STK1100 våren 2023

Bootstrapping, Stokastisk simulering og Monte Carlo-integrasjon

Svarer til deler av notatet
«Bootstrap og simulering»

Matematisk institutt
Universitetet i Oslo

Estimering

Vi antar at $x_1, x_2, ..., x_n$ er observerte verdier av uavhengige og identisk fordelte (u.i.f.) stokastiske variabler $X_1, X_2, ..., X_n$ og at X_i-ene har en fordeling som avhenger av en parameter θ

Vi vil estimere verdien til θ på grunnlag av observasjonene våre

Til det bruker vi en estimator $\hat{\theta} = \hat{\theta}(X_1, X_2, ..., X_n)$

På grunnlag av de observerte x_i-ene får vi estimatet $\hat{\theta} = \hat{\theta}(x_1, x_2, ..., x_n)$

Standardfeil

Når vi rapporterer resultatet av en undersøkelse, bør vi ikke nøye oss med å oppgi et estimatet.
Vi bør også si noe om hvor presist estimatet er. Det er da vanlig å oppgi (et estimat for) standardavviket

Standardavviket $\sigma_{\hat{\theta}} = \sqrt{V(\hat{\theta})}$ til en estimator $\hat{\theta}$ blir vanligvis kalt standardfeilen til estimatoren

Ofte vil $\sigma_{\hat{\theta}} = \sqrt{V(\hat{\theta})}$ avhenge av en eller flere ukjente parmetere. Hvis vi estimerer disse, får vi den estimerte standardfeilen $s_{\hat{\theta}}$

Bootstrap

For enkle situasjoner kan vi finne et uttrykk for standardfeilen $\sigma_{\hat{\theta}} = \sqrt{V(\hat{\theta})}$ til en estimator

Men hvis estimatoren og/eller fordelingen til X_i-ene er komplisert, kan det være vanskeligere å finne et slikt uttrykk

Da kan vi bruke stokastisk simulering til å finne et estimat $s_{\hat{\theta}}$ for standardfeilen

Vi ser først på en metode som kalles parametrisk bootstrap

Anta at X_i-ene har tetthet/punktsannsynlighet $f(x;\theta)$

Ut fra de observerte x_i-ene får vi estimatet $\hat{\theta}$

For $b = 1, 2, ..., B$ gjør vi nå følgende:

- Genererer et bootstrap-utvalg $x_1^*, x_2^*, ..., x_n^*$ fra tettheten/punktsannsynligheten $f(x;\hat{\theta})$
- Beregner estimatet $\hat{\theta}_b^*$ ut fra bootstrap-utvalget (på samme måte som $\hat{\theta}$ ble beregnet ut fra de opprinnelige observasjonene)

Bootstrap-estimatet for standardfeilen er da

$$s_{\hat{\theta}} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_b^* - \bar{\theta}^*\right)^2}$$

Eksempel (jf. oppgave 8.28)

Vi har observasjonene (antall skritt per sekund):

0.95 0.85 0.92 0.95 0.93 0.86 1.00 0.92 0.85 0.81
0.78 0.93 0.93 1.05 0.93 1.06 1.06 0.96 0.81 0.96

Vi vil anta at disse er observasjoner av $X_1,...,X_{20}$ som er uavhengige og $N(\mu,\sigma^2)$-fordelte , og vi lar $Y_1 < ... < Y_{20}$ være X_i-ene gitt i stigende rekkefølge

Vi ser på tre estimatorer for μ :

- Gjennomsnittet $\hat{\mu}_1 = \frac{1}{20}\sum_{i=1}^{20} X_i$
- Empirisk median $\hat{\mu}_2 = (Y_{10} + Y_{11})/2$
- 10% trimmet gjennomsnitt $\hat{\mu}_3 = \frac{1}{16}\sum_{j=3}^{18} Y_j$

Estimatene blir:

$\hat{\mu}_1 = 0.926$ $\hat{\mu}_2 = 0.930$ $\hat{\mu}_3 = 0.925$

Vi kan bestemme standardfeilene ved bootstrap:

```
import numpy as np
import scipy.stats as stats
import random
import math
x=[0.95,0.85,0.92,0.95,0.93,0.86,1.00,0.92,0.85,0.81,0.78,0.93,0.93,1.05,0.93,1.06,1.06,0.96,0.81,0.96]
mean=np.mean(x)
median=np.median(x)
trmean=stats.trim_mean(x,0.1)

n=len(x)
m=np.mean(x)
s=np.std(x)
B=10000
meanvec=np.zeros(B+1)
medianvec=np.zeros(B+1)
trmeanvec=np.zeros(B+1)
ind=np.arange(0,B)
for i in ind:
    xstar=stats.norm.rvs(size=n,loc=m,scale=s)
    meanvec[i]=np.mean(xstar)
    medianvec[i]=np.median(xstar)
    trmeanvec[i]=stats.trim_mean(xstar,0.1)
```

Vi finner standardfeilene:

