STATS 230: Computational Statistics
Bootstrap

Babak Shahbaba

Department of Statistics, UCI

Introduction

- In this lecture, we discuss bootstrap as a computational method to quantify uncertainty when it is difficult or impossible to do so analytically
- We fist discuss the main idea behind bootstrap, and explain parametric and nonparametric approaches
- Next, for inferential problems we show how bootstrap can be used to find confidence intervals
- Finally, for predictive models we show the application of bootstrap for evaluating performance
- For more details, refer to " An Introduction to the Bootstrap" by Efron and Tibshirani (1998)
- You should also read Chapter 9 from Givens and Hoeting (2013)

Bootstrap

- Performing statistical inference typically involves focusing on a feature of the underlying distribution of data
- This is usually expressed as a functional of the distribution function, F,

$$\theta = t(F)$$

for example, $\theta = \int x dF$, is the mean of the distribution

- Given a sample of size n, $\mathbf{X} = (X_1, \ldots, X_n)$, where

$$X_1, \ldots, X_n \sim F$$

our inference (within the frequentist framework) is then based on a statistical function $s(\mathbf{X})$ of the data, for example, $\overline{X}$, and its corresponding distribution

Bootstrap

- Given a set of observations $\mathbf{x} = (x_1, \ldots, x_n)$, i.e., a realization of $\mathbf{X}$, we can find a point estimate $\hat{\theta} = s(\mathbf{x})$; however, we need to quantify our uncertainty (e.g., standard error)
- Finding the distribution of $s(\mathbf{X})$ analytically might be difficult or impossible (e.g., if F is unknown)
- To address this issue, we can use bootstrap to approximate the distribution of $s(\mathbf{X})$

Nonparametric Bootstrap

- Given a set of observations, we can estimate F using the empirical distribution function $\hat{F}$, which puts probability $1/n$ on each observed data point
- We could then consider specifically the "plug-in" estimate $\hat{\theta} = t(\hat{F})$ as a point estimate for θ, but in what follows we consider general estimators of the form $s(\mathbf{X})$
- To account for the uncertainty of the estimator, we could approximate the distribution of $s(\mathbf{X})$ by that of $s(\mathbf{X}^*)$, where $\mathbf{X}^* \sim \hat{F}$
- For small datasets, we might be able to tabulate all possible values of $\mathbf{X}^*$, apply the function s, and find its distribution
- We then approximate the distribution of $s(\mathbf{X})$ by the distribution of $s(\mathbf{X}^*)$
- In general, however, we rely on Monte Carlo simulations

Nonparametric Bootstrap

- We simulate "bootstrap" samples, $\mathbf{x}^*$ from $\hat{F}$
- Each bootstrap sample is generated by sampling n iid data points from the observed data with probability $1/n$ (i.e., with replacement): $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$
- That is, we are treating the observed data as a pseudo population from which we generate pseudo datasets $\mathbf{x}^*$
- These bootstrap samples are regarded as realizations of the random variable $\mathbf{X}^* \sim \hat{F}$

Nonparametric Bootstrap

Algorithm 6.1 in Efron and Tibshirani (1998)

Select B (usually 25-200) independent bootstrap samples $\mathbf{x}^{*(1)}, \ldots, \mathbf{x}^{*(B)}$, where each sample includes n data points drawn with replacement from $\mathbf{x}$

For each bootstrap sample, evaluate the corresponding *bootstrap replication*,

$$\hat{\theta}^{*(b)} = s(\mathbf{x}^{*(b)}), \qquad b = 1, \ldots, B$$

We use $\hat{\theta}^{*(1)}, \ldots, \hat{\theta}^{*(B)}$ to approximate the sampling distribution of $\hat{\theta}$; for example, we use the standard deviation of the B bootstrap replications to estimate the standard error of $\hat{\theta}$

$$\hat{\text{se}}_B = \{\sum_{b=1}^{B} (\hat{\theta}^{*(b)} - \overline{\theta}^*)^2/(B-1)\}^{1/2}$$

