainty in the estimator $\hat{\theta}$ computed on n samples.

In reality, we do not actually have access to several independent re-samples from p^*, and therefore would like some way to approximate these re-samples using our one datasets. The solution taken by the bootstrap method is to approximate hypothetical actual re-samples by subsampling n points *with replacement* from the dataset that we have. The bootstrap algorithm creates many subsampled datasets and computes $\hat{\theta}$ on each, resulting in a collection of values for our estimator $\hat{\theta}$ that can be used to estimate uncertainty in $\hat{\theta}$—for example, by computing the empirical variance.

Bootstrap

Given data $x_1, \ldots, x_n$.
$B =$ number of bootstrap samples.

For $b = 1, \ldots, B$:

1. Sample $x'_1, \ldots, x'_n$ with replacement from $x_1, \ldots, x_n$.
2. Compute $\hat{\theta}^{(b)} = \hat{\theta}(x'_1, \ldots, x'_n)$

Output $\{\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)}\}$.

Example 26.2. We can use the bootstrap algorithm to estimate the mean of a 1D geometric distribution from samples. Even with only $n = 15$ data points, we are able to estimate the mean fairly well based on $B = 1000$ bootstrap samples:

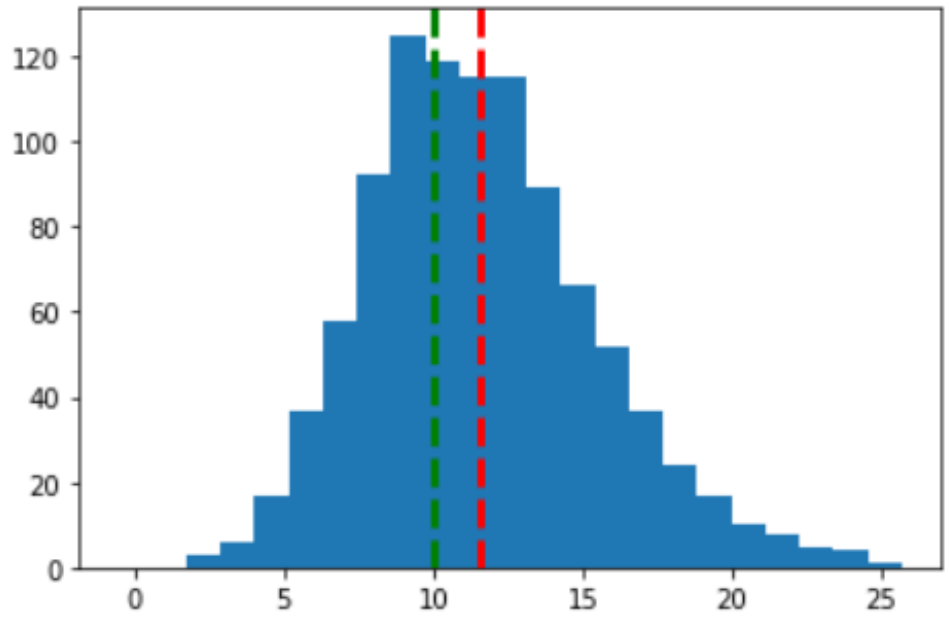

Figure 26.1: Histogram of $B = 1000$ bootstrap estimates of the mean of a 1D geometric distribution. The red line indicates the empirical mean on the original sample of size $n = 15$, and the green line indicates the true mean of the distribution.

26.3 Conditions for the Bootstrap to Work

Although the bootstrap is an extremely general approach to estimating uncertainty, we would nevertheless like to understand the conditions under which it works the best.

First, note that while $\hat{\theta}$ does not need to be unbiased, the bootstrap works best when $\hat{\theta}$ is "smooth." For instance, bootstrap estimates for the median or for logistic regression improve more slowly with the amount of data n than those for, say, the mean. One intuition for why this is true is that these estimators are less "smooth" in that their values are dominated by a few points in the dataset.

In general, the bootstrap works well for most parametric estimators. **Parametric** means that there is a fixed number of parameters which is small compared to the amount of data, n. Some examples of **nonparametric** things include decision trees, neural networks (since the number of model parameters is usually on the same order as the data), and kernels. Nonparametric methods are generally able to fit the data perfectly, and thus sampling with replacement ends up being approximately the same as subsampling the data *without replacement* which can hinder the performance of

the bootstrap. Note that by "works well" we mean that the error in the bootstrap estimate is lower order than the estimate itself. For example, under reasonable assumptions, the standard deviation tends to be $O(1/\sqrt{n})$ and the bootstrap error on the standard deviation estimate is $O(1/n)$. This means that the error on the bootstrap estimate goes down faster than the estimate itself goes down, and as n gets larger the bootstrap estimate gets better.