

Assignment 8

Due June 20th at 12:00

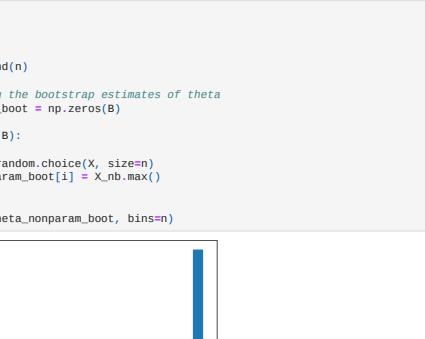
15

Please note:

- Read the instructions in the exercise PDF and in this notebook carefully.
- Add your solutions only at YOUR CODE HERE / YOUR ANSWER HERE, and remove the corresponding `raise NotImplementedError()`.
- Do not change the provided code and text, if not stated.
- Do not add or delete cells.
- Do not import additional functionality.
- Before submitting: Please make sure, that your notebook can be executed from top to bottom. Menu -> Kernel -> Restart & Run all.

Exercise 1 (Bootstrap, 2+1+1+2+3 points)

a)



b)

```
In [1]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

```
In [2]: B=5000
n=25
```

data points

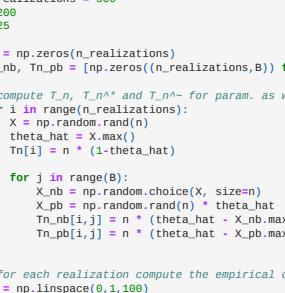
X=np.random.rand(n)

array holding the bootstrap estimates of theta

theta_nonparam_boot = np.zeros(B)

for i in range(B):
 # draw
 X_nb = np.random.choice(X, size=n)
 theta_nonparam_boot[i] = X_nb.max()

_ = plt.hist(theta_nonparam_boot, bins=n)



c)

$$P(X_i \mid \theta) = \begin{cases} \frac{1}{\theta} & \text{if } X_i \geq \theta \\ 0 & \text{else} \end{cases} \quad (1)$$

$$P(X \mid \theta) = \begin{cases} \frac{1}{\theta} & \text{if } X_i \geq \theta \quad \forall i = 1, \dots, n \\ 0 & \text{else} \end{cases} \quad (2)$$

$$= \begin{cases} \frac{1}{\theta} & \text{if } \max_i X_i \geq \theta \\ 0 & \text{else} \end{cases} \quad (3)$$

$$\Rightarrow \hat{\theta}_n = \arg \max_{\theta} P(X \mid \theta) = \max_i X_i \quad (4)$$

$$F_{\hat{\theta}_n}(z) = P(\hat{\theta}_n \leq z) \quad (5)$$

$$= P(\max_i X_i \leq z) \quad (6)$$

$$= \begin{cases} 0 & \text{if } z < \max_i X_i \\ 1 & \text{else} \end{cases} \quad (7)$$

$$\Rightarrow \text{Rewrite as } \hat{P}(X_i \leq z) \quad (8)$$

```
In [3]: B=5000
n=25
```

data points

X=np.random.rand(n)

array holding the bootstrap estimates of theta

theta_nonparam_boot = np.zeros(B)

for i in range(B):
 # draw
 X_nb = np.random.choice(X, size=n)
 theta_nonparam_boot[i] = X_nb.max()

_ = plt.hist(theta_nonparam_boot, bins=n)

3000

2500

2000

1500

1000

500

0

0.725 0.750 0.775 0.800 0.825 0.850 0.875 0.900

d)

$$\begin{aligned} P(T_n^* \leq 0) &= 1 - P(T_n^* \geq 0) \\ &= 1 - P(\theta_n \leq \theta_n) \\ &= 1 - P(\max_i X_i \notin X^*) \\ &= 1 - \prod_j^n P(X_j^* \neq \max_i X_i) \\ &= 1 - \prod_j^n 1 - \frac{1}{n} \\ &= 1 - \left(1 - \frac{1}{n}\right)^n \end{aligned} \quad (1)$$

e)

$$\begin{aligned} \limsup_{n \rightarrow \infty} \sup_{t \in \mathbb{R}} |P(T_n \leq t) - P(T_n^* \leq t)| &\geq 1 - e^{-1} \\ \Leftrightarrow \sup_{t \in \mathbb{R}} |P(T_n \leq t) - P(T_n^* \leq t)| &\geq \left(1 - \frac{1}{n}\right)^n \\ \Leftrightarrow \sup_{t \in \mathbb{R}} |P(T_n \leq t) - P(T_n^* \geq 0)| &\geq P(T_n^* \geq 0) \end{aligned} \quad (2)$$