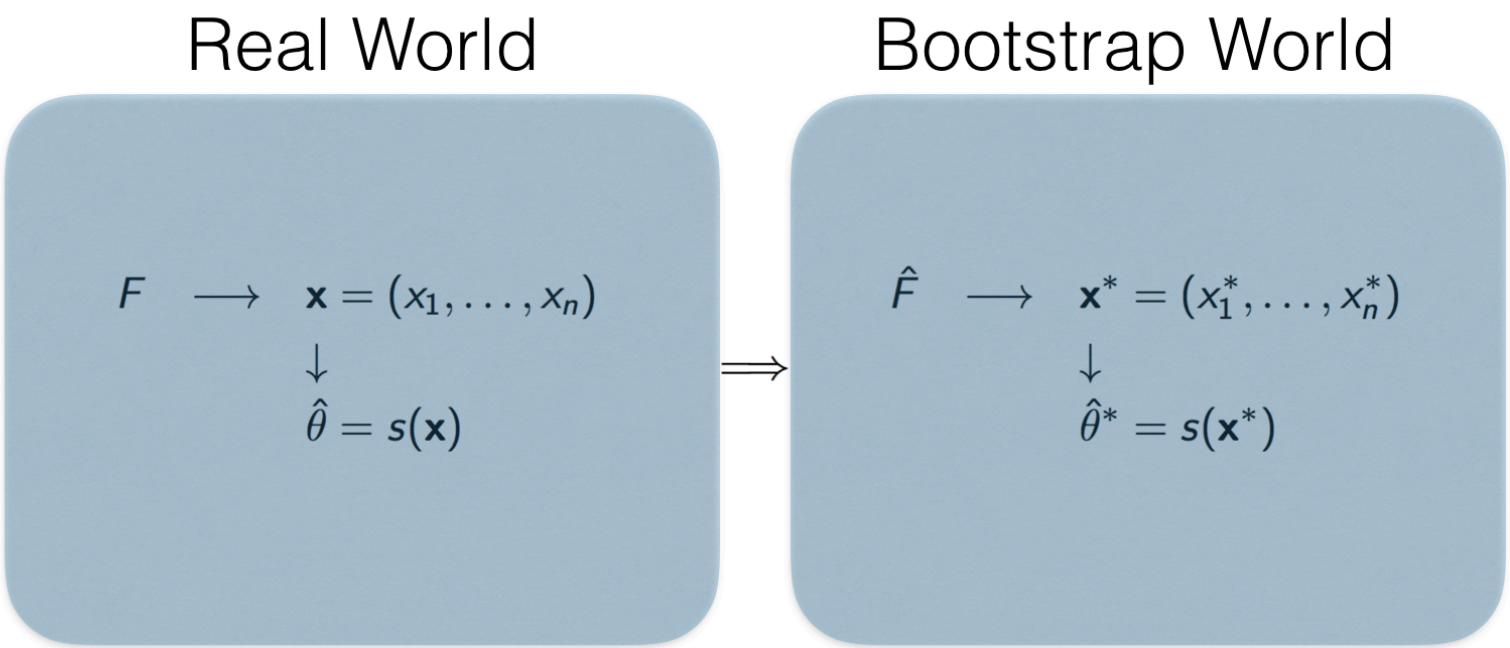
where $\overline{\theta}^* = \sum_{b=1}^{B} \hat{\theta}^{*(b)}/B$

Parametric Bootstrap

- Sometimes, we assume a parametric distribution $F(x, \eta)$ for the data
- In such cases, we can use the observed data to estimate the model parameters (e.g., using method of moment or MLE), then simulate the bootstrap samples from $F(x, \hat{\eta})$
- For example, if we assume $X \sim N(\mu, \sigma^2)$, then we simulate bootstrap samples from $N(\overline{x}, S^2)$
- The remaining steps are as before
- Like any other parametric method, the quality of inference will be negatively affected if the parametric assumptions are wrong, i.e, does not properly represent the underlying mechanism of the data generating process

