Datorintensiva metoder i matematisk statistik

Gunnar Englund

Innehåll

Kapitel 1

Inledning

1.1 Bakgrund

Datorernas intåg har inneburit att man inom matematisk statistik har fått möjlighet att använda ett antal metoder som tidigare var antingen okända eller orealistiska att använda p g a den stora beräkningsvolymen. Detta har inneburit att man kunnat ersätta orealistiskt idealiserade statistiska modeller med mer realistiska modeller.

1. Vi vill kunna analysera datamaterial utan restriktiva antaganden om hur våra data uppkommit, t ex att data kommer från normalfördelning, exponentialfördelning etc.

2. Vi kommer att slippa trassliga analytiska beräkningar av stickprovsvariablers fördelningar etc.

3. Vi kommer i stället att göra omfattande (men hanterliga) kalkyler och simuleringar på dator.

1.2 Innehåll

De metoder som kommer att behandlas är framför allt

1. Bootstrap och jackknife

2. Simulering – speciellt Markov Chain Monte Carlo (MCMC)

3. Val av modell – Model selection

Om man kort vill beskriva Bootstrap och Jackknife kan man säga att de fungerar på så sätt att man ur sina mätdata skapar fiktiva datauppsättningar (ofta genom simuleringar) och sen studerar hur punktskattningar förändras när man på detta sätt ”ruckat” på sina mätdata.

I Jackknife (fällkniv/universalverktyg) plockar man helt enkelt bort en observation i taget och studerar hur mycket skattningen då förändras. Man skulle

kunna uttrycka det så att man ser efter hur mycket varje enskild observation betyder för skattningen. Förändras skattningen mycket då en enstaka observation plockats bort ur data har man ju all anledning att tro att skattningen är osäker. Detta ger en möjlighet att skatta standardavvikelsen för skattningen – det s k medelfelet.

I bootstrap (lyfta sig själv i håret/stövelskaften – jämför Baron Münchhausen) lottar man i princip fram nya data genom dragning med återläggning från de ursprungliga data. Genom att studera hur skattningen varierar mellan dessa fiktiva datauppsättningar kan man få en god uppfattning om hela fördelningen av stickprovsvariabeln inklusive t ex dess standardavvikelse, väntevärde, percentiler i fördelningen etc.

Vi kommer att lägga tonvikten vid Bootstrap, medan Jackknife kommer att spela en mer underordnad roll. Båda är exempel på "Återsamplingsmetoder" (Resampling methods) dit man också brukar hänföra metoder som korsvalidering (som används för modell-val).

Bradley Efron hittade på bootstrap 1979, medan jackknife är från 1940-talet. Bootstrap-metodiken har väckt intresse inom vissa "avnämarområden" såsom genetik, signalbehandling och ekonomi, men är fortfarande förvånansvärt okänd bland statistiker även om forskning pågår.

När det gäller Simulering (Monte Carlo-metoder) så är tekniken där att anlysera komplicerade funktioner av stokastiska variabler genom att lotta fram utfall av dem. Detta gör att man kan slippa besvärliga analytiska beräkningar t ex för fördelningen för stickprovsvariabler. Speciellt MCMC (Markov Chain Monte Carlo) är synnerligen användbar och har revolutionerat delar av matematisk statistik (framför allt Bayesiansk statistik). Det finns också viktiga tillämpningar inom statistisk fysik och optimering.

Modell-val handlar om det viktiga problemet att välja den "bästa" av ett antal föreslagna modeller. Av speciellt intresse är korsvalidering som praktiskt innebär att man (då man har n observationer) för varje föreslagen modell måste skatta ingående parametrar n gånger.

Materialet i dessa föreläsningsanteckningar är hämtade från flera källor bl a

1. "An Introduction to the Bootstrap" av Efron och Tibshirani
2. "Computer Intensive Statistical Methods" av Urban Hjorth
3. Material skrivet av Ola Hössjer till en motsvarande kurs i Lund
4. "Markov Chain Monte Carlo in Practice" av Gilks, Richardson och Spiegelhalter

De genomräknade exempel som finns redovisade har skrivits i Matlab, men naturligtvis går det bra att implementera algoritmerna i andra program-språk.

1.3 Uppläggning av materialet

I kapitel 2 behandlas simulering (Monte Carlo-metoder) där man tar fram fördelningen för funktioner av fler-dimensionella stokastiska variabler. Kapitel 3 behandlar inferens och är mer att se som en uppfriskning av material som behandlats i grundkurser. Kapitel 4 behandlar bootstrap och i kapitel 5 och 6 ges ett antal exempel på mer eller mindre komplicerade tillämpningar av bootstrap. Kapitel 7 behandlar hur bootstrap fungerar för medelvärde och median. Kapitel 8 visar hur bootstrap kan användas för att skatta systematiska felet hos skattningar. Kapitel 9 inför jackknife och visar hur denna metod kan användas för att skatta systemenatiskt fel och standardavvikelse hos skattningar. Kapitel 10 behandlar hur bootstrap kan användas i mer komplicerade datastrukturer som flera oberoende stickprov, regressionsproblem och tidsserier. Kapitel 11 ger ett bevis för att bootstrap ger korrekt resultat då data bara kan anta ett ändligt antal värden. Kapitlet visar också på sambandet mellan jackknife och bootstrap. Kapitel 12, 13 och 14 presenterar tre olika metoder att konstruera konfidensintervall med hjälp av bootstrap. Kapitel 15 behandlar Bayesianska metoder helt allmänt. Dessa metoder är av stort intresse men brukar ej behandlas i elementära statistikkurser. Avsikten är att dessa ska kunna praktiskt implementeras med Markov Chain Monte Carlo-metoderna (MCMC) som presenteras i kapitel 17. MCMC är en mycket användbar teknik att simulera godtyckliga fler-dimensionella fördelningar. Det mellanliggande kapitel 16 behandlar översiktligt modellvalsmetoder, dvs hur man jämför olika föreslagna statistiska modeller med varandra på ett rättvisande sätt. Av speciellt intresse är då korsvalidering.

Kapitel 2

Om simulering

2.1 Inledning

Ordet simulera betyder ju i vardagsspråket "låtsas", "efterhärma" eller "fuska" medan simulering i tekniska och matematiska sammanhang innebär att man ersätter verkligheten med en matematisk modell och utför experiment i denna. Vi kommer att ägna oss åt s k Monte Carlo-simulering, dvs att lotta fram värden på stokastiska variabler för att ersätta komplicerade analytiska beräkningar av t ex stickprovsvariablers fördelningar.

Vi återkommer till simulering i kapitel 17 när vi sysslar med Markov Chain Monte Carlo-simulering framför allt för att simulera fler-dimensionella stokastiska variabler.

2.2 Slumptal

Vi kommer att behöva generera tal som uppför sig som oberoende utfall av stokastiska variabler med specificerad fördelning. Man skulle kunna tänka sig att som slumpmekanism ha något fysikaliskt fenomen som t ex radioaktivt sönderfall som enligt modern fysik är exempel på genuin slump, men vi kommer i stället att använda oss av s k pseudo-slumptal dvs sekvenser som uppträder med "tillräcklig slumpmässighet". Dessa deterministiska följder av pseudoslumptal $x_0, x_1, x_2, \cdots$ kan genereras med olika algoritmer. Den vanligaste typen är kongruensalgoritmer av typen

$$x_{n+1} = (ax_n + b) \mod c$$

(där a, b och c är givna heltal) dvs den rest man får då man dividerar $ax_n + b$ med c. Man dividerar de erhållna x_n:en med c för att erhålla slumptal på intervallet $(0, 1)$. I en tidigare version av Matlab användes algoritmen $x_{n+1} = (7^7 x_n) \mod (2^{31} - 1)$ men algoritmen har förändrats mellan olika versioner.

Matlab levererar sådana pseudoslumptal med funktionen `rand` och enligt manualen för version 5 gäller att
"This generator can generate all the floating point numbers in the closed interval $[2^{-53}, 1 - 2^{-53}]$. Theoretically, it can generate over 2^{1492} values before

repeating itself." Det anges dock inte vilken algoritm som egentligen används.

Genom att anropa funktionen med `rand(n,m)` får man en $n \times m$-matris av sådana pseudoslumptal.

Man kan även styra sekvensen av pseudoslumptal och kan därigenom upprepa exakt samma sekvens. Detta kan vara användbart om man önskar upprepa exakt samma simulering. Läsaren hänvisas till Matlabs manualer eller till online-hjälpen i Matlab.

Dessa pseudo-slumptal uppträder alltså som om de vore oberoende likformigt fördelade på intervallet $(0, 1)$. Vi kommer inte att vara särskilt bekymrade över att de "egentligen" är deterministiska utan kommer att lita på att de har tillräcklig slumpmässighet vad gäller fördelning och oberoende för att tjäna våra syften.

2.3 Inversmetoden

Om man har tillgång till oberoende likformigt fördelade stokastiska variabler (dvs $R(0, 1)$-fördelade) så kan man i princip generera vilken en-dimensionell fördelning som helst i enlighet med följande sats.

Sats 2.1 Låt $F(x)$ vara en fördelningsfunktion och definiera "inversen"

$$F^{-1}(y) = \inf\{x : F(x) \geq y\},\ 0 \leq y \leq 1.$$

Om U är $R(0, 1)$ och vi låter $X = F^{-1}(U)$ så gäller att $P(X \leq x) = F(x)$, dvs X har F till fördelningsfunktion. □

Inversen F^{-1} överensstämmer med den vanliga inversen om F är kontinuerlig strikt växande, men fungerar också om F har språng (dvs punktmassor i sannolikhetsfördelningen) eller är konstant på ett intervall (dvs det saknas massa på motsvarande intervall). Se figur 2.1 för en illustration.

Bevis: Av definitionen framgår att $\{F^{-1}(y) > x\} = \{y > F(x)\}$ vilket ger

$$P(X > x) = P\left(F^{-1}(U) > x\right) = P(U > F(x)) = 1 - F(x)$$

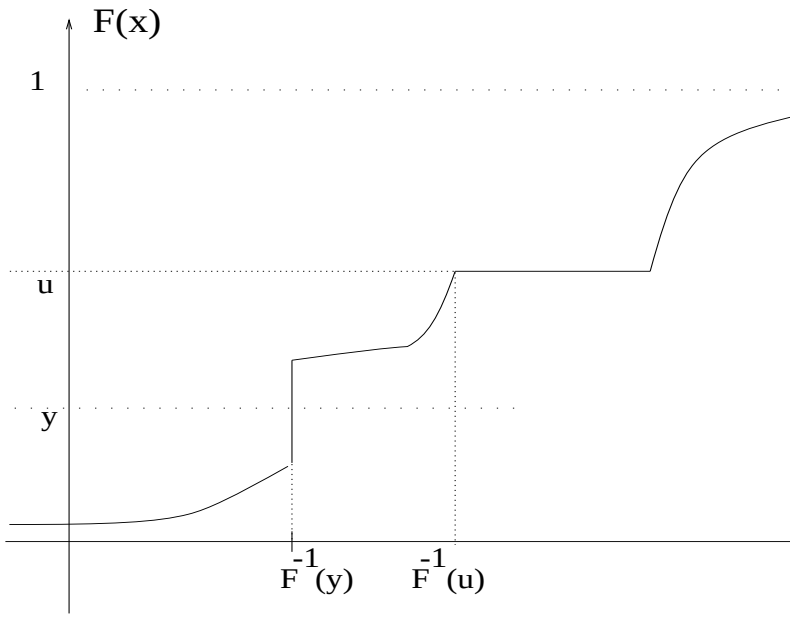
dvs $P(X \leq x) = F(x)$ □

Exempel 2.1 Exponentialfördelningen $\text{Exp}(\theta)$ har fördelningsfunktionen

$$F(x) = 1 - e^{-x/\theta} \text{ för } x \geq 0.$$

Denna har inversen $F^{-1}(y) = -\theta \ln(1 - y)$ och alltså kan vi se att

$$X = -\theta \ln(1 - U) \text{ är } \text{Exp}(\theta) \text{ om } U \text{ är } R(0, 1).$$



Figur 2.1: Inversmetoden

Eftersom även $1-U$ är $R(0,1)$ så gäller också att $X = -\theta \ln(U)$ är $\text{Exp}(\theta)$.

Matlabs `Stats`-modul har funktionen `exprnd` som genererar exponentialfördelade slumptal med hjälp av ovanstående metod. □

Inte alla fördelningar har lättinverterade fördelningsfunktioner. Normalfördelningens $\Phi(x)$ är t ex inte så lätt att invertera, men då finns det andra knep som de som visas i avsnitt 2.7 på sidan 11. I Matlab:s standardmodul finns funktionen `randn` som genererar N(0,1)-fördelade slumptal och i `Stats`-modulen finns funktionen `normrnd`.

2.4 Diskreta fördelningar

För diskreta fördelningar F som alltså lägger massan $P(X = x_k)$ i x_k för $k = 1, 2, \cdots$ kan man använda sig av att ta

$$F^{-1}(u) = x_j \text{ om } \sum_{k=1}^{j-1} P(X = x_k) \leq u < \sum_{k=1}^{j} P(X = x_k),\ j = 1, 2, \cdots$$

Exempel 2.2 Om $P(X = 1) = 0.4, P(X = 7) = 0.2$ och $P(X = 10) = 0.4$ kan man låta

$$X = F^{-1}(U) = \begin{cases} 1 & \text{om } 0 \leq U < 0.4 \\ 7 & \text{om } 0.4 \leq U < 0.6 \\ 10 & \text{om } 0.6 \leq U < 1 \end{cases}$$

□

2.5 Matlabs slumpstalsgeneratorer

Standardmodulen i Matlab har funktionerna `rand` och `randn` för $R(0,1)$ respektive den standardiserade normalfördelningen.

Matlabs `Stats`-modul har en hel uppsättning slumptalsgeneratorer för olika fördelningar (både diskreta och kontinuerliga):

```
Random Number Generators.
betarnd     - Beta random numbers.
binornd     - Binomial random numbers.
chi2rnd     - Chi square random numbers.
exprnd      - Exponential random numbers.
frnd        - F random numbers.
gamrnd      - Gamma random numbers.
geornd      - Geometric random numbers.
hygernd     - Hypergeometric random numbers.
lognrnd     - Lognormal random numbers.
mvnrnd      - Multivariate normal random numbers.
nbinrnd     - Negative binomial random numbers.
ncfrnd      - Noncentral F random numbers.
nctrnd      - Noncentral t random numbers.
ncx2rnd     - Noncentral Chi-square random numbers.
normrnd     - Normal (Gaussian) random numbers.
poissrnd    - Poisson random numbers.
random      - Random numbers from specified distribution.
raylrnd     - Rayleigh random numbers.
trnd        - T random numbers.
unidrnd     - Discrete uniform random numbers.
unifrnd     - Uniform random numbers.
weibrnd     - Weibull random numbers.
```

Om man med Matlab vill simulera någon annan fördelning får man själv skriva en `m`-fil som med t ex inversmetoden genererar slumptal med den önskade fördelningen.

2.6 Funktioner av oberoende stokastiska variabler

Antag att vi kan generera slumptal $(x_1, x_2, \cdots, x_d) = \boldsymbol{x}$ som är utfall av oberoende stokastiska variabler $(X_1, X_2, \cdots, X_d) = \boldsymbol{X}$ där X_i:na har en givna fördelningar (i de flesta tillämpningar samma fördelning). Denna generering skall kunna upprepas så att $\boldsymbol{x}$-vektorerna är att betrakta som oberoende.

Vi är intresserade av fördelningen för $g(X_1, X_2, \cdots, X_d)$ för en specificerad (men kanske komplicerad) reellvärd funktion g. Vad vi har i tankarna är att t

ex låta g-funktionen vara en punktskattning av en parameter θ. Detta betyder att $g(X_1, X_2, \cdots, X_d)$ i så fall är motsvarande stickprovsvariabel.

Detta kan lätt göras med simulering. Man skaffar sig ett stort antal (kanske 10000 eller 100000) framlottade uppsättningar om d slumptal vardera där dessa har rätt fördelning. För varje sån uppsättning beräknar vi g:s värde och gör ett histogram över resultatet. Detta histogram (uppfattat som sannolikhetsfördelning) kommer att väl approximera den sanna fördelningen för $g(X_1, X_2, \cdots, X_d)$.

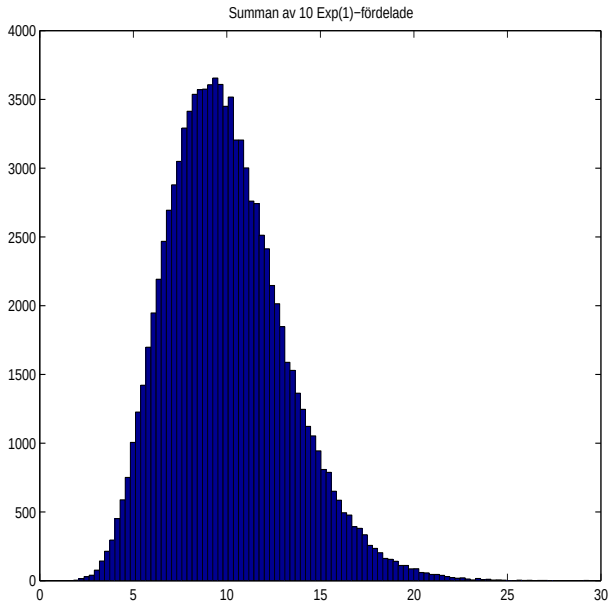
Exempel 2.3 Om vi simulerar 100000 st uppsättningar av 10 st oberoende Exp(1)-fördelade stokastiska variabler $X_1, X_2, \cdots, X_{10}$ och bildar

$$S_{10} = g(X_1, X_2, \cdots, X_{10}) = \sum_{i=1}^{10} X_i$$

så har vi på detta sätt simulerat fördelningen för summan av 10 oberoende Exp(1)-fördelade variabler. I Matlab blir detta:

```
x=exprnd(1,10,100000);
y=sum(x);
hist(y,100)
```

där sista kommandot gör ett histogram med 100 st ”fack”. Denna operation tog ca 20 sek på en Pentium. Resultatet framgår av figuren 2.2.



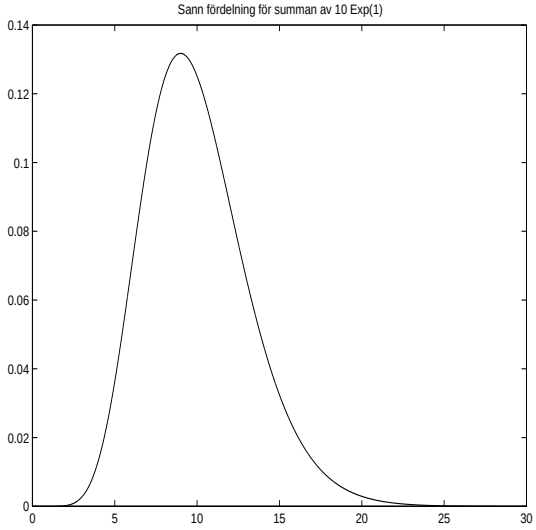
Figur 2.2: Simulerad fördelning för summan av 10 Exp(1)-fördelade

Man kan visa att S_{10} är $\Gamma(10,1)$-fördelad. Detta kan göras med hjälp av faltning eller via Poisson-processen. $\Gamma(10,1)$-fördelningen har tätheten

$$f(x) = \frac{x^9}{9!}e^{-x},\ x \geq 0.$$

Det är lätt att plotta denna i Matlab med hjälp av `gampdf`-funktionen (gamma probability density function) som finns i `Stats`-modulen och resultatet syns i figur 2.3

```
u=[0:0.001:30];
p=gampdf(u,10,1);
plot(u,p)
```



Figur 2.3: Sann fördelning för summan av 10 Exp(1)-fördelade

Här vet vi det rätta svaret och simulering är onödig, men man noterar hur väl simuleringarna överensstämmer med den sanna fördelningen. Det centrala var

1) att vi lätt kunde generera 100000 stickprov av storlek 10 vardera

2) i vart och ett av dessa 100000 stickprov kunde beräkna g-funktionen

3) att dessa operationer gick någorlunda raskt.

I detta fall var simuleringen obehövlig men med en mer komplicerad g-funktion eller med en mer komplicerad fördelning för X:n skulle vi ställas inför oöverstigliga analytiska problem om vi försökte finna den sanna fördelningen. Simuleringen är mer en fråga om tålamod och i någon mån listiga algoritmer. □

2.7 Normalfördelning

Som angavs ovan är normalfördelningen lite trasslig att simulera ifrån med hjälp av Inversmetoden, men det finns flera metoder att kringgå detta.

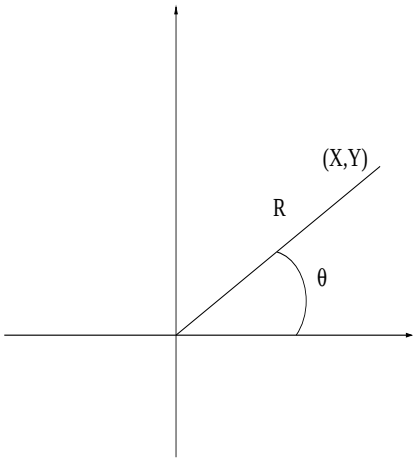
2.7.1 Box-Mullers metod:

En vanlig metod att generera oberoende $N(0,1)$-fördelade slumptal är den s k Box-Mullers metod.

Om X och Y är oberoende N(0,1) så gäller att $R = \sqrt{X^2 + Y^2}$ och $\Theta = \arg(X, Y)$ oberoende. Detta inses av att gå över till polära koordinater (r, θ) från (x, y). Vi skriver alltså

$$X = R\cos(\Theta),\ Y = R\sin(\Theta)$$

enligt figur 2.4.



Figur 2.4: Box-Mullers metod

Man har då vid variabelbytet Jacobianen r dvs $dxdy = r\,drd\theta$. Alltså får vi

$$f_{X,Y}(x,y)dxdy = f_X(x)f_Y(y)dxdy = \frac{1}{2\pi}e^{-\frac{x^2+y^2}{2}}dxdy = re^{-\frac{r^2}{2}}\frac{1}{2\pi}drd\theta =$$

$$= f_{R,\Theta}(r,\theta)drd\theta$$

dvs att R har tätheten

$$f_R(r) = re^{-r^2/2},\ r \geq 0$$

och att Θ har tätheten $f_\Theta(\theta) = 1/(2\pi)$, $0 \leq \theta \leq 2\pi$. Dessutom ser man att R och Θ är oberoende! Alltså får vi

$$F_R(r) = \int_0^r ue^{-u^2/2}du = 1 - e^{-r^2/2},\ r \geq 0.$$

Denna fördelning kallas Rayleigh-fördelningen och fördelningsfunktionen är lätt inverterbar. Vi får

$$F_R^{-1}(y) = \sqrt{-2\ln(1-y)}.$$

Alltså kan vi generera två stycken oberoende N(0,1)-fördelade variabler X och Y ur två stycken $R(0,1)$-fördelade variabler U_1 och U_2 genom

$$X = \cos(2\pi U_1)\sqrt{-2\ln(1-U_2)}$$

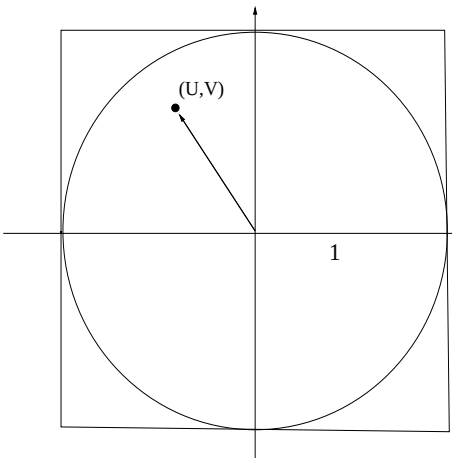
$$Y = \sin(2\pi U_1)\sqrt{-2\ln(1-U_2)}$$

Här kan vi ersätta $1-U_2$ med U_2 om vi så vill!

Vi kommer genomgående att låta $N(m,\sigma^2)$ beteckna normalfördelningen med väntevärde m och varians σ^2. Om vi vill generera slumptal från denna fördelning skaffar vi oss $N(0,1)$-fördelade slumptal Z enligt Box-Mullers metod och bildar sen $X = m + \sigma Z$ som ju får avsedd fördelning.

2.7.2 Alternativ metod

En alternativ metod att generera normalfördelade slumptal är att först välja U och V som oberoende likformigt fördelade på intervallet $[-1,1]$. Om de hamnar inom den i kvadraten inskrivna cirkeln enligt figur 2.5 har man därigenom genererat en likformigt fördelad slumpmässig riktning.



Figur 2.5: Metod för att få likformig slumpmässig riktning.

Skulle (U,V) hamna utanför cirkeln, vilket innebär att $U^2+V^2 > 1$ får man lotta fram nya värden på U och V tills man fått en punkt inne i cirkeln. Man väljer sedan en radie R i enlighet med Rayleigh-fördelningen och flyttar sig på strålen genom $(0,0)$ och (U,V) så att punktens avstånd blir R. Den sålunda

genererade punkten (X, Y) har oberoende $N(0,1)$-fördelade koordinater X och Y. Eftersom cirkelns area är π och kvadratens är 4 måste man generera ett ffg$(\pi/4) \approx$ ffg(0.7854)-fördelat antal (U, V)-par. Fördelen med detta förfarande är att man inte behöver beräkna cosinus och sinus för vinkeln Θ som i Box-Mullers metod vilket kan snabba upp simuleringen.

Funktionen `randn` i Matlab ger $N(0,1)$-fördelade slumptal, men eftersom den är hårdkodad kan man inte se vilken metod Matlab använder sig av.

2.8 Allmän fler-dimensionell fördelning

Att simulera en godtycklig fler-dimensionell fördelning där de ingående variablerna är beroende är ett trixigt problem, men vi återkommer till det i samband med kapitel 17 om Markov Chain Monte Carlo-simulering.

I speciella situationer kan beroendet mellan X_i:na i $\boldsymbol{X} = (X_1, X_2, \cdots, X_d)$ skapas genom att de enskilda X_i:na är funktioner av oberoende variabler $Z_1, Z_2, \cdots$ men där samma Z-variabler ingår i flera X_i-variabler. T ex kan man ha en konstruktion av typen

$$\begin{cases} X_1 &= g_1(Z_1, Z_2, Z_3) \\ X_2 &= g_2(Z_1, Z_2, Z_3) \\ X_3 &= g_3(Z_1, Z_2, Z_3) \end{cases}$$

t ex $X_1 = Z_1$, $X_2 = Z_1 + Z_2$ och $X_3 = Z_1 + Z_2 + Z_3$. Genom att generera oberoende Z-variabler och plugga in dessa i g-funktionerna får man rätt beroendestruktur för X-variablerna. Ett exempel på ovanstående konstruktion utgör de fler-dimensionella normalfördelningarna som kan ses som uppbyggda av linjärkombinationer av oberoende N(0,1)-fördelade variabler.

För den intresserade kan nämnas att det finns en fler-dimensionell generalisering av invers-metoden enligt 2.3 som använder en s k copula, dvs en fler-dimensionell fördelning på enhetskuben där de olika komponenterna har ett specificerat beroende men där marginalfördelningarna är likformiga. Genom en operation liknande inversmetodens kan man sen generera fler-dimensionella variabler med specificerade marginalfördelningar och den specificerad beroendestrukturen. Den intresserade hänvisas till Nelsen, R.B, (1999) *An Introduction to Copulas*, Springer, New York. Svårigheten är att välja copulan så att den avspeglar vad man tror om beroendestrukturen.

2.8.1 Fler-dimensionella normalfördelningar

Om vi vill simulera en fler-dimensionell normalfördelning med given väntevärdesvektor och kovariansmatris går detta förhållandevis lätt.

Genom Box-Muller-metoden kan vi generera oberoende N(0,1)-fördelade stokastiska variabler. Vi vill generera fler-dimensionella (t ex d-dimensionella)

normalfördelade variabler med föreskriven väntevärdesvektor $\boldsymbol{m}$ och kovariansmatris $\boldsymbol{C}$. Om inte $\boldsymbol{C}$ är en diagonalmatris är de ingående komponenterna beroende, men beroendet är av en mycket speciell typ – en slags linjärt beroende som gör att vi kan konstruera d-dimensionella slumptal med rätt fördelning.

Eftersom $\boldsymbol{C}$ är en kovariansmatris måste den vara symmetrisk och positivt semi-definit (dvs bara ha icke-negativa egenvärden).

Choleski-metoden innebär att vi först hittar en d-dimensionell matris $\boldsymbol{A}$ som uppfyller $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^T$. Om vi sedan tar d oberoende N(0,1)-fördelade variabler $Z_1, Z_2, \cdots, Z_d$ och bildar kolonnvektorn $\boldsymbol{Z}=(Z_1,\ Z_2,\ \cdots,\ Z_d)^T$ så gäller att

$$\boldsymbol{X} = \boldsymbol{m} + \boldsymbol{A}\boldsymbol{Z}$$

blir d-dimensionellt normalfördelad med väntevärdesvektor $\boldsymbol{m}$ och kovariansmatris $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^T$. Man kan säga att de fler-dimensionella normalfördelningarna är precis vad man kan få genom att bilda $\boldsymbol{X} = \boldsymbol{m} + \boldsymbol{A}\boldsymbol{Z}$ där $\boldsymbol{Z}$ är oberoende $N(0,1)$.

Att hitta $\boldsymbol{A}$ (som fungerar som ett slags kvadratrot till $\boldsymbol{C}$) är lätt. I Matlab finns rutinen `sqrtm` som hittar den (faktiskt unika) positivt semi-definita symmetriska matris $\boldsymbol{C}^{1/2}$ som uppfyller $\boldsymbol{C} = \boldsymbol{C}^{1/2}\boldsymbol{C}^{1/2}$. Det finns även andra val av $\boldsymbol{A}$-matris som kan vara lämpliga. Man kan t ex låta $\boldsymbol{A}$ vara en triangulär matris. Att skriva $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^T$ innebär då en LR-faktorisering av $\boldsymbol{C}$. Faktum är att konstruktionen av $\boldsymbol{A}$ som en triangulär matris innebär att man i princip utför det som i linjär algebra kallas Gram-Schmidts ortogonaliseringsförfarande.

2.9 Exempel: Korrelationskoefficient

Låt

$$\boldsymbol{U} = ((X_1, Y_1), (X_2, Y_2), \cdots, (X_{20}, Y_{20}))$$

där (X_i, Y_i), $i = 1, 2, \cdots, 20$ är oberoende likafördelade två-dimensionella stokastiska variabler. Vi låter

$$\widehat{\rho}(\boldsymbol{U}) = \frac{\sum_{i=1}^{20}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{20}(X_i - \bar{X})^2 \sum_{i=1}^{20}(Y_i - \bar{Y})^2}}$$

där $\bar{X} = \sum_{i=1}^{20} X_i/20$ och $\bar{Y} = \sum_{i=1}^{20} Y_i/20$.

Speciellt är vi kanske intresserade av fallet att (X_i, Y_i) är två-dimensionellt normalfördelade, t ex

$$N_2\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right).$$

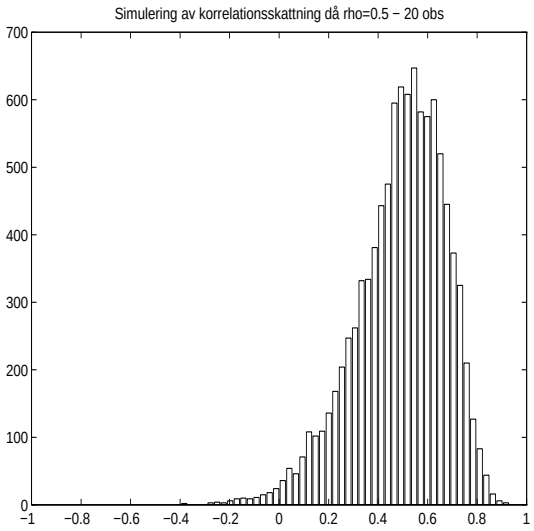
Faktum är att fördelningen för $\widehat{\rho}(\boldsymbol{U})$ i denna situation inte beror av $E(X)$, $E(Y)$, $V(X)$ och $V(Y)$ så att vi tagit dessa till 0, 0, 1 respektive 1 innebär ingen inskränkning.

Man inser att tanken är att med hjälp av punktskattningen $\widehat{\rho}$ skatta korrelationskoefficienten

$$\rho(X,Y) = \frac{C(X,Y)}{D(X)D(Y)}.$$

Vi vill alltså undersöka denna stickprovsvariabel för att kunna analysera motsvarande punktskattning.

Hur fungerar detta? Är skattningen väntevärdesriktig? Vad har den för standardavvikelse? Vad är dess fördelning?

