Bootstrap and Resampling Methods

Lecture 11: Bootstrap in Statistical Learning

Elia Lapenta (ENSAE)

Introduction

Overview of today's class

- To learn an unknown function we often need to select some **tuning parameters**
  - **The bootstrap** can be used for this task
- **Boosting**: method that can reduce the "bias" of a weak learner
  - We will explore the connection between the **Wild Bootstrap** and the L_2-boosting for *linear* estimators
- **Bagging**: method based on the bootstrap to stabilize a high-variance estimator

Throughout this class we will focus on a prototype estimation method: **Smoothing Splines**

Overview of Smoothing Spline Estimation

Smoothing Spline Estimation

We observe $\{Y_i, X_i\}_{i=1}^n$ with $X_i \in [0, 1]$. The model is

$$Y_i = f_0(X_i) + \varepsilon_i \text{ with } \mathbb{E}\{\varepsilon|X\} = 0$$

We want to estimate f_0 **nonparametrically**, i.e. without assuming that f has a specific functional form (e.g., it is linear in X). We just assume that f_0 is a *smooth* function.

The **Smoothing Spline** method estimates f as

$$\widehat{f}_\lambda := \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n [Y_i - f(X_i)]^2 + \lambda \int_0^1 \left| f^{(2)}(x) \right|^2 dx$$

where λ is a penalization parameter, $f^{(2)}$ is the second derivative of f, and

$$\mathcal{F} := \text{ Class of functions that are twice continuously differentiable}$$

Smoothing Spline Estimation: The Role of λ

$$\widehat{f}_\lambda := \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} [Y_i - f(X_i)]^2 + \lambda \int_0^1 \left| f^{(2)}(x) \right|^2 dx$$

λ is a **penalization parameter**

- The larger λ the higher the penalty from *variations* of f
- $\lambda = 0 \Rightarrow$ no penalty, and we will have $Y_i = \widehat{f}_\lambda(X_i)$ (**overfitting**)
- $\lambda = \infty \Rightarrow \widehat{f}_\lambda$ is a linear function

λ avoids overfitting **but** introduces a **regularization bias**: **How do we select λ ?**

Selection of λ by Bootstrap

An Ideal Objective Function

Given the available data $\{Y_i, X_i\}_{i=1}^n$ we define the L_2 risk as

$$\widehat{R}(\lambda) := \mathbb{E}_X \left[f_0(X) - \widehat{f}_\lambda(X)\right]^2 + \sigma_\varepsilon^2 = \mathbb{E}_{Y,X} \left[Y - \widehat{f}_\lambda(X)\right]^2$$

where $\mathbb{E}_X$ considers as random **only** X but **not** $\widehat{f}_\lambda$, $\mathbb{E}_{Y,X}$ considers as random only (Y, X) but not $\widehat{f}_\lambda$, and σ_ε^2 is the variance of ε.

- Notice that $\widehat{R}(\lambda)$ is random: it depends on the data because $\widehat{f}_\lambda$ depends on $\{Y_i, X_i\}_{i=1}^n$

Our **ideal** selection of λ would minimize the expectation of the L_2 risk:

$$\mathbb{E}_{sample}\, \widehat{R}(\lambda)$$

where $\mathbb{E}_{sample}$ is the expectation with respect to the sample $\{Y_i, X_i\}_{i=1}^n$

The Bootstrap principle will help us creating an estimator of $\mathbb{E}_{sample}\widehat{R}$

Bootstrap Counterpart of the L_2 Risk

$$\mathbb{E}_{sample}\,\widehat{R}(\lambda) = \mathbb{E}_{sample}\,\mathbb{E}_{Y,X}\left[Y - \widehat{f}_\lambda(X)\right]^2$$

