BIOSTAT 572

5/15/23

Lecture 15

Jackknife
Bootstrap
Jackknife +

AoNPS
Ch 3

Project - grading guidelines

Bootstrap, Jackknife, J++

estimate bias, Var, CDF of $\theta = T$

estimator

Given • data • $\mathcal{D} = \{x^{1:n}\}$ → Ex: θ = Variance, median, some parameter

$\mathcal{D} = \{(x^{1:n}, y^{1:n})\}$ → θ = prediction of y at x^{n+1} new point

• "estimator" estimation algorithm

$T(F) = \theta$ ← true value

↑ true data distribution

T_n = estimated value from $\mathcal{D}$

Ex: θ = variance $(F) \equiv \sigma^2$

$T_n = \frac{1}{n}\sum_{i=1}^{n}(x^i - \bar{x})^2 = \hat{\sigma}^2$

$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x^i$

T_{-i} = estimated value from $\mathcal{D}_{-i}$

$\mathcal{D} \setminus \{x^i\}$

(or $\mathcal{D} \setminus \{(x^i, y^i)\}$)

$F \xrightarrow{T} T(F) = \theta$

$\mathcal{D} = F_n \longmapsto T_n$

Jackknife estimate bias

$$\bar{T} = \frac{1}{n}\sum_{i=1}^{n} T_{-i}$$

for $i = 1:n$
  calculate $T(\mathcal{D}_{-i}) = T_{-i}$
calculate T

"run" T $n+1$ times
Easy
Efficient for $T = Lx$ or $T = Ly$ linear predictor

$$\text{bias} = E_{F^n}[T_n - \theta] \Rightarrow \theta = E[T_n] - \text{bias}$$

$\widehat{\text{bias}}_J = (n-1)(\bar{T} - T_n) \Rightarrow \hat{T}_J = T_n - \widehat{\text{bias}}_J = T_n - \frac{n-1}{n}\sum T_{-i} + (n-1)T_n =$

$\widehat{\text{bias}}_J$: Estimated by J

"de biased T_n"

$= nT_n - \frac{n-1}{n}\sum T_{-i}$

Why? Assume bias of T_n: $\text{bias} = \frac{a}{n} + \frac{b}{n^2} + O\left(\frac{1}{n^3}\right)$

Ex: $\hat{\sigma}^2$: $\text{bias}\,\hat{\sigma}^2 = \frac{1}{n}\sigma^2$

$$\text{bias}(T_{-i}) = \frac{a}{n-1} + \frac{b}{(n-1)^2} + \cdots$$

$$\text{bias}\,T_{Jack} = E[T_J - \theta] = n\cdot\text{bias}\,T_n - \frac{n-1}{n}\sum \text{bias}(T_{-i}) =$$

$$= n\left(\frac{a}{n} + \frac{b}{n^2} + \cdots\right) - (n-1)\left(\frac{a}{n-1} + \frac{b}{(n-1)^2} + \cdots\right) = a - a + b\left(\frac{1}{n} - \frac{1}{n-1}\right) + \cdots \approx -b/n^2 + \cdots$$

$\frac{a}{n-1} + \frac{b}{(n-1)^2} + \cdots$: bias T_{-i}

Bootstrap: estimates $\mathrm{Var}(T_n)$, $\mathrm{CDF}(T_n)$

Algorithm

number Bootstrap samples

for $b = 1:B$

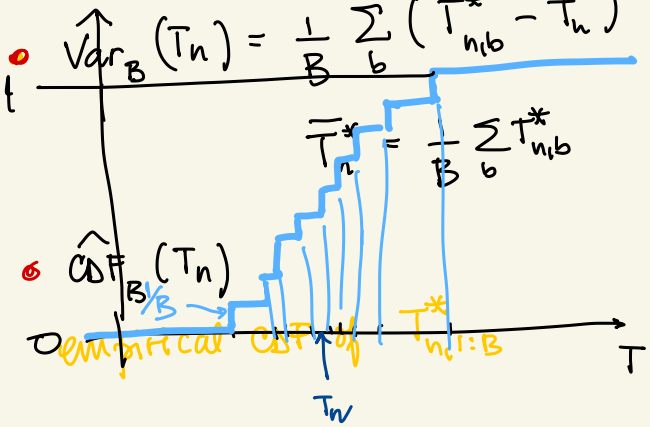
  draw $\mathcal{D}_b^* = X_*^{1:n} \sim \mathcal{D}$ iid with replacement

  $T_{n,b}^* = T(\mathcal{D}_b^*)$

$$\widehat{\mathrm{Var}}_B(T_n) = \frac{1}{B}\sum_b \left(T_{n,b}^* - \bar{T}_n^*\right)^2$$

sample var of $T_{n,b}^*$

$$\bar{T}_n^* = \frac{1}{B}\sum_b T_{n,b}^*$$



- Confidence Intervals

Ex 1: $\theta = E_F[X]$

$T_n = \bar{X} = \frac{1}{n}\sum x_i$

$n = 100$

$B = 50$

$\mathrm{Var}\,\bar{X} = \frac{\sigma^2}{n}$ $\quad \sigma^2 = \mathrm{Var}\,X$

$\mathcal{D} = \{x^1, \dots x^n\}$

→ for $b = 1:B$

sample $\mathcal{D}^*_b$: $x^*_{1:n} \sim \mathcal{D}$ with replacement $\quad (1-\frac{1}{e})n \approx$ distinct samples

compute $\bar{X}^*_{n,b} = \frac{1}{n}\sum x^*_i$ ← Should be weighted mean

$\hat{\mathrm{Var}}_B \bar{X} = \frac{1}{50}\sum \left(x^*_{n,b} - \bar{X}^*_n\right)^2$