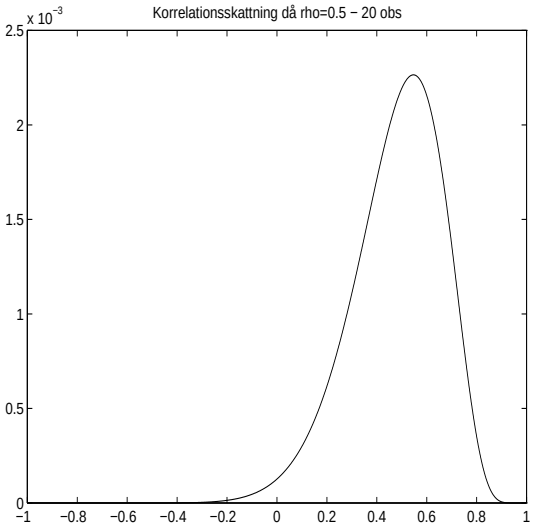


Figur 2.6: Simulering av korrelationsskattning då ρ=0.5 (20 observationer)

Ett sätt att undersöka detta är att simulera fördelningen, dvs lotta fram ett stort antal (säg B st) oberoende utfall av vardera 20 talpar $\boldsymbol{u} = ((x_1, y_1), (x_2, y_2), \cdots, (x_{20}, y_{20}))$ från denna fördelning och sen studera fördelningen av $\widehat{\rho}(\boldsymbol{u})$. För att kunna simulera måste fördelningen vara helt specificerad och vi måste alltså välja ett värde på ρ och vi tar $\rho = 0.5$. Om vi kallar de B framlottningarna $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_B$ (vardera alltså bestående av 20 2-dimensionella vektorer) så kommer (för stora B) fördelningen av

$$\widehat{\rho}(\boldsymbol{u}_1), \widehat{\rho}(\boldsymbol{u}_2), \cdots, \widehat{\rho}(\boldsymbol{u}_B)$$

att approximera fördelningen för $\widehat{\rho}(\boldsymbol{U})$. Faktum är att genom att låta $B \to \infty$ kan vi få fördelningen för $\widehat{\rho}(\boldsymbol{U})$ approximerad hur bra som helst! Speciellt kan vi beräkna intressanta storheter som t ex $E(\widehat{\rho}(\boldsymbol{U}))$ eller $D(\widehat{\rho}(\boldsymbol{U}))$. Vi har i och för sig gjort detta bara för en viss helt specificerad fördelning (i vårt fall med $\rho = 0.5$), men alternativet att ta fram den exakta fördelningen för $\widehat{\rho}(\boldsymbol{U})$ är svårt (för att inte säga omöjligt). Man kan naturligtvis upprepa förfarandet för en uppsättning intressanta ρ-värden t ex i närheten av ett observerat värde ur ett datamaterial. Förhoppningen är då att man kan få en ungefärlig uppfattning



Figur 2.7: "Sann" fördelning av korrelationsskattning då ρ=0.5 (20 observationer) via Fisher-transformationen

om uppträdandet av punktskattningen. Resultatet för $\rho = 0.5$ visas i figur 2.6

I fallet med korrelationskoefficient från två-dimensionellt normalfördelade data, så finns approximativa lösningar som bygger på den s k Fisher-transformationen

$$h(x) = \frac{1}{2} \log \left(\frac{1+x}{1-x} \right).$$

Man kan nämligen visa att $h(\widehat{\rho}(\boldsymbol{U}))$ är approximativt $\mathrm{N}(h(\rho), 1/(n-3))$, dvs normalfördelad med väntevärde $h(\rho)$ och varians $1/(n-3)$. Figuren 2.7 ("facit") är gjord med hjälp av denna transformation.

Framtagningen av Fisher-transformationen har krävt stor finurlighet. Att simulera fördelningen kan göras helt maskinmässigt och kräver bara en någorlunda snabb dator och/eller tålamod. Dessutom fungerar simuleringen även om vi skulle ha haft en annan fördelning än 2-dimensionell normalfördelning – det approximativa resultatet rörande Fisher-transformationen gäller egentligen bara 2-dimensionell normalfördelning. Man kan alltså enkelt ersätta svåra kalkyler och tankearbete med simuleringar.

2.10 Fördelningskonvergens

Vi har en d-dimensionell stokastisk variabel $\boldsymbol{X} = (X_1, X_2, \cdots, X_d)$ och vill beräkna fördelningen för $g(\boldsymbol{X})$ där g är en reellvärd funktion. Om vi kan generera B oberoende utfall $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_B$ av $\boldsymbol{X}$ så vet vi enligt stora talens lag

att

$$\lim_{B\to\infty} \frac{1}{B}\sum_{i=1}^{B} g(\boldsymbol{x}_i) = E\left(g(\boldsymbol{X})\right)$$

Alltså kan vi beräkna $E(g(\boldsymbol{X}))$ genom att simulera.

Mer allmänt gäller att hela fördelningen för simuleringarna konvergerar i fördelningsmening mot den sanna fördelningen för $g(\boldsymbol{X})$.

I praktiken gör vi ett histogram över $g(\boldsymbol{x}_1), g(\boldsymbol{x}_2), \cdots, g(\boldsymbol{x}_B)$ och detta kommer att väl approximera fördelningen för $g(\boldsymbol{X})$. Man kan alltså genom flitigt framlottande av oberoende utfall av $\boldsymbol{X}$ få fram fördelningen för $g(\boldsymbol{X})$ och då också få uppskattningar av t ex $E(g(\boldsymbol{X}))$ eller $D(g(\boldsymbol{X}))$ genom att göra motsvarande kalkyler i histogrammet.

Om man i Matlab har B en-dimensionella simuleringsresultat i vektorn `x` får man då en approximation av $E(X)$ genom `mean(x)` och av $D(X)$ genom `std(x)`. Om vi simulerar en d-dimensionell fördelning och lägger de B simulerade d-dimensionella resultaten i matrisen `x` som alltså har dimensionen $B \times d$ där respektive simulerad vektor ligger radvis kan man beräkna t ex $E(g(\boldsymbol{X}))$ genom

```
result=[];
for i=1:B
 result(i)=g(x(i,:));
end ;
mean_g=mean(result)
```

om funktionen `g(u)` beräknar funktionen g ur radvektorn `u`.

Att man kan få fram hela fördelningen för $g(\boldsymbol{X})$ kan man inse genom att låta

$$h_u(\boldsymbol{x}) = \begin{cases} 1 & \text{om } g(\boldsymbol{x}) \le u \\ 0 & \text{annars} \end{cases}$$

och notera att (igen med hjälp av stora talens lag)

$$\lim_{B\to\infty} \frac{1}{B}\sum_{i=1}^{B} h_u(\boldsymbol{x}_i) = E\left(h_u(\boldsymbol{X})\right) = P(g(\boldsymbol{X}) \le u).$$

Detta fungerar för alla u. Man kan också notera att summan $\sum_1^B h_u(\boldsymbol{x}_i)$ är antalet av $g(\boldsymbol{x}_1), g(\boldsymbol{x}_2), \cdots, g(\boldsymbol{x}_B)$, som är $\le u$. Detta sett som funktion av u brukar kallas den empiriska fördelningen. Om man vill tänka på den som en sannolikhetsfördelning svarar den mot att man lagt massan $1/B$ i vardera av värdena $g(\boldsymbol{x}_1), g(\boldsymbol{x}_2), \cdots, g(\boldsymbol{x}_B)$.

Man kan visa att denna konvergens inte bara sker för varje fixt u utan för alla u på en gång, dvs sedd som en stokastisk process med "tid" u där $-\infty < u < \infty$.

Även om g skulle vara ta värden i R^d fungerar detta helt oförändrat. Man kan se det som att den d-dimensionella fördelningen för $g(\boldsymbol{X})$ approximeras med den fördelning som simuleringsresulaten ger, dvs den fördelning som lägger massan $1/B$ i vardera av de framsimulerade värdena.

Alltså kan vi även skatta storheter som $E(X_1\sqrt{X_2})$ genom att beräkna

$$\frac{1}{B}\sum_{i=1}^{B} x_{i1}\sqrt{x_{i2}}$$

där det i:te framsimulerade värdet är $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ för något "stort" värde på B. På motsvarande sätt kan vilken aspekt som helst av den simulerade fördelningen beräknas genom att beräkna motsvarande kvantitet för de simulerade vektorerna.

Med hjälp av denna typ av resultat kan en massa mycket intressanta storheter rörande fördelningen beräknas. Även hela fördelningen, fördelningen för en viss komponent i vektorn eller någon aspekt av denna, t ex väntevärde, varians, median etc. kan erhållas.

För att illustrera detta synnerligen användbara faktum ger vi några exempel. Antag därför att vi har genererat $\boldsymbol{x}_i$ för $i = 1, 2, \cdots, B$ där B "stort" och där $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})$. Vi har alltså en "lång" följd av simulerade värden.

Exempel 2.4 *Några exempel*
Om vi är intresserade av $E(X_3)$ dvs väntevärdet av den tredje komponenten i den d-dimensionella $\boldsymbol{X}$-vektorn så erhåller vi

$$E(X_3) = \lim_{B\to\infty} \frac{1}{B}\sum_{i=1}^{B} x_{i3}$$

och på motsvarande sätt

$$E(X_3^2) = \lim_{B\to\infty} \frac{1}{B}\sum_{i=1}^{B} x_{i3}^2$$

eller mer allmänt

$$E(g(X_3)) = \lim_{B\to\infty} \frac{1}{B}\sum_{i=1}^{B} g(x_{i3}).$$

Om vi tar $g(u) = 1$ för $u \le z$ och 0 för $u > z$ får vi

$$P(X_3 \le z) = \lim_{B\to\infty} \frac{1}{B}\sum_{i=1}^{B} g(x_{i3}) = \lim_{B\to\infty} \frac{1}{B}(\text{antal } x_{i3} \le z)$$

Om vi vill beräkna $E(X_1X_2)$ erhåller vi

$$E(X_1X_2) = \lim_{B\to\infty} \frac{1}{B}\sum_{i=1}^{B} x_{i1}x_{i2}$$

och skulle vi vilja beräkna $P(X_1 \leq 5; X_2 \leq 3)$ erhåller vi

$$P(X_1 \leq 5; X_2 \leq 3) = \lim_{B\to\infty} \frac{1}{B}(\text{antal par där } x_{i1} \leq 5 \text{ och } x_{i2} \leq 3).$$

eller mer generellt kan vi få en approximation av den simultana fördelningen för (X_1, X_2). □

Vi har velat vara lite övertydliga i ovanstående exempel för att peka på den enorma rikedom av resultat vi kan erhålla om vi lyckas erhålla en lång simulering.

Som mått på skillnaden mellan den sanna fördelningen $H(x)$, $x \in R$ för $g(\boldsymbol{X})$ och den s k empiriska fördelningen $\widehat{H}_B$ för $g(\boldsymbol{x}_1), g(\boldsymbol{x}_2), \cdots, g(\boldsymbol{x}_B)$ (dvs en fördelning som lägger massan $1/B$ i vardera av $g(\boldsymbol{x}_1), g(\boldsymbol{x}_2), \cdots, g(\boldsymbol{x}_B)$) kan man ta

$$D_B = \max_u |H(u) - \widehat{H}_B(u)|$$

som alltså mätter maximala skillnaden i vertikal ledd mellan den sanna fördelningsfunktionen och den fördelning som $g(\boldsymbol{x}_1), g(\boldsymbol{x}_2), \cdots, g(\boldsymbol{x}_B)$ har. Det är ju denna s k empiriska fördelning H_B som histogrammet över framlottningarna beskriver. I själva verket beror teststorheten D_B inte av fördelningen H om H är kontinuerlig. Att det är så beror på att om X har fördelningsfunktionen F är $F(X)$ likformigt fördelad på intervallet $[0, 1]$ – ett faktum som vi använde oss av i inversionsmetoden. Detta utnyttjas i Kolmogorov-Smirnovs test som testar om data verkligen kommer från en föreskriven fördelning. I tabellverk anger man percentiler för D_B för olika B som alltså kan användas för hypotesprövning av H_0: "Data kommer från F". Man kan också visa att

$$\lim_{B\to\infty} P\left(D_B < \frac{z}{\sqrt{B}}\right) = 1 - 2\sum_{j=1}^{\infty}(-1)^{j-1}\exp(-2j^2z^2) \approx 1 - 2e^{-2z^2}$$

som alltså säger att fördelningarna ligger mycket nära varandra för stora B.

Kapitel 3

Om inferens

Vi börjar med den enklaste typen av situation och inför lite beteckningar. Senare återkommer vi till lite mer komplicerade situationer då observationerna inte är oberoende (typiskt för stokastiska processer) och då de inte är likafördelade (typiskt för regression).

Inferens handlar ju om att kunna dra slutsatser från mätdata och för att bli konkreta ger vi ett exempel som vi kommer att återkomma till flera gånger.

Exempel 3.1 *Livslängder*
Vi har mätt upp livslängderna för 10 identiska komponenter och därvid fått värdena

5.6 19.7 1.7 3.3 5.1 0.7 15.6 5.4 3.3 0.1

som vi också betecknar med $x_1, x_2, \cdots, x_{10}$.

Vi vill kunna göra ett uttalande om medellivslängden θ av denna typ av komponent t ex i form av ett 90%-igt konfidensintervall.

Medellivslängden skulle vi ju skatta med $\widehat{\theta}(x_1, x_2, \cdots, x_{10}) = \bar{x} = 6.05 =$ aritmetiska medelvärdet av de observerade livslängderna. Men hur kommer vi till ett konfidensintervall? □

Definition 3.1 *Huvudsituation 1*
Vi har n numeriska observationer $x_1, x_2, \cdots, x_n$ som vi ser som utfall av oberoende likafördelade stokastiska variabler $X_1, X_2, \cdots, X_n$ som alla har fördelningen F. De tänkbara fördelningarna F tillhör en mängd M. Det finns vidare en parameter θ som är en funktion (funktional) av F, dvs $\theta = T(F)$.

För att skatta θ har vi en punktskattning $\widehat{\theta}(x_1, x_2, \cdots, x_n)$ som är att betrakta som dels en funktion av n argument och dels ett numeriskt värde.

För att analysera punktskattningens egenskaper studerar vi stickprovsvariabeln $\widehat{\theta}(X_1, X_2, \cdots, X_n)$ vars fördelning alltså beror av F. □

Ofta låter man M, dvs den tänkbara mängden fördelningar för observationerna vara definierad som en s k parametrisk familj, dvs att fördelningen F är bestämd av θ:s värde. Vi vill dock gärna kunna frigöra oss från detta restriktiva antagande. Vi ser därför M som en vid klass av fördelningar. Parametern θ vi är intresserad av skall vara definierad för alla fördelningar i M, men helst för en ännu vidare klass av fördelningar.

3.1 Några exempel

För att konkretisera och klarlägga vad man egentligen gjorde i grundkursen ges här ett antal exempel. Man skall inte låta sig förvillas av att det mesta är enkelt och välbekant utan noga tänka igenom kopplingen till definitionen ovan.

Exempel 3.2 $P(X_i = 1) = \theta$ och $P(X_i = 0) = 1 - \theta$. Vi har $\theta = E(X)$ och skattar θ med

$$\widehat{\theta}(x_1, \cdots, x_n) = \bar{x} = \text{andelen 1:or i stickprovet}$$

Vi vet då att

$$\widehat{\theta}(X_1, \cdots, X_n) = \bar{X} = \frac{1}{n}\sum_1^n X_i = \frac{Y}{n}$$

där Y är $\text{Bin}(n, \theta)$. Detta betyder att

$$E(\widehat{\theta}(X_1, \cdots, X_n)) = \theta \text{ och } V(\widehat{\theta}(X_1, \cdots, X_n)) = \frac{\theta(1-\theta)}{n}.$$

□

Exempel 3.3 X_i är $\text{Exp}(\theta)$ och där

$$\theta = E(X) = \int_{-\infty}^{\infty} x f(x) dx$$

där $f(x) = F'(x) = \frac{1}{\theta}\exp(-x/\theta)$ för $x \geq 0$ och $f(x) = 0$ för $x < 0$.

Vi skattar lämpligen θ med $\widehat{\theta}(x_1, \cdots, x_n) = \bar{x}$ och får alltså

$$\widehat{\theta}(X_1, X_2, \cdots, X_n) = \bar{X} = \frac{1}{n}\sum_1^n X_i$$

Här vet man att $\sum_1^n X_i$ är $\Gamma(n, 1/\theta)$ dvs har en gammafördelning med täthetsfunktionen

$$f(x) = \frac{x^{n-1}}{(n-1)!\theta^n}\exp(-x/\theta) \text{ för } x > 0.$$

□

Exempel 3.4 X_i är normalfördelad $\mathrm{N}(\theta, \sigma_0^2)$ där $\sigma_0 = 0.1$ (dvs känd spridning). Återigen skattar vi θ med $\widehat{\theta} = \bar{x}$ och vi vet att $\widehat{\theta}(X_1, X_2, \cdots, X_n) = \bar{X}$ är $\mathrm{N}(\theta, \sigma_0^2/n)$. □

Exempel 3.5 X_i är normalfördelad $\mathrm{N}(m, \theta^2)$ och vi skattar θ med

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

$(n-1)\widehat{\theta}^2(X_1, X_2, \cdots, X_n)/\theta^2$ är då $\chi^2(n-1)$-fördelad.

Exempel 3.6 X_i har en kontinuerlig fördelning beskriven av F och θ = medianen i fördelningen, dvs lösningen till ekvationen $F(\theta) = 1/2$. Vi kan skatta θ med medianen av $(x_1, x_2, \cdots, x_n)$ dvs det mittersta i storleksordning (om n är udda) och medelvärdet av de två mittersta (om n är jämnt).

Definition 3.2 *Generalisering av huvudsituation 1*
Vi kan låta data vara fler-dimensionella (säg d-dimensionella), dvs att vi har observationerna $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ där $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ som ses som oberoende utfall av $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \cdots, X_{id})$ som är en d-dimensionell stokastisk variabel med en d-dimensionell fördelning beskriven av F. Observera att typiskt är de olika komponenterna av $\boldsymbol{X}$-vektorn beroende stokastiska variabler. Även här har vi en parameter $\theta = T(F)$. □

Som tillämpning av detta studerar vi skattning av korrelationskoefficient.

Exempel 3.7 *Korrelationskoefficient*
Vi har 2-dimensionella observationer $(x_1, y_1), (x_2, y_2), \cdots, (x_{20}, y_{20})$ som är utfall av $(X_1, Y_1), (X_2, Y_2), \cdots, (X_{20}, Y_{20})$ som är oberoende likafördelade med fördelningsfunktion $F(x, y),\ (x, y) \in R^2$. Parametern ρ är korrelationskoefficienten dvs

$$\rho = \frac{C(X, Y)}{D(X)D(Y)}$$

där alltså ρ är något som kan räknas ut med hjälp av F. Vi kan skatta ρ med

$$\widehat{\rho} = \frac{\sum_{i=1}^{20}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{20}(x_i - \bar{x})^2 \sum_{i=1}^{20}(y_i - \bar{y})^2}}.$$

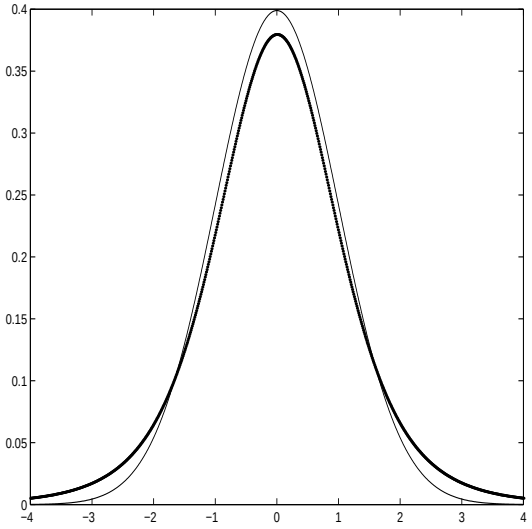
□

3.1.1 Störparametrar

En vanlig komplikation är att fördelningen F inte är helt bestämd av parametern θ. Ofta ser man då parametern θ som flerdimensionell, t ex $\boldsymbol{\theta} = (\theta_1, \theta_2)$ där vi är intresserade av just θ_1. Man brukar då kalla θ_2 för en störparameter (nuisance parameter) som egentligen inte är av primärt intresse men som påverkar stickprovsvariabelns fördelning.

Exempel 3.8 X_i är $\mathrm{N}(\theta, \sigma^2)$ där vi är intresserade av θ medan σ är att betrakta som störparameter.

Vi skattar återigen θ med $\widehat{\theta}(x_1, \cdots, x_n) = \bar{x}$ och vi har att $\widehat{\theta}(X_1, \cdots, X_n) = \bar{X}$ är $\mathrm{N}(\theta, \sigma^2/n)$. Man ser att denna fördelning beror av störparametern σ och därför brukar man ju (t ex i konstruktionen av konfidensintervall för θ) studera en listigt vald transformation av stickprovsvariabeln $\bar{X}$ vars fördelning inte beror av störparametern σ eller av θ.



Figur 3.1: t-fördelning med 5 frihetsgrader (tjockare linjen) samt för jämförelse N(0,1)-fördelningens täthet.

Vi skattar σ med s där

$$s^2(x_1, x_2, \cdots, x_n) = \frac{1}{n-1} \sum_1^n (x_i - \bar{x})^2.$$

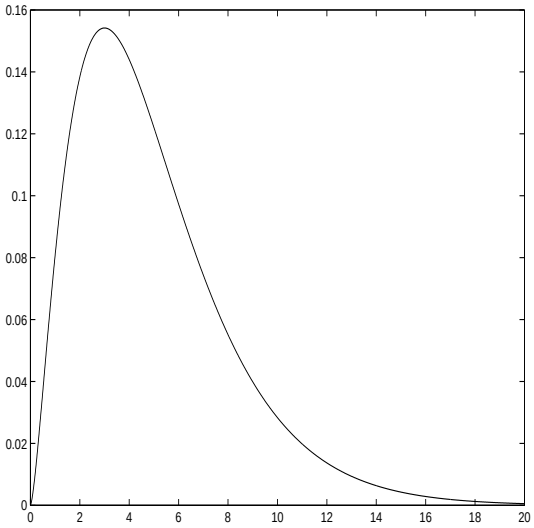
Vi vet att

$$T = \frac{\widehat{\theta}(X_1, \cdots, X_n) - \theta}{s(X_1, \cdots, X_n)/\sqrt{n}} = \frac{\bar{X} - \theta}{s(X_1, \cdots, X_n)/\sqrt{n}}$$

är t-fördelad med $n-1$ frihetsgrader (dvs $t(n-1)$-fördelad). Som exempel visar figur 3.1 $t(5)$-fördelningen.

Denna typ av variabel T kallas pivot-variabel vars fördelning alltså är fix och "känd" oavsett F:s utseende inom den tillåtna klassen M av tänkbara fördelningar. Pivot betyder "svängtapp" men en bättre svensk översättning vore kanske "grundbult". □

Exempel 3.9 X_i är $\mathrm{N}(m, \theta^2)$ där alltså m är att betrakta som störparameter. Vi skulle lämpligen skatta m med $\bar{x}$ och θ med $s(x_1, \cdots, x_n)$ som i föregående exempel.



Figur 3.2: $\chi^2(5)$-fördelningens täthet.

Man har alltså stickprovsvariablen

$$s^2(X_1, X_2, \cdots, X_n) = \frac{1}{n-1}\sum_1^n (X_i - \bar{X})^2$$

och man kan bevisa att

$$T = \frac{(n-1)s^2(X_1, X_2, \cdots, X_n)}{\sigma^2}$$

är $\chi^2(n-1)$-fördelad. Som exempel visas $\chi^2(5)$-fördelningens täthet i figur 3.2.

Att få fram en figur som 3.2 i Matlab är mycket enkelt. Följande kod ger resultatet

```
x=[0:0.01:20];
y=chi2pdf(x,5);
plot(x,y)
```

□

I ovanstående exempel har vi gjort det ytterst restriktiva antagandet att våra observationer kommer från normalfördelningar. Om man vill släppa på detta i allmänhet helt orealistiska antagande faller konstruktionen ihop som ett korthus. Man kan dock ibland motivera metodiken med att den är "robust" dvs förhållandevis okänslig för avvikelser vad gäller fördelningsantaganden.

Exempel 3.10 För X_i antar vi att $E(X_i) = \theta$ och $V(X_i) = \sigma^2$ men utan att anta att den är normalfördelad. Här är de tänkbara fördelningarna inte begränsade till en viss parametrisk klass. Genom att tillämpa Centrala Gränsvärdessatsen får man en approximativ uppfattning om fördelningen för $\widehat{\theta} = \bar{X}$, dvs att $\bar{X}$ är approximativt $\mathrm{N}(\theta, \sigma^2/n)$ och kan beräkna t ex approximativa konfidensintervall. Oftast blir vi dessutom tvungna att skatta σ ur data. Notera dock att denna approximerande normalfördelning t ex är symmetrisk. Med bootstrap kommer vi att se att denna metodik mycket enkelt kan starkt förbättras. □

3.2 Konstruktion av skattningar

Hur man konstruerar skattningen $\widehat{\theta}(x_1, x_2, \cdots, x_n)$ ligger egentligen utanför denna diskussion. I allmänhet tas dessa fram med Minsta Kvadratmetoden eller Maximum Likelihoodmetoden. Vi kommer egentligen inte alls att bry oss särskilt om hur skattningen konstruerats utan kommer att betrakta den som given.

3.2.1 Empirisk fördelning och plug-in-skattningar

Av ett särskilt intresse är dock de s k plug-in-skattningarna och vissa resultat rörande bootstrap förutsätter att man har just en sådan skattning.

Vi såg ovan att θ betraktades som något som kunde beräknas ur F, dvs θ är en funktional $\theta = T(F)$. Vi antar att denna funktional T är definierad för en vid klass av sannolikhetsfördelningar, dvs även för fördelningar utanför klassen M som utgör den statistiska modellen.

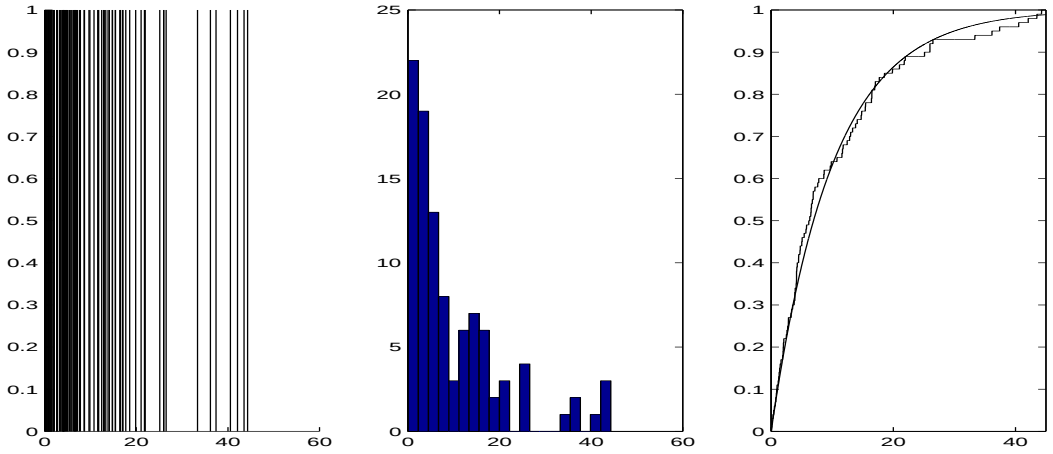
Definition 3.3 *Empirisk fördelning och plug-in-skattning*
Vi låter $\widehat{F}_n$ vara fördelningen som lägger massan $1/n$ i vardera av observationerna $x_1, x_2, \cdots, x_n$. Detta kallas den empiriska fördelningen för observationerna.

Plug-in-skattningen av θ är $\widehat{\theta}(x_1, x_2, \cdots, x_n) = T(\widehat{F}_n)$ dvs samma uttryck (funktional) av den empiriska fördelningen $\widehat{F}_n$ som θ är av F. □

Empiriska fördelningar och plug-in-skattningar kommer alltså att vara mycket centrala i fortsättningen. Empiriska fördelningen för 100 observationer är egentligen en diskret punktmassefördelning med massan 1/100 i vardera av observationerna. Skulle 7 observationer ha samma värde hamnar massan 7/100

i motsvarande punkt. Skulle man illustrera empiriska fördelningen med hjälp av ett stolpdiagram blir det hela svåröverskådligt.

Exempel 3.11 *Empirisk fördelning*
Vi har 100 observationer. Jag väljer att generera dessa från Exp(10)-fördelningen med hjälp av `exprnd` genom Matlab-koden `x=exprnd(10,100,1)`. Om man illustrerar dessa med hjälp av ett stolpdiagram ser det ut som i vänstra figuren i figur 3.3.
En överskådligare bild av datamaterialet får man med hjälp av ett histogram som man skapar i Matlab med `hist(x,20)` där data delats in i 20 klasser som i mittersta bilden i figur 3.3.



Figur 3.3: Stolp-diagram och histogram över de 100 Exp(10)-fördelade observationerna samt empirisk fördelningsfunktion med "sanna" fördelningsfunktionen dvs $1 - \exp(-x/10)$.

Histogrammet är alltså inte exakt en bild av den empiriska fördelningen. Histogrammet är dock så mycket lättare att tolka visuellt att vi kommer genomgående att illustrera diskreta fördelningar med hjälp av histogram. Om man normerar så att ytan av staplarna i histogrammet blir 1 kan histogrammet uppfattas som en sannolikhetsfördelning beskriven av en täthetsfunktion med styckvis konstant täthet.

Ytterligare ett alternativ att illustrera empiriska fördelningen är att rita upp motsvarande fördelningsfunktion som i högra bilden i figur 3.3. □

3.2.2 Exempel på plug-in-skattningar

Många, men inte alla, av de vanligaste skattningarna är faktiskt plug-in-skattningar som framgår av följande exempel.