$S_{\hat{\mu}_1}=0.0197$ $S_{\hat{\mu}_2}=0.0230$ $S_{\hat{\mu}_3}=0.0202$

Parametrisk bootstrap forusetter at den modellen vi bruker for dataene gir en rimelig god beskrivelse av fordelingen til X_i-ene

For ikke-parametrisk bootstrap gjør vi ingen forutsetninger om fordelingen til X_i-ene

Da trekker vi bootstrap-utvalget fra den empiriske fordelingsfunksjonen

$$\hat{F}(x) = \frac{1}{n}\{\text{antall } x_i \leq x\}$$

Merk at $\hat{F}$ gir sannsynlighet 1/*n* til hver x_i

Trekking fra $\hat{F}$ svarer derfor til trekning fra $x_1, x_2, ..., x_n$ med tilbakelegging

Framgangsmåten for ikke-parametrisk bootstrap er dermed som følger:

For $b = 1, 2, ..., B$:

- Trekk *n* verdier $x_1^*, x_2^*, ..., x_n^*$ med tilbakelegging fra $x_1, x_2, ..., x_n$
- Beregn estimatet $\hat{\theta}_b^*$ ut fra bootstrap-utvalget

Bootstrap-estimatet for standardfeilen er

$$s_{\hat{\theta}} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_b^* - \bar{\theta}^*\right)^2}$$

For eksemplet har vi kommandoene:

```
import numpy as np
import scipy.stats as stats
import random
import math
x=[0.95,0.85,0.92,0.95,0.93,0.86,1.00,0.92,0.85,0.81,0.78,0.93,0.93,1.05,0.93,1.06,1.06,0.96,0.81,0.96]
mean=np.mean(x)
median=np.median(x)
trmean=stats.trim_mean(x,0.1)

n=len(x)
m=np.mean(x)
s=np.std(x)
B=10000
meanvec=np.zeros(B+1)
medianvec=np.zeros(B+1)
trmeanvec=np.zeros(B+1)
ind=np.arange(0,B)
for i in ind:
    xstar=random.choices(x,k=n)
    meanvec[i]=np.mean(xstar)
    medianvec[i]=np.median(xstar)
    trmeanvec[i]=stats.trim_mean(xstar,0.1)
```

Vi finner standardfeilene

$S_{\hat{\mu}_1}=0.0199$ $S_{\hat{\mu}_2}=0.0181$ $S_{\hat{\mu}_3}=0.0219$

Simulering av tilfeldige tall på [0,1]

Datamaskiner kan generere en følge av tall, såkalt «pseudotilfeldige» tall, som for (nesten) alle praktiske formål ligner på tilfeldige tall på intervallet [0,1]

Formelt svarer et tilfeldig tall på [0,1] til en stokastisk variabel *U* som er uniformt fordelt på [0,1]

Hvis *U* er uniformt fordelt på [0,1] , har vi at

$$f_U(u) = \begin{cases} 1 & \text{for } 0 \le u \le 1 \\ 0 & \text{ellers} \end{cases}$$

$$F_U(u) = \begin{cases} 0 & u < 0 \\ u & 0 \le u \le 1 \\ 1 & u > 1 \end{cases}$$

Hvordan kan vi ut fra tilfeldige tall på [0,1] generere en kontinuerlig fordelt stokastisk variabel *X*, som har en gitt fordeling?

To vanlige metoder (det fins flere)

- Inversjonsmetoden
- Forkastningsmetoden

Inversjonsmetoden kan vi bruke når vi har et eksplisitt uttrykk for den inverse av den kumulative fordelingen til *X*

Forkastningsmetoden kan vi bruke også når vi ikke har et uttrykk for den inverse av den kumulative fordelingen (Vi vil ikke se på denne, men metoden er beskrevet i notatet)

Inversjonsmetoden (jf. oblig 2)

Vi vil generere en kontinuerlig stokastisk variabel X som har kumulativ fordelingsfunksjon $F(x)$. Her er $F(x)$ en strengt voksende kumulativ fordelingsfunksjon.