- $(Y, X) \sim P^{Y,X}$, $(Y_i, X_i) \sim$ iid $P^{Y,X}$, and $\widehat{f}_\lambda$ depends on the sample data $\{Y_i, X_i\}_{i=1}^n$
- By the bootstrap principle we replace the population with our sample $\{Y_i, X_i\}_{i=1}^n \Rightarrow$ $P^{Y,X}$ is replaced by

$$\mathbb{P}_n(A) = \frac{1}{n}\sum_{i=1}^n 1\{(Y_i, X_i) \in A\}$$

Hence: $(Y^*, X^*) \sim \mathbb{P}_n$, $\quad (Y_i^*, X_i^*) \sim$ iid $\mathbb{P}_n$, $\quad \widehat{f}_\lambda^*$ depends on $\{Y_i^*, X_i^*\}_{i=1}^n$

$$\mathbb{E}^*_{sample}\mathbb{E}^*_{Y^*,X^*}\left[Y^* - \widehat{f}^*_\lambda(X^*)\right]^2 \text{ where } \mathbb{E}^*_{Y^*,X^*}\left[Y^* - \widehat{f}^*_\lambda(X^*)\right]^2 = \frac{1}{n}\sum_{i=1}^n\left[Y_i - \widehat{f}^*_\lambda(X_i)\right]^2$$

Bootstrap Counterpart of the L_2 Risk , ct'ed

$$\mathbb{E}^*_{sample}\,\mathbb{E}^*_{Y^*,X^*}\left[Y^* - \widehat{f}^*_\lambda(X^*)\right]^2 = \mathbb{E}^*_{sample}\frac{1}{n}\sum_{i=1}^{n}\left[Y_i - \widehat{f}^*_\lambda(X_i)\right]^2$$

with $\widehat{f}^*_\lambda$ that depends on $\{Y^*_i, X^*_i\}_{i=1}^n$ and $\mathbb{E}^*_{sample}$ expectation considering $\{Y^*_i, X^*_i\}_{i=1}^n$ iid drawn with replacement from $\{Y_i, X_i\}_{i=1}^n$.

We approximate $\mathbb{E}^*_{sample}$ by the Monte-Carlo Algorithm:

- Extract with replacement n observations from $\{Y_i, X_i\}_{i=1}^n$ to get $\{Y^*_i, X^*_i\}_{i=1}^n$
- Compute $\widehat{f}^*_\lambda$ and $\widehat{R}^*_b(\lambda) = (1/n)\sum_{i=1}^n \left[Y_i - \widehat{f}^*_\lambda(X_i)\right]^2$
- Repeat the above steps B times to get $\{\widehat{R}^*_b(\lambda) : b = 1, \ldots, B\}$ and approximate

$$\mathbb{E}^*_{sample}\,\mathbb{E}^*_{Y^*,X^*}\left[Y^* - \widehat{f}^*_\lambda(X^*)\right]^2 \approx \frac{1}{B}\sum_{b=1}^{B}\widehat{R}^*_b(\lambda)\,. \quad \text{So} \quad \lambda^* = \arg\min_\lambda \frac{1}{B}\sum_{b=1}^{B}\widehat{R}^*_b(\lambda)$$

L_2 boosting and Wild Bootstrap

L_2 Boosting and Smoothing Splines

L_2 boosting is a method used to reduce the bias of an estimator. It does so by extracting information from the residuals in an iterative fashion.

$$\widehat{f}^Y := \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} [Y_i - f(X_i)]^2 + \lambda \int_0^1 \left| f^{(2)}(x) \right|^2 dx$$

L_2 boosting algorithm

Initial Compute $\widehat{f}^Y$ (weak learner) and set $\widehat{f}_0 = \widehat{f}^Y$.

For $m = 1, \ldots, M$

Step 1 Compute the residuals $\widehat{\varepsilon}_i = Y_i - \widehat{f}_{m-1}(X_i)$

Step 2 Compute $\widehat{f}^{\widehat{\varepsilon}}$ by minimizing the above objective function where Y_i is replaced by $\widehat{\varepsilon}_i$.