```
In [3]: n_realizations = 500
B=500
n=25
```

Tn = np.zeros(n_realizations)

Tn_nb, Tn_pb = [np.zeros(n_realizations, B) for _ in range(2)]

compute T_n, T_n^* and T_n^*- for param, as well as nonparam. bootstrap

for i in range(B):
 X_nb = np.random.rand(n)
 theta_hat = X_nb.max()
 Tn[i] = n * (1 - theta_hat)

for i in range(B):
 X_nb = np.random.choice(X, size=n)
 X_pb = np.random.rand(n) * theta_hat
 Tn_nb[i, :] = n * (theta_hat - X_nb.max())
 Tn_pb[i, :] = n * (theta_hat - X_pb.max())

for each realization compute the empirical cumulative distribution function (ecdf) of T_n across realizations and, for each realization, the ecdf of T_n^* and T_n^*- across bootstrap samples

zs = np.linspace(0,1,100)

Fn_nb, Fn_pb = [np.zeros(zs.shape[0], n_realizations) for _ in range(2)]

for i, z in enumerate(zs):
 Fn_nb[i] = 1/n_realizations * np.sum(Tn < z)
 for i in range(n_realizations):
 Fn_nb[i, :] = 1/B * np.sum(Tn_nb[i, :] < z)
 Fn_pb[i, :] = 1/B * np.sum(Tn_pb[i, :] < z)

plot the empirical distribution function with uncertainty bands across realizations

plt.plot(zs, Fn_nb, label = 'Fn')

plt.plot(zs, Fn_nb[zs,:].mean(), label = 'nonparam')

plt.fill_between(zs, [Fn_nb[zs,:].mean() - Fn_nb[zs,:].std(), Fn_nb[zs,:].mean() + Fn_nb[zs,:].std()], for iz in range(zs.shape[0])), alpha = 0.4, color = 'tab:orange')

plt.plot(zs, Fn_pb[zs,:].mean(), label = 'param')

plt.fill_between(zs, [Fn_pb[zs,:].mean() - Fn_pb[zs,:].std(), Fn_pb[zs,:].mean() + Fn_pb[zs,:].std()], for iz in range(zs.shape[0])), alpha = 0.4, color = 'tab:green')

plt.legend()

cm = matplotlib.colors.LinearSegmentedColormap.from_list('my_cmap', [(0, 'red'), (1, 'blue')])

Out[3]: <matplotlib.legend.Legend at 0x7f0fb819db50>

0.8

0.6

0.4

0.2

0.0

0.0 0.2 0.4 0.6 0.8 1.0

Had to think of

Pin Dr as sum

of weights of

misclassified data points.

sloppy notation imo

however that's the

intuition

1.5

for the

effort.

1

1

a)

$$\begin{aligned} Z_t &= \sum_i D_i(t) \exp(-y_i \alpha_t h_i(x_i)) \\ &= \sum_i D_i(t) \exp(-\alpha_t y_i h_i(x_i)) \\ &= \sum_{i: h_i(x_i) = 1} D_i(t) \exp(-\alpha) + \sum_{i: h_i(x_i) = -1} D_i(t) \exp(\alpha) \\ &= \sum_{i: h_i(x_i) = 1} D_i(t) \exp\left(\frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\right) + \sum_{i: h_i(x_i) = -1} D_i(t) \exp\left(\frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\right) \\ &= P_{i \sim D_t}(y_i = h_t(x_i)) \exp\left(-\frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\right) + P_{i \sim D_t}(y_i \neq h_t(x_i)) \exp\left(\frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\right) \\ &= (1 - \epsilon_t) \exp\left(-\frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\right) + \epsilon_t \exp\left(\frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\right) \\ &= (1 - \epsilon_t) \sqrt{\frac{\epsilon_t}{1 - \epsilon_t}} + \epsilon_t \sqrt{\frac{1 - \epsilon_t}{\epsilon_t}} \\ &= 2 \sqrt{\epsilon_t(1 - \epsilon_t)} \end{aligned} \quad (1)$$

Zt = sum(Di(t) * exp(-y * alpha_t * hi(xi)))

= sum(Di(t) * exp(-alpha * y * hi(xi)))

= sum(Di(t) * exp(-alpha_t * y * hi(x)))

= sum(Di(t) * exp(-1/2 * ln((1-epsilon_t)/epsilon_t) * hi(x)))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \neq h_t(x_i))

= 1 - 2 * P_{i \sim D_t}(y_i \ne