Ex: $\theta = y^{n+1}$ at x^{n+1} true prediction

$\mathcal{F}$ = RForest, Neural Net, Nearest Neighbor, NW (kernel regression)

Train: train $T(\mathcal{D}) \to T_n = f_n$ a predictor

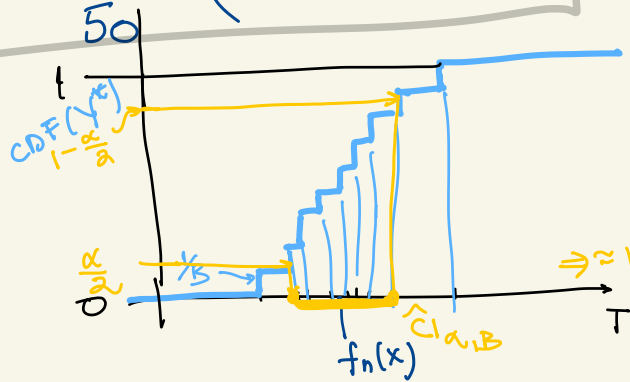
for $b = 1:B$: sample $\mathcal{D}^*_b$, train $T(\mathcal{D}^*_b) \to f^*_{n,b}$

"Test" new x : $\hat{y} = f_n(x)$ prediction

$Y^*(x) = \{y^*_b = f^*_{n,b}(x)\}_{b=1:B}$

$\mathrm{Var}\,\hat{y}$: sample var of $Y^*(x)$

$\hat{CI}_{\alpha,B}(y^*)$: Interquantile $[\frac{\alpha}{2}, 1-\frac{\alpha}{2}]$ of $Y^*(x)$



$\Rightarrow \approx n\frac{\alpha}{2} \times 2$ removed

"Pivotal CI"

Variations

- Parametric Bootstrap
  - If $T_n = \theta$ a parameter $\Rightarrow F_\theta$ model

  Ex: density estimation by Mixture Models

  sample $\mathcal{D}_b^* \sim F_\theta$ iid $\notin \mathcal{D}$
- Normal CI use $\sqrt{\widehat{\text{Var}}_B(T_n)}$ when T_n asymptotically normal
- studentized
- Bag of Little Bootstraps (BLB)

- **Bag of little Bootstraps** [arXiv: 1112.5016]

for large n

Idea use $n' \ll n$, repeat bootstrap K times

for $k = 1:K$ ~ "folds"

- sample $\mathcal{D}_k^*$ of size n' without replacement
  (okay also take a block $\mathcal{D}_k^*$ from $\mathcal{D}$)
- do bootstrap on $\mathcal{D}_k^*$ with sample size n

  for $b = 1:B$

  1. sample $(n_{i,k,b}) \sim \text{multinomial}\left(n, \left[\frac{1}{n'}, \dots \frac{1}{n'}\right]\right)$ for $i \in \mathcal{D}_k^*$ → "$\mathcal{D}_{k,b}^*$" virtual
  2. estimate $T_{n,k,b}^*(\mathcal{D}_{k,b}^*)$ ← fast because only n' distinct samples

- $\bar{T}_{n,k}^* = \frac{1}{B}\sum_b T_{n,k,b}^*$

  estimate Var, CDF, CI (T_n) from $\mathcal{D}_k^*$ ⇒ $\hat{\text{Var}}_{B,k}^*$ (e.g.; random $\mathcal{D}_k^*$; Bootstrap; "fold k")

- $\hat{\text{Var}}_{BLB}(T_n) = \frac{1}{K}\sum_k \hat{\text{Var}}_{Bk}^*$

BLB

Theory $n' = \Omega(\sqrt{n})$, e.g. $n' = n^{\alpha}$, $\alpha > 0.5$

$K \sim \frac{n}{n'}$

B – as usual for Bootstrap e.g. $\approx 50\text{-}100$

Experimentally $K \approx$ constant e.g. 50

Computation savings from fast recomputation of $T^{*}_{n,b,k}$ $\sim n'$

Jackknife + ⊂ Conformal Prediction

Given $\{(x^{1:n}, y^{1:n}) \} = \mathcal{D}$

T training alg $\quad T(\mathcal{D}) = f_n \equiv T_n$

f predictor $\quad T(\mathcal{D}_{-i}) = f_{-i}$

Train compute f_n

—"— f_{-i}, for $i = 1:n$

Test given x want CI for $y(x)$

$f_n(x)$

Compute "LOO residuals"

$$R_i = |y^i - f_{-i}(x^i)|$$