Bootstrap world vs. real world

In general, we are trying to replicate the real world in the bootstrap world

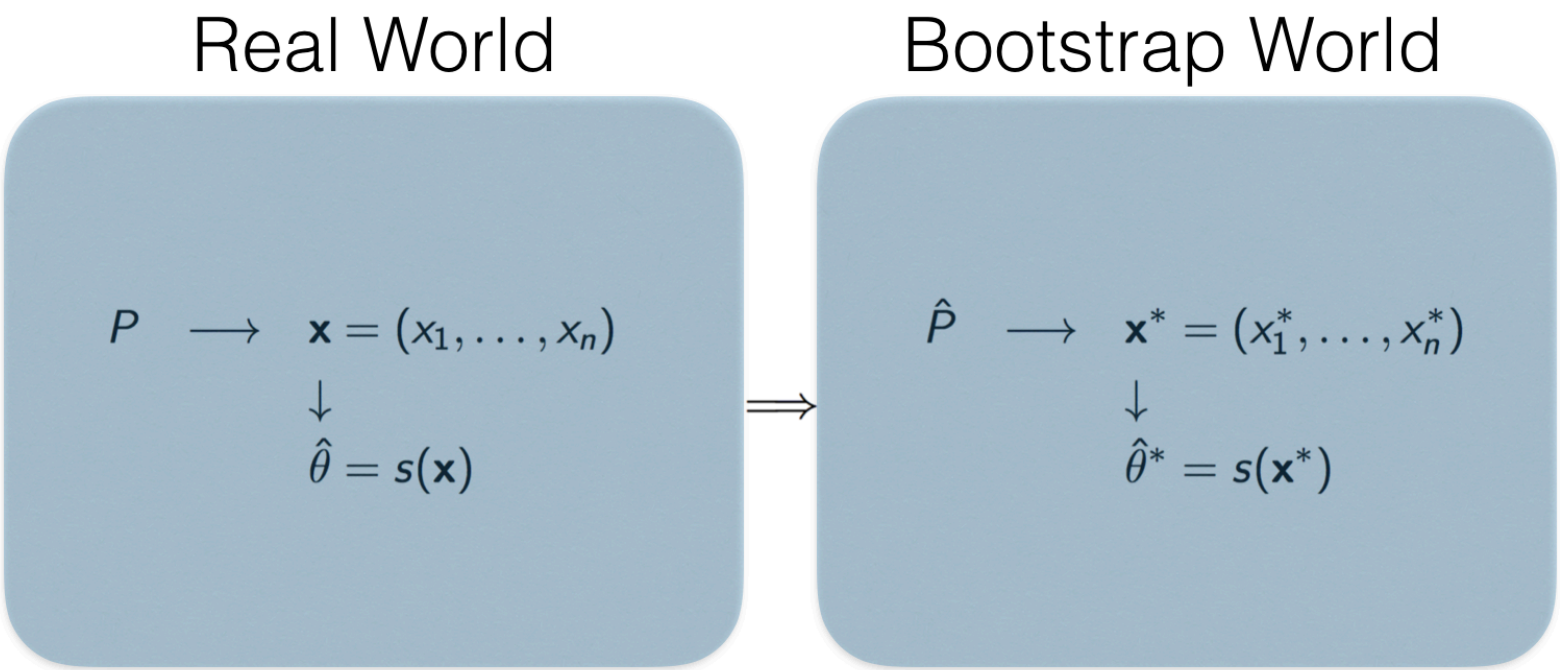


Based on Figure 8.1 in Efron and Tibshirani (1998)

Bootstrap with complex data structure

- Note that in general, each data point could be a vector so F puts $1/n$ probability on each observed vector
- In this case, when we draw a bootstrap sample, we select the whole vector
- Also, in general, data might have a complex underlying structure (e.g., repeated measurements)
- In such cases we should make sure that the bootstrap world mimics the real world
- That is, we assume that there is an unknown probability mechanism, P, responsible for the observed data and use an estimate $\hat{P}$ to generate bootstrap samples

Bootstrap with complex data structure



Based on Figure 8.3 in Efron and Tibshirani (1998)

Bootstrap for regression models

- For regression models, we observe a set of covariates $\mathbf{z}_i = (z_1, \ldots, z_p)$ and a corresponding response variable y_i
- Therefore, each observation can be considered as a vector $\mathbf{x}_i = (\mathbf{z}_i, y_i)$
- To preserve the underlying structure of data, each bootstrap sample is a vector $\mathbf{x}^* = (\mathbf{z}, y)^*$ drawn with replacement from the observed data
- We then fit the assumed regression model to the bootstrap sample to obtain bootstrap replications of regression coefficients: $\boldsymbol{\beta}^*$