Exempel 3.12 Väntevärdet:

$$\theta = E(X) = T(F) = \int x dF(x) =$$

$$= \begin{cases} \int x f(x) dx & \text{om fördelningen har täthet} \\ \sum_k x_k p(x_k) & \text{om det är diskret fördelning} \end{cases}$$

och då blir plug-in-skattningen

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = T(\widehat{F}_n) = \int_{-\infty}^{\infty} x d\widehat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} x_i = \bar{x}$$

dvs den helt sedvanliga skattningen. □

Exempel 3.13 Variansen
På samma sätt får vi för

$$\sigma^2 = V(X) = \int_{-\infty}^{\infty} (x - E(X))^2 \, dF(x) =$$

$$= \int_{-\infty}^{\infty} \left(x - \int_{-\infty}^{\infty} u dF(u) \right)^2 dF(x) = T(F)$$

att plug-in-skattningen blir

$$\widehat{\sigma}^2 = T(\widehat{F}_n) = \int_{-\infty}^{\infty} \left(x - \int_{-\infty}^{\infty} u d\widehat{F}_n(u) \right)^2 d\widehat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

dvs den icke-väntesvärdeskorrigerade skattningen. Den traditionella "väntevärdes-korrigerade" skattningen $s^2 = \sum_{i=1}^{n} (x_i - \bar{x})^2/(n-1)$ är faktiskt inte en plug-in-skattning. Detta kan inses genom att om vi "dubblerade" vårt stickprov, dvs hade stickprovet $x_1, x_1, x_2, x_2, \cdots, x_n, x_n$ så skulle den empiriska fördelningen $\widehat{F}$ vara oförändrad, medan stickprovsvariansen skulle bli

$$s^2 = \frac{1}{2n-1} \sum_{i=1}^{n} 2(x_i - \bar{x})^2 = \frac{1}{n - 1/2} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

dvs skulle förändras. □

Exempel 3.14 Median
Låt θ = medianen för F dvs θ är lösningen till ekvationen $F(\theta) = 1/2$. Vi får då plug-in-skattningen $\widehat{\theta}$ = medianen för $\widehat{F}_n$, dvs en (lämpligt vald) lösning till ekvationen $\widehat{F}_n(\widehat{\theta}) = 1/2$. Alltså är plug-in-skattningen av medianen helt enkelt medianen i datamaterialet. □

Exempel 3.15 Percentil
Låt $\theta = \alpha$-percentilen för F dvs θ är lösningen till ekvationen $F(\theta) = \alpha$. Vi får då plug-in-skattningen $\widehat{\theta} = \alpha$-percentilen för $\widehat{F}_n$, dvs en (lämpligt vald) lösning till ekvationen $\widehat{F}_n(\widehat{\theta}) = \alpha$. □

Exempel 3.16 Kovarians
Vi har två-dimensionella observationer $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ och vill skatta kovariansen i den bakomliggande två-dimensionella fördelningen F där $F(x, y) = P(X \leq x; Y \leq y)$, $(x, y) \in R^2$. Med $\theta = C(X, Y)$ är plug-in-skattningen

$$\widehat{\theta}((x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

□

Exempel 3.17 Korrelationskoefficient
Med två-dimensionella data som i föregående exempel och

$$\theta = \rho(X, Y) = C(X, Y)/\sqrt{V(X)V(Y)} =$$

är plug-in-skattningen av θ

$$\widehat{\theta}((x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)) = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_1^n (x_i - \bar{x})^2 \sum_1^n (y_i - \bar{y})^2}}$$

dvs precis vad man brukar skatta korrelationen med. □

Exempel 3.18 Trimmat medelvärde – Winsorization
För att minska inflytandet av extrema observationer som kanske beror på grova mätfel eller felavläsningar används ibland "trimmade medelvärden" (trimmed means) där man väljer att bortse från de mest extrema observationerna innan man bildar medelvärde. Man kan t ex rensa bort $\alpha = 25\%$ av observationerna i vardera ändan och bilda medelvärde av de kvarvarande. Motsvarande parameter θ i den bakomliggande fördelningen F är alltså väntevärdet i den betingade fördelningen givet att observationen hamnat mellan $F^{-1}(\alpha)$ och $F^{-1}(1 - \alpha)$ och alltså är (för $0 < \alpha < 1/2$)

$$\theta = \frac{\int_{F^{-1}(\alpha)}^{F^{-1}(1-\alpha)} x dF(x)}{1 - 2\alpha} = T(F).$$

Notera att θ är ett uttryck (funktional) i F. Plug-in-skattningen ur observationer $x_1, x_2, \cdots, x_n$ är samma uttryck fast för empiriska fördelningen $\widehat{F}_n$, dvs $T(\widehat{F}_n)$. Man rensar bort $\alpha \cdot n$ observationer i vardera änden av de storleksordnade observationerna och bildar medelvärde av de kvarvarande. Om vi kallar de storleksordnade värdena för $x_{(1)}, x_{(2)}, \cdots, x_{(n)}$ blir plug-in-skattningen

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = \frac{1}{n(1 - 2\alpha)} \sum_{i=\alpha \cdot n+1}^{(1-\alpha)n} x_{(i)}$$

om $\alpha \cdot n$ är ett heltal. □

Som nämnts kommer det i allmänhet att vara så att vi använder plug-in-skattningar, men vi gör vid enstaka tillfällen avsteg från detta. Plug-in-skattningar är egentligen inte unika eftersom en parameter θ kan tolkas som olika funktionaler beroende på synsätt. Om vi t ex har observationer från $R(0,\theta)$-fördelningen så kan vi antingen tolka θ som högra ändpunkten för stödet för fördelningen eller också kan vi se θ som $2E(X)$. För just $R(0,\theta)$-fördelningen sammanfaller dessa tolkningar men för en annan fördelning skulle de skilja sig åt. I de allra flesta fall är det dock så att parametern θ har en "uppenbar" tolkning och det är den funktional som motsvarar denna tolkning som vi betraktar.

3.3 Egenskaper hos stickprovsvariabeln

Genomgående för ovanstående exempel är att parametern θ är definierad för alla fördelningar i den (mer eller mindre vida) klassen M av tillåtna fördelningar, men även är definierad för (i stort sett) alla sannolikhetsfördelningar. Att få en uppfattning om sannolikhetsfördelningen för stickprovsvariabeln är fundamentalt vilket framgår av följande:

3.3.1 Systematiskt fel – bias

Frågan om punktskattningen har ett systematiskt fel (bias) avgörs av

$$b_F(\widehat{\theta},\theta) = E(\widehat{\theta}(X_1,\cdots,X_n)) - \theta.$$

Skattningen kallas väntevärdesriktig (unbiased) om $b_F(\widehat{\theta},\theta)$ är 0 för alla fördelningar F (dvs för alla θ) i den aktuella fördelningsklassen M. Vi vill gärna ha skattningar som är väntevärdesriktiga och i den mån de inte är det skulle vi vilja kunna modifiera dem så att det systematiska felet försvinner eller åtminstone minskar.

3.3.2 Medelfel

Ett mycket användbart och ofta använt mått på osäkerheten i $\widehat{\theta}(x_1,x_2,\cdots,x_n)$ är det s k medelfelet $d(\widehat{\theta})$ som är en skattning av den sanna standardavvikelsen $D(\widehat{\theta}(X_1,\cdots,X_n))$. Ofta måste man, för att få ett numeriskt värde på denna standardavvikelse, sätta in skattningar av parametrar.

När det gäller Binomial-situationen i exempel 3.2 är

$$V(\bar{X}) = \frac{\theta(1-\theta)}{n} \text{ dvs } D(\bar{X}) = \sqrt{\frac{\theta(1-\theta)}{n}}.$$

Ett lämpligt sätt att beräkna medelfelet är då att ta

$$d(\widehat{\theta}) = \sqrt{\frac{\widehat{\theta}(1-\widehat{\theta})}{n}}$$

med insättning av det numeriska värdet på $\widehat{\theta}(x_1, \cdots, x_n)$.

På samma sätt anger man ofta medelfelet för $\bar{x}$ som $s/\sqrt{n}$ eftersom $V(\bar{X}) = \sigma^2/n$ dvs $D(\bar{X}) = \sigma/\sqrt{n}$ och eftersom s är en skattning av σ.

För en allmän skattning $\widehat{\theta}$ är det ofta svårt att ange ett (ens approximativt) värde på medelfelet. T ex finns inget enkelt slutet analytiskt uttryck som funktion av våra observationer för medelfelet ens för så enkla skattningar som median eller korrelationskoefficient. Notera att medelfelet är en skattning och alltså skall beräknas numeriskt utifrån observationerna.

Ofta är det tillräckligt att som mått på osäkerheten i skattningen ange ett numeriskt värde på medelfelet. Om man kan ta fram ett hyfsat värde på medelfelet $d(\widehat{\theta})$ och man dessutom tror att skattningen är approximativt väntevärdesriktig och approximativt normalfördelad kan man med hjälp av normalapproximation konstruera det approximativa 95%-iga konfidensintervallet $\widehat{\theta} \pm 1.96 d(\widehat{\theta})$ – mer om detta senare.

Med hjälp av bootstrap och även jackknife kommer vi att se att beräkning av medelfel för godtyckligt komplicerade skattningar är ytterst enkel. Vi kommer också att se att för skattningen $\bar{x}$ ger jackknife och även bootstrap "rätt" medelfel, dvs $s/\sqrt{n}$, men det saknar egentligen intresse eftersom vi för just $\bar{x}$ kan ange medelfelet. Den stora poängen är att man lätt kan få medelfelsskattningar även i mer komplicerade situationer på ett rent mekaniskt sätt.

3.4 Stickprovsvariabelns fördelning

Vi ser att vi är intresserade av att beräkna stickprovsvariabelns fördelning, t ex dess fördelningsfunktion

$$G(z) = P(\widehat{\theta}(X_1, X_2, \cdots, X_n) \leq z), z \in R.$$

Man kan notera att $G(z)$ för fixt z är att betrakta som ett komplicerat uttryck som beror av F, dvs vi kan se detta som en funktional S_z av F där alltså

$$G(z) = P(\widehat{\theta}(X_1, X_2, \cdots, X_n) \leq z) = S_z(F)$$

Vi har alltså en avbildning (funktional) $S_z : F \to G(z)$ som är av stort intresse.

För konstruktion av konfidensintervall är det ibland intressant att kunna beräkna inte bara stickprovsvariabelns fördelning utan också fördelningen för pivot-variabler. T ex studerar man i exponentialfördelningsfallet ovan

$$G(z) = P(n\widehat{\theta}(X_1, X_2, \cdots, X_n)/\theta \leq z) = P(\sum_1^n X_i/\theta \leq z) = S_z(F)$$

som alltså är $\Gamma(n, 1)$-fördelningen enligt vad vi såg i exempel 3.3.

I normalfördelningsfallet med okänd spridning är vi intresserade av

$$G(z) = P\left(\frac{\bar{X} - \theta}{s(X_1, X_2, \cdots, X_n)/\sqrt{n}} \leq z\right) = S_z(F)$$

som vi vet är $t(n-1)$-fördelad oavsett vilken fördelning F är i familjen av normalfördelningar.

Ovanstående pivot-variabler är ofta "approximativa" pivot-variabler även om data inte skulle komma från exponential- respektive normalfördelningar men i sådana situationer är det ofta svårt att ta fram deras fördelningar. Med bootstrap kommer vi i stället att få fram fördelningarna med hjälp av simuleringar utan att behöva göra några antaganden om datas fördelningar. Man kan i detta sammanhang påpeka att i båda fallen ovan utgör $G(z)$ komplicerade funktionaler av de bakomliggande fördelningarna för X-variablerna. Även percentiler för G-fördelningen dvs lösningar z_α till ekvationen $G(z_\alpha) = \alpha$ är att betrakta som funktionaler av den bakomliggande fördelningen F. Det är ofta dessa percentiler som ingår i konfidensintervall som behandlas i nästa avsnitt.

3.5 Konfidensintervall

Definition 3.4 *Konfidensintervall*
Ett konfidensintervall (intervallskattning) med konfidensgrad $1-\alpha$ är ett numeriskt intervall

$$(a(x_1, x_2, \cdots, x_n), b(x_1, x_2, \cdots, x_n))$$

sådant att

$$P\left(a(X_1, X_2, \cdots, X_n) \leq \theta \leq b(X_1, X_2, \cdots, X_n)\right) = 1 - \alpha$$

I allmänhet tar man fram funktionerna $a(\cdot)$ och $b(\cdot)$ genom att studera fördelningen för en pivot-variabel, ta fram percentiler i denna fördelning varefter man "löser ut" θ.

3.6 Några exempel på konfidensintervall

Återigen tar vi ett par välkända exempel som får illustrera metodiken. Läsaren uppmanas att noga tänka igenom vad som egentligen görs och inte låta sig förvillas av att exemplen är välbekanta och enkla!

Av praktiska skäl kommer ibland percentiler att räknas åt vänster och inte, som i grundkursen, åt höger.

3.6.1 Normalfördelade observationer

Vi antar alltså att data kommer från en normalfördelning $N(m, \sigma^2)$.

Exempel 3.19 Som exempel kan vi ta konstruktionen av konfidensintervallet $\bar{x} \pm t_{\alpha/2}(n-1)s/\sqrt{n}$ för m.

Med $G(z) = P(T \leq z)$ där T är $t(n-1)$-fördelad väljer vi alltså tal $z_{\alpha/2}$ och $z_{1-\alpha/2}$ så att

$$G(z_{\alpha/2}) = \alpha/2 \text{ och } G(z_{1-\alpha/2}) = 1 - \alpha/2$$

och får eftersom $T = (\bar{X} - m)/(s/\sqrt{n})$ är $t(n-1)$-fördelad att

$$P\left(z_{\alpha/2} \leq \frac{\bar{X} - m}{s/\sqrt{n}} \leq z_{1-\alpha/2}\right) = 1 - \alpha$$

som efter elementär omformning ger

$$P\left(\bar{X} - z_{1-\alpha/2}\frac{s}{\sqrt{n}} \leq m \leq \bar{X} - z_{\alpha/2}\frac{s}{\sqrt{n}}\right) 1 - \alpha$$

Med sedvanliga beteckningar har vi $z_{\alpha/2} = -t_{\alpha/2}(n-1)$ och $z_{1-\alpha/2} = t_{\alpha/2}(n-1)$ som ger intervallet. □

Man kan här påpeka att det numeriska konfidensintervallet

$$\bar{x} \pm t_{\alpha/2}(n-1)s(x_1, \cdots, x_n)/\sqrt{n}$$

är att betrakta som ett utfall av det stokastiska intervallet

$$\bar{X} \pm t_{\alpha/2}(n-1)s(X_1, X_2, \cdots, X_n)/\sqrt{n}$$

och att konfidensgraden $1 - \alpha$ (i allmänhet 95%) är sannolikheten att det stokastiska intervallet täcker m, dvs är en utsaga om säkerheten i metoden och egentligen inte en utsaga om det konkreta numeriska intervallet som erhålls.

Exempel 3.20 Om parametern vi vill beräkna konfidensintervall för i stället är variansen σ^2 så utnyttjar vi att om X_i:na är $\mathrm{N}(m, \sigma^2)$ så är

$$(n-1)s^2(X_1, X_2, \cdots, X_n)/\sigma^2$$

$\chi^2(n-1)$-fördelad och vi får (med percentildefinition enligt grundkursen) att

$$P\left(\chi^2_{1-\alpha/2}(n-1) \leq \frac{(n-1)s^2(X_1, X_2, \cdots, X_n)}{\sigma^2} \leq \chi^2_{\alpha/2}(n-1)\right) = 1 - \alpha$$

som ger intervallet

$$\left(s\sqrt{\frac{n-1}{\chi^2_{\alpha/2}(n-1)}}, s\sqrt{\frac{n-1}{\chi^2_{1-\alpha/2}(n-1)}}\right).$$

□

3.6.2 CGS-baserade intervall

Som tidigare nämnts kan man ofta med hjälp av Centrala gränsvärdessatsen (CGS) ta fram approximativa konfidensintervall. Ofta gäller ju att man har skattningar som man asymptotiskt kan visa (eller göra troligt) är approximativt normalfördelade. Har man en sådan skattning $\widehat{\theta}$ som alltså uppfyller att $\widehat{\theta}(X_1, X_2, \cdots, X_n)$ är approximativt $\mathrm{N}(\theta, d^2(\widehat{\theta}))$ där $d(\widehat{\theta})$ är medelfelet så kan man ge konfidensintervall av typen $\widehat{\theta} \pm \lambda_{\alpha/2} d(\widehat{\theta})$. Användningen av dessa intervall är ytterst spridd, men ofta kan man tycka att t ex n är för litet för att man skall kunna hänvisa till CGS.

Exempel 3.21 Om vi bara antar för våra 10 livslängder i exempel 3.1 att de är utfall av oberoende likafördelade variabler så kan vi alltså tillgripa en approximativ metod som innebär användning av Centrala gränsvärdessatsen. Vi antar alltså att $\bar{X} = \sum_1^{10} X_i/10$ är approximativt normalfördelad $\mathrm{N}(\theta, \sigma^2/10)$ och vi ser att vi tvingas skatta σ^2 med $s^2(x_1, \cdots, x_{10})$ dvs $s^2 = 41.9$ och sedan anta att $\bar{X}$ är approximativt $N(\theta, 41.9/10) = N(\theta, 4.19)$. Vi har här alltså ett medelfel $s/\sqrt{10} = \sqrt{4.19}$. Vi använder detta för att få

$$1 - \alpha \approx P\left(-\lambda_{\alpha/2} \leq \frac{\bar{X} - \theta}{\sqrt{4.19}} \leq \lambda_{\alpha/2}\right)$$

som ger med $1-\alpha$ =90% intervallet $\bar{x} \pm \lambda_{0.05}\sqrt{4.19} \approx 6.05 \pm 3.37 = (2.68, 9.42)$.

Detta intervall känns ju otroligt osäkert eftersom

1) n=10 är mycket litet för att motivera användning av CGS som ju är ett gränsvärdesresultat som gäller då $n \to \infty$

2) skevheten i observationerna antyder att CGS-approximationen är ytterst dålig

3) intervallet är symmetriskt kring $\bar{x} = 6.05$ medan data antyder en skevhet i fördelningen.

4) det är tänkbart att intervallets undre gräns kan bli negativ (om s vore lite större).

□

3.6.3 Exponentialfördelade livslängder

Om vi kan anta att våra 10 livslängder kommer från en specifik parametrisk familj kan vi konstruera att "exakt" konfidensintervall, dvs som har exakt konfidensgrad om modellen är korrekt!

Vi antar därför att $X_1, \cdots, X_{10}$ är $\mathrm{Exp}(\theta)$ och om vi skattar θ med

$$\widehat{\theta}(x_1, \cdots, x_{10}) = \bar{x}$$

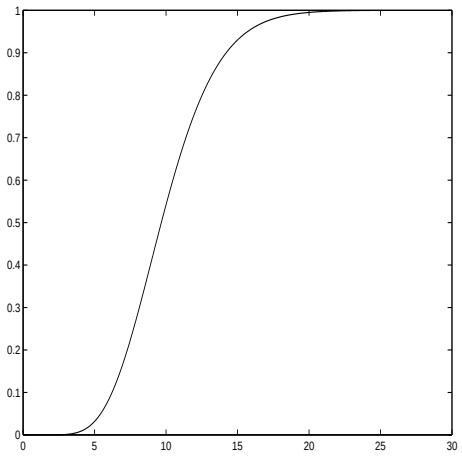
så ser vi i exempel 2.3 på sid 9 att $\sum_1^{10} X_i/\theta$ är $\Gamma(10,1)$-fördelad.
Vi tar talen $\Gamma_{\alpha/2}(10,1)$ respektive $\Gamma_{1-\alpha/2}(10,1)$ så att

$$G(\Gamma_{\alpha/2}(10,1)) = \alpha/2 \text{ och } G(\Gamma_{1-\alpha/2}(10,1)) = 1-\alpha/2$$

där G är fördelningsfunktionen för $\Gamma(10,1)$-fördelningen, dvs

$$G(z) = 1 - \sum_{k=0}^{9} \frac{z^k}{k!} e^{-z},\ z > 0$$

som finns i figur 3.4. $\Gamma_{0.05}(10,1)$ respektive $\Gamma_{0.95}(10,1)$ ser man i figur 3.4 är ca 5 respektive ca 15. Percentilerna beräknas "exakt" i Matlab med anropen `gaminv(0.95,10,1)` respektive `gaminv(0.05,10,1)`. Funktionen `gaminv` finns i modulen `Stats` i Matlab. Resultatet blir 5.4254 respektive 15.7052.



Figur 3.4: Fördelningsfunktion för $\Gamma(10,1)$-fördelningen, dvs för summan av 10 oberoende Exp(1)-fördelade stokastiska variabler

Detta ger

$$1-\alpha = P\left(\Gamma_{\alpha/2}(10,1) \le \frac{\sum_1^{10} X_i}{\theta} \le \Gamma_{1-\alpha/2}(10,1)\right) =$$

$$= P\left(\frac{10\bar{X}}{\Gamma_{1-\alpha/2}(10,1)} \le \theta \le \frac{10\bar{X}}{\Gamma_{\alpha/2}(10,1)}\right)$$

dvs vi får konfidensintervallet (90%-igt)

$$(10\bar{x}/\Gamma_{0.95}(10,1), 10\bar{x}/\Gamma_{0.05}(10,1)) \approx (10\cdot 6.05/15.7052, 10\cdot 6.05/5.4254) \approx$$

$$\approx (3.85, 11.15)$$

Man kan här notera att

1) Intervallet är exakt om exponentialfördelningsantagandet är korrekt. Skulle data komma från en annan fördelning så är det inte alls säkert att intervallet är speciellt bra. Intervallet byggde ju helt på att $\sum_{i=1}^{10} X_i/\theta$ var $\Gamma(10,1)$-fördelad. Vi visste ju att X_i/θ var Exp(1)-fördelad och att därför $\sum_1^{10} X_i/\theta$ var $\Gamma(10,1)$.

2) Vi fick vara lite kunniga (och ha lite tur) när vi kunde få fram fördelningen för $\sum_1^{10} X_i/\theta$. Hade vi som modell haft någon annan fördelning än exponentialfördelningen Exp(1) för X_i/θ hade det kunnat vara besvärligt att ta fram $\alpha/2$ och $1-\alpha/2-$punkterna i fördelningen för $\sum_1^{10} X_i/\theta$.

3) Även om vi inte skulle kunna få fram den exakta fördelningen för $\sum_1^{10} X_i/\theta$ för någon helt specificerad fördelning för X_i/θ:na så skulle vi ju kunna simulera denna, men detta skulle vara helt omöjligt om vi inte ens känner fördelningen för X_i/θ.

Kapitel 4

Bootstrap

Vi skall i detta kapitel beskriva metodiken och de bakomliggande idéerna bakom bootstrap och i kapitel 5 samt 6 visas hur denna ytterst kraftfulla metodik kan tillämpas i ett antal exempel.

4.1 Idéer bakom Bootstrap

Vi såg att vi fick våra numeriska observationer $x_1, x_2, \cdots, x_n$ ur vår bakomliggande fördelning F genom att de sågs som utfall av $X_1, X_2, \cdots, X_n$ som var oberoende likafördelade med fördelningen F. Vi hade den numeriska skattningen $\widehat{\theta}(x_1, x_2, \cdots, x_n)$ och motsvarande stokastiska variabel $\widehat{\theta}(X_1, X_2, \cdots, X_n)$, dvs stickprovsvariabeln. Vi använder $\widehat{\theta}$ som skattning av $\theta = T(F)$, där alltså parametern θ ses som ett uttryck (funktional) i F.

Vi är intresserade av

$$P\left(\widehat{\theta}(X_1, X_2, \cdots, X_n) \leq z\right)$$

eller kanske av

$$P\left(\widehat{\theta}(X_1, X_2, \cdots, X_n)/\theta \leq z\right)$$

eller kanske av

$$G(z) = P\left(\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta \leq z\right)$$

I det följande koncentrerar vi oss på $G(z)$. Om vi kunde generera nya friska datauppsättningar av storlek n ur F skulle vi kunna få fram fördelningen för $\widehat{\theta}(X_1, X_2, \cdots, X_n)$ beskriven av t ex $G(z)$ genom flitig simulering. Detta är ju inte möjligt eftersom vi inte kände fördelningen F och därmed inte heller θ – vi såg ju θ som en funktional $T(F)$ av fördelningen F.

Vad man nu gör är att skaffa sig en "kopia" av det ursprungliga försöket (som ju innebar lottning med hjälp av den okända fördelningen F) genom att skatta F med $\widehat{F}_n$ på lämpligt sätt ur data och sen lotta fram nya fiktiva datauppsättningar med hjälp av $\widehat{F}_n$.

Genom att denna "kopia" av den ursprungliga datagenereringen är helt specificerad behöver vi inte ens utföra några kalkyler för att få fram fördelningen

av stickprovsvariabeln i kopian, eftersom vi har full kontroll över slumpmekanismen, ty vi känner $\widehat{F}_n$. Detta betyder att vi kan simulera denna kopia och därigenom ta reda på stickprovsvariabelns fördelning i kopian precis hur bra som helst genom flitig simulering.

Vi ser alltså $G(z)$ som en (komplicerad) funktional av F som vi kan kalla $S_z(F)$. Eftersom θ också är en funktional av F som vi tidigare kallat $T(F)$ ser vi att

$$S_z(F) = G(z) = P(\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta \leq z) = P(\widehat{\theta}_F - T(F) \leq z)$$

där vi betecknat $\widehat{\theta}(X_1, X_2, \cdots, X_n)$ med $\widehat{\theta}_F$ för att understryka att just F är den fördelning som genererade $X_1, X_2, \cdots, X_n$. Notera att på motsvarande sätt är $\widehat{\theta}_{\widehat{F}_n}$ stickprovsvariablen $\widehat{\theta}$ fast applicerad på variabler som genereras av $\widehat{F}_n$ i stället för från F.

Vi är egentligen intresserade av $G(z)$ dvs $S_z(F)$ och F har vi information om i form av skattningen $\widehat{F}_n$ och vi tror nu att $\widehat{F}_n$ tillräckligt mycket liknar F så att t ex funktionalen $S_z(F) \approx S_z(\widehat{F}_n)$. Detta i analogi med att vi tror att $\widehat{\theta}(x_1, x_2, \cdots, x_n)$ väl approximerar θ – det är ju därför vi skattar θ med $\widehat{\theta}$.

Vi kommer genomgående att beteckna variabler genererade av $\widehat{F}_n$ med X_1^*, X_2^*, $\cdots$, X_n^* och på motsvarande sätt beteckna observationer (dvs numeriska utfall av dessa variabler) med $x_1^*, x_2^*, \cdots, x_n^*$. Vi låter också $\widehat{\theta}^*$ beteckna stickprovsvariabeln då data genererats från fördelningen $\widehat{F}_n$ dvs

$$\widehat{\theta}^* = \widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*).$$

På samma sätt som vi ser variablerna $X_1, X_2, \cdots, X_n$ och de numeriska observationerna $x_1, x_2, \cdots, x_n$ som genererade av (den sanna fördelningen) F ser vi $X_1^*, X_2^*, \cdots, X_n^*$ respektive $x_1^*, x_2^*, \cdots, x_n^*$ som variablerna respektive observationerna genererade av $\widehat{F}_n$.

Definition 4.1 *Grundläggande bootstrap-hypotes*
Vi antar att "kopian" där data genereras av $\widehat{F}_n$ tillräckligt liknar det ursprungliga försöket där data genererades av F att $S(\widehat{F}_n) \approx S(F)$ för en "vid" klass av funktionaler S. □

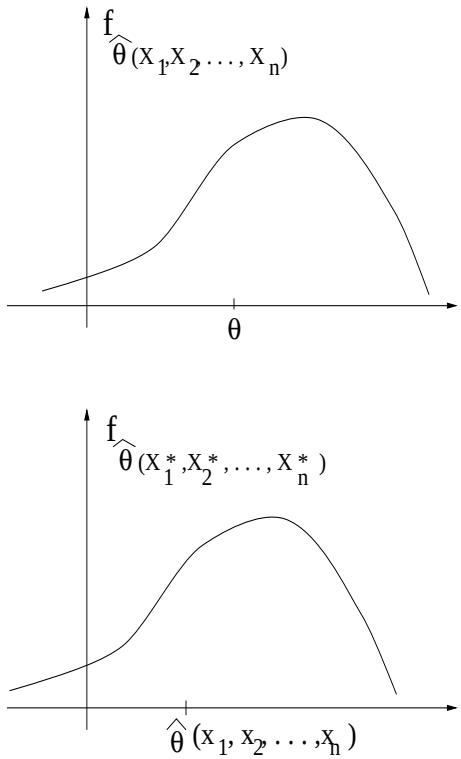
Detta innebär att vi tror att

$$P(\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta \leq z) = P(\widehat{\theta}_F - T(F) \leq z) \approx$$

$$\approx P(\widehat{\theta}_{\widehat{F}_n} - T(\widehat{F}_n) \leq z) = P(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) - \widehat{\theta}(x_1, x_2, \cdots, x_n) \leq z)$$

där vi antagit att $\widehat{\theta}$ är en plug-in-skattning, dvs att $\widehat{\theta}(x_1, x_2, \cdots, x_n) = T(\widehat{F}_n)$.

Vi antar alltså att fördelningen för $\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta$ väl approximeras av fördelningen för $\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) - \widehat{\theta}(x_1, x_2, \cdots, x_n)$, där $X_1^*, X_2^*, \cdots, X_n^*$ är



Figur 4.1: Jämförelse av fördelningarna för $\widehat{\theta}(X_1, X_2, \cdots, X_n)$ och $\widehat{\theta}^* = \widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$

generade av $\widehat{F}_n$ (givet observationerna $x_1, x_2, \cdots, x_n$). Detta illustreras i figur 4.1.
Eftersom fördelningarna är approximativt lika kommer alla aspekter av dessa fördelningar att likna varandra, som t ex väntevärde, varians, percentiler, median etc.

Senare kommer vi att ge en del resultat som visar att denna grundläggande bootstrap-hypotes faktiskt är sann i vissa situationer som gränsvärdesresultat när $n \to \infty$ då

1) skattningen är av enkel typ, t ex $\bar{x}$ eller median$(x_1, x_2, \cdots, x_n)$

2) den bakomliggande fördelningen F är enkel, t ex bara lägger massa i ett ändligt antal punkter.

Denna typ av satser är dock av ett rent teoretiskt intresse eftersom vi vill kunna använda bootstrap just för ett ändligt n och då situationen är så komplicerad att satserna ej är tillämpliga.

4.2 Icke-parametrisk och parametrisk bootstrap

Skattningen $\widehat{F}_n$ av F kan göras på olika sätt

Definition 4.2 *Icke-parametrisk bootstrap*
Icke-parametrisk bootstrap innebär att vi skattar F med

$$\widehat{F}_n = \text{empiriska fördelningen för våra observationer,}$$

dvs den sannolikhetsfördelning som lägger massan $1/n$ i vardera av de n observationerna $x_1, x_2, \cdots, x_n$. □

Definition 4.3 *Parametrisk bootstrap*
Parametrisk bootstrap är tillämplig då vi tror att F tillhör en parametrisk familj med en eller flera parametrar. Vi skattar numeriskt de ingående parametrarna och låter $\widehat{F}_n$ vara den medlem i den parametriska familjen som ges av just den fördelningen. □

Exempel 4.1 *Parametrisk bootstrap i exponentialfördelning*
Om vi antar att X_i är Exp(θ) så tar vi med parametrisk bootstrap $\widehat{F}_n$ som Exp($\bar{x}$)-fördelningen. Nya data skulle i livslängdsexemplet (exempel 3.1) alltså genereras från Exp(6.05)-fördelningen eftersom vi skattade θ med $\bar{x} = 6.05$.□

Exempel 4.2 *Parametrisk bootstrap för korrelationskoefficient-skattning*
Vi vill skatta korrelationskoefficienten ρ ur två-dimensionella data $(x_1, y_1), (x_2, y_2), \cdots, (x_{20}, y_{20})$ med

$$\widehat{\rho} = \frac{\sum_{i=1}^{20}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{20}(x_i - \bar{x})^2 \sum_{i=1}^{20}(y_i - \bar{y})^2}}$$

och antag att vi erhåller $\widehat{\rho} = 0.49$. Vi tror nu att data kommer från en tvådimensionell normalfördelning och vill utföra parametrisk bootstrap. Vi genererar då nya datauppsättningar med hjälp av

$$N_2\left(\begin{pmatrix}\bar{x}\\ \bar{y}\end{pmatrix}, \begin{pmatrix}\frac{1}{20}\sum_1^{20}(x_i-\bar{x})^2 & \frac{1}{20}\sum_1^{20}(x_i-\bar{x})(y_i-\bar{y})\\ \frac{1}{20}\sum_1^{20}(x_i-\bar{x})(y_i-\bar{y}) & \frac{1}{20}\sum_1^{20}(x_i-\bar{x})^2\end{pmatrix}\right).$$

I enlighet med en tidigare kommentar i avsnitt 2.9 på sidan 14 kan vi för en analys av fördelningen för $\widehat{\rho}$ lika gärna generera nya data med

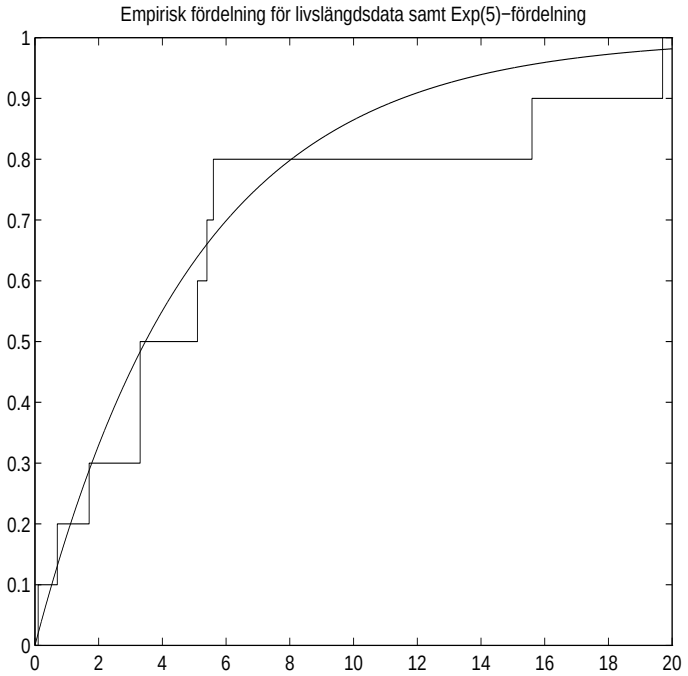
$$N_2\left(\begin{pmatrix}0\\ 0\end{pmatrix}, \begin{pmatrix}1 & 0.49\\ 0.49 & 1\end{pmatrix}\right).$$

Vi använder alltså denna fördelning och genererar nya data bestående av 20 talpar och skattar ρ i denna datauppsättning. Detta förfarande upprepas sen B gånger och resultatet utgör en (godtyckligt bra) approximation av den parametriska bootstrap-fördelningen. □

Parametrisk bootstrap är mest tillämplig (och intressant) när vi har en komplicerad stickprovsvariabel som vi inte klarar av att analytiskt ta fram fördelningen för även för helt specificerade F.