Step 3 Update $\widehat{f}_m = \widehat{f}_{m-1} + \widehat{f}^{\widehat{\varepsilon}}$

L_2 boosting and the Wild Bootstrap

L_2 boosting can be seen as a **bias correction** based on the **Wild Bootstrap**

- First define $\widehat{f}^n := (\widehat{f}(X_1), \ldots, \widehat{f}(X_n))^T$. From the theory of smoothing splines (*Reproducing Kernel Hilbert Spaces/Support Vector Machines*) we know that

$$\widehat{f}^n = \mathcal{S}Y^n$$

  where $Y^n := (Y_1, \ldots, Y_n)^T$ and $\mathcal{S}$ is a matrix that depends only on the data $\{X_i\}_{i=1}^n$
- So, the 1st iteration in the L_2 boosting is

$$\widehat{f}_1^n = \mathcal{S}Y^n + \mathcal{S}(Y^n - \mathcal{S}Y^n) = \mathcal{S}Y^n + \mathcal{S}(I - \mathcal{S})Y^n$$

L_2 boosting and the Wild Bootstrap , ct'ed

- Now, generate an artificial sample by the **Wild Bootstrap**
$$Y_i^* = \widehat{f}(X_i) + \xi_i \widehat{\varepsilon}_i$$
where $\widehat{\varepsilon}_i = Y_i - \widehat{f}(X_i)$ and $\{\xi_i\}_{i=1}^n$ are iid bootstrap weights independent from the sample with $\mathbb{E}\xi = 0$ and $\mathbb{E}\xi^2 = 1$. In vector form
$$Y^{*n} = \widehat{f}^n + (\xi\widehat{\varepsilon})^n$$
with $Y^{*n} = (Y_1^*, \dots, Y_n^*)^T$ and $(\xi\widehat{\varepsilon})^n = (\xi_1\widehat{\varepsilon}_1, \dots, \xi_n\widehat{\varepsilon}_n)^T$.
- Since $\widehat{f}^n = \mathcal{S}Y^n$, the smoothing spline estimator in the bootstrap world is
$$\widehat{f}^{*n} = \mathcal{S}Y^{*n} = \mathcal{S}\widehat{f}^n + \mathcal{S}(\xi\widehat{\varepsilon})^n = \mathcal{S}\mathcal{S}Y^n + \mathcal{S}(\xi\widehat{\varepsilon})^n$$
- So, the **Bias in the bootstrap world** is
$$E_\xi^* \widehat{f}^{*n} - \widehat{f}^n = \mathcal{S}\mathcal{S}Y^n - \mathcal{S}Y^n = -\mathcal{S}(I - \mathcal{S})Y^n$$
because $E_\xi^* \mathcal{S}(\xi\widehat{\varepsilon})^n = 0$ (as $E_\xi^*\xi = 0$).

L_2 boosting and the Wild Bootstrap , ct'ed

- We can use the **bias in the bootstrap world** to estimate $Bias(\widehat{f}^n)$. So, the *bias corrected* estimator is

$$\widehat{f}^n - \widehat{Bias}(\widehat{f}^n) = \mathcal{S}Y^n + \mathcal{S}(I - \mathcal{S})Y^n$$

  which corresponds to the **boosted estimator after one boosting step**.

- So, the **boosted estimator** after one boosting iterations corresponds to a **bootstrap bias-corrected estimator** where the bias is estimated by the Wild bootstrap

- We can also generalize this to **any number of boosting steps**!

Note that the above procedure is valid for any type of estimator that is linear in Y^n. Such a correspondence does not hold for estimators that are not linear in Y^n (e.g., neural nets)

The Role of M (=number of iterations)

Idea behind the L_2 Boosting:

1. Extract information from the residuals (i.e. estimate the model that considers the residuals as a response variable)
2. Add this piece of information to the "initial" estimate

How does the estimator behave when we increase the booting iterations?

- If we iterate too many times we run in **overfitting**!

Let us analyze the Mean Squared Error of $\widehat{f}$ as a function of M. Let $\{\lambda_k : j = 1, \ldots, K\}$ be the eigenvalues of the matrix $\mathcal{S}$. We have $\lambda_k \in [0, 1]$.