Bootstrap for regression models

- Alternatively, we can consider the probability model $P = (\boldsymbol{\beta}, F)$, where $\boldsymbol{\beta}$ is the vector of regression coefficients and F is the probability distribution of the error term
- We can then use the approximate $\hat{P} = (\hat{\boldsymbol{\beta}}, \hat{F})$ to generate bootstrap samples
- The following algorithm shows this approach for linear regression models
- Similar approach can be used for other regression models in general

Bootstrap for linear regression models

Use the observed data to find the least square estimate $\hat{\boldsymbol{\beta}}$

Find the residuals (approximate errors)

$$e_i = y_i - \mathbf{z}_i \hat{\boldsymbol{\beta}} \qquad i = 1, \ldots, n$$

Select B (usually 25-200) independent bootstrap samples $\mathbf{e}^{*(1)}, \ldots, \mathbf{e}^{*(B)}$, where each sample includes n draws with replacement from $\mathbf{e} = (e_1, \ldots, e_n)$

For each bootstrap sample, we then find the corresponding vector of response variable,

$$\mathbf{y}^{*(b)} = \mathbf{z}\hat{\boldsymbol{\beta}} + \mathbf{e}^{*(b)}, \qquad b = 1, \ldots, B$$

For each bootstrap sample, evaluate the corresponding *bootstrap replication*,

$$\hat{\boldsymbol{\beta}}^{*(b)} = (\mathbf{z}^\top \mathbf{z})^{-1} \mathbf{z}^\top \mathbf{y}^{*(b)}, \qquad b = 1, \ldots, B$$

Use the standard deviation of the B bootstrap replications to estimate the standard error of $\hat{\boldsymbol{\beta}}$ as before

Bootstrap for regression models

- Note that in the above algorithm we kept $\mathbf{z}$ fixed over bootstrap samples since in regression models the covariates are assumed to be fixed
- Compared to bootstrapping vectors, bootstrapping residuals is more sensitive to our modeling assumptions
- Also, note that in this case we don't need Monte Carlo simulations to find the bootstrap standard error since it has a closed form which is the same as the usual estimate of standard error
- However, we can use a similar approach for more complex regression models where there is no closed form or it is difficult to find

Bias estimation

- For the parameter of interest $\theta = t(F)$, we define the bias of an estimator $s(\mathbf{X})$ as follows

$$\text{bias}_F = E_F[s(\mathbf{X})] - t(F)$$

- We can use the bootstrap to estimate the bias of the estimator,

$$\text{bias}_{\hat{F}} = E_{\hat{F}}[s(\mathbf{X}^*)] - t(\hat{F})$$

  that is, the bootstrap estimate of bias, is the plug-in estimate of the bias

- Note that our estimate $\hat{\theta} = s(\mathbf{x})$ may not be the same as the plug-in estimate $t(\hat{F})$; our estimate of bias is a plug-in estimate regardless

- In practice, we approximate $\text{bias}_{\hat{F}}$ using the Monte Carlo simulation

$$\widehat{\text{bias}}_B = \overline{\theta}^* - t(\hat{F})$$

Bias estimation

- After we estimate the bias, we could use it to obtain a new bias-corrected estimator

$$\tilde{\theta} \quad = \quad \hat{\theta} - \widehat{\text{bias}}_B$$

- Using $\widehat{\text{bias}}_B = \overline{\theta}^* - \hat{\theta}$, we have

$$\tilde{\theta} \quad = \quad 2\hat{\theta} - \overline{\theta}^*$$

- However, this could be dangerous since $\widehat{\text{bias}}_B$ might have high variability resulting in relatively high standard error for $\tilde{\theta}$

Confidence interval

- Denote the cumulative distribution function of $\hat{\theta}^*$ as $\hat{G}$
- The $1-2\alpha$ percentile interval of the estimator $\hat{\theta}$ can be obtained from the α and $1-\alpha$ percentiles of $\hat{G}$

$$[\hat{\theta}_\ell, \hat{\theta}_u] = [\hat{G}^{-1}(\alpha), \hat{G}^{-1}(1-\alpha)] = [\hat{\theta}^{*(\alpha)}, \hat{\theta}^{*(1-\alpha)}]$$