Vi kommer mest att använda icke-parametrisk bootstrap och bara då och då den parametriska som ju lider kraftigt av att den är helt avhängig av att den statistiska modellen är helt specificerad så när som på en eller ett par parametrar.

För livslängdsdata ser empiriska fördelningens fördelningsfunktion ut enligt figur 4.2 där även fördelningsfunktionen för Exp(5)-fördelningen ritats in - livslängderna har i verkligheten lottats fram enligt den fördelningen.



Figur 4.2: Empirisk fördelning för livslängdsdata samt för Exp(5)-fördelning

Vid en icke-parametrisk bootstrap lottas alltså nya datauppsättningar fram i enlighet med den empiriska fördelningen för observationerna. Detta i kontrast mot det ursprungliga försöket (som vi försöker efterlikna) där data lottats fram enligt Exp(5)-fördelningen i livslängdsexemplet. Eftersom empiriska fördelningen ganska väl avspeglar den sanna fördelningen (åtminstone om antalet observationer n är någorlunda stort – i verkligheten gäller ju att empiriska fördelningen konvergerar mot den sanna fördelningen då $n \to \infty$) borde data genererade enligt den empiriska fördelningen likna de data vi skulle få med den sanna bakomliggande fördelningen. Det är detta som är essensen i bootstrap–hypotesen. I en verklig situation känner vi inte den bakomliggande fördelningen – vi vet inte ens vilken parametrisk familj den tillhör. All vår

information om fördelningen kommer från vårt stickprov och den empiriska fördelningen innehåller all denna information.

Lottning enligt den empiriska fördelningen $\widehat{F}_n$ innebär att välja värdena $x_1, x_2, \cdots, x_n$ vardera med sannolikhet $1/n$. Att lotta upprepade gånger är samma sak som att dra värdena med återläggning. Detta innebär att framlottning av nya data med hjälp av empiriska fördelningen är mycket enkel. I Matlabs `Stats`-modul utför funktionen `bootstrp` detta och beräknar också skattningen för de på detta sätt framlottade stickproven.

4.3 Icke-parametrisk bootstrap

Vi genererar alltså ett nytt stickprov av storlek n genom att använda empiriska fördelningen $\widehat{F}_n$ som lägger massan $1/n$ i vardera av de ursprungliga observationerna $x_1, x_2, \cdots, x_n$. Detta stickprov kallas bootstrap-stickprovet och vi betecknar det genomgående med $x_1^*, x_2^*, \cdots, x_n^*$. Vi betraktar detta som utfall av $X_1^*, X_2^*, \cdots, X_n^*$ där alltså

$$P(X_i^* = x_k) = 1/n \text{ för } k = 1, 2, \cdots, n \text{ och } i = 1, 2, \cdots, n$$

Som påpekats betyder det att $x_1^*, x_2^*, \cdots, x_n^*$ fås genom dragning med återläggning ur de ursprungliga observationerna $x_1, x_2, \cdots, x_n$. Denna framlottning är lätt att genomföra rent praktiskt i t ex Matlab – mer om detta i avsnitt 4.4 på sidan 45.

Fördelningen för $X_1^*, X_2^*, \cdots, X_n^*$ beror av våra observationer $x_1, x_2, \cdots, x_n$. Detta betyder att olika aspekter av fördelningen för $X_1^*, X_2^*, \cdots, X_n^*$, t ex fördelningen för $\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$ (eller dess väntevärde, varians, median, 90%-percentil etc) är en komplicerad funktion av $x_1, x_2, \cdots, x_n$. Det explicita uttrycket för denna funktion behöver vi dock aldrig i praktiken härleda eftersom vi lätt kan simulera fördelningen för $\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$.

Det faktum att t ex väntevärdet $E(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*))$, dvs egentligen det betingade väntevärdet

$$E\left(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) \middle| X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n\right)$$

är en funktion av $x_1, x_2, \cdots, x_n$ innebär att det i sig är en punktskattning. Att dennas värde i praktiken inte beräknas exakt utan bara med hjälp av simuleringar är egentligen ointressant – det centrala är om den kan visas skatta något intressant (i detta fall ge information om det systematiska felet för $\widehat{\theta}$). På samma sätt kommer standardavvikelsen $D(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*))$ att skatta standardavvikelsen $D(\widehat{\theta}(X_1, X_2, \cdots, X_n))$ dvs vi kommer att erhålla en skattning av medelfelet med hjälp av bootstrap. Detta medelfel beror (naturligtvis) av våra ursprungliga observationer $x_1, x_2, \cdots, x_n$. Denna i allmänhet mycket komplicerade funktion tar vi inte fram explicit utan simulerar fram med hjälp

av bootstrap-stickproven. Notera att $D(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*))$ skall tolkas som den betingade standardavvikelsen

$$D\left(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)\middle| X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n\right).$$

Vi är som alltid främst intresserade av fördelningen för stickprovsvariabeln dvs för $\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$. Observera att stickprovet $X_1^*, X_2^*, \cdots, X_n^*$ genererat av $\widehat{F}_n$ då betraktas som slumpmässigt precis som i den ursprungliga stickprovsvariabeln $\widehat{\theta}(X_1, X_2, \cdots, X_n)$ där data genererades av F. Observationerna $x_1, x_2, \cdots, x_n$ betraktas härvid som fixa och givna.

Den exakta fördelningen för $\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$ är i allmänhet i stort sett omöjlig att få fram exakt om inte n är mycket litet, men det fina är att vi lätt kan simulera den hur bra som helst eftersom vi har full kontroll över slumpmekanismen.

Detta betyder att fördelningen för $\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$ kan beräknas hur bra som helst oavsett hur komplicerad $\widehat{\theta}$ än är! Man lottar fram B st (kanske $B = 1000$ eller $B = 10000$) datauppsättningar med hjälp av $\widehat{F}_n$, vardera av storlek n, skattar θ med hjälp av $\widehat{\theta}$ för var och en av dessa och får på så sätt B st observationer av $\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$. Ett histogram över dessa approximerar den sanna fördelningen för $\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$ hur bra som helst bara vi tar B tillräckligt stor.

Dock kan komplicerade skattningar ta lång tid att beräkna och detta kan lägga hinder i vägen för att ta B hur stor som helst. Vidare måste dessa skattningar lagras i minnet och det kan också ställa till problem. Själva framlottningen av stickprovet görs med dragning utan återläggning och detta brukar inte vara någon begränsande faktor rent datatekniskt t ex i Matlab eftersom det är en så enkel operation.

När väl fördelningen för $\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$ approximerats tillräckligt väl kan man sen lätt (approximativt eftersom B är ändligt) beräkna godtyckliga aspekter av denna fördelning, t ex väntevärdet, standardavvikelsen, medianen eller en percentil i fördelningen.

4.3.1 Ett trivialt exempel

Följande är ett mycket trivialt exempel på icke-parametrisk bootstrap men som ändå kan vara illustrativt.

Vi studerar en opinionsundersökningssituation där parametern p=andelen som skulle svara "ja" på en enkel ja/nej-fråga. Vi tar ett urval om t ex 1000 personer ur den (oändliga populationen) och erhåller svaren $x_1, x_2, \cdots, x_{1000}$ där $x_i = 1$ om person i svarat "ja" och $x_i = 0$ annars. Som modell ser man dessa svar som utfall av $X_1, X_2, \cdots, X_{1000}$ som antas oberoende likafördelade med $P(X_i = 1) = p$ och $P(X_i = 0) = 1-p$. Man inser att $Y = X_1+X_2+\cdots+X_{1000}$ är $\text{Bin}(1000, p)$.

Antag att vi fått $y = 300$ "ja"-svar. Vi skattar p med $\widehat{p} = 300/1000 = 0.3$ och motsvarande stickprovsvariabel är $\widehat{p} = Y/1000$ och man inser att $E(Y/1000) = p$ och $V(Y/1000) = p(1-p)/1000$, dvs att skattningen är väntevärdesriktig och att standardavvikelsen är $D(Y/1000) = \sqrt{p(1-p)/1000}$. Denna standardavvikelse innehåller ju tyvärr den okända parametern och som mått på osäkerheten anger man då medelfelet $d(\widehat{p}) = \sqrt{\widehat{p}(1-\widehat{p})/1000} = \sqrt{0.3 \cdot 0.7/1000} \approx 0.0145$ som kan ses som ett mått på osäkerheten i skattningen $\widehat{p} = 0.30$. Med hjälp av normalapproximation av binomialfördelningen skulle man t ex kunna ge det approximativt 95%-iga konfidensintervallet $0.30 \pm 1.9600 \cdot 0.0145 = 0.30 \pm 0.0284$.

Med bootstrap skulle detta kunna utföras på följande alternativa sätt. Vi skapar en fiktiv population bestående av vårt stickprov, dvs med 300 st 1:or och 700 0:or. Vi drar sedan ett stort antal (säg B=10000) stickprov med återläggning vardera av storlek 1000 ur denna fiktiva population. Notera att detta är precis vad vi gör då vi använder oss av icke-parametrisk bootstrap. Vi skattar p i vardera av dessa B stickprov och erhåller skattningarna $\widehat{p}_1^*, \widehat{p}_2^*, \cdots, \widehat{p}_B^*$. Man inser att dessa är oberoende och har samma fördelning som $Y^*/1000$ där Y^* är $\text{Bin}(1000, 0.30)$. Alltså har de en standardavvikelse som är $\sqrt{0.30 \cdot 0.7/1000} = 0.0145$ dvs samma värde som dök upp som medelfel i ovanstående "traditionella" analys. Detta resultat kan vi faktiskt få fram helt mekaniskt och utan någon som helst kunskap om vare sig binomialfördelning eller statistisk analys överhuvudtaget genom att beräkna spridningen av de numeriska skattningarna $\widehat{p}_1^*, \widehat{p}_2^*, \cdots, \widehat{p}_B^*$ om vi har B mycket stort genom att beräkna

$$\widehat{se}_{boot,B} = \sqrt{\frac{1}{B-1}\sum_{k=1}^{B}(\widehat{p}_k^* - \widehat{p}_\cdot^*)^2}$$

där $\widehat{p}_\cdot^* = \frac{1}{B}\sum_1^B \widehat{p}_k^*$. Vi ser också att om $B \to \infty$ kommer $\widehat{se}_{boot,B} \to 0.0145$ dvs det "sanna" medelfelet!

I denna situation är det naturligtvis helt onödigt att utföra denna bootstrap eftersom vi kan räkna exakt på situationen. Vi kunde faktiskt här t o m analysera de statistiska egenskaperna hos bootstrap-skattningarna men formeln ovan för $\widehat{se}_{boot,B}$ fungerar oavsett detta! Principen att vi skapar en kopia av den ursprungliga lottningen ur populationen (som gav upphov till våra 1000 svar) genom att skapa en situation med en känd sammansättning av populationen bestämd av våra observationer (dvs med 30% "ja"-svar) där vi kan flitigt upprepa lottningsförfarandet och där vi kan analysera spridningen av dessa lottningsresultat vad gäller den studerade skattningen är i all enkelhet tanken bakom bootstrap.

Vi skapar alltså med hjälp av det ursprungliga stickprovet en fiktiv population ur vilken vi skapar ett stort antal nya stickprov. Dessa stickprov analyseras med hjälp av den punktskattning vi valt för det ursprungliga stickprovet. Vi erhåller då ett stort antal skattningar och genom att studera dessa

(t ex vad gäller variabilitet, genomsnitt, percentiler etc) kan vi få en uppfattning om den ursprungliga punktskattningens statistiska egenskaper. T ex skulle vi kunna göra konfidensintervall utan att fundera på normalapproximationer etc utan i stället använda oss av percentiler hämtade ur det numeriska material bootstrap-skattningarna gett upphov till. Denna metodik är i ovanstående situation utan intresse, men notera att metodiken fungerar oavsett hur komplicerad skattningen än är. Det är här styrkan i bootstrap kommer att visa sig!

4.4 Bootstrap i Matlab

Matlab har en som nämnts en rutin `bootstrp` som utför bootstrap, dvs som ur ett givet stickprov lottar fram nya stickprov medelst dragning med återläggning och applicerar den valda punktskattnings-funktionen på dessa framlottade stickprov. Den ingår i modulen `Stats` som innehåller en mängd `m`-filer för simulering, statistisk analys etc.

Följande är resultatet av kommandot `help bootstrp`:

```
BOOTSTRP Bootstrap statistics.
   BOOTSTRP(NBOOT,BOOTFUN,D1,...) draws NBOOT bootstrap data
       samples and analyzes them using the function, BOOTFUN.
       NBOOT must be a positive integer.
   BOOTSTRAP passes the (data) D1, D2, etc. to BOOTFUN.

   [BOOTSTAT,BOOTSAM] = BOOTSTRP(...) Each row of BOOTSTAT
       contains the results of BOOTFUN on one bootstrap sample.
       If BOOTFUN returns a matrix,
   then this output is converted to a long vector
       for storage in BOOTSTAT.
   BOOTSAM is a matrix of indices into the row
```

Ett typiskt anrop när våra observationer finns i vektorn `x` och vi vill göra 1000 bootstrap-stickprov är alltså

```
boot=bootstrp(1000,'estimate',x);
```

där funktionen (`m`-filen) `estimate` skattar den intresanta storheten θ ur datavektorn `x` med anropet `estimate(x)`. Vektorn `boot` innehåller alltså de 1000 skattningarna, dvs Matlab har genererat 1000 stickprov ur vektorn `x` vardera av samma storlek som `x`, har sedan applicerat funktionen `estimate` på vart och ett av dessa "fiktiva" stickprov och sen lagrat resultatet i vektorn `boot`.

Om skattningen $\widehat{\theta}(x_1, x_2, \cdots, x_n)$ utförs av funktionen `estimate` med anropet `estimate(x)` får man alltså den simulerade bootstrap-fördelningen i vektorn `boot`. Är man sedan intresserad av en skattning av medelfelet för $\widehat{\theta}$ får man den genom

```
se_boot=std(boot)
```

eftersom detta beräknar standardavvikelsen för `boot`, dvs standardavvikelsen i bootstrap-fördelningen. Är vi i stället intresserade av väntevärdet i bootstrap-fördelningen får vi detta genom `mean(boot)` och skulle vi vilja ha 5%-percentilen i bootstrap-fördelningen får man det genom `prctile(boot,5)`. Vill man se hela bootstrap-fördelningen kan man använda sig av kommandot `hist(boot)`. Dessa utgör i och för sig bara approximationer av motsvarande storheter i bootstrap-fördelningen eftersom vi endast gjort ett ändligt antal (t ex 1000) bootstrap-genereringar. Detta approximationsfel kan dock göras hur litet som helst genom att göra tillräckligt många bootstrap-genereringar. Den enda begränsande faktorn är vårt tålamod och möjligen minneskapaciteten hos vår dator. Notera speciellt att det inte spelar någon roll hur komplicerad $\widehat{\theta}$ är – det centrala är att vi har en funktion `estimate` som beräknar den för olika stickprov.

Den sanna bootstrap-fördelningen kan alltså erhållas precis hur noggrant som helst, men man bör påpeka att vad vi egentligen är intresserade av är motsvarande storheter för den bakomliggande fördelningen som genererat våra ursprungliga observationer. Vad bootstrap-hypotesen säger är att dessa storheter för den bakomliggande fördelningen väl approximeras med motsvarande storheter i bootstrap-fördelningen. Vi återkommer senare med en analys i enkla situationer av om denna hypotes fungerar.

4.5 Mer om bootstrap-fördelningen

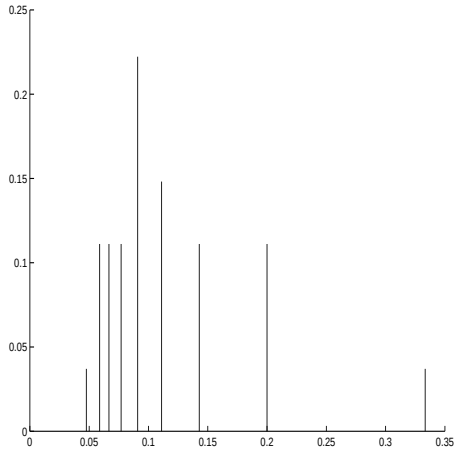
Om vi har n observationer $x_1, x_2, \cdots, x_n$ kan vi ur detta generera n^n tänkbara (icke-parametriska) bootstrap-stickprov om vi tar hänsyn till ordningen. Dessa har vardera sannolikheten $1/n^n$. I princip skulle en exakt fördelning för $\widehat{\theta}(X_1^*, \cdots, X_n^*)$ kunna beräknas genom rå kalkyl, där man beräknar dess värde för vart och ett av de n^n tänkbara argument-uppsättningarna.

Exempel 4.3 *Exakt bootstrap fördelning*
Om vi fått observationerna (x_1, x_2, x_3) =(1,3,7) så kan vi konstruera 3^3 bootstrapstickprov nämligen:

(1,1,1), (1,1,3), (1,1,7), (1,3,1), (1,3,3), (1,3,7), (1,7,1), (1,7,3), (1,7,7),
(3,1,1), (3,1,3), (3,1,7), (3,3,1), (3,3,3), (3,3,7), (3,7,1), (3,7,3), (3,7,7),
(7,1,1), (7,1,3), (7,1,7), (7,3,1), (7,3,3), (7,3,7), (7,7,1), (7,7,3), (7,7,7)

som vardera har sannolikheten $1/3^3 = 1/27$. Om vi t ex har konstruerat skattningen $\widehat{\theta} = 1/(x_1 + x_2 + x_3)$ har vi alltså för $x_1^* + x_2^* + x_3^*$ de tänkbara värdena 3, 5, 7, 9, 11, 13, 15, 17 och 21 vardera med sannolikheterna 1/27, 3/27, 3/27, 4/27, 6/27, 3/27, 3/27, 3/27 och 1/27 och fördelningen för $\widehat{\theta}^*$ blir som i figur 4.3.

□



Figur 4.3: Sann bootstrap-fördelning

Notera att ovanstående innebär att det för $n = 10$ finns 10^{10}=10 miljarder tänkbara stickprov vilket innebär att det i praktiken är omöjligt att beräkna den exakta bootstrap-fördelningen. Många av dessa stickprov är bara permutationer av varandra och detta skulle kunna användas för att minska beräkningsbördan speciellt om $\widehat{\theta}$ är symmetrisk i sina argument, vilket i denna situation är lämpligt – skattningproblemet är ju formulerat helt permutationsinvariant. Följande sats visar att antalet distinkta bootstrap-stickprov är $\binom{2n-1}{n}$ om alla observationerna är distinkta.

Sats 4.1 *Dragning med återläggning utan hänsyn till ordning.*
Vi har N distinkta föremål föremål och drar n med återläggning. Antalet distinkta sätt detta går att göra på om vi ej tar hänsyn till ordning är

$$\binom{N+n-1}{n}$$

Antalet distinkta bootstrap-stickprov reducerar sig till fallet $N = n$ dvs vi har maximalt $\binom{2n-1}{n}$ stycken olika bootstrap-stickprov. □

Som exempel kan vi ta tre föremål a, b, c och kan då få följande resultat då man ej tar hänsyn till ordning: (a, a, a), (a, a, b), (a, a, c), (b, b, b), (b, b, a), (b, b, c), (c, c, c), (c, c, a), (c, c, b) samt (a, b, c) dvs 10 st som ju stämmer med uttrycket för $n = 3$.

Detta "fjärde" fall av dragning med och utan återläggning, med och utan hänsyn till ordning brukar i allmänhet utelämnas i grundkurser.
Bevis:
Placera ut $N+1$ vertikala streck på en rad. Mellanrummen mellan dessa får stå för de olika elementen från mängden. Vi skall nu välja n st med återläggning

och detta kan vi symbolisera genom att placera ut n st $*$-or emellan strecken. T ex står $|**||*|$ för (a,a,c), $|***|||$ för (a,a,a) och $|*|*|*|$ för (a,b,c) i ovanstående exempel.

Det skall alltså placeras ut n st $*$-or och $N-1$ st streck mellan det vänstraste och det högraste strecket. Detta kan göras på

$$\binom{n+N-1}{n} \text{ sätt}$$

genom att vi väljer positioner för de n $*$-orna bland de $n+N-1$ möjliga positionerna. □

Notera att även om man skulle utnyttja denna symmetri fullt ut finns för $n=10$ totalt $\binom{19}{10}$=92378 distinkta stickprov och skattningen skall beräknas för samtliga dessa. Dessutom måste vi hålla reda på hur sannolika dessa distinkta stickprov är, dvs hur sannolika de är som resultat av bootstrap-genereringen – de är ju typiskt olika sannolika beroende på hur många distinkta värden de innehåller. I princip ger detta upphov till en fördelning med (säg) 92378 st punktmassor och alla dessa skall alltså beräknas. Det fiffiga är att denna komplicerade kalkyl inte behöver utföras utan vi simulerar i stället fördelningen.

4.6 Sammanfattning

Vi kan se det ursprungliga försöket som att den bakomliggande fördelningen F ger upphov till våra observationer $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$ och dessa ger vår skattning $\widehat{\theta}(\boldsymbol{x})$. För att kunna analysera denna situation vad avser t ex osäkerheten i skattningen $\widehat{\theta}(\boldsymbol{x})$ ser vi våra data som utfall av $\boldsymbol{X}=(X_1,X_2,\cdots,X_n)$ och vi ser alltså även skattningen $\widehat{\theta}(\boldsymbol{x})$ som ett utfall av stickprovsvariabeln $\widehat{\theta}(\boldsymbol{X})$. Man kan se det som att vi genom att införa modellens stokastiska $\boldsymbol{X}$ får en möjlighet att fundera över hur data kunde ha sett ut. För att bedöma t ex osäkerheten i $\widehat{\theta}(\boldsymbol{x})$ vill vi veta osäkerheten i $\widehat{\theta}(\boldsymbol{X})$. Hade vi möjlighet att simulera fördelningen för $\boldsymbol{X}$ skulle vi ju kunna få fram fördelningen för $\widehat{\theta}(\boldsymbol{X})$, men detta är ju inte möjligt eftersom detta skulle kräva att vi visste fördelningen F. Genom att skatta F med $\widehat{F}_n$ ur våra observationer $\boldsymbol{x}$ skapar vi en kopia av det ursprungliga försöket. Denna kopia kan simuleras godtyckligt många gånger.

Följande figur är en illustration till ovanstående:

$$\theta=T(F) \Leftarrow \quad F \Rightarrow \quad \boldsymbol{x}=(x_1,\cdots,x_n) \Rightarrow \quad \widehat{\theta}(\boldsymbol{x}) \qquad\qquad \text{medelfel } d(\widehat{\theta})$$

$$\theta=T(F) \Leftarrow \quad F \Rightarrow \quad \boldsymbol{X}=(X_1,\cdots,X_n) \Rightarrow \quad \widehat{\theta}(\boldsymbol{X}) \Rightarrow \quad P(\widehat{\theta}(\boldsymbol{X})\le z) \Rightarrow \quad D(\widehat{\theta}(\boldsymbol{X}))$$

$$\widehat{\theta}(\boldsymbol{x})=T(\widehat{F}_n) \Leftarrow \quad \widehat{F}_n \Rightarrow \quad \boldsymbol{X}^*=(X_1^*,\cdots,X_n^*) \Rightarrow \quad \widehat{\theta}(\boldsymbol{X}^*) \Rightarrow \quad P(\widehat{\theta}(\boldsymbol{X}^*)\le z) \Rightarrow \quad D(\widehat{\theta}(\boldsymbol{X}^*))$$

Vi vill få fram t ex medelfelet $d(\widehat{\theta})$ som skattning av $D(\widehat{\theta}(\boldsymbol{X}))$. För att få fram $D(\widehat{\theta}(\boldsymbol{X}))$ skulle vi behöva fördelningen $P(\widehat{\theta}(\boldsymbol{X}) \leq z),\ z \in R$. Denna skulle kunnat ha simulerats om F var känd, men det är den ju inte. Dock är $\widehat{F}_n$ är känd så där kan vi simulera fördelningen för $\widehat{\theta}(\boldsymbol{X}^*)$ genom att generera B bootstrapstickprov $\boldsymbol{x}^{*1}, \boldsymbol{x}^{*2}, \cdots, \boldsymbol{x}^{*B}$. Dessa ger värden $\widehat{\theta}(\boldsymbol{x}^{*1}), \widehat{\theta}(\boldsymbol{x}^{*2}), \cdots, \widehat{\theta}(\boldsymbol{x}^{*B})$ och dessas fördelning approximerar fördelningen för $\widehat{\theta}(\boldsymbol{X}^*)$ i princip hur bra som helst genom att välja B "stort". Det möjliggör att t ex beräkna $D(\widehat{\theta}(\boldsymbol{X}^*))$ med godtycklig precision. Enligt bootstrap-hypotesen kommer fördelningen för $\widehat{\theta}(\boldsymbol{X})$ att någorlunda väl approximeras av fördelningen för $\widehat{\theta}(\boldsymbol{X}^*)$. Alltså gäller även att $D(\widehat{\theta}(\boldsymbol{X}))$ väl approximeras av $D(\widehat{\theta}(\boldsymbol{X}^*))$. Vi kan alltså beräkna $D(\widehat{\theta}(\boldsymbol{X}^*))$ numeriskt ur våra bootstrap-simuleringar och använda detta som skattning av $D(\widehat{\theta}(\boldsymbol{X}))$, dvs som medelfelsskattning.

Kapitel 5

Några exempel på Bootstrap

5.1 Inledning

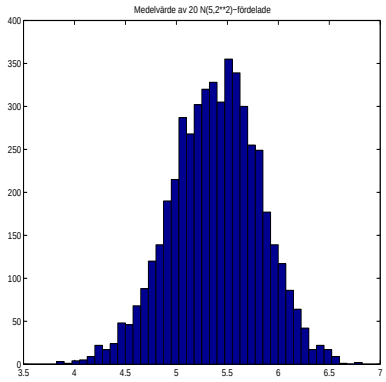
I detta kapitel presenterar vi ett antal enkla exempel på hur bootstrap kan användas och hur lätt det är att göra detta i Matlab. Fullt medvetet är de flesta exempel artificiella i så måtto att vi känner den sanna fördelningen för data och använder så enkla skattningar att det går att räkna fram den exakta fördelningen för stickprovsvariabeln eller åtminstone simulera den. Detta är fullt avsiktligt eftersom vi gärna vill kunna checka av att bootstrap ger ett någorlunda realistiskt resultat i situationer vi kan kontrollera. Man bör dock ha i minnet att den stora poängen med bootstrap är att vi vill kunna använda den i situationer där vi helt saknar denna möjlighet antingen därför att vi inte känner den bakomliggande fördelningen och/eller för att stickprovsvariabeln är så komplicerad att vi helt saknar möjlighet att räkna analytiskt på den. När vi ställs inför verkliga data är det ingen som berättar för oss vilken fördelning eller fördelningsfamilj våra data kommer från. Vi har bara en parameter vi vill studera och förhoppningsvis en punktskattning av denna. Att vi i de följande exemplen i allmänhet känner den "sanna" fördelningen för våra data skall på intet vis påverka hur vi analyserar data, men däremot är denna kunskap av stort intresse när det gäller att se om våra bootstrap-resultat verkar stämma.

5.2 Väntevärde i normalfördelning

Vi genererar 20 st $\mathrm{N}(5, 2^2)$-fördelade observationer och vill skatta θ=väntevärdet med $\bar{x}$ dvs med Matlabfunktionen `mean`. Utifrån detta stickprov genererar vi 5000 bootstrap-stickprov och skattar θ med $\bar{x}^*$, dvs beräknar aritmetiska medelvärdet i vart och ett av dessa 5000 stickprov om 20 observationer vardera.

I Matlab görs detta med

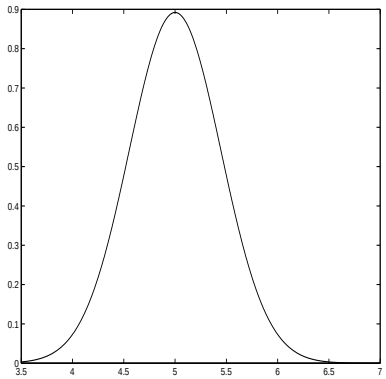
```
x=normrnd(5,2,1,20);
thetahat=mean(x)              % gav  5.3841
stdhat=std(x)                 % gav 1.9905
```



Figur 5.1: Bootstrapfördelning för medelvärde av 20 st $N(5, 2^2)$-fördelade

```
boot=bootstrp(5000,'mean',x); % skapar bootstrap-skattn.
mean(boot)                    % gav  5.3865
hist(boot,30)                 % ger histogram med 30 ‘‘fack’’
se_hat=std(boot)              % gav  0.4353
```

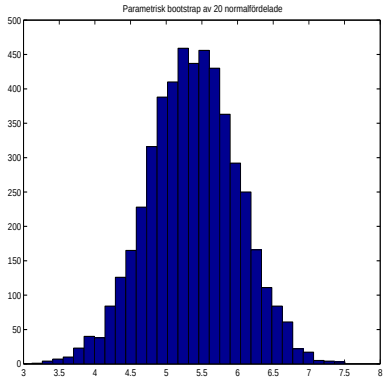
Rutinen `bootstrp` genererar alltså 5000 stickprov vardera av storlek 20 utifrån `x` och applicerar rutinen `mean` på vart och ett av dessa samt lagrar resultatet i vektorn `boot`. Att genomsnittet av de 5000 skattningarna i `boot` blev 5.3865 (dvs nästan $\bar{x} = 5.3841$) är ingen tillfällighet – det avspeglar att $\bar{x}$ är en väntevärdesriktig skattning av θ=väntevärdet.



Figur 5.2: Tätheten för N(5,1/5)-fördelningen (“facit”)

Att `se_hat`=0.4353 (dvs att spridningen av de 5000 bootstrap-skattningarna är 0.4353) avspeglar att $V(\bar{X}) = \sigma^2/20 = 4/20 = 1/5 \approx 0.4472^2$ eftersom `se_hat` är en skattning av $D(\bar{X})$. Detta syns i figur 5.1 genom att histogrammet över vektorn `boot` ser ut som den sanna fördelningen av $\bar{X}$, dvs $\mathrm{N}(5, 2^2/20)$=N(5,1/5)-fördelningen. Detta dock med den skillnaden att histogrammet borde ha sin tyngdpunkt i 5.3841 i stället för i 5. Bootstrap-genereringen har ju gjorts med hjälp av empiriska fördelningen för $x_1, x_2, \cdots, x_{20}$ som

hade sin tyngdpunkt i $\bar{x} = 5.3841$. P g a det begränsade antalet simuleringar hamnade den verkliga tyngdpunkten i histogrammet i 5.3865 och inte i 5.3841.



Figur 5.3: Parametrisk bootstrap av 20 normalfördelade

Den traditionella medelfelsskattningen $s/\sqrt{n}$ blir i vårt fall $1.9905/\sqrt{20}$=0.4451. I kapitel 7 kommer vi att visa att medelfels-skattningen $D(\bar{X}^*)$ uppfyller

$$D(\bar{X}^*) = \sqrt{\frac{n-1}{n}}\frac{s}{\sqrt{n}}$$

dvs i vårt fall $0.4451 \cdot \sqrt{19/20} = 0.4338$. Ur våra 5000 bootstrap-stickprov erhöll vi `se_hat`=0.4353. Orsaken till att vi inte fick exakt 0.4338 är att vi inte beräknat den exakta standardavvikelsen $D(\bar{X}^*)$ utan approximerat den med `se_hat` ur våra 5000 simuleringar. Hade vi gjort fler simuleringar hade vi kunnat undvika detta approximationsfel.

Skulle parametrisk bootstrap ha använts skulle vi ha genererat nya uppsättningar av data med hjälp av $\mathrm{N}(5.3841, 1.9905^2)$-fördelningen med Matlabkoden

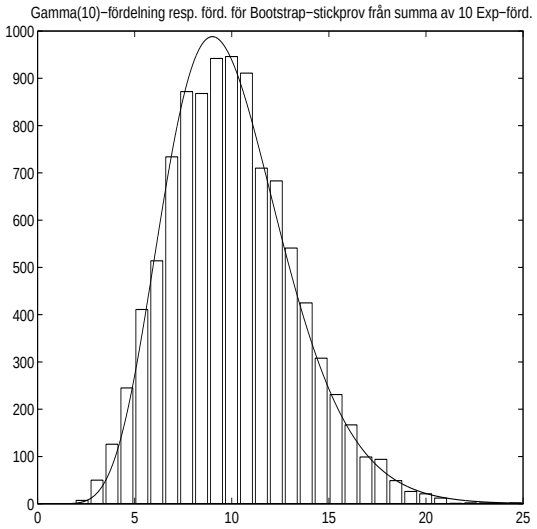
```
y=normrnd(5.3841,1.9905,10,5000);
estimate=mean(x);
hist(estimate,30)
```

som gav histogrammet 5.3. Den sanna fördelningen för $\bar{X}$ var i detta fallet $\mathrm{N}(5, 0.4472^2)) = N(5, 1/5)$-fördelningen (se figur 5.2). I förhållande till den sanna fördelningen är båda bootstrap-fördelningarna aningen förflyttade i sidled och något omskalade, men det viktiga är att fördelningsformen är i princip den korrekta. Detta med skift och omskalning är något vi kan komma tillrätta med genom att göra bootstrap av

$$(\bar{x} - \theta)/(s/\sqrt{n}) \text{ dvs av } (\bar{x}^* - \bar{x})/(s^*/\sqrt{n})$$

där $\bar{x}^*$ och s^* är aritmetiskt medelvärde respektive stickprovsstandardavvikelse för bootstrap-stickproven. Mer om detta i samband med konstruktion av konfidensintervall med pivot-metoden som beskrivs i kapitel 12.