La $U \sim uniform[0,1]$ og sett $X = F^{-1}(U)$

Da er den kumulative fordelingen til X gitt ved

$$F_X(x) = P(X \le x) \quad = P(F^{-1}(U) \le x)$$

$$= P(U \le F(x)) \quad = F(x)$$

så X har kumulativ fordeling $F(x)$

Eksempel: Cauchy-fordelingen

Standard Cauchy fordelingen har tetthet

$$f(x) = \frac{1}{\pi(1+x^2)}$$

Den kumulative fordelingen er

$$F(x) = \frac{1}{2} + \frac{1}{\pi}\arctan(x)$$

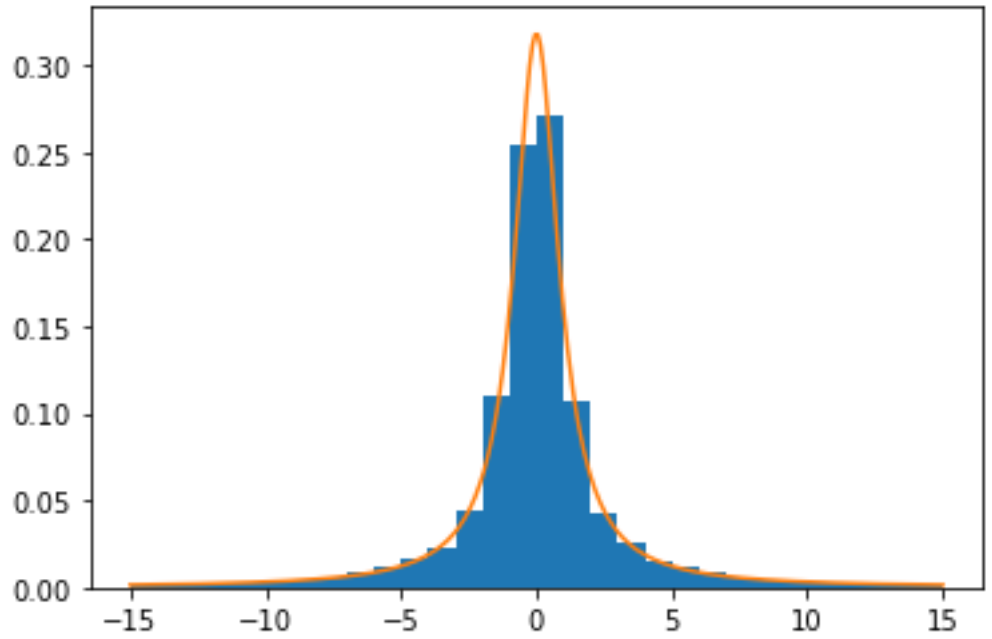
der $\arctan(x) = \tan^{-1}(x)$

Den inverse av den kumulative fordelingen er

$$F^{-1}(u) = \tan\left[\pi(u - 1/2)\right]$$

Python:

```
import numpy as np
import scipy.stats as stats
import math
import matplotlib.pyplot as plt
u=stats.uniform.rvs(0,1,size=10000)
x=np.tan(math.pi*(u-0.5))
plt.hist(x,bins=30,density=True,range=[-15,15])
xp=np.linspace(-15,15,300)
plt.plot(xp,1/(math.pi*(1+xp**2)))
```



Monte Carlo-integrasjon

Vi er interessert i å bestemme integralet

$$\theta = \int_{-\infty}^{\infty} \cos(x) e^{-x^2/2}\, dx$$

Merk at vi kan skrive

$$\theta = \int_{-\infty}^{\infty} \sqrt{2\pi}\cos(x)\frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx \quad = \int_{-\infty}^{\infty} \sqrt{2\pi}\cos(x) f(x) dx$$

der $f(x)$ er standardnormaltettheten

Altså er

$$\theta = E\left\{\sqrt{2\pi}\cos(X)\right\}$$

Der $X \sim f$

Vi kan estimere θ med

$$\hat{\theta} = \bar{Y} = \frac{1}{M}\sum_{i=1}^{M} Y_i \qquad \text{der} \qquad Y_i = \sqrt{2\pi}\cos(X_i)$$

og $X_1, X_2, ..., X_M$ er u.i.f med tetthet $f(x)$

Python:

```
import numpy as np
import scipy.stats as stats
import math
M=10000
x=stats.norm.rvs(size=M,loc=0,scale=1)
y=np.sqrt(2*math.pi)*np.cos(x)
theta_hat=np.mean(y)
```

Vi kan lett bestemme en feilmargin for estimatet $\hat{\theta}$

Sett $S = \sqrt{\frac{1}{M-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$

Da er $\frac{\bar{Y} - \theta}{S / \sqrt{M}}$ tilnærmet standardnormalfordelt

Et 100(1-α)% konfidensintervall for θ er gitt ved

$$\bar{y} \pm z_{\alpha/2}\frac{s}{\sqrt{M}}$$

Python:

```
alpha=0.01
s=np.std(y)
n=len(y)
feilmargin=stats.norm.ppf(1-alpha/2)*s/np.sqrt(n)
```

Generelt ser vi på et integral av formen

$$\theta = \int_{-\infty}^{\infty} g(x)dx$$

Det kan vi skrive

$$\theta = \int_{-\infty}^{\infty} \frac{g(x)}{f(x)} f(x)dx$$

der $f(x)$ er en tetthet og $f(x) > 0$ hvis $g(x) > 0$

Da er

$$\theta = E\left\{g(X)/f(X)\right\}$$

Der $X \sim f$

Vi kan estimere θ med

$$\hat{\theta} = \frac{1}{M}\sum_{i=1}^{M} Y_i \qquad \text{der} \qquad Y_i = g(X_i)/f(X_i)$$

og $X_1, X_2, ..., X_M$ er u.i.f med tetthet $f(x)$

Dette kalles Monte Carlo-integrasjon