Then, (see Buhlmann and Yu , 2003)

$$MSE = Bias^2(\widehat{f}_M) + Var(\widehat{f}_M)$$

with $Bias(\widehat{f}_M) \sim diag((1-\lambda_k)^{2M})$ and $Var(\widehat{f}_M) \sim [1-(1-\lambda_k)^{M+1}]^2$

Bagging

Bagging as a Variance Reduction Technique

Bagging= bootstrap aggregating.

Main Idea: Extract bootstrap samples from the original sample, compute your estimator on each bootstrap sample, and then average out the results.

- Draw with replacement n observations from your sample $\{Y_i, X_i\}_{i=1}^n$ to get $\{Y_i^*, X_i^*\}_{i=1}^n$
- compute the estimator $\widehat{f}^*$ on the bootstrap sample
- Repeat the above steps B times to obtain $\{\widehat{f}_b^* : b = 1 \ldots, B\}$
- Average the results:

$$\widehat{f}_{Bagged} = \frac{1}{B} \sum_{b=1}^{B} \widehat{f}_b^*$$

Why Bagging Works?

If we could, it would be better to use $\mathbb{E}_{sam}\widehat{f}(x)$ instead of $\widehat{f}(x)$.

- In fact, as already noticed

$$\textbf{L_2 risk of } \widehat{f} = \mathbb{E}_{Y,X}\left[f(X)-\widehat{f}(X)\right]^2+\sigma_\varepsilon^2=\mathbb{E}_{Y,X}\left[Y-\widehat{f}(X)\right]^2$$

- Next, notice that

$$\begin{aligned}
\mathbb{E}_{sam}\mathbb{E}_{Y,X}\left[Y-\widehat{f}(X)\right]^2 =& \mathbb{E}_{sam}\mathbb{E}_{Y,X}\left[Y-\mathbb{E}_{sam}\widehat{f}(X)+\mathbb{E}_{sam}\widehat{f}(X)-\widehat{f}(X)\right]^2 \\
=& \mathbb{E}_{sam}\mathbb{E}_{Y,X}\left[Y-\mathbb{E}_{sam}\widehat{f}(X)\right]^2+\mathbb{E}_{sam}\mathbb{E}_{Y,X}\left[\mathbb{E}_{sam}\widehat{f}(X)-f(X)\right]^2 \\
\geq& \mathbb{E}_{Y,X}\left[Y-\mathbb{E}_{sam}\widehat{f}(X)\right]^2 = \textbf{ L_2 risk of } \mathbb{E}_{sample}\widehat{f}
\end{aligned}$$

where the cross-product in the second equality is

$$\mathbb{E}_{sam}\mathbb{E}_{Y,X}\left\{\left[Y-\mathbb{E}_{sam}\widehat{f}(X)\right]\left[\mathbb{E}_{sam}\widehat{f}(X)-\widehat{f}(X)\right]\right\}=0$$

Why Bagging Works? Ct'ed

- So, if we could use $\mathbb{E}_{sam}\widehat{f}(X)$ we would get a smaller risk (L_2 error) than if we used $\widehat{f}(x)$
- In practice we cannot compute $\mathbb{E}_{sam}$, so we approximate it by the bootstrap:
$$\mathbb{E}^*_{sam}\widehat{f}^*(x)$$
where $\mathbb{E}^*_{sam}$ is the expectation that considers each observation in $\{Y_i^*, X_i^*\}_{i=1}$ drawn with replacement from $\mathbb{P}_n$ and $\widehat{f}^*$ is the estimator based on the bootstrapped sample $\{Y_i^*, X_i^*\}$.
- Similarly as in Slide 6, we approximate $\mathbb{E}^*_{sam}\widehat{f}^*(x)$ by the Monte-Carlo algorithm:
$$\mathbb{E}^*_{sam}\widehat{f}^*(x) \approx \frac{1}{B}\sum_{b=1}^{B}\widehat{f}^*_b(x)$$