5.3 Väntevärde i exponentialfördelning



Figur 5.4: $\Gamma(10,1)$-fördelning resp. bootstrap-fördelning för $\bar{X}^*/\bar{x}$

Vi använder de 10 livlängdsdata ur exempel 3.1 (i verkligheten alltså genererade från en Exp(5)-fördelning) och upprepar proceduren – notera att Matlabkoden är helt oförändrad. Vi erhöll `mean(x)=6.05` och `std(x)=6.4732` och histogrammet 5.4 där vi skalat om med 10/6.05 och för en jämförelse lagt in den "sanna" fördelningen dvs $\Gamma(10,1)$-fördelningen. Denna division med $\bar{x}$ har återigen samband med att $\bar{X}/\theta$ är en pivotvariabel för exponentialfördelningen – mer om detta i samband med pivot-metoden att konstruera konfidensintervall. Matlab-koden blir

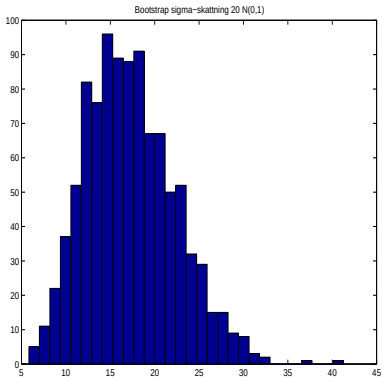
```
m=mean(x)
s=std(x)
boot=bootstrp(1000,'mean',x);
bootskal=boot/m ;
hist(bootskal,30)
```

För Exp(θ)-fördelningen utgör $\bar{X}/\theta$ en pivot-variabel, vars fördelning inte beror av θ och $\sum_1^n X_i/\theta$ har en $\Gamma(n,1)$-fördelning oavsett θ:s värde. Bootstrapfördelningen för $\bar{X}^*/\bar{x}$ kommer att likna fördelningen för pivot-variabeln och detta ger oss möjlighet att skatta percentiler i fördelningen för pivot-variabeln. Dessa percentiler ingår i konfidensintervallet för θ i denna situation – se avsnitt 3.6.3 på sidan 34. Detta fungerar på exakt samma sätt för alla skalinvarianta fördelningar, dvs sådana där X/θ har en viss ospecificerad men fix fördelning. Vi kommer alltså att kunna beräkna vettiga konfidensintervall för alla dessa situationer med samma metodik. Vi återkommer till vad detta har för konsekvenser i kapitel 6 som mer nogrant analyserar livslängdsexemplet

samt empiriskt undersöker hur bra dessa konfidensintervall blir vad avser den sanna konfidensgraden.

Om vi här skulle vilja använda oss av en parametrisk bootstrap i stället för icke-parametrisk bootstrap, dvs generera nya datauppsättningar av storlek 10 med hjälp av Exp($\bar{x}$)=Exp(6.05)-fördelningen och gjorde en motsvarande omskalning skulle vi (så när som på att ändligt antal simuleringar utförs) få precis exakt rätt fördelning!! Man bör dock betänka att detta beror på att den "sanna" fördelningen var just exponential-fördelningen medan den icke-parametriska varianten inte förutsatte någon kunskap alls om den "sanna" fördelningen.

5.4 Variansskattning



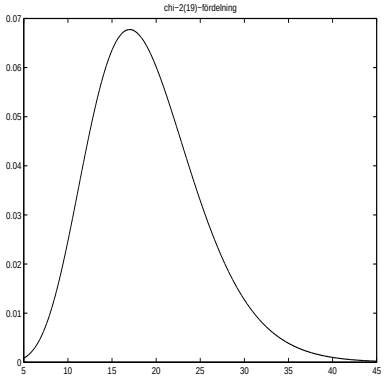
Figur 5.5: Bootstrap-fördelning för σ^2-skattning baserat på 20 $N(0,1)$-fördelade

Vi genererar 20 st N(0,1)-fördelade observationer och skattar variansen ($\sigma^2 = 1$ alltså) med plug-in-skattningen

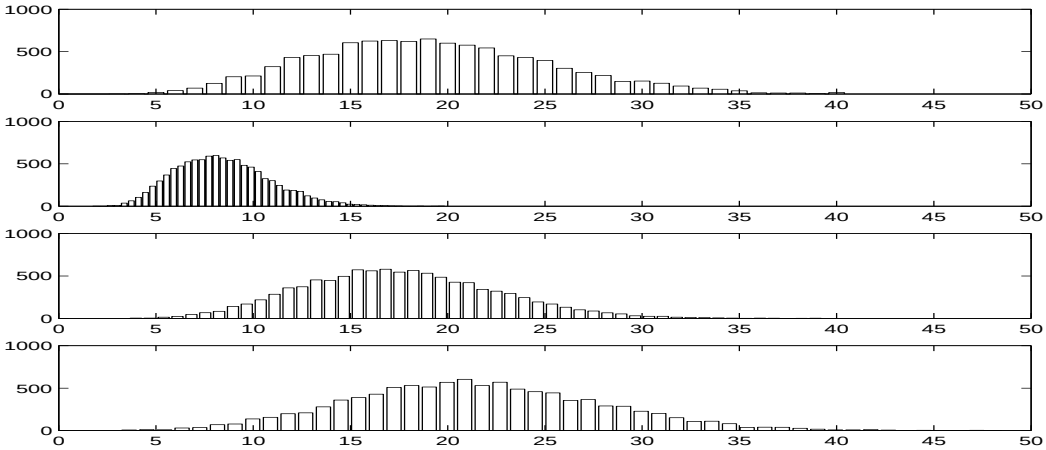
$$\widehat{\sigma}^2 = \frac{1}{20}\sum_{i=1}^{20}(x_i - \bar{x})^2$$

Vi kör sedan bootstrap (1000 st) och det resulterande histogrammet syns i figur 5.5. Matlab-koden blir:

```
x=normrnd(0,1,20,1); %ger 20 N(0,1) slumptal som kolonn-vektor
var(x,1) %ger skattningen 0.9128 av variansen (division med n=20)
boot=bootstrp(1000,'var',x,1);
mean(boot)  % gav 0.8651
std(boot)   % gav 0.2535
hist(boot,30)
```



Figur 5.6: $\chi^2(19)$-fördelningens täthet



Figur 5.7: Bootstrapfördelning för $20\widehat{\sigma}^2$ för 4 olika stickprov vardera bestående av 20 observationer från $N(0,1)$.

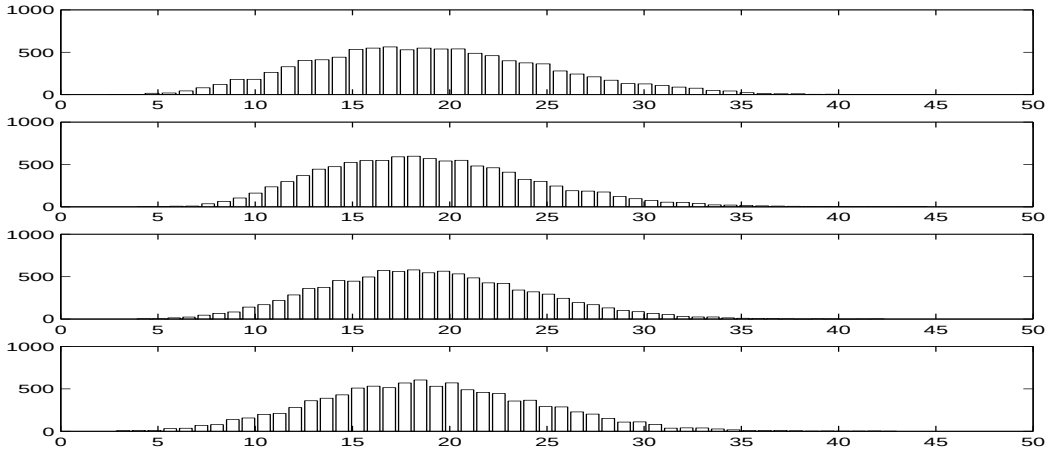
I figur 5.5 redovisas histogrammet av de 1000 bootstrap-värdena multiplicerade med 20. I figur 5.6 visas tätheten för en $\chi^2(19)$-fördelad variabel som den alltså bör likna.
Som bekant har $\chi^2(f)$-fördelningen väntevärdet f och variansen $2f$, vilket gör att den sanna standardavvikelsen är

$$D(\widehat{\sigma}^2) = \sigma^2\sqrt{2 \cdot 19}/20 \approx \sigma^2 0.308$$

att jämföra med 0.2535 som bootstrap gav. Om vi skattar den sanna standardavvikelsen $0.308\sigma^2$ med $0.308\widehat{\sigma}^2 = 0.308 \cdot 0.9128$ erhålls det "sanna" medelfelet 0.2813 som är obetydligt högre än bootstrap-skattningen av medelfelet dvs 0.2535.

I figur 5.7 redovisas bootstrap-fördelningen för 4 olika stickprov. Notera hur de är förskjutna i förhållande till varandra beroende på vad skattningen $\widehat{\sigma}^2$ av

σ^2 råkat bli i respektive stickprov. För de fyra stickproven var skattningarna 1.0000, 0.4453, 0.9261 respektive 1.1281. Notera att stickprov nr 2 som gav en extremt liten skattning ger upphov till en "hoptryckt" bootstrap-fördelning. Det centrala är dock att fördelningsformen blir den korrekta dvs med en relativt tung svans åt höger precis som $\chi^2(19)$-fördelningen som de alltså borde likna. Om man gjort en omskalning och utfört en bootstrap av $20\widehat{\sigma}^2/\sigma^2$ hade fördelningen blivit väsentligt mer stabil.

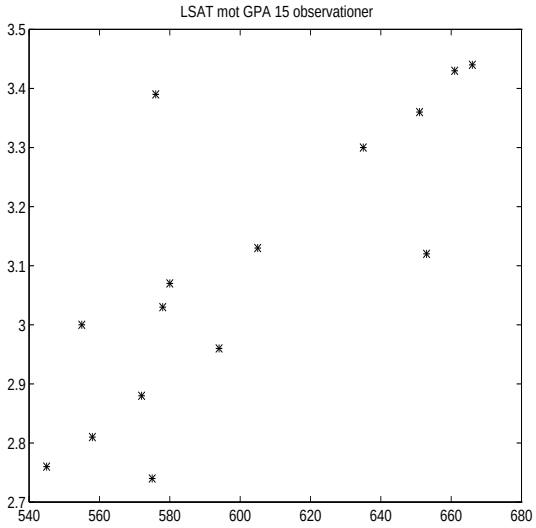


Figur 5.8: Bootstrapfördelning för $20\widehat{\sigma}^2/\sigma^2$ för de 4 stickproven i figur 5.7.

Detta hade i praktiken betytt att vi simulerat fördelningen för $20\widehat{\sigma}^{2*}/\widehat{\sigma}^2$ eller rent numeriskt skalat om respektive axel genom division med $\widehat{\sigma}^2$ för respektive stickprov. Resultatet redovisas i figur 5.8 där man tydligt ser att bootstrapfördelningen för samtliga 4 stickprov blir väldigt lika. Detta hänger ihop med att i denna situation har vi gjort bootstrap av en pivot-variabel vars fördelning är mer stabil. I princip borde de idealt ge $\chi^2(19)$-fördelningen om data kommer från $N(m, \sigma^2)$-fördelningen. Man kan beräkna konfidensintervall för σ^2 enligt metoden beskriven i exempel 3.20 på sidan 33 genom att ta percentiler från dessa bootstrapfördelningar. Mer om detta i kapitel 12 på sidan 155. Skulle data komma från någon annan fördelning än normalfördelningen skulle den omskalade bootstrap-fördelningen enligt figur 5.8 "automatiskt" bli anpassad till den bakomliggande fördelningen.

5.5 Klassiskt exempel om korrelation

Ett klassiskt exempel som gavs av Efron är två-dimensionella data på LSAT (Law School Aptitude Test - ett nationellt prov med inriktning på juridik) och GPA (Grade Point Average – dvs betygsmedelvärde från grundexamen) för ett antal amerikanska juridik-fakulteter. I ett material om 82 talpar dras slumpmässigt ett urval om 15 st (se figur 5.9) och uppgiften är att ur dessa 15 talpar skatta korrelationskoefficienten i totalmaterialet om 82 st.



Figur 5.9: LSAT mot GPA 15 observationer

Vi skattar ρ på det traditionella sättet genom att med $n = 15$ ta

$$\widehat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Notera att detta faktiskt är en plug-in-skattning! Vi har ju

$$\rho = \rho(X, Y) = \frac{C(X, Y)}{\sqrt{V(X)X(Y)}} = \frac{E\big((X - EX)(Y - EY)\big)}{\sqrt{E\left((X - EX)^2\right) E\left((Y - EY)^2\right)}}$$

Även i det "enklaste" fallet med två-dimensionell normalfördelning kan man inte få fram den exakta fördelningen för $\widehat{\rho}$, men genom Fisher-transformationen

$$\widehat{\psi} = \frac{1}{2} \log \frac{1 + \widehat{\rho}}{1 - \widehat{\rho}}$$

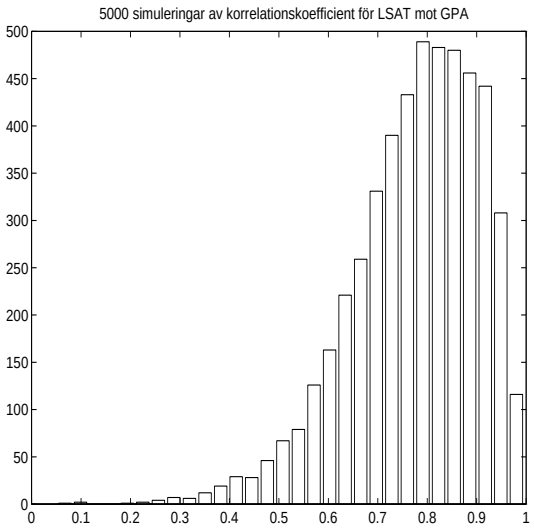
(en s k approximativt variansstabiliserande transformation) så kan man visa att $\widehat{\psi}$ är approximativt normalfördelad med väntevärde

$$\psi = \frac{1}{2} \log \frac{1 + \rho}{1 - \rho}$$

och varians $1/(n - 3)$. Detta kan då användas för att konstruera ett (approximativt) konfidensintervall för ψ som sen kan tranformeras (via den inversa transformationen)

$$\rho = \frac{e^{2\psi} - 1}{e^{2\psi} + 1}$$

till ett approximativt konfidensintervall för ρ.



Figur 5.10: Bootstrap-fördelning för skattning korrelationskoefficient för data i figur 5.9

Den variansstabiliserande transformationen fungerar (förvånansvärt) bra för normalfördelade stickprov, men man tappar helt kontrollen på fördelningen för $\widehat{\rho}$ om data kommer från någon annan två-dimensionell fördelning. Ett typiskt fenomen för $\widehat{\rho}$ är att skattningen har en skev fördelning särskilt om $|\rho|$ är nära 1. Den variansstabiliserande transformationen tar hand om denna skevhet på ett föredömligt sätt. Att för en allmän skattning hitta en sådan variansstabiliserande transformation är inte speciellt lätt. Det s k "modifierade percentil-intervallet" enligt BC_a-metoden (beskrivna i kapitel 14) hittar sådana variansstabiliserande transformationer på ett automatiskt sätt i en viss (ganska stor) klass av problem.

Hur görs nu bootstrap i denna situation? Jo, om vi t ex har 15 talpar, så genererar vi ett bootstrap-stickprov om 15 observationer tagna från de observerade talparen. Man genererar alltså återigen 15 observationer från den empiriska fördelningen – här den fördelning som lägger massan 1/15 i vardera av de 15 talparen. Sen räknar vi ut korrelationen i vardera av dessa (kanske 1000 st) bootstrap-stickprov om vardera 15 talpar, och skattar $D(\widehat{\rho})$ ur den observerade standardavvikelsen bland de 1000 korrelationsskattningarna ur bootstrap-stickproven. Man kan också (vi återkommer med modifikationer senare) skatta fördelningen för $\widehat{\rho}$ med fördelningen för $\widehat{\rho}^*$, dvs i praktiken genom att ta histogrammet för de observerade korrelationsskattningarna från Bootstrap-stickproven och därigenom få konfidensintervall för ρ.

Dessa LSAT/GPA-data finns inlagda i Matlab och man får tillgång till dem genom kommandot `load lawdata`. Matlab-rutinen `corrcoef` skattar hela korrelationsmatrisen. När vi nu vill använda rutinen `bootstrp` så kan man notera formuleringen "`If BOOTFUN returns a matrix, then this output is converted to a long vector for storage in BOOTSTAT`", dvs vi får som

resultat rader som innehåller hela matrisen – i detta fall $2 \cdot 2 = 4$ värden för varje bootstrap-skattning. Vi utför bootstrap och får ett histogram genom Matlabkoden:

```
boot=bootstrp(5000,'corrcoef',lsat,gpa);
hist(boot(:,2),30);
```

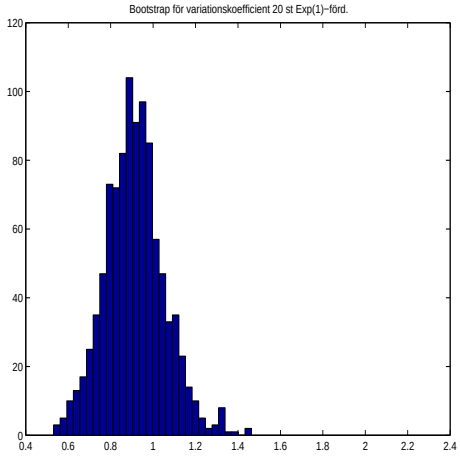
eftersom vi vill ha just 2:dra kolumnen i `boot`-vektorn.

5.6 Variationskoefficient

Här är ytterligare ett exempel där man först gjort en `m`-fil `varcoeff.m` som innehåller en funktion som skattar variationskoefficienten $D(X)/E(X)$ för en vektor `x` med $s/\bar{x}$. Här används funktionen `std` som skattar σ med division med $n-1$ för omväxlings skull. Notera dock att detta innebär att vi inte använder oss av en plug-in-skattning.

```
function v=varcoeff(x)
v=std(x)/mean(x);
```

Vi använder oss åter av våra 10 livslängdsdata och gör 5000 bootstrap-stickprov och bootstrap-fördelningen visas i figur 5.11.

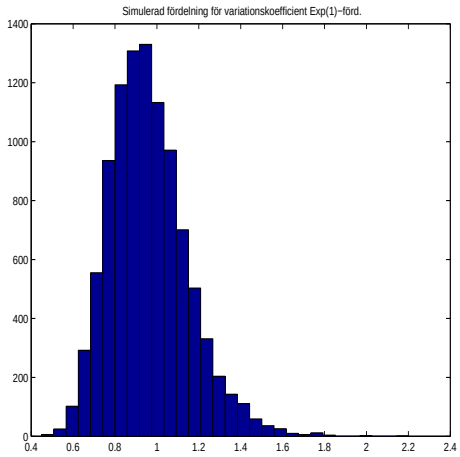


Figur 5.11: Bootstrap-fördelning för variationskoefficient för 20 st Exp(1)-fördelade

Matlab-koden blir

```
boot=bootstrp(5000,'varcoeff',x);
mean(boot)                          % gav 0.9820
std(boot)                           % gav 0.2177
mean(x)                             % gav 6.05
```

```
std(x)                          % gav 6.4732
std(x)/mean(x)                  % gav 1.0700
hist(boot,30)
```



Figur 5.12: Simulerad fördelning för variationskoefficient för 20 Exp(1)-fördelade

Man erhåller alltså skattningen 1.0700 och medelfels-skattningen 0.2177.

Med den Exp(5)-fördelning vi valt för våra data är ju det sanna värdet på $D(X)/E(X)$=1 och vår skattning hamnar mindre än ett medelfel i från det sanna värdet vilket ju är vad man kan förvänta sig. Tolkningen av medelfel är ju (liksom för standardavvikelsen som den skattar) att huvuddelen av utfallen hamnar ± 2 medelfel från det sanna värdet.

Notera att det skulle vara lite knepigt att beräkna medelfelet för $s/\bar{x}$ exakt. Detta skulle kräva att vi kunde beräkna fördelningen för

$$\frac{\sqrt{\sum_{i=1}^{10}(X_i - \bar{X})^2/9}}{\sum_{i=1}^{10} X_i/10}$$

då $X_1, X_2, \cdots, X_{10}$ är oberoende Exp(5) – en inte helt lätt uppgift även om den naturligtvis skulle kunna utföras med hjälp av simulering. Matlab-koden för denna simulering blir

```
x=exprnd(5,10,10000);
y=varcoeff(x);
hist(y,30)
mean(y) % gav 0.9225
std(y)  % gav 0.2362
```

Man kan notera att medelfelsskattningen med bootstrap var 0.2177 och med hjälp av simuleringarna var "facit" 0.2362 – en ganska god överensstämmelse.

Skattningen 1.07 ligger också mindre än ett medelfel från det sanna värdet 1 och från det sanna väntevärdet i fördelningen som ju enligt simuleringarna är ca 0.9225. Fördelningarna i figur 5.11 och figur 5.12 liknar (med lite god vilja) varandra någorlunda – man kan här mest förundra sig över att det ens går att komma i närheten av den sanna fördelningen med så få data som 10 i en så variabel fördelning som exponentialfördelningen. Att den "sanna" fördelningen har en tung svans åt höger avspeglar att vissa stickprov innehåller enstaka stora observationer (som "blåser upp" s mycket mer än $\bar{x}$). Vårt stickprov om 10 innehöll dock ingen sådan extrem observation.

5.7 Trimmat medelvärde

Antag att vi har 50 observationer från en Exp(1)-fördelning och stryker de 10 minsta och 10 största samt beräknar aritmetiska medelvärdet av de 30 i mitten. Detta är alltså ett exempel på ett trimmat medelvärde där vi har $\alpha = 0.2$ enligt exempel 3.18 på sidan 29. Följande `m`-fil utför detta

```
function w=winsor(x,alpha) ;
y=sort(x) ;
n=length(x) ;
y=y(floor(alpha*n)+1:ceil(n*(1-alpha)));
w=mean(y) ;
```

Skattningen blev 0.9740. Sanna värdet på parametern är som nämndes i exempel 3.18

$$\theta = \frac{\int_{F^{-1}(\alpha)}^{F^{-1}(1-\alpha)} x dF(x)}{1-2\alpha}$$

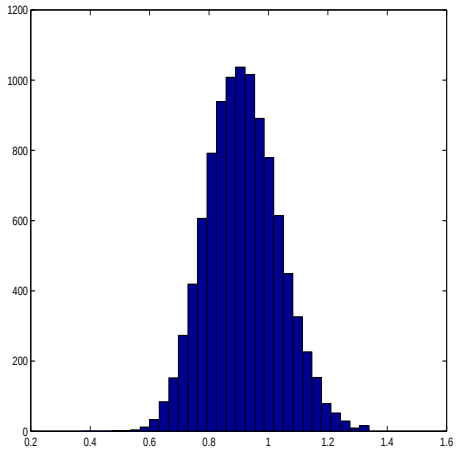
som ger $\theta = \frac{1}{0.8}(0.9 \cdot (1 + \ln(0.9)) - 0.1 \cdot (1 + \ln(0.1))) \approx 0.8307$ som alltså utgör facit. Att analytiskt beräkna medelfelet för skattningen är svårt men att skatta den med bootstrap är lätt. En bootstrap av skattningsförfarandet utförs av Matlabkoden

```
boot=bootstrp(10000,'winsor',x,0.2);
```

och resultatet syns i figur 5.13.
I denna fördelning beräknades standardavvikelsen med `std(boot)` till 0.1213 och detta är alltså skattningen av medelfelet med bootstrap. Den "sanna" fördelningen simulerades och resultatet framgår av figur 5.14. Ur simuleringarna erhölls "sanna" standardavvikelsen 0.1243. Vi har alltså fått en alldeles utmärkt skattning av den sanna standardavvikelsen med hjälp av bootstrap.

Simuleringen utfördes med koden

```
y=exprnd(1,50,10000);
exact=[];
for i=1:10000,
   exact=[exact winsor(y(:,i),0.2)];
end;
```



Figur 5.13: Bootstrapfördelning för trimmat medelvärde av 50 oberoende Exp(1)-fördelade där 10 i vardera ändan plockats bort.

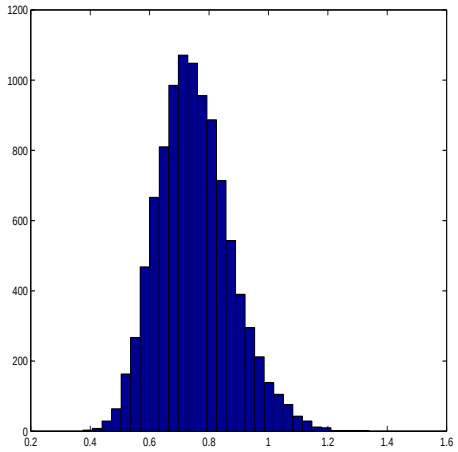
Notera att utan kunskap om den bakomliggande fördelningen hade simuleringen av den "sanna" fördelningen för $\widehat{\theta}$ ej kunnat genomföras, men med bootstrap var det en rent maskinell operation. F ö hade man fått ännu mer tillförlitliga resultat av bootstrap om man normerat sina observationer genom att dividera med $\bar{x}$ innan bootstrap utfördes. Detta beror på att $E(X)$ är en ren skalfaktor och att därför $\widehat{\theta}/E(X)$ har en fördelning som ej beror av $E(X)$ dvs ej av parametern i exponentialfördelningen. Man hade alltså kunnat välja $\widehat{\theta}/E(X)$ som pivot-variabel. Motsvarande bootstrap-kvatitet är $\widehat{\theta}(X_1^*, \cdots, X_n^*)/\bar{x}$ och denna är alltså lämpligare att utföra bootstrap av.

5.8 Medelfel för principalkomponenter

Vi skall här presentera ett mer komplicerat exempel som behandlar s k principalkomponenter hämtat ur Efrons bok. Syftet är inte att lära ut principalkomponenter utan att visa att även mycket komplicerade situationer lätt kan analyseras med bootstrap.

88 personer har fått göra 5 prov var (mekanik, vektoranalys, algebra, analys och statistik) som vardera ger 0-100 poäng. De två första proven gjordes utan tillgång till lärobok och de tre sista med tillgång till lärobok. Data redovisas i tabell 5.1.

Man ser ur data att provresultaten för de olika eleverna verkar vara kraftigt beroende – en del elever är helt enkelt duktigare än andra. Man kan fråga sig om "allmän begåvning" förklarar det mesta av den variation vi ser i data eller om det är så att olika elever har olika begåvningsprofil. Detta är ett exempel på en frågeställning som ibland dyker upp då man har denna typ av multivariata data. Ett sätt att försöka kvantifiera detta är metoden med principalkomponenter. Även om användningen av principalkomponent-metoder är



Figur 5.14: Simulerad fördelning för trimmat medelvärde av 50 oberoende Exp(1)-fördelade där 10 i vardera ändan plockats bort.

vida sprid framför allt inom tillämpningar inom psykologi och pedagogik kan man ha stora invändningar mot hela metodiken – den ligger t ex bakom begreppet intelligenskvot. Vi bortser här från dessa filosofiska och metodologiska svårigheter med tolkningar utan ser det hela som ett renodlat statistiskt problem.

Man har beräknat kovariansmatrisen med element

$$G_{jk} = \frac{1}{88}\sum_{i=1}^{88}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k), \quad j,k = 1,2,3,4,5$$

där x_{ij} =elev nr i:s resultat på prov nr j, $j = 1,2,3,4,5$ och $i = 1,2\cdots,88$ och fått den empiriska kovariansmatrisen

$$G = \begin{pmatrix} 302.3 & 125.8 & 100.4 & 105.1 & 116.1 \\ 125.8 & 170.9 & 84.2 & 93.6 & 97.9 \\ 100.4 & 84.2 & 111.6 & 110.8 & 120.5 \\ 105.1 & 93.6 & 110.8 & 217.9 & 153.8 \\ 116.1 & 97.9 & 120.5 & 153.8 & 294.4 \end{pmatrix}$$

Detta utförs i Matlab av `cov(X)` om vi har våra data lagrade i `X` med en rad för varje observation (dvs elev) och en kolumn för varje variabel (dvs prov). Denna empriska kovariansmatris beskriver då det (linjära) beroendet mellan de olika provresultaten. Som alternativ kunde man ha beräknat korrelationsmatrisen med Matlab-rutinen `corrcoef`.

Vidare har man beräknat egenvärden och egenvektorer till kovariansmatrisen med hjälp av Matlab-rutinen `eig` och fått

$$\widehat{\lambda}_1 = 679.2, \quad \widehat{\boldsymbol{v}}_1 = (0.5050, 0.368, 0.346, 0.451, 0.535)$$

Mek	Vek	Alg	Ana	Sta
77	82	67	67	81
63	78	80	70	81
75	73	71	66	81
55	72	63	70	68
63	63	65	70	63
53	61	72	64	73
51	67	65	65	68
59	70	68	62	56
62	60	58	62	70
64	72	60	62	45
52	64	60	63	54
55	67	59	62	44
50	50	64	55	63
65	63	58	56	37
31	55	60	57	73
60	64	56	54	40
44	69	53	53	53
42	69	61	55	45
62	46	61	57	45
31	49	62	63	62
44	61	52	62	46
49	41	61	49	64
12	58	61	63	67
49	53	49	62	47
54	49	56	47	53
54	53	46	59	44
44	56	55	61	36
18	44	50	57	81
46	52	65	50	35
32	45	49	57	64
30	69	50	52	45
46	49	53	59	37
40	27	54	61	61
31	42	48	54	68
36	59	51	45	51
56	40	56	54	35
46	56	57	49	32
45	42	55	56	40
42	60	54	49	33
40	63	53	54	25
23	55	59	53	44
48	48	49	51	37
41	63	49	46	34
46	52	53	41	40

Mek	Vek	Alg	Ana	Sta
46	61	46	38	41
40	57	51	52	31
49	49	45	48	39
22	58	53	56	41
35	60	47	54	33
48	56	49	42	32
31	57	50	54	34
17	53	57	43	51
49	57	47	39	26
59	50	47	15	46
37	56	49	28	45
40	43	48	21	61
35	35	41	51	50
38	44	54	47	24
43	43	38	34	49
39	46	46	32	43
62	44	36	22	42
48	38	41	44	33
34	42	50	47	29
18	51	40	56	30
35	36	46	48	29
59	53	37	22	19
41	41	43	30	33
31	52	37	27	40
17	51	52	35	31
34	30	50	47	36
46	40	47	29	17
10	46	36	47	39
46	37	45	15	30
30	34	43	46	18
13	51	50	25	31
49	50	38	23	9
18	32	31	45	40
8	42	48	26	40
23	38	36	48	15
30	24	43	33	25
3	9	51	47	40
7	51	43	17	22
15	40	43	23	18
15	38	39	28	17
5	30	44	36	18
12	30	32	35	21
5	26	15	20	20
0	40	21	9	14

Tabell 5.1: Resultat i 5 prov (Mekanik, Vektoranalys, Algebra, Analys, Statistik) för 88 elever

$$\widehat{\lambda}_2 = 199.8, \quad \widehat{\boldsymbol{v}}_2 = (-0.749, -0.207, 0.076, 0.301, 0.548)$$

$$\widehat{\lambda}_3 = 102.6, \quad \widehat{\boldsymbol{v}}_3 = (-0.300, 0.416, 0.145, 0.597, -0.600)$$

$$\widehat{\lambda}_4 = 83.7, \quad \widehat{\boldsymbol{v}}_4 = (0.296, -0.783, -0.003, 0.518, -0.176)$$

$$\widehat{\lambda}_5 = 31.8, \quad \widehat{\boldsymbol{v}}_5 = (0.079, 0.189, -0.924, 0.286, 0.151)$$

där egenvärdena är sorterade i storleksordning.

Skattningen innebär att vi ser data som en punktsvärm i $\boldsymbol{R}^5$ och försöker anpassa en ellipsoid till denna punktsvärm. Egenvektorerna svarar mot huvudaxlarna i ellipsoiden och egenvärdena anger längderna av dessa huvudaxlar.

Dessa egenvärden och egenvektorer kan ges tolkningen

1) Allmän begåvning (alla komponenter i $\widehat{\boldsymbol{v}}_1$ är positiva)

2) Prov med bok – Prov utan bok (postiva vikter i $\widehat{\boldsymbol{v}}_2$ för de med lärobok och negativa för de utan lärobok)

3) Matematik – Tillämpad matematik (postiva vikter för vektoranalys, algebra och analys, negativa komponenter i $\widehat{\boldsymbol{v}}_3$ för mekanik och statistik)

Man är intresserad av hur mycket som faktor 1 (allmän begåvning) förklarar av den totala variationen och ett mått på detta är

$$\theta = \frac{\lambda_1}{\sum_{i=1}^{5} \lambda_i}$$

som utgör en (komplicerad) parameter. Den skattas med

$$\widehat{\theta} = \frac{\widehat{\lambda}_1}{\sum_{i=1}^{5} \widehat{\lambda}_i} = 0.619.$$

θ är en aspekt (funktional) av den bakomliggande 5-dimensionella fördelningen F som vi ser våra data som utfall av. Skattningen $\widehat{\theta}$ är en plug-in-skattning av denna parameter. θ har en viss statistisk tolkning, men detta är på sätt och vis oväsentligt. Vi kan se det som att skattningen $\widehat{\theta}$ är given och att undrar över t ex dess medelfel. Att denna skattning är en särdeles komplicerad funktion av våra observationer gör inte framtagandet av medelfels-skattningen med bootstrap speciellt komplicerad.

Tolkningen av θ är att den mäter hur väl variablititen av de 5-dimensionella observationerna förklaras av första principal-komponenten, dvs i vilken utsträckning de ligger längs egenvektorn som hör till det största egenvärdet. En extremt förenklad modell för de olika elevernas resultat är att bortsett från slumpfel

$$\boldsymbol{x}_i = Q_i \boldsymbol{v}$$

där $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{i5})$ är elev nr i:s resultat på de 5 proven och $\boldsymbol{v} = (v_1, v_2, \cdots, v_5)$ är en uppsättning tal gemensamma för alla studenter. Denna vektor svarar mot egenvektorn som hör till det största egenvärdet – den första principal-komponenten. Q_i är ett tal som representerar elev nr i:s "allmänna begåvning". Om "allmän begåvning" förklarar hela variationen i data så är $\theta = 1$ och det skulle innebära att punkterna ligger på en rät linje i R^5. Geometriskt skulle ellipsoiden i R^5 vara urartad till en linje. I enlighet med denna enkla modell skulle resultatet för elev i på de olika proven bestämmas helt av talet Q_i.

Om θ är "stor" dvs nära 1 betyder "allmän begåvning" mycket för att förklara variationen i testresultat. Detta i analogi med att en korrelationskoefficient på $\rho = 1$ innebär att observationerna ligger på en rät linje medan ρ "nära" 1 innebär att de är väl samlade kring linjen. Vi är alltså intresserade av att beräkna ett konfidensintervall för θ. För att beräkna detta behöver vi medelfelet för skattningen $\widehat{\theta}$. Detta är naturligtvis rysligt komplicerat att göra analytiskt, men med bootstrap är det förhållandevis enkelt. Vi lottar då fram 88 personer genom dragning med återläggning från de 88 observationerna (som ju består av de 5 provresultaten). Dessa ger upphov till G^*=empiriska kovariansmatrisen för bootstrap-stickprovet, och ur denna räknas de storlekssorterade egenvärdena $\widehat{\lambda}_1^*, \widehat{\lambda}_2^*, \cdots, \widehat{\lambda}_5^*$ ut där alltså $\widehat{\lambda}_1^* \geq \widehat{\lambda}_2^* \geq \cdots \geq \widehat{\lambda}_5^*$ och vi får

$$\widehat{\theta}^* = \frac{\widehat{\lambda}_1^*}{\sum_{i=1}^{5} \widehat{\lambda}_i^*}.$$

För att tolkningen av första principalkomponenten skall svara mot "allmän begåvning" skall egenvektorn svarande mot största egenvärdet genomgående ha enbart positiva komponenter – annars svarar inte den mot "allmän begåvning".

Detta görs sedan B ggr och ur de observerade värdena på $\widehat{\theta}^*$ kan vi skatta medelfelet $\widehat{se}$ för $\widehat{\theta}$ och t ex göra ett konfidensintervall av typen $\widehat{\theta} \pm 1.9600\widehat{se}$ för att få ett approximativt 95%-igt intervall.

Man erhöll $\widehat{se}_{boot,200} = 0.047$ som ger det 95%-iga konfidensintervallet $0.619 \pm 1.9600 \cdot 0.047 = (0.5269, 0.7111)$.

Notera att ett försök att räkna exakt på fördelningen för denna skattning $\widehat{\theta} = \widehat{\lambda}_1/(\sum_{i=1}^5 \widehat{\lambda}_i)$ ens om vi antar att data genererats enligt en 5-dimensionell normalfördelning skulle vara helt ogörligt.

Skulle vi vilja göra detta är nog det enda genomförbara alternativet att

1) skatta kovariansmatrisen och väntevärdesvektorn ur data

2) Simulera denna 5-dimensionella normalfördelning och se vad $\widehat{\theta}$ får för fördelning.

Detta skulle faktiskt precis innebära en parametrisk bootstrap!

5.9 Kurvanpassning – Low S

I detta avsnitt skall vi presentera en intressant metodik för kurvanpassning kallad "Low S" och visa hur man kan skatta intressanta storheter inklusive konfidensintervall för dessa.

Om man har data i form av talpar (x_i, y_i), $i = 1, 2, \cdots, n$ vill man gärna försöka anpassa en modell av typen $Y_i = g(x) + \varepsilon_i$ där $E(\varepsilon_i) = 0$ och (möjligen) $V(\epsilon_i) = \sigma^2$ dvs försöka finna en funktion $g(x)$ som beskriver data. Ibland kan man välja att ta g-funktionen från en parametrisk familj av funktioner, t ex $g(x) = \beta_0 + \beta_1 x$ eller $g(x) = \beta_0 + \beta_1 x + \beta_2 x^2$ eller $g(x) = \beta_0 \exp(\beta_1 x) + \beta_2 x$ där alltså t ex $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)$ utgör en parameter-vektor. Man kan skatta $\boldsymbol{\beta}$-vektorn med t ex Minsta Kvadratmetoden. Att ansätta en g-funktion av någon viss av ovanstående typer kan i en verklig situation dock kännas restriktivt och man skulle gärna vilja kunna slippa bestämma sig för en specifik funktionsklass för g-funktionen. Den i detta avsnitt beskrivna tekniken kan vara ett alternativ där man dessutom med bootstrap kan analysera intressanta aspekter av den resulterande g-funktionen (regressions-funktionen).

g-funktionen kan uppfattas som ett betingat väntevärde $g(x) = E(Y|x)$ och man skulle kunna frestas ta

$$\widehat{g}(x) = \frac{\text{summa av } y_i \text{ med } x_i = x}{\text{antal } x_i = x}$$

men en sådan lokal anpassnings-strategi skulle ge en mycket orolig regressionsfunktion $g(x)$. Vad man då kan göra är följande som innebär en kompromiss

mellan en lokal anpassning och en global. Man försöker med viktad minsta kvadratmetodik lokalt anpassa en linje $y = \beta_0 + \beta_1 x$ där bara observationerna i närheten av x ingår i anpassningen. För en skattning av $g(x)$ används bara en viss proportion α av observationerna som har x_i-värden i närheten av x. Ett vanligt val är $\alpha = 0.3$.

Man använder sig av följande algoritm för att skatta $g(x)$ för ett fixt x:

1. Välj för ett visst x de αn observationer som har minst $|x_i - x|$ dvs de observationer som ligger närmast x. Dessa observationers index kallar vi $N(x)$. $N(x)$ består alltså av αn element.

2. Anpassa med en viktad minsta kvadrat-metod en linje $y = \beta_{x,0} + \beta_{x,1} x$ till dessa. Vi väljer alltså $\widehat{\beta}_{x,0}$ och $\widehat{\beta}_{x,1}$ som de värden som minimerar

$$Q = \sum_{i \in N(x)} w_{x,i}(y_i - \beta_{x,0} - \beta_{x,1} x_i)^2$$

   där vi väljer vikterna $w_{x,i}$ som t ex

$$w_{x,i} = 1 - \frac{|x_i - x|}{\max_{j \in N(x)} |x_j - x|}$$

   dvs att vikten för en viss observation avtar linjärt med hur långt från x som x_i ligger.

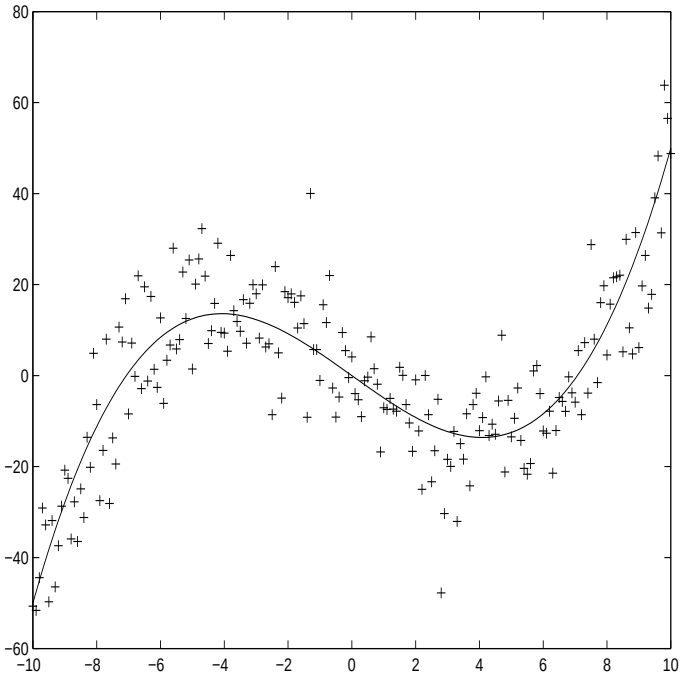
3. Vi väljer $\widehat{g}(x) = \widehat{\beta}_{x,0} + \widehat{\beta}_{x,1} x$

4. Gå vidare till nästa x-värde och upprepa ovanstående.

Om vi lagrar vikterna $w_{x,i}$ i vektorn `W`, x_i:na respektive de y_i:n som ligger i $N(x)$ i kolonn-vektorerna `x` respektive `y` så erhålls den viktade MK-skattningen `bx` med Matlabkoden

```
X=[ones(size(x)) x];
bx=inv(X'*diag(W)*X)*X'*diag(W)*y
```

där alltså `X` innehåller den aktuella design-matrisen.
Som exempel genererades observationer $Y_i = 0.1x_i^3 - 5x_i + \varepsilon_i$ där $x_i = 10 - 0.1i$ för $i = 0, 1, 2, \cdots, 200$, dvs löpte över intervallet -10 till 10 i steg om 0.1 genom `x=-10:0.1:10` och ε_i var oberoende $N(0, 10^2)$. Resultatet framgår av figur 5.15 där även det "sanna" regressionssambandet $y = g(x) = 0.1x^3 - 5x$ är inritat. Om man anpassar en kurva enligt ovanstående metodik erhölls en anpassning enligt figur 5.16. Notera att denna goda anpassning erhölls utan någon kunskap om det sanna regressions-sambandet. Om man nu vill skatta någon egenskap för regressions-sambandet kan detta ofta göras ur den anpassade kurvan. Som exempel kan man ta skillnaden mellan det lokala maximat och minimat eller läget för inflexions-punkten (där alltså andra-derivatan=0) etc. Man kan i och för sig göra motsvarande skattningar i en parametrisk modell där man skattar parametrarna ur data och sedan skattar t ex ovanstående storheter ur den



Figur 5.15: Simulerade data enligt $y = 0.1x^3 - 5x$

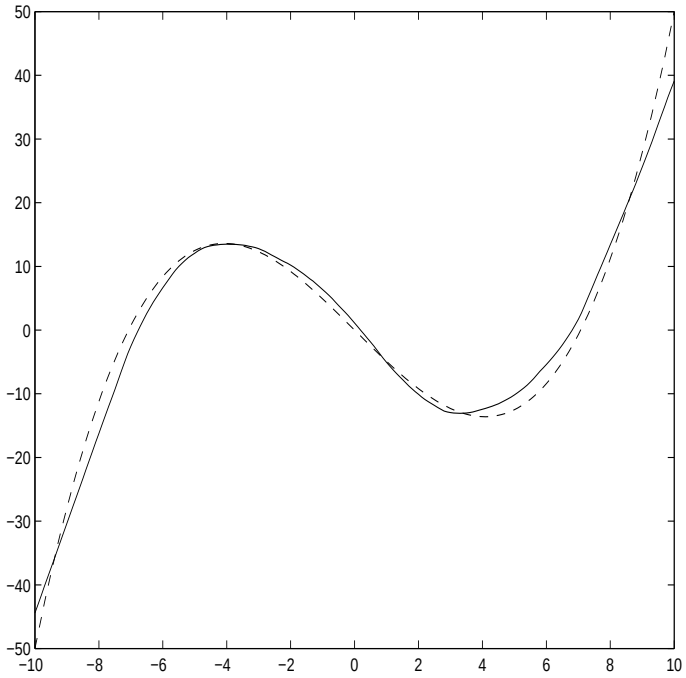
anpassade modellen. Notera dock hur avgörande det är att man då bestämmer sig för en "korrekt" parametrisk modell.
Om man nu t ex vill skatta en storhet som skillnaden mellan det lokala maximat och minimat (som alltså i verkligheten är $g(-\sqrt{5/0.3}) - g(\sqrt{5/0.3}) \approx 27.2166$) så erhålls detta ur den anpassade kurvan till 26.55266. Hur skulle man nu bete sig om man vill ha en medelfels-skattning för denna skattning? Med bootstrap skattar man regressions-sambandet genom att dra talpar från den ursprungliga uppsättningen och skattar den intressanta storheten max-min ur dessa bootstrap-stickprov. Datorn jobbade med detta under någon timme och erhöll 1000 bootstrap-stickprov. Figur 5.17 visar 25 av dessa anpassade kurvor. Alla 1000 kurvorna var av liknande typ.

Den `m`-fil `loess` som användes hade följande utseende. Talparen (x_i, y_i) var lagrade i vektorn `talpar` och dessutom fanns det ett gitter av x-värden, för vilka vi önskade skatta g-funktionen, lagrade i vektorn `z`. Notera att rutinen måste fungera även om x_i-värdena är "oordnade" eftersom rutinen `bootstrap` skickar "osorterade" argument till skattningsrutinen. Man måste alltså se till att skattnings-rutinen klarar av även detta. För tydlighetens skull har koden givits förklarande kommentarer.

`function kurva=loess(talpar,alpha);` (funktions-anrop)

`global z ;` (gör att man får tillgång till "yttre" vektorn `z` som innehåller gittret på vilket vi önskar $\widehat{g}$-funktionen)

`x=talpar(:,1);` (plockar fram x_i:vektorn ur `talpar`)



Figur 5.16: Anpassning med Low S-metodik samt "sanna regressions-sambandet" $y = 0.1x^3 - 5x$ (streckad)

`y=talpar(:,2);` (plockar fram y_i:vektorn ur `talpar`)

`s=length(x);` (tar fram storleken av x_i:vektorn)

`t=length(z) ;` (tar fram storleken av z:vektorn)

`antal=floor(s*alpha);` (tar fram storleken av $N(x)$-mängden)

`for j=1:t,` (`for`-loop för att gå igenom z-värdena)

`[dummy I]=sort(abs((x-z(j))));` (`I` kommer att innehålla indexen för de ordnade avstånden från z_j)

`xtemp=x(I(1:antal));` (`xtemp` innehåller de `antal` x_i:na med minst avstånd till z_j)

`ytemp=y(I(1:antal));` (`ytemp` innehåller motsvarande y_i:värden för de `antal` x_i:n med minst avstånd från z_j)

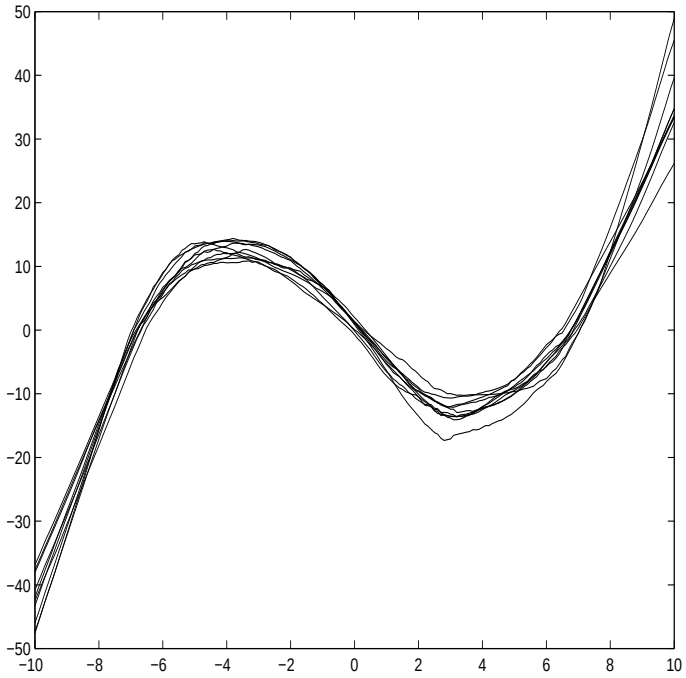
`W=(1-abs((xtemp-z(j))/max(abs(xtemp-z(j)))) ;` (`W` innehåller vikterna $w_{z(j),i}$ som en vektor)

`X=[ones(antal,1) xtemp];` (`X` innehåller den aktuella design-matrisen)

`b=inv(X'*diag(W)*X)*X'*diag(W)*ytemp;` (skattar $\beta_{z,0}$ och $\beta_{z,1}$)

`kurva(j)=b(j,1)+b(j,2)*z(j);` (skapar `kurva` som innehåller regressionsskattningen $\widehat{g}$ för gittret `z`)

`end ;`



Figur 5.17: Anpassningar enligt Low S för 25 bootstrap-stickprov

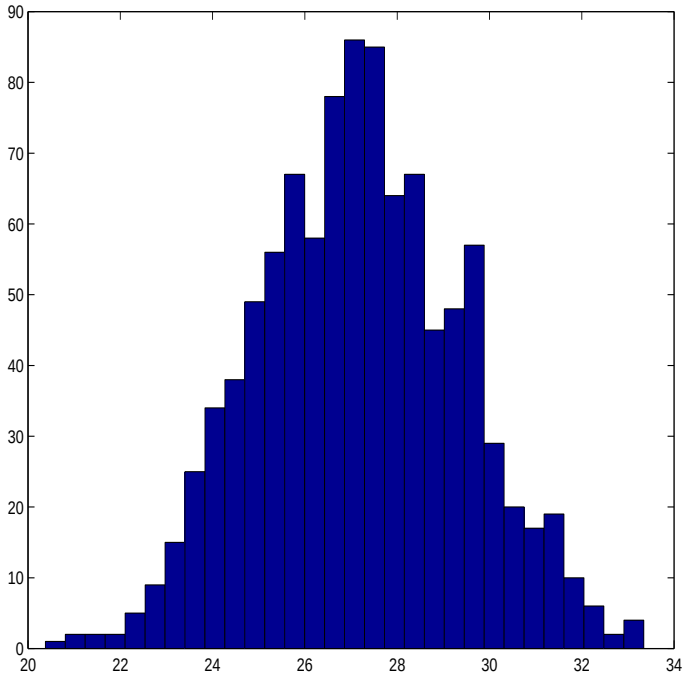
Man gör sedan bootstrap på detta förfarande med Matlab-koden
`boot=bootstrp(B,'loess',talpar,0.3)`
där parametern `alpha` tagits till 0.3. Matrisen `boot` kommer att innehålla B rader (ett för varje bootstrap-stickprov) med värdena för skattningen av g-funktionen på gittret givet av `z`:vektorn i de olika kolumnerna.

Om man ur dessa kurvor tar skillnaden mellan max och min för vart och ett erhålls figur 5.18.

Ur materialet erhåller man medelfels-skattningen för max-min till 2.1808 och ett 95%-igt konfidensintervall skulle bli $26.55 \pm 1.9600 \cdot 2.1808 \approx (22.28, 30.83)$ om vi tror att skattningen är approximativt normalfördelad, vilken den enligt bootstrap-fördelningen ser ut att vara. Som belysning av styrkan i detta angreppssätt kan man ju fundera på hur man med traditionella metoder skulle kunna få ett konfidensintervall för denna typ av parameter (lokalt max-lokalt min) för en ospecificerad kurva där vi har mätningar med brus.

5.10 Problem med bootstrap

Bootstrap lämpar sig inte för att skatta alla aspekter av den bakomliggande fördelningen F. Ett enkelt exempel är om vi försöker skatta antalet punktmassor i fördelningen F. Funktionalen "räkna antalet punktmassor" är för irreguljär.



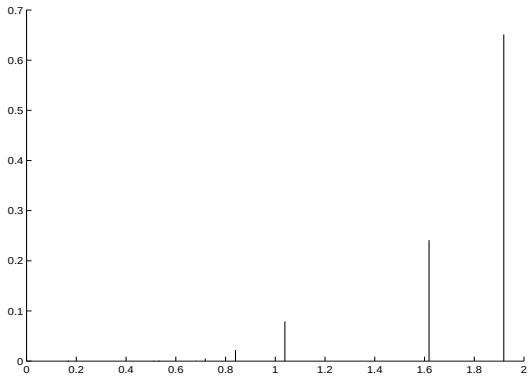
Figur 5.18: Bootstrap-fördelning för skillnad mellan lokalt max och lokalt min

Ett mindre artificiellt exempel är om man försöker skatta θ där observationerna kommer från en $R[0,\theta]$-fördelning. Om vi har observationerna $x_1, x_2, \cdots, x_n$ är plug-in-skattningen av θ att ta $\max(x_1, x_2, \cdots, x_n)$. Detta förutsätter att vi tolkar parametern som högra ändpunkten för stödet för fördelningen F, dvs den "högraste" punkt där massa finns. Denna skattning kommer vid bootstrap typiskt att med sannolikhet

$$1 - \left(1 - \frac{1}{n}\right)^n \to 1 - e^{-1} \approx 0.632 \text{ då } n \to \infty$$

ge värdet maximum$(x_1, x_2, \cdots, x_n)$ eftersom ovanstående är sannolikheten att den maximala observationen skall komma med i bootstrap-stickprovet. På samma sätt är sannolikheten att man får den näst största observationen som den största i bootstrap-stickprovet ungefär $e^{-1} - e^{-2} \approx 0.2325$. Den empiriska fördelningen är helt enkelt en dålig skattning av den bakomliggande fördelningen F:s utseende längst ut till höger.

Generellt kommer skattningen av maximum i bootstrap-stickproven att anta det k:te största värdet $x_{(k)}$ i det ursprungliga stickprovet med sannolikhet $(1-(k-1)/n)^n - (1-k/n)^n$ dvs approximativt $\exp(-(k-1)) - \exp(-k)$ för $k = 1, 2, \cdots, n$. Denna fördelning för ett konkret stickprov av storlek 10 från $R(0,2)$-fördelningen visas i figur 5.19 som alltså bara har massa i de ursprungliga 10 observationerna. Denna punktmasse-fördelning liknar ju inte



Figur 5.19: Bootstrap-fördelning för maximum av 10 st $R(0, 2)$-fördelade.

särskilt mycket den sanna fördelningen för $\max(X_1, \cdots, X_{10})$ som har tätheten

$$f(x) = \frac{10x^9}{2^{10}} \text{ för } 0 \leq x \leq 2$$

när de bakomliggande variablerna kommer från $R(0, 2)$-fördelningen. Det betyder att antagligen kommer bara vissa "snälla" funktionaler på dessa två fördelningar ger liknande resultat. Dock kommer antagligen väntevärdet och möjligen standardavvikelsen i bootstrap-fördelningen att ge realistiska resultat.

Fenomenet att bootstrap-fördelningen bara har massa i några få punkter dyker också upp då man försöker skatta medianen eftersom medianen i bootstrap-stickprovet kommer att bli någon av de ursprungliga observationerna åtminstone om man har ett udda antal observationer. Detta behandlas lite mer ingående i avsnitt 7.4.2 på sidan 92.

I de flesta av exemplen som givits i detta kapitel är dock bootstrap-fördelningen inte så urartad som för maximum och median utan är mer utspridd. Notera dock att enligt resonemangen i avsnitt 4.5 är bootstrap-fördelningen alltid en diskret fördelning vid icke-parametrisk bootstrap. Den har dock i allmänhet ett mycket stort antal punktmassor och har alltså bättre möjligheter att kunna väl approximera (den i allmänhet kontinuerliga) fördelningen för stickprovsvariabeln.

Kapitel 6

Livslängdsexemplet

I detta kapitel behandlas lite mer utförligt ett exempel på hur man med bootstrap-metodik kan få fram tillförlitliga konfidensintervall utan att lägga några restriktiva antaganden på hur data har uppkommit. Exemplet behandlas så ingående att det är lämpligt att det får ett helt eget kapitel. Det skall dock ses som bara ett (förhoppningsvis) illustrativt exempel på den enorma styrkan som finns i bootstrap-metodiken. Detta exempel kan ses som en introduktion till konstruktionen av konfidensintervall med pivot-metoden som behandlas i kapitel 12 på sida 155. Den otålige läsaren kan därför hoppa över detta kapitel och återvända till det efter läsning av kapitel 12.

Vi återvänder till livslängdsproblemet (dvs exempel 3.1 på sidan 21) där vi observerat de 10 livslängderna 5.6, 19.7, 1.7, 3.3, 5.1, 0.7, 15.6, 5.4, 3.3, 0.1 och vill ha ett 90%-igt konfidensintervall för θ =medellivslängden.

6.1 Skalinvarians

Vi vill släppa på antagandet om exponentialfördelning, men ändå ha det fallet som specialfall för att t ex kunna checka av hur vår metod fungerar. Man kan också säga att metoden att få fram konfidensintervall är kalkerad på hur man fick fram konfidensintervallet i exponentialfördelningssituationen i avsnitt 3.6.3 på sidan 34, men att metoden fungerar helt oförändrad även om data kommer från en annan fördelning.

Vi antar att våra data kommer från en skalinvariant familj, dvs att med $\theta = E(X)$ har vi

$$P\left(\frac{X}{\theta} \leq z\right) = H(z) \text{ dvs } F(x) = P(X \leq x) = H(x/\theta)$$

för någon fördelningsfunktion H där vi alltså har $F(x) = H(x/\theta)$. Vår parameter θ är alltså en funktional av F som vi kan kalla $T(F)$.

I exponentialfördelningsfallet är $H(z) = 1 - e^{-z}, \quad z > 0$, men vi kan ju få en hel del annat också (t ex Weibull-fördelningar etc). Man kan uttrycka det så att vi tagit X/θ som pivot-variabel (dvs fördelningen för denna beror ej av

θ). Med ett specifikt H skulle vi som kunna härleda (med faltning och annat eländigt trassel) funktionen G där

$$P\left(\frac{\bar{X}}{\theta} \leq z\right) = G(z)$$

och kunna hitta tal $z_{\alpha/2}$ och $z_{1-\alpha/2}$ sådana att

$$G(z_{\alpha/2}) = \alpha/2 \text{ och } G(z_{1-\alpha/2}) = 1 - \alpha/2$$

för att sen få konfidensintervallet

$$\left(\frac{\bar{x}}{z_{1-\alpha/2}}, \frac{\bar{x}}{z_{\alpha/2}}\right)$$

eftersom

$$1 - \alpha = G(z_{1-\alpha/2}) - G(z_{\alpha/2}) = P\left(z_{\alpha/2} \leq \frac{\bar{X}}{\theta} \leq z_{1-\alpha/2}\right) =$$

$$= P\left(\frac{\bar{X}}{z_{1-\alpha/2}} \leq \theta \leq \frac{\bar{X}}{z_{\alpha/2}}\right).$$

Vi känner inte H (och därmed inte G) och hur skall vi då få fram $z_{\alpha/2}$ och $z_{1-\alpha/2}$? Notera att även om vi känner H kan det vara lite trixigt att härleda G och därmed $z_{\alpha/2}$ och $z_{1-\alpha/2}$. Ovanstående metod är precis den vi använde för det exakta intervallet då vi antog att data kom från en exponentialfördelning i avsnitt 3.6.3 på sidan 34. Då hade vi tur och var skickliga nog att inse att G i princip beskrev en gammafördelning.

Vi noterar att F (där $F(x) = H(x/\theta)$) genererar vårt stickprov och att man kan betrakta

$$G(z) = P\left(\frac{\bar{X}}{\theta} \leq z\right)$$

för ett fixt z som en (mycket komplicerad) funktional $S(F)$ av F dvs

$$S(F) = G(z) = P\left(\frac{\bar{X}}{\theta} \leq z\right) = P\left(\frac{\bar{X}}{T(F)} \leq z\right).$$

Vi bildar den empiriska fördelningen för våra data dvs

$$\widehat{F}_{10} := \text{massan } \frac{1}{10} \text{ i vardera av } x_1, x_2, \cdots, x_{10}.$$

Om vi nu bildar $S(\widehat{F}_{10})$, dvs samma funktional S fast av vår empiriska fördelning $\widehat{F}_{10}$ så ser vi att detta ger (notera att det är plug-in-skattningen av $G(z)$)

$$S(\widehat{F}_{10}) = P\left(\frac{\bar{X}^*}{T(\widehat{F}_{10})} \leq z\right) = P\left(\frac{\bar{X}^*}{\bar{x}} \leq z\right) = G^*(z)$$

där $\bar{X}^*$ är medelvärdet av bootstrap-stickprovet $\boldsymbol{X}^* = (X_1^*, X_2^*, \cdots, X_n^*)$, dvs det vi fått med framlottning enligt $\widehat{F}_{10}$. Man bör här särskilt notera att det står

$$\bar{x} = \int_{-\infty}^{\infty} x d\widehat{F}_{10}(x) = T(\widehat{F}_{10})$$

i nämnaren i $S(\widehat{F}_{10})$ eftersom funktionalen $S(F)$ innehöll

$$\theta = E(X) = \int_{-\infty}^{\infty} x dF(x) = T(F)$$

i nämnaren.

Vi antar nu enligt bootstrap-hypotesen att $G^*(z) \approx G(z)$ (dvs att $S(\widehat{F}_{10}) \approx S(F)$) och kan alltså få skattningar $z^*_{\alpha/2}$ av $z_{\alpha/2}$ och $z^*_{1-\alpha/2}$ av $z_{1-\alpha/2}$ genom att lösa ekvationerna $G^*(z^*_{\alpha/2}) = \alpha/2$ respektive $G^*(z^*_{1-\alpha/2}) = 1 - \alpha/2$.

G^* är besvärlig att beräkna exakt i allmänhet, men vi kan simulera den och därigenom försöka få fram fördelningen för

$$\frac{\bar{X}^*}{\bar{x}}$$

som G^* beskriver och detta är lätt. Vi lottar helt enkelt fram stickprov av storlek 10 från $\widehat{F}_{10}$, vilket innebär att vi lottar fram 10 st från $x_1, x_2, \cdots, x_{10}$ med dragning med återläggning. Vi beräknar $\bar{x}^* =$ medelvärdet av detta stickprov om 10 data, och slutligen bildar vi $\bar{x}^*/\bar{x}$. Om vi gör detta ett stort antal gånger (säg B gånger) kan vi approximera G^* på ett bra sätt med G^*_B – den empiriska fördelningen för $\bar{x}^*/\bar{x}$ bland våra B bootstrap-stickprov. Alltså kan vi få vettiga skattningar $z^*_{\alpha/2,B}$ av $z^*_{\alpha/2}$ och $z^*_{1-\alpha/2,B}$ av $z^*_{1-\alpha/2}$ som i sin tur är skattningar av $z_{\alpha/2}$ och $z_{1-\alpha/2}$. Dessa skattningar får vi genom att i princip ta nr $[(B+1)\alpha/2]$ respektive $[(B+1)(1-\alpha/2)]$ i storleksordning av de erhållna B värdena på $\bar{x}^*/\bar{x}$. Om för $\alpha = 0.10$ tar $B = 9999$ kan vi välja nr 500 respektive nr 9500 av de storleksordnade värdena på $\bar{x}^*/\bar{x}$.

6.2 Sammanfattning

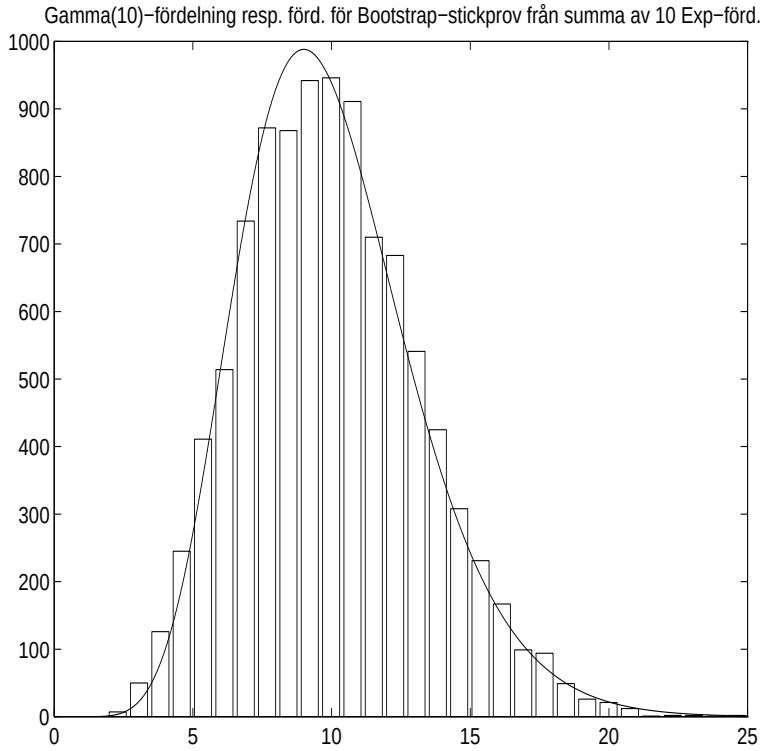
Vi skaffar oss B (kanske 9999 st) bootstrap-stickprov $\boldsymbol{x}^*_1, \boldsymbol{x}^*_2, \cdots, \boldsymbol{x}^*_B$ vardera om 10 observationer och studerar den empiriska fördelningen för $\bar{x}^*/\bar{x}$, som vi kan kalla $G^*_B(z)$ och får ur denna $z^*_{\alpha/2,B}$ och $z^*_{1-\alpha/2,B}$ samt konstruerar konfidensintervallet

$$\left(\frac{\bar{x}}{z^*_{1-\alpha/2,B}}, \frac{\bar{x}}{z^*_{\alpha/2,B}} \right).$$

Hur funkar detta då i praktiken? Med de data vi hade får vi med $B = 9999$ intervallet

$$(6.05/1.49, 6.05/0.54) = (4.05, 11.19)$$

eftersom vi fick $G_B^*(0.54) = 0.05$ och $G_B^*(1.49) = 0.95$, dvs nr 500 respektive nr 9500 i storleksordning av de 9999 värdena blev 0.54 respektive 1.49. På analogt sätt skulle vi få ett 95%-igt intervall till (3.76, 13.00).



Figur 6.1: $\Gamma(10, 1)$-fördelning resp. bootstrap-fördelning

Notera att detta betyder att vi har skattat $\Gamma_{0.95}(10, 1)/10 = 1.57$ med 1.49 och $\Gamma_{0.05}(10, 1)/10 = 0.542$ med 0.54.

Man kan inte nog poängtera att dessa skattningar av $\Gamma_{0.95}(10, 1)$ och $\Gamma_{0.05}(10, 1)$ på intet sätt utnyttjar kunskapen att faltningen av exponentialfördelningar blir en Γ-fördelning. Ja, att data "i verkligheten" kommer från en exponentialfördelning utnyttjas inte överhuvudtaget! Allt som utnyttjas är att X/θ har en viss (icke-specificerad) fördelning och att den operation som utförs är att falta denna fördelning kommer bara indirekt in i vår kalkyl genom att vi applicerar Matlab-rutinen `mean` på de olika bootstrap-stickproven.

I Matlab blir koden (där vektorn `x` innehåller våra observationer)

```
m=mean(x);
boot=bootstrp(9999,'mean',x);
boot=boot/m ;
bootsort=sort(boot);
zlow=bootsort(500);
zhigh=bootsort(9500);
```

```
conflow=m/zhigh
confhigh=m/zlow
```

Skulle våra observationer ha kommit från någon annan fördelning än exponentialfördelningen skulle Matlab-koden alltså vara helt oförändrad och vad vi skulle få som resultat vad gäller percentiler (och även konfidensintervall) skulle vara "automatiskt" anpassat till den aktuella fördelningen.

6.3 Jämförelse mellan metoder

Är detta ett vettigt intervall? Ja, det verkar ju stämma rätt bra med intervallet ovan som vi beräknade under exponentialfördelningsantagandet. Data i exemplet var i verkligheten taget från en exponentialfördelning med väntevärde 5. Vi hade alltså genererat 10 utfall av Exp(5)-fördelade stokastiska variabler, så "facit" var att data verkligen kom från en exponentialfördelning. Dock måste erkännas att vi valde ett stickprov som någorlunda fångade upp det karaktäristiska hos Exp(5)-fördelningen. Detta kan man se av att den empiriska fördelningen för data rätt väl approximerar Exp(5)-fördelningen i figur 4.2 på sidan 41.

Detta medför också att fördelningen för $\sum_{i=1}^{n} X_i^*/\bar{x}$ rätt väl approximerar $\Gamma(10, 1)$-fördelningen. Vad man mest kan förundras över är att man i figur 6.1 har lyckats få en approximation av $\Gamma(10, 1)$-fördelningen med en metod som bara bestod i att lotta fram nya stickprov från den ursprungliga datauppsättningen. Att få fram histogrammet var en rent maskinell operation utförd på 10 numeriska värden, medan härledningen av den "sanna" tätheten krävde stora analytiska kunskaper.

Dessa konfidensintervall sammanfattas i följande tabell:

Konfidensintervall för livslängds data (Exponentialfördelade)

	90% undre	90% övre	95% undre	95% övre
Exakt metod	3.85	11.15	3.54	12.62
Bootstrap	4.05	11.19	3.76	13.00
CGS	2.68	9.42	2.03	10.06

Man slås av hur väl pivot-metoden approximerar det exakta intervallet, medan CGS-approximationen är rätt skruttig.

6.4 Empirisk kontroll av metoderna:

Ja, det såg ju bra ut för just dessa data (och de var ju "valda" så att $\widehat{F}_{10} \approx$ Exp(5)-fördelningen), men vi kan undersöka saken vidare.

Vi skulle kunna testa och se hur det fungerar generellt om vi hade exponentialfördelade data, dvs lotta fram 10 data enligt en känd exponentialfördelning Exp(θ), beräkna konfidensintervall med ovanstående metoder och kolla om intervallen innehåller det sanna värdet på θ. Om vi lottar fram ett mycket

stort antal sådana datauppsättningar och beräknar konfidensintervall för varje enskild datauppsättning med hjälp av ovanstående bootstrap-metodik, och sen kollar hur många intervall som hamnar "snett" (till höger respektive till vänster om det sanna värdet på θ) så skulle detta utgöra en empirisk kontroll av metoden åtminstone om data kommer från en exponentialfördelning. Frekvenstolkningen av konfidensintervall innebär ju att $\alpha/2$ av intervallen skall missa till vänster och $\alpha/2$ skall missa till höger. Möjligen skulle man dessutom vilja studera längden av dessa intervall.

Följande resultat erhölls då vi simulerade 1000 st datauppsättningar om 10 respektive 100 st Exp(1)-fördelade observationer vardera, beräknade 90%-iga konfidensintervall för θ enligt ovanstående metod (baserat på 500 bootstrap av $\bar{x}^*/\bar{x}$ vardera). Varje beräkning i Matlab av konfidensintervallet tog ca 30 sekunder så hela beräkningen tog ca 5 timmar på en PC (486-a).

Konfidensintervall för exponentialfördelade data – 90%-iga intervall

Felmarginal (Medelfel): För CGS ±0.4%, för pivot-metoden ±1.4%

Metod	n	Missade undre gränsen	Missade övre gränsen
Bootstrap	10	9.5%	7.9%
CGS	10	2.6%	15.1%
Bootstrap	100	4.9%	6.1%
CGS	100	2.0%	8.1%

Det borde varit 5% i respektive ända. Gränserna i dubbelsidiga symmetriska 90%-iga konfidensintervall konstrueras ju så att med sannolikhet 5% skall sanna parametervärdet ligga till vänster om intervallet och med sannolikhet 5% till höger. Ett "bra" 90%igt symmetriskt konfidensintervall skall alltså vara sådant att det hamnar till höger respektive till vänster om det sanna parametervärdet med sannolikhet 5% vardera. Här har vi simulerat stickprov och p g a det begränsade antalet stickprov skulle inte ens en "exakt" metod få 5% "missar" i vardera änden. De observerade siffrorna på andelen "missar" är därför behäftade med osäkerhet.

Man ser ur tabellen att intervallen med CGS-metoden är typiskt för långa åt vänster och för korta åt höger, dvs för få respektive för många missar. Dessutom blir en del av vänstergränserna < 0 vilket är "ofysikaliskt". Det blir lite bättre om $n = 100$ än om $n = 10$ men fortfarande består för CGS-metoden fenomenet att intervallet är symmetriskt medan stickprovsvariabeln $\bar{X}$ har en skev fördelning. Detta gör att övertäckningsgraden blir fel i vardera änden även om den totala övertäckningsgraden "råkade" ligga mycket nära den nominella om 10%. Detta är faktiskt inte någon tillfällighet utan ett allmänt fenomen för CGS-baserade intervall.

När man jämför detta med Bootstrap-resultaten, ser man att dessa är aningen konservativa (speciellt för $n = 10$), men att antalet "missar" är någorlunda jämt fördelade på övre och undre konfidensgränsen. Detta är en ytterst viktig egenskap, eftersom slutsatserna av ett intervall skulle kunna bli helt fel om vi placerade "felmassan" asymmetriskt. Bootstrap-metoden tar alltså i viss mån hänsyn till den skevhet som finns i data och gör konfidensintervallet "skevt"

för att kompensera för detta. Att det enskilda konfidensintervallet tog 30 sek att beräkna utgör ju inget som helst problem. Om man kanske ägnat dagar eller veckor åt att samla in data, så är 30 sekunders exekveringstid inte mycket att orda om.

Enda orsaken till att vi inte "fläskade" till med fler än 500 bootstrap i varje stickprov berodde på att vi ville analysera så många stickprov att variabiliteten i övertäckningsgraden inte var för stor. Man har ju i princip ett Bin(1000, 0.05)-fördelat antal intervall som "bör" hamna snett i vardera ändan. Denna felprocent har alltså ett medelfel om

$$\sqrt{\frac{0.05(1-0.05)}{1000}} \approx 0.7\%$$

så den "naturliga" variabiliteten i andelarna intervall som missar parametervärdet är av denna storleksordning. Med andra ord är avvikelser på $\pm 2 \cdot 0.7\% = \pm 1.4\%$-enheter från de nominella i vardera änden vad man kan förvänta sig bara av det skälet att vi gjort ett begränsat antal uppsättningar mätdata. För CGS-fallet är osäkerheten $\pm 0.4\%$ eftersom 10000 stickprov simulerats.

Exponentialfördelningen är lite "elak" eftersom den är mycket skev, och man kan kanske mest förundra sig över att så lite som 10 data kan ge ett så hyfsat konfidensintervall överhuvudtaget, speciellt med metoder som på inget sätt utnyttjar några speciella egenskaper hos exponentialfördelningen. Uppträdandet för $n = 100$ är ytterst imponerande. Vi behöver ju i beräkningen av konfidensintervallet inte ens fundera över fördelningsantaganden och än mindre om fördelningen för $\sum_{i=1}^{10} X_i$.

6.4.1 Weibull – formparameter = 2

Om man går över till en annan fördelning som är mindre skev, t ex Weibullfördelningen med formparameter 2, dvs en fördelning av typen

$$P(X \leq z) = 1 - e^{-z^2}, \quad z \geq 0$$

så fungerar Bootstrapmetodiken helt oförändrad eftersom den inte på något sätt bygger på något fördelningsantagande. Matlab-koden är t ex helt oförändrad.

Resultat för Weibull-fördelade med formparameter 2 – 90%-iga intervall

Felmarginal: För CGS och "Exakta" metoden $\pm 0.4\%$, för pivot-metoden $\pm 1.4\%$

Metod	n	Missade undre gränsen	Missade övre gränsen
"Exakt" metod	10	0.13%	0.05%
CGS	10	4.8%	9.1%
Bootstrap	10	9.4%	6.2%
"Exakt" metod	100	0.12%	0%
CGS	100	2.3%	2.8%
Bootstrap	100	5.0%	3.6%

Man kan speciellt notera hur dåligt det går för den ”Exakta” metoden, dvs den som var exakt för exponentialfördelade data, men som går alldeles snett då data i verkligheten kom från en annan fördelning.

Att CGS-metoden fungerar så pass bra som den gör för $n = 10$ beror mer på ”tur” än ”skicklighet”. Detta framgår med tydlighet om man tittar på fallet $n = 100$ som ju borde passa denna metod bättre. CGS-metoden har en tendens att gå för långt åt vänster, men eftersom Weibull-2-fördelningen har så liten spridning motverkas detta av att s är så liten. I den övre ändan har intervallet en tendens att bli för kort, men den lägre skevheten hjälper här till att få övertäckningsgraden bättre, dvs närmare den nominella om 5%.

Vidare kan man se att Pivot-metoden ger för kort intervall (för många missar) i undre ändan, vilket avspeglar att G^* inte riktigt orkar med att bli tillräckligt skev i övre ändan. Detta beror antagligen på att en del stickprov saknar extremt stora observationer.

6.4.2 Weibull – formparameter = 0.75

Vi väljer nu en ”elakare” fördelning än t o m exponentialfördelningen, och vi fastnade för Weibullfördelningen med formparameter 0.75 som är ännu skevare än exponentialfördelningen Vi har alltså en fördelning av typen

$$P(X \leq z) = 1 - e^{-z^{0.75}}, \quad z \geq 0.$$

Weibull-fördelade med formparameter 0.75 – 90%-igt intervall

Felmarginal: För CGS och ”Exakta” metoden ±0.4%, för pivot-metoden ±1.4%

Metod	n	Missade undre gränsen	Missade övre gränsen
”Exakta” metoden	10	9.7%	11.9%
Normalapproximation	10	1.4%	19.4%
Bootstrap	10	8.9%	10.0%
”Exakta” metoden	100	9.7%	10.1%
Normalapproximation	100	2.6%	9.6%
Bootstrap	100	6.1%	6.5%

Att den ”Exakta” metoden fungerar hyfsat för $n = 10$ beror på att Weibullfördelningen med formparameter 0.75 ”liknar” exponentialfördelningen ganska mycket för rätt stor andel av stickproven, men med $n = 100$, då stickproven bättre avspeglar att fördelningen har ändrats från exponentialfördelning, förbättras inte den ”Exakta” metoden som den borde ha gjort med tanke på den större stickprovsstorleken. Den går helt enkelt inte tillräckligt långt ut i fördelningen, vilket innebär att intervallet blir för kort och övertäckningsgraden blir alltså för liten. Notera att CGS-intervallet (normalapproximation) förbättras högst avsevärt då $n = 100$ eftersom normalapproximationen då är bättre. Fenomenet att felmassan för CGS-metoden är asymmetriskt fördelad består dock.

Pivot-metoden fungerar bra speciellt för $n = 100$. Att den fungerar något sämre för $n = 10$ där intervallen är lite för korta beror på att en hel del av stickproven saknar extrema observationer och bootstrap-metoderna kan då inte känna av att de verkliga percentilerna ska ligga långt ut. Dessa stickprov avspeglar helt enkelt inte förekomsten av extrema observationer i den bakomliggande fördelningen. Då $n = 100$ blir tillräckligt många observationer extrema för att bootstrap-metoderna skall fungera för de allra flesta stickprov. För normalapproximation (CGS) blir resultatet naturligtvis ännu sämre än i exponentialfördelningsfallet.

6.5 Slutsatser

1) CGS-metoden kan ej ta hänsyn till skevhet (asymmetri) i fördelningen så ”felmassan” α fördelas fördelas ojämt mellan intervalländarna.

2) En ”exakt” metod under ett visst fördelningsantagande kan bli urusel om antagandet är fel.

3) Att få fram en ”exakt” metod kräver stor finurlighet och tur.

4) Bootstrap-metoderna

   a) tvingar oss inte att formulera en restriktiv modell för data

   b) kräver inte finurlighet i härledningen av stickprovsvariablers fördelningar

   c) är (nästan) helt automatisk

   d) tycks ge bra resultat i situationer där vi kan räkna exakt eller simulera

   e) borde därför ge bra resultat även i situationer där vi ej kan räkna exakt.

Kapitel 7

Bootstrap av medelvärde och median

7.1 Inledning

I detta kapitel analyserar vi hur bootstrap fungerar vad gäller medelfel för det aritmetiska medelvärdet $\bar{x}$, där man kan räkna exakt, samt även i någon mån medianen. Att analysera hur bootstrap fungerar för $\bar{x}$ är mest av teoretiskt intresse som ett enkelt fall där vi kan räkna exakt och alltså kan se att bootstrap fungerar. För medianen kommer man att få fram förvånande precisa resultat åtminstone i de specialfall man kan räkna exakt på.

7.2 Aritmetiskt medelvärde

För att skatta $\theta = E(X)$ ur observationerna $x_1, x_2, \cdots, x_n$ använder vi oss av plug-in-skattningen $\widehat{\theta}(x_1, x_2, \cdots, x_n) = \bar{x}$. Modellen är som vanligt att vi ser $x_1, x_2, \cdots, x_n$ som utfall av de oberoende likafördelade stokastiska variablerna $X_1, X_2, \cdots, X_n$ som alla har fördelningen F. Vi vet att

$$V(\widehat{\theta}(X_1, X_2, \cdots, X_n)) = V(\bar{X}) = \frac{\sigma^2}{n}$$

där $\sigma^2 = V(X_i)$. σ^2 är ju okänd men kan skattas med

$$s^2(x_1, x_2, \cdots, x_n) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

och man brukar ange det s k medelfelet $\widehat{se}$(=standard error)

$$\widehat{se} = d(\widehat{\theta}) = \frac{s(x_1, x_2, \cdots, x_n)}{\sqrt{n}}$$

som alltså är en skattning av $D(\widehat{\theta}(X_1, \cdots, X_n))$. Detta numeriska värde är ett mått på osäkerheten i $\bar{x}$. Ofta används $\bar{x} \pm 1.96d(\widehat{\theta})$ som ett approximativt

95%-igt konfidensintervall för θ. Detta bygger på en användning av Centrala gränsvärdessatsen där man räknar som om $(X_1 + X_2 + \cdots + X_n)/n$ vore $\mathrm{N}(\theta, s^2(x_1, x_2, \cdots, x_n)/n)$-fördelad.

7.2.1 Icke-parametrisk bootstrap av medelvärde

Vi genererar (givet observationerna $x_1, x_2, \cdots, x_n$) alltså ett slumpmässigt stickprov $X_1^*, X_2^*, \cdots, X_n^*$ där vi för $i = 1, 2, \cdots, n$ låter $P(X_i^* = x_j) = 1/n$ för $j = 1, 2, \cdots, n$ och att dessutom $X_1^*, X_2^*, \cdots, X_n^*$ är oberoende.

Just för $\widehat{\theta} = \bar{x}$ går det att räkna exakt på väntevärde och varians för stickprovsvariablen som hör till bootstrap-skattningen

$$\widehat{\theta}^* = \widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)$$

under den fördelningen som X_i^*:na har dvs $P(X_i^* = x_j) = 1/n$ för $j = 1, 2, \cdots, n$. Att bilda väntevärde (respektive varians) med avseende på denna fördelning betcknar vi med $E_{\widehat{F}_n}$ respektive $V_{\widehat{F}_n}$. Detta väntevärde respektive varians är alltså att betrakta som det betingade väntevärdet respektive variansen givet observationerna $x_1, x_2, \cdots, x_n$. Vi får

$$E_{\widehat{F}_n}(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)) = E_{\widehat{F}_n}\left(\frac{1}{n}\sum_{i=1}^{n} X_i^*\right) = \frac{1}{n}\sum_{i=1}^{n} E_{\widehat{F}_n}(X_i^*) = E_{\widehat{F}_n}(X_1^*).$$

Vi har

$$E_{\widehat{F}_n}(X_1^*) = \sum_{j=1}^{n} x_j P(X_1^* = x_j) = \sum_{j=1}^{n} x_j \frac{1}{n} = \bar{x}$$

och (p g a att $X_1^*, X_2^*, \cdots, X_n^*$ är oberoende och likafördelade)

$$V_{\widehat{F}_n}(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)) = V_{\widehat{F}_n}\left(\frac{1}{n}\sum_{i=1}^{n} X_j^*\right) = \frac{1}{n^2}\sum_{j=1}^{n} V_{\widehat{F}_n}(X_j^*) = \frac{V_{\widehat{F}_n}(X_1^*)}{n}.$$

Vi har vidare

$$V_{\widehat{F}_n}(X_1^*) = E_{\widehat{F}_n}\left((X_1^* - E_{\widehat{F}_n}(X_1^*))^2\right) = E_{\widehat{F}_n}\left((X_1^* - \bar{x})^2\right) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

och alltså får vi

$$V_{\widehat{F}_n}(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)) = \frac{1}{n^2}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{n-1}{n} \cdot \frac{s^2(x_1, x_2, \cdots, x_n)}{n}$$

som ger

$$\widehat{se}_{boot} = D(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)) = \frac{s(x_1, x_2, \cdots, x_n)}{\sqrt{n}}\sqrt{\frac{n-1}{n}} = d(\widehat{\theta})\sqrt{\frac{n-1}{n}}$$

som alltså (så när som på den oväsentliga faktorn $\sqrt{(n-1)/n}$) överensstämmer med medelfelet $d(\widehat{\theta})$ enligt ovan.

Just för $\widehat{\theta} = \bar{x}$ kan man visa att (likformigt i u)

$$\left| P\left(\frac{\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) - \bar{x}}{s(X_1^*, X_2^*, \cdots, X_n^*)/\sqrt{n}} \le u \right) - P\left(\frac{\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta}{s(X_1, X_2, \cdots, X_n)/\sqrt{n}} \le u \right) \right| \to 0$$

med sannoliket 1 då $n \to \infty$, dvs för nästan alla utfall $x_1, x_2, \cdots, x_n, \cdots$ av $X_1, X_2, \cdots, X_n, \cdots$. Den första sannolikheten skall som alltid tolkas som den betingade sannolikheten

$$P\left(\frac{\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) - \bar{x}}{s(X_1^*, X_2^*, \cdots, X_n^*)/\sqrt{n}} \le u \middle| X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n \right).$$

7.2.2 Parametrisk bootstrap

Den parametriska bootstrap-fördelningen konvergerar på motsvarande sätt för $\widehat{\theta} = \bar{x}$ mot den sanna fördelningen då $n \to \infty$.

Om t ex X_i:na är oberoende Exp(θ)-fördelade så innebär ju en parametrisk bootstrap att vi genererar $X_1^*, X_2^*, \cdots, X_n^*$ med Exp($\bar{x}$)-fördelningen. Eftersom $X_i^*/\bar{x}$ är Exp(1)-fördelad gäller att

$$\frac{\sum_{i=1}^n X_i^*}{\bar{x}} \text{ är } \Gamma(n, 1)$$

dvs att $\sum_{i=1}^n X_i^*$ är $\Gamma(n, 1/\bar{x})$. Å andra sidan är $\sum_{i=1}^n X_i$ ju $\Gamma(n, 1/\theta)$ och eftersom $1/\bar{x} \to 1/\theta$ då $n \to \infty$ (med sannolikhet 1) så gäller att (likformigt i u)

$$\left| P\left(\sum_{i=1}^n X_i^*/\bar{x} \le u \middle| X_1 = x_1; X_2 = x_2; \cdots; X_n = x_n \right) - P\left(\sum_{i=1}^n X_i/\theta \le u \right) \right| \to 0$$

då $n \to \infty$ med sannolikhet 1, dvs för "nästan alla" sekvenser $x_1, x_2, \cdots$.

Som tidigare påpekats saknar denna typ av teoretiska satser egentligt intresse eftersom vi vill kunna använda bootstrap (både parametrisk och icke-parametrisk) just för ändligt n och för väsentligt mer komplicerade stickprovsvariabler än $\bar{x}$.

7.3 Median

Medelfelsberäkningen för $\bar{x}$ saknar annat än teoretiskt intresse men situationen blir en annan om vi låter $\widehat{\theta}(x_1, x_2, \cdots, x_n) = \text{median}(x_1, x_2, \cdots, x_n)$ och försöker räkna ut medelfelet $\widehat{se}$ för denna. För en allmän fördelning finns helt enkelt inget explicit uttryck för $V(\text{median}(X_1, X_2, \cdots, X_n))$. Dock finns approximativa uttryck som bygger på normalfördelningen nämligen att variansen är $1/(4nf^2(\widetilde{m}))$ där $\widetilde{m}$ är den sanna medianen och f är tätheten för X_i:na.

För bootstrap-variansen $V_{\widehat{F}}(\text{median}(X_1^*, X_2^*, \cdots, X_n^*))$ kan man med viss möda konstruera explicita uttryck (se nedan) men detta är helt oväsentligt – vi kan ju ändå simulera denna fördelning godtyckligt väl genom att utföra flitig bootstrap. Vi beräknar då (för t ex $B = 1000$ eller $B = 10000$)

$$\widehat{se}^2_{boot,B} = \frac{1}{B-1}\sum_{b=1}^{B}(\widehat{\theta}(\boldsymbol{x}^{*b}) - \widehat{\theta}^*(\cdot))^2$$

där $\boldsymbol{x}^{*b} = (x_1^{*b}, x_2^{*b}, \cdots, x_n^{*b})$ är det b:te framlottade bootstrap-stickprovet och $\widehat{\theta}(\boldsymbol{x}^{*b}) = \text{median}(\boldsymbol{x}^{*b})$ för $b = 1, 2, \cdots, B$ och

$$\widehat{\theta}^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B}\theta(\boldsymbol{x}^{*b}).$$

Vi lottar fram B bootstrap-stickprov, skattar medianen i vart och ett av dessa och studerar dessa medianers spridning kring sitt gemensamma medelvärde. Eftersom vi vet att $\widehat{se}_{boot,B} \to \widehat{se}_{boot} = D_{\widehat{F}}(\text{median}(X_1^*, \cdots, X_n^*))$ då $B \to \infty$ ser vi att vi genom flitig simulering kan konsistent skatta den sanna bootstrap-standardavvikelsen med hjälp av simuleringarna. Den besvärliga exakta beräkningen av medelfelsskattningen $\widehat{se}_{boot}$ är alltså inte nödvändig för medianen liksom inte heller för några andra skattningar. För att kunna kontrollera att det hela fungerar är det dock trevligt att vi för medianen kan slippa göra den approximation som själva simuleringen av bootstrap-fördelningen innebär när vi försöker beräkna standardavvikelsen i bootstrap-fördelningen.

7.3.1 Exakt uttryck för medelfel med bootstrap för medianen

Vi antar för enkelhets skull att stickprovet innehåller ett udda antal distinkta observationer dvs att $n = 2r + 1$ för något heltal r.

Uppenbart är att $\widehat{\theta}(\boldsymbol{x}^*) = \text{medianen}(\boldsymbol{x}^*)$ är någon av de ursprungliga data x_i då vi har ett udda antal observationer i vårt stickprov.

Om vi sorterar stickprovet i storleksordning får vi $(x_{(1)}, x_{(2)}, \cdots, x_{(n)})$.

Kalla medianen i bootstrap-stickprovet för $\widetilde{m}^*$, dvs att

$$\widehat{\theta}(\boldsymbol{x}^*) = \text{medianen}(\boldsymbol{x}^*) = \widetilde{m}^* = x^*_{(r+1)}.$$

Vi får

$$P(\widetilde{m}^* \le x_{(i)}) = P(r+1 \text{ eller fler av } X_1^*, X_2^*, \cdots, X_n^* \le x_{(i)})$$

och alltså om vi låter $x_{(0)} = -\infty$

$$P(\widetilde{m}^* = x_{(i)}) = P(\widetilde{m}^* \le x_{(i)}) - P(\widetilde{m}^* \le x_{(i-1)})$$

Om vi låter U_i= antalet i bootstrap-stickprovet $\le x_{(i)}$ så är U_i då $\text{Bin}(n, i/n)$.

På samma sätt är V_i= antalet $\leq x_{(i-1)}$ ju Bin$(n,(i-1)/n)$.

Alltså får vi

$$P\left(\widetilde{m}^* = x(i)\right) = P(U_i \geq r+1) - P(V_i \geq r+1) = P(V_i \leq r) - P(U_i \leq r)$$

som beräknas med `binocdf(r,n,(1-1)/n)-binocdf(r,n,i/n)` där vi använt Matlabfunktionen `binocdf` som beräknar fördelningsfunktionen för binomialfördelningen.

Med 9 data blir $9 = 2r+1$ dvs $r = 4$ och medianskattningen blir $x_{(5)}$.

Vi får för $n = 9$ att med $p(i) = P(\widetilde{m}^* = x_{(i)})$ är

$$p(1) = p(9) = 0.0014,\ p(2) = p(8) = 0.0289,$$

$$p(3) = p(7) = 0.1145,\ p(4) = p(6) = 0.2207,\ p(5) = 0.2690$$

Detta är beräknat med följande Matlab-funktion där argumentet är ett udda tal n:

```
function p=pexact(n);
r=floor(n/2);
p(1)=1-binocdf(r,n,1/n);
p(n)=p(1);
for i=2:r+1;
        p(i)=binocdf(r,n,(i-1)/n)-binocdf(r,n,i/n);
        p(n+1-i)=p(i);
end;
```

Denna komplicerade kalkyl visar alltså hur vi i princip kan få fram ett exakt uttryck på $\widehat{se}_{boot}$, dvs den "ideala" bootstrap-skattningen av medelfelet för medianen för $B = \infty$. Detta eftersom vi får $P(\tilde{m}^* = x_{(i)}) = p(i)$ och allså känner den sanna bootstrap-fördelningen.

Med hjälp av de upprepade simuleringarna inte behöver beräkna denna exakt utan kommer att kunna få fram den som den observerade spridningen mellan de B bootstrap-skattningarna, dvs vi beräknar $\widehat{se}_{boot,B}$ för något stort B.

7.3.2 Vissa problem med medianen

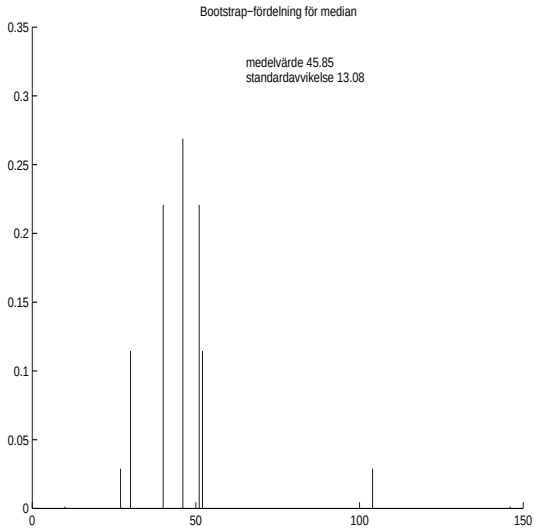
En tumregel som brukar anges (t ex i Efrons bok) är att $B = 200$ brukar räcka för att skatta medelfel. Detta är aningen optimistiskt åtminstone vad gäller medianen.

Vi tar ett konkret exempel (hämtat ur Efrons bok) med 9 observationer

52 104 146 10 51 30 40 27 46

som har $\bar{x}$=56.22 och $s = 42.4758$ samt median=46. Efron hävdar att $B = 200$ räcker för att skatta medelfelet för medianen.

Den sanna fördelningen för medianen i bootstrap-stickproven visas i figur 7.1 och den har väntevärde 45.86 och standardavvikelse 13.08. Notera att det finns (nästan osynliga) punktmassor 0.0014 i punkterna 10 och 146 men dessa små punktmassor har ganska stort inflytande på standardavvikelsen.



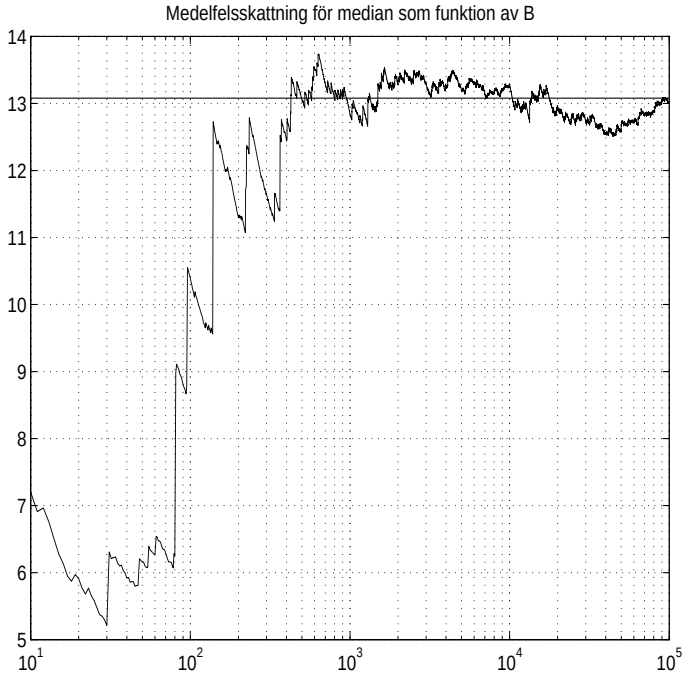
Figur 7.1: Exakt bootstrap-fördelning för medianen av 9 observationer. Väntevärde 45.85, standardavvikelse 13.08

När vi alltså simulerar bootstrap-fördelningen är det fördelningen i figur 7.1 vi försöker simulera och när vi försöker beräkna medelfelet $\widehat{se}_{boot}$ är det standardavvikelsen i denna fördelning vi försöker skatta.

Vi genererade B=100000 bootstrap-stickprov från dessa data och skattade medianen i vardera. I gränsen $B = \infty$ är medelfelet 13.08 och detta är alltså den sanna standardavvikelsen i figuren ovan. Man ser ur figur 7.2 att konvergensen mot detta värde är ytterst långsam för detta datamaterial.
Medianen är en osedvanligt "elak" parameter när det gäller så här små stickprov som $n = 9$, men det kanske kan ge anledning till en viss skepsis om att man klarar sig med små värden på B.

Svårigheten med medianen är att vi försöker simulera en fördelning med endast några få punktmassor för att beräkna variansen. Eftersom vi har små punktmassor i "ändarna" och att dessa kan ligga "långt ut", så förstår man att det krävs många simuleringar om man vill vara säker på att få ett bra värde på variansen. Hacken i figur 7.2 över "Medelfelsskattning som funktion av B" beror antagligen på att vissa av bootstrap-stickproven gett medianskattningar i de extrema punkterna. Plötsligt ökar då variansen dramatiskt. Ju fler simuleringar vi gör, desto mindre hack får vi, men konvergensen är långsam.

Med lite perspektiv på vad vi faktiskt uträttat, kan man dock tycka att en halvhyfsad ($\pm 10\%$) skattning av medelfelet duger. Vi bör dessutom inte tappa



Figur 7.2: Medelfelsskattning för median som funktion av B=antalet bootstrap-stickprov

bort det faktum att vi erhållit en vettig skattning av medelfelet för medianen av 9 data. I avsnitt 7.4.2 kommer vi nämligen att åtminstone i ett specialfall kunna visa att bootstrap ger en rimlig skattning av medelfelet.

7.4 Skattar bootstrap-medelfelet standardavvikelsen?

Det centrala är egentligen att vi vill få en bra skattning av standardavvikelsen för $\widehat{\theta}$, dvs av $D_F(\widehat{\theta}(X_1, X_2, \cdots, X_n))$, men detta kräver att vi bildar väntevärde över F, dvs den bakomliggande fördelningen som genererat våra data. Den har vi ju ingen kunskap om annat än i form av våra data $x_1, x_2, \cdots, x_n$ dvs i form av den empiriska fördelningen $\widehat{F}_n$. Man kan nu undra hur bra det går att skatta standardavvikelsen för stickprovsvariabeln $\widehat{\theta}$ i denna bakomliggande fördelning genom att studera bootstrap-skattningen av medelfelet för våra observerade data i situationer där vi vet svaret.

Som nämnts är medelfelsskattningen med bootstrap $\widehat{se}_{boot} = D_{\widehat{F}}(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*))$ att uppfatta som en betingad standardavvikelse givet observationerna, dvs som

$$D_{\widehat{F}}(\widehat{\theta}^*) = D\left(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) \middle| X_1 = x_1; X_2 = x_2; \cdots, X_n = x_n\right).$$

Detta utgör alltså en funktion av $x_1, x_2, \cdots, x_n$ som vi skulle kunna kal-

la $g(x_1, x_2, \cdots, x_n)$. En av de stora poängerna med bootstrap är att denna komplicerade funktion g inte behöver tas fram analytiskt utan vi simulerar den. Vi betraktar nu $g(X_1, X_2, \cdots, X_n)$ dvs vi ser det ursprungliga stickprovet som slumpmässigt och vill beräkna $E(g(X_1, X_2, \cdots, X_n))$. Detta innebär alltså att vi studerar hur medelfelsskattningen med bootstrap fungerar som punktskattning av den sanna standardavvikelsen $D(\widehat{\theta}(X_1, X_2, \cdots, X_n))$ där vi bildar väntevärde över alla stickprov vi kunde ha fått.

För att kunna göra kalkylerna lite lättare väljer vi att i stället studera $\widehat{se}^2_{boot}$ dvs $V_{\widehat{F}}(\widehat{\theta}^*)$ i stället för $D_{\widehat{F}}(\widehat{\theta}^*)$.

7.4.1 Aritmetiskt medelvärde

I fallet $\widehat{\theta} = \bar{x}$ så är ju $\widehat{se}^2_{boot} = \sum_{i=1}^n (x_i - \bar{x})^2/n^2$. Denna medelfelsskattning beror alltså av de ursprungliga observationerna. Vi nu ser dessa som stokastiska dvs betraktar $\widehat{se}^2_{boot}$ som en stickprovsvariabel och denna blir alltså

$$\widehat{se}^2_{boot} = \frac{1}{n^2} \sum_{i=1}^{n} (X_i - \bar{X})^2.$$

Notera hur vi nu ser $\widehat{se}^2_{boot}$ som en funktion av variablerna $X_1, \cdots, X_n$ dvs som stickprovsvariabel. Vi tittar nu på väntevärdet för denna stickprovsvariabel dvs beräknar

$$E_F(\widehat{se}^2_{boot}) = E_F\left(\sum_{i=1}^{n} (X_i - \bar{X})^2/n^2\right) =$$

$$= \frac{n-1}{n^2} E_F\left(\frac{1}{n-1}\sum_{i=1}^{n} (X_i - \bar{X})^2\right) = \frac{n-1}{n^2} E_F(s^2) = \frac{n-1}{n}\frac{\sigma^2}{n} = \frac{n-1}{n} V_F(\bar{X})$$

där vi utnyttjat att $E_F(s^2) = \sigma^2$. Detta betyder att vi med bootstrapmetodiken får fram (så när som på faktorn $(n-1)/n$) variansen för stickprovsvariabeln $\bar{X}$ åtminstone i genomsnitt – dvs i genomsnitt över alla de tänkbara stickprov vi kunde ha fått. Notera att i ovanstående kalkyl E_F innebär en väntevärdesbildning över fördelningen F. Vad ovanstående kalkyl alltså visar är att $\widehat{se}^2_{boot}$ är ett (nästan) väntevärdesriktigt sätt att skatta variansen för stickprovsvariabeln $\bar{X}$.

7.4.2 Median

Vi skulle på motsvarande sätt vilja kunna analysera medianen, men tyvärr finns ingen enkel formel för $V\,(\text{median}(X_1, X_2, \cdots, X_n))$ utan vi får titta på någon situation där man kan räkna exakt på medianen. En sådan situation är om $X_1, X_2, \cdots, X_n$ är oberoende exponentialfördelade Exp(1).

Låt $X_{(1)}, X_{(2)}, \cdots, X_{(n)}$ vara $X_1, X_2, \cdots, X_n$ sorterade i storleksordning. På grund av "minneslösheten" hos exponentialfördelningen och att minimum av

oberoende exponentialfördelade är exponentialfördelad där "intensiteterna" adderas gäller att

$$X_{(1)} \text{ är Exp}(1/n) \text{ och } X_{(i)} - X_{(i-1)} \text{ är Exp}(1/(n+1-i)) \text{ för } i = 2, 3, \cdots, n$$

samt att de alla är oberoende.

Alltså gäller för Exp(1)-fördelade variabler

$$E\left(\text{median}(X_1, X_2, \cdots, X_n)\right) = E(X_{(n+1)/2}) = \sum_{i=1}^{(n+1)/2} \frac{1}{n-i+1}$$

och

$$V\left(\text{median}(X_1, X_2, \cdots, X_n)\right) = \sum_{i=1}^{(n+1)/2} \frac{1}{(n-i+1)^2}$$

om n är udda (då är ju medianen entydigt bestämd – annars blir den medelvärdet av de två mittersta vilket komplicerar kalkylen aningen). Medianen är då $X_{(r)}$ där $r = (n+1)/2$. För $n = 9$ respektive $n = 101$ ger dessa formler värdena 0.7456 och 0.6981 för väntevärdet respektive 0.1162 och 0.0099 för variansen för $X_{((n+1)/2)}$). Dessa värden utgör alltså de sanna väntevärdena respektive varianserna för stickprovsvariabeln. Dessa värden skulle vi vilja kunna skatta med bootstrap-metodik utan att utnyttja antagandet om exponentialfördelning.

I syfte att undersöka om man kan erhålla detta med bootstrap-metodik, genererades 10000 stickprov av storlek 9 resp 101 (udda antal eftersom vi ville ha en unik "mittobservation" som skulle utgöra medianen) från Exp(1)-fördelningen. I varje sådant stickprov beräknades den teoretiska bootstrap-fördelningen för medianen med hjälp av den sannolikhetsfördelning som togs fram i avsnitt 7.3.1.

Fördelningen för $\widehat{\theta}^*$ (dvs bootstrap-fördelningen) blir alltså att $P(\widehat{\theta}^* = x_{(i)}) = p_i$ för $i = 1, 2, \cdots, n$ där p_i:na beräknades i avsnitt 7.3.1 på sidan på sidan 89 och $x_{(i)}$ var den i:te i storleksordning av $x_1, x_2, \cdots, x_n$. Väntevärdet i bootstrapfördelningen är alltså

$$E_{\widehat{F}}(\widehat{\theta}^*) = \sum_{i=1}^{n} p_i x_{(i)}$$

och vi får

$$\widehat{se}^2_{boot} = E_{\widehat{F}}((\widehat{\theta}^*)^2) - (E(\widehat{\theta}^*))^2 = \sum_{i=1}^{n} p_i x^2_{(i)} - (\sum_{i=1}^{n} p_i x_{(i)})^2.$$

Dessa utgör alltså funktioner av $x_1, x_2, \cdots, x_n$ och skulle kunna betraktas som två punktskattningar. Vi betraktar nu motsvarande stickprovsvariabler, dvs motsvarande funktioner av $X_1, X_2, \cdots, X_n$. Det är dessa stickprovsvariabler som vi analyserar genom att ta deras väntevärden och detta svarar mot att bilda medelvärde över alla ursprungliga observationer vi kunde ha fått.

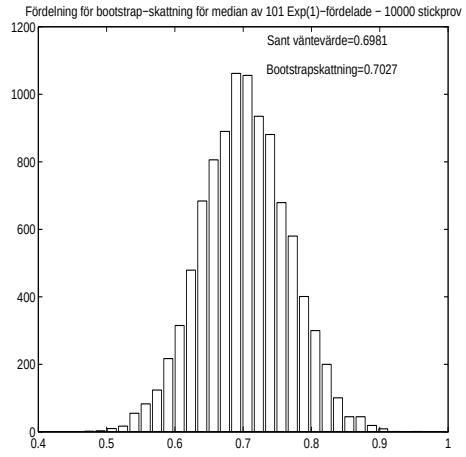
För varje enskilt stickprov erhölls alltså en separat bootstrap-fördelning där punktmassorna hela tiden var desamma men de hade olika placering beroende på hur stickprovet såg ut. I dessa 10000 bootstrap-fördelningar beräknades väntevärde respektive varians. Variansen svarar alltså mot kvadraten på medelfelsskattningen. Vi erhöll alltså 10000 olika bootstrap-fördelningar och hur väntevärde och varians varierade i dessa för $n = 101$ visas i figur 7.3 respektive 7.4.

Egentligen skulle man kunna få fram variansen för medianen i varje stickprov med bootstrap-generering med något stort B, men enligt vad som syntes ovan i figur 7.2 behöver B vara stor för just medianen och här kan man alltså få fram väntevärdes- och variansskattningen för $B = \infty$ analytiskt och för att göra kalkylen hanterlig används denna "genväg" till bootstrap-metodens medelfelsskattning.

Analys av median för Exp(θ) med bootstrappning

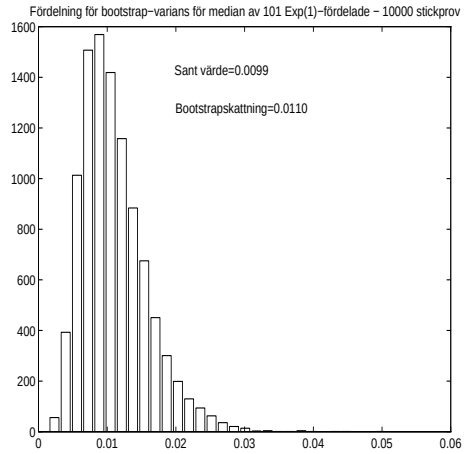
Antal	Sant väntev.	Bootstrap väntev.	Sant var.	Bootstrapvar.
9	0.7456	0.7974	0.1162	0.1552
101	0.6981	0.7027	0.0099	0.0110

Kolumnerna "Sant väntev." respektive "Sant var" anger alltså exakta värdena på väntevärde respektive varians enligt formlerna ovan. "Bootstrap väntev." respektive "Bootstrap var." är det genomsnittliga bootstrap-väntevärdet respektive bootstrap-variansen (alltså beräknade över i princip alla tänkbara ursprungliga observationer). För ett enskilt ursprungligt stickprov kan väntevärde repektive varians i bootstrap-fördelningen se annorlunda ut. Figurerna 7.3 och 7.4 visar variabiliteten i dessa beroende på hur det ursprungliga stickprovet råkade bli!



Figur 7.3: Fördelning för bootstrap-väntevärde för median av 101 Exp(1)-fördelade. Sant väntevärde=0.6981. Genomsnitt av bootstrap-väntevärden=0.7027

Enligt asymptotisk teori så är $X_{((n+1)/2)}$ asymptotiskt $N\left(\widetilde{m}, 1/(4nf^2(\widetilde{m}))\right)$ där $\widetilde{m}$ är medianen. Här ingår tätheten för den (i allmänhet okända) bakomlig-



Figur 7.4: Fördelning för bootstrap-varians för median av 101 Exp(1)-fördelade. Sann varians=0.0099. Genomsnitt av bootstrap-varians=0.0110

gande fördelningen F. I specialfallet Exp(θ)-fördelningen som vi studerar är $\widetilde{m} = \theta \ln 2$ och vi får

$$f(\widetilde{m}) = e^{-\theta \ln 2/\theta} = e^{-\ln 2} = \frac{1}{2}$$

så vi får att medianskattningen $X_{((n+1)/2)}$ är approximativt $N(\ln 2, 1/n)$. Detta ger $N(0.6931, 0.0099)$ för $n = 101$ och $N(0.6931, 0.111)$ för $n = 9$. Detta stämmer utomordentligt bra med de exakta värdena vad gäller varianserna. Faktiskt är dessa varianser bättre än vad Bootstrap-metoden ger. Man bör dock tänka på att normalapproximationen bygger på detaljerad kunskap om exponentialfördelningens täthet. Bootstrap-metodiken däremot fungerar helt oberoende av fördelningsantaganden. Vi kommer att återvända till medianen för exponentialfördelade i avsnitt 8.3.2 på sidan 105 då vi kommer att se att en väntevärdes-korrigering med bootstrap är ytterst effektiv.

Kapitel 8

Bias-skattning med bootstrap

8.1 Inledning

Bias betyder bristande väntevärdesriktighet, dvs ett systematiskt fel i skattningen. I detta kapitel tas metoder fram som möjliggör skattning av detta systematiska fel. När man väl skattat det systematiska felet vill man naturligtvis ofta använda sig av denna skattning för att korrigera sin ursprungliga skattning så den blir väntevärdesriktig.

Definition 8.1 *Bias och väntevärdesriktighet*
Om vi har en skattning $\widehat{\theta} = \tau(\boldsymbol{x})$ av $\theta = t(F)$ låter vi

$$bias_F = bias_F(\tau, \theta) = E_F\left(\tau(\boldsymbol{X})\right) - \theta =$$

$$= E_F(\tau(\boldsymbol{X})) - t(F)$$

vara det systematiska felet (bias) för τ.

Bias anger hur mycket skattningen av parametern i genomsnitt överstiger parametern. En väntevärdesriktig skattning har $bias_F = 0$ oavsett F (dvs oavsett θ:s värde). □

Man bör kanske påpeka att vi här inte nödvändigtvis betraktar enbart plug-in-skattningar utan $\widehat{\theta} = \tau(\boldsymbol{x})$ kan vara någon annan typ av skattning. Dock är vi speciellt intresserade av fallet att τ är en plug-in-skattning $\tau = t(\widehat{F})$.

Väntevärdesriktiga skattningar, dvs som uppfyller $E_F(\widehat{\theta}) = \theta = t(F)$ oavsett F (dvs oavsett värdet på θ) är att föredra. Dessa har alltså bias 0. Plug-in-skattningar är inte nödvändigtvis väntevärdesriktiga, men de har oftast liten bias.

För att med bootstrap skatta det systematiska felet (bias) för θ-skattningen $\tau(\boldsymbol{x})$ använder vi oss av plug-in-skattningen av $bias_F$, dvs vi betraktar bias för θ-skattningen $\tau(\boldsymbol{x})$ som den parameter som skall skattas! Vi har alltså en funktional av F som för en viss skattning τ är det systematiska felet som τ har som skattning av $\theta = t(F)$. Plug-in-skattningen av detta systematiska fel är alltså samma funktional fast applicerad på $\widehat{F}$.

Definition 8.2 *Bootstrap-skattning av bias*
Bootstrap-skattningen av bias för τ som skattning av $\theta = t(F)$ definierar vi som plug-in-skattningen av $bias_F = E_F(\tau(\boldsymbol{X})) - t(F)$ dvs som

$$\widehat{bias}_{\widehat{F}}(\tau, \theta) = E_{\widehat{F}}\left(\tau(\boldsymbol{X}^*)\right) - t(\widehat{F}).$$

Speciellt är

$$\widehat{bias}_{\widehat{F}}(\widehat{\theta}, \theta) = E_{\widehat{F}}\left(\widehat{\theta}(\boldsymbol{X}^*)\right) - \widehat{\theta}(\boldsymbol{x})$$

om $\widehat{\theta}$ är plug-in-skattningen $t(\widehat{F})$. □

Notera att detta betyder att vi för plug-in-skattningar skattar fördelningen för $\widehat{\theta}(\boldsymbol{X}) - \theta$ med den observerade fördelningen för

$$\widehat{\theta}^* - \widehat{\theta}(\boldsymbol{x}) = \widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) - \widehat{\theta}(x_1, x_2, \cdots, x_n)$$

som vi kan simulera fram med bootstrap-generering.

$\widehat{bias}_{\widehat{F}}$ är alltså tyngdpunkten i fördelningen för $\widehat{\theta}(\boldsymbol{X}^*) - \widehat{\theta}(\boldsymbol{x})$ vars fördelning vi, enligt den grundläggande bootstrap-hypotesen, tror någorlunda väl approximerar fördelningen för $\widehat{\theta}(\boldsymbol{X}) - \theta$. Om alltså $\widehat{\theta}(\boldsymbol{X})$ i genomsnitt överstiger θ (positiv bias) så borde alltså på motsvarande sätt $\widehat{\theta}(\boldsymbol{X}^*)$ överstiga $\widehat{\theta}(\boldsymbol{x})$

Observera att andra termen i $\widehat{bias}_{\widehat{F}}$ är plug-in-skattningen av $\theta = t(F)$, dvs $t(\widehat{F})$ så detta förutsätter att vi har tillgång till en plug-in-skattning av θ. Skulle man till äventyrs inte kunna ta fram en plug-in-skattning av θ kan man naturligtvis "fuska" genom att behandla $\tau(\boldsymbol{x})$ som om den vore en plug-in-skattning. Det är dock ganska ovanligt att inte man inte kan ange en plug-in-skattning – de flesta parametrar utgör ju någon aspekt av den bakomliggande fördelningen F som genererat observationerna.

I allmänhet kan vi inte beräkna $E_{\widehat{F}}\left(\tau(\boldsymbol{X}^*)\right)$ exakt, men vi har våra simuleringar att tillgå och kan skatta den med aritmetiska medelvärdet av våra bootstrap-skattningar.

Vi skaffar oss B oberoende bootstrap-stickprov $\boldsymbol{x}^{*1}, \boldsymbol{x}^{*2}, \cdots, \boldsymbol{x}^{*B}$. Sen beräknar vi skattningarna $\tau^*(b) = \tau(\boldsymbol{x}^{*b}), \quad b = 1, 2, \cdots, B$ för alla dessa B stickprov. Vi approximerar bootstrap-förväntan $E_{\widehat{F}}\left(\tau(\boldsymbol{X}^*)\right)$ med aritmetiska medelvärdet av dessa skattningar, dvs med

$$\tau_B^*(\cdot) = \frac{1}{B}\sum_{i=1}^{B} \tau^*(b) = \frac{1}{B}\sum_{i=1}^{B} \tau(\boldsymbol{x}^{*b})$$

eftersom detta enligt stora talens lag konvergerar mot $E_{\widehat{F}}\left(\tau(\boldsymbol{X}^*)\right)$ då $B \to \infty$.

Vi skattar slutligen bias baserad på dessa B replikationer med

$$\widehat{bias}_B = \tau_B^*(\cdot) - t(\widehat{F}).$$

eller med $\widehat{bias}_B = \widehat{\theta}^*_B(\cdot) - \widehat{\theta}(\boldsymbol{x})$ om vi har en plug-in-skattning.

Om våra observationer ligger i vektorn `x` och plug-in-skattningen $\widehat{\theta}$ beräknas med `m`-filen `estimate`, dvs genom `estimate(x)` så får man skattningen `bias_est` av det systematiska felet (bias) genom

```
boot=bootstrp(B,'estimate',x);
bias_est=mean(boot)-estimate(x)
```

med B "stort", t ex 1000 eller 10000.

8.1.1 Bias för skattningar av väntevärde och varians

För vissa skattningar, t ex av väntevärde och varians, kan vi studera hur bootstrap-skattningen av bias fungerar eftersom vi kan räkna exakt på dessa. Detta är ju dock av begränsat teoretiskt intresse – vi vill kunna använda bias-skattningarna just i så komplicerade situationer att vi inte kan göra en teoretisk analys. Att bootstrap fungerar bra i dessa "check"-situationer ger oss dock förhoppningen att den fungerar även i de komplicerade situationerna.

Exempel 8.1 *Väntevärde*
Om $\tau(\boldsymbol{x}) = \bar{x}$ och $t(F) = \theta = E_F(X)$ så vet vi att $\tau(\boldsymbol{x})$ är väntevärdesriktig eftersom $E_F(\bar{X}) = \theta$. Man inser lätt att bootstrap-skattningen $\widehat{bias}_{\widehat{F}} = 0$ eftersom $E_{\widehat{F}}(\bar{X}^*) = \bar{x}$ som medför att $\widehat{bias}_{\widehat{F}} = \bar{x} - t(\widehat{F}) = \bar{x} - \bar{x} = 0$ eftersom $\bar{x}$ är plug-in-skattningen av $\theta = E_F(X)$. □

Exempel 8.2 *Varians*
Variansskattningen

$$\widehat{\sigma}^2(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

har en bias$=-\sigma^2/n$ ty om $E_F(X_i) = m$ gäller att

$$\sum_{1}^{n}(X_i - \bar{X})^2 = \sum_{1}^{n}(X_i - m + m - \bar{X})^2 =$$

$$= \sum_{1}^{n}(X_i - m)^2 - 2(\bar{X} - m)\sum_{1}^{n}(X_i - m) + n(\bar{X} - m)^2 = \sum_{1}^{n}(X_i - m)^2 - n(\bar{X} - m)^2$$

som ger

$$E_F(\widehat{\sigma}^2(\boldsymbol{X})) = E_F(\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2) =$$

$$= \frac{1}{n}\left(\sum_{1}^{n} E(X_i - m)^2 - nE((\bar{X} - m)^2)\right) = \frac{1}{n}\left(nV(X_1) - nV(\bar{X})\right) = \sigma^2 - \frac{\sigma^2}{n}.$$

Detta är orsaken till att vi oftast använder oss av den "väntevärdeskorrigerade" skattningen

$$s^2 = \frac{1}{n-1}\sum_{1}^{n}(x_i - \bar{x})^2$$

av σ^2 i stället för $\widehat{\sigma}^2$.

Ovanstående ger att bootstrap-skattningen av bias är

$$\widehat{bias}_{\widehat{F}} = -\frac{1}{n^2}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

eftersom detta är plug-in-skattningen av $bias_F(\widehat{\sigma}^2, \sigma^2) = -\sigma^2/n$. □

I ovanstående exempel kan vi alltså beräkna den teoretiska bootstrap-skattningen av bias, men för en allmän skattning har vi ju alltid våra simuleringar av bootstrap-fördelningen att tillgå.

8.2 Bias-korrektion

När vi nu har en skattning av bias är det naturligt att korrigera vår skattning med hänsyn till denna.

Definition 8.3 *Bias-korrigerad skattning*
Om τ är en skattning av $\theta = t(F)$ är den med bootstrap bias-korrigerade skattningen $\bar{\tau}$

$$\bar{\tau} = \tau - \widehat{bias} = \tau - \widehat{bias}_{\widehat{F}} = \tau(\boldsymbol{x}) - E_{\widehat{F}}(\tau(\boldsymbol{X}^*)) + \widehat{\theta}(\boldsymbol{x})$$

Om τ är en plug-in-skattning dvs $\tau(\boldsymbol{x}) = \widehat{\theta}(\boldsymbol{x}) = t(\widehat{F})$ blir

$$\bar{\theta} = 2\widehat{\theta}(\boldsymbol{x}) - E_{\widehat{F}}(\widehat{\theta}(\boldsymbol{X}^*))$$

den bias-korrigerade skattningen. □

I praktiken kan ju inte $E_{\widehat{F}}(\widehat{\theta}(\boldsymbol{X}^*))$ beräknas exakt men vi har ju skattningen $\widehat{bias}_B = \widehat{\theta}^*_B(\cdot) - \widehat{\theta}(\boldsymbol{x})$ baserad på de B bootstrap-stickproven och detta ger den (skattade) bias-korrigerade skattningen

$$\bar{\theta}_B = 2\widehat{\theta}(\boldsymbol{x}) - \widehat{\theta}^*_B(\cdot).$$

Detta beror på att $E_{\widehat{F}}(\widehat{\theta}(\boldsymbol{X}^*))$ utgör tyngdpunkten i bootstrap-fördelningen och denna skattas naturligen med aritmetiska medelvärdet av våra bootstrap-skattningar. I Matlab får man alltså den bias-korrigerade skattningen `est_korr` av en plug-in-skattning genom

```
boot=bootstrp(B,'estimate',x);
est_korr=2*estimate(x)-mean(boot)
```

Exempel 8.3 *Bias-korrigering av variansskattningen*
Om vi bias-korrigerar

$$\widehat{\theta} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2,$$

(dvs plug-in-skattningen av σ^2) så erhåller vi den bias-korrigerade skattningen

$$\bar{\theta} = \widehat{\theta}(\boldsymbol{x}) - \widehat{bias}_{\widehat{F}} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 - (-\frac{1}{n^2}\sum_{i=1}^{n}(x_i - \bar{x})^2) =$$

$$= \frac{n+1}{n^2}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Om vi gör approximationen

$$\frac{n+1}{n^2} = \frac{1}{n}\left(1 + \frac{1}{n}\right) \approx \frac{1}{n}\cdot\frac{1}{1-\frac{1}{n}} = \frac{1}{n-1}$$

med användning av $1/(1-x) \approx 1+x$ så ser vi att vi hamnar mycket nära den sedvanliga "väntevärdes-korrigerade" σ^2-skattningen

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

genom att bias-korrigera den ursprungliga skattningen $\widehat{\sigma}(\boldsymbol{x})$. Här behövde vi inga simuleringar för att se vad bias-korrigeringen innebar, men notera att denna (nästan korrekta) korrigering sker helt automatiskt. Om vi bias-korrigerat med hjälp av bootstrap-simuleringarna hade vi alltså utan någon som helst kunskap om skattningens egenskaper fått en (nästan perfekt) korrekt väntevärdes-korrigerad skattning genom en rent mekanisk operation.

Man kan f ö påpeka att om man skattar systematiska felet med jackknife-metoden (som beskrivs i kapitel 9) så erhåller man genom bias-korrigering detta exakt just för variansen. Detta beror på att man faktiskt har avpassat jackknife-skattningen av bias på ett sådant sätt att det skall stämma exakt för just variansen. Mer om detta när vi behandlar jackknife-skattningen av bias.□

8.2.1 Skall man bias-korrigera?

Man kan tycka att det verkar tämligen självklart att man bör korrigera för ett systematiskt fel (bias). Problemet är att om man kan skatta bias och så korrigerar sin skattning och som slutresultat tar den biaskorrigerade skattningen så får ju denna antagligen andra statistiska egenskaper än den ursprungliga icke-korrigerade skattningen, t ex har den antagligen ett annat medelfel. Bootstrap gav ju en medelfelsskattning för den ursprungliga skattningen.

Speciellt är det intressant att ställa bias i relation till medelfelet i den ursprungliga skattningen som vi mekaniskt skattar med $\widehat{se}_B$ ur våra bootstrapstickprov. Om det systematiska felet (bias) är litet i förhållande till medelfelet finns det faktiskt ingen större anledning att bias-korrigera. Detta följer av följande överväganden.

För en godtycklig konstant a gäller:

$$E\left((X-a)^2\right) = V(X) + (E(X)-a)^2$$

som alltså t ex säger att

$$E\left((\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta)^2\right) =$$

$$= V(\widehat{\theta}(X_1, X_2, \cdots, X_n)) + \left(E(\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta)^2\right) =$$

$$= se_F^2(\widehat{\theta}) + (bias_F(\widehat{\theta}, \theta))^2.$$

Storheten $E\left((\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta)^2\right)$ kallas medelkvadratfelet (MSE – Mean Square Error).

Man ser med hjälp av Taylorutvecklingen

$$\sqrt{1+x} = (1+x)^{1/2} \approx 1 + \frac{1}{2}x$$

att

$$\sqrt{MSE} = \sqrt{se_F^2(\widehat{\theta}) + bias_F(\widehat{\theta}, \theta)^2} = se_F(\widehat{\theta})\sqrt{1 + \left(\frac{bias_F}{se_F}\right)^2} \approx$$

$$\approx se_F(\widehat{\theta})\left(1 + \frac{1}{2}\left(\frac{bias_F}{se_F}\right)^2\right).$$

$\sqrt{MSE}$ kallas "Root Mean Square Error" och är ett mått på osäkerheten som både tar hänsyn till bias och medelfel. Om bias är liten i förhållande till medelfelet finns det alltså liten anledning att bias-korrigera utan $\sqrt{MSE} \approx se_F$. T ex om $bias_F = se_F/4$ gäller att $\sqrt{MSE} \approx 33/32 se_F \approx 1.03 se_F$ och att det alltså är rätt meningslöst att bias-korrigera.

8.2.2 Bias-skattning med Double bootstrap

Ett annan lösning av problemet att beräkna medelfel för den bias-korrigerade skattningen är att se hela förfarandet med bias-korrektion som ingående i skattningsförfarandet! För en föreslagen ursprunglig skattning $\widehat{\theta}$ är det alltså "egentligen" den väntevärdes-korrigerade skattningen $\bar{\theta} = \widehat{\theta} - \widehat{bias}_{\widehat{F}}$ vi är ute efter att analysera vad gäller t ex medelfel.