- In practice, we order the B bootstrap replications and use the $(B\alpha)$th and $B(1-\alpha)$th values to approximate the confidence interval
- The percentile intervals are transformation-preserving for any monotone transformation $\phi = m(\theta)$

Prediction error

- We typically evaluate predictive models based on their prediction error presented as the expectation of an assumed loss function, L,

$$\text{Err} = E[L(y, \hat{y})]$$

  where y is the observed value of the response variable, and $\hat{y}$

- For regression models, we usually set $L(y, \hat{y}) = (y - \hat{y})^2$

- For classification models, we usually set $L(y, \hat{y}) = 1$ when $y \neq \hat{y}$, and zero otherwise; this is called the 0-1 loss function

Prediction error

- We use the observed data to estimate error
- Building a predictive model based on the observed data, $\mathbf{x}$, and evaluating it based on the same data will provide optimistic estimates of prediction error
- The optimistic prediction error on the the training data set itself is called the "apparent error", which we denote as $\mathrm{Err}(\mathbf{x}, \hat{F})$
- To avoid this issue, we usually use an independent test set to estimate prediction error
- If the sample size is large enough, we can divide the observed data into two independent training and test sets

Cross-validation

- When the sample size is relatively small, we recycle the data using K-fold cross-validation (CV)
  - Split the data into K roughly equal parts
  - For $k = 1, \ldots, K$, treat the kth part as the test set to evaluate the model trained on the remaining $K - 1$ parts
  - Obtain the CV estimate of prediction error as

$$\widehat{\mathrm{Err}}_{CV} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

    where $\hat{y}_i$ is the prediction using the parts that do not include the ith observation

- Setting $K = n$ is called "leave-one-out"

Bootstrap estimate of prediction error

- We can use bootstrap to estimate prediction error
- For this,
  - We could simply create B bootstrap samples
  - Use each bootstrap sample to train the predictive model
  - Evaluate the resulting B models based on the original data to obtain B estimates of prediction error: $\mathrm{Err}(\mathbf{x}^*, \hat{F})$
  - Use their average as our bootstrap estimate
- The above approach, however, doesn't work well

Bootstrap estimate of prediction error

- A more refined estimate of prediction error is based on using bootstrap to estimate the amount of "optimism"
- We can obtain an estimate of optimism as follows
  - We evaluate each of the B model based on its corresponding bootstrap sample: $\mathrm{Err}(\mathbf{x}^*, \hat{F}^*)$
  - For each bootstrap sample, find the difference $o^{(b)} = \mathrm{Err}(\mathbf{x}^*, \hat{F}) - \mathrm{Err}(\mathbf{x}^*, \hat{F}^*)$
  - The average of these differences provides an estimate of optimism:

$$\hat{o}(\hat{F}) = \frac{1}{B} \sum_{b=1}^{B} o^{(b)}$$

- Our refined estimate of prediction error is the sum of the apparent error (i.e., prediction error on the training dataset) and the above estimate of optimism

$$\mathrm{Err}(\mathbf{x}, \hat{F}) + \hat{o}(\hat{F})$$

The .632 bootstrap estimator

- We can design a better bootstrap estimation of error by following cross-validation
- To this end, for each observations we use the predictions from bootstrap samples not containing that observations
- For leave-one-out bootstrap estimate, we have

$$\widehat{\mathrm{Err}}^{(1)} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} L(y_i, \hat{y}_i^{*(b)})$$

  where C^{-i} is the set of indices of bootstrap samples not including i
- We remove all the terms with $|C^{-i}| = 0$; If possible, we can use a large enough B to avoid such cases

The .632 bootstrap estimator

- The above estimate would be upward biased
- The .632 estimator address this issue by using a point between this estimate and apparent error

$$\widehat{\mathrm{Err}}_{.632} = .368\mathrm{Err}(\mathbf{x}, \hat{F}) + .632\widehat{\mathrm{Err}}^{(1)}$$

- Roughly, 0.632 is the average proportion of unique observations appearing in each bootstrap sample
- See Efron and Tibshirani (1998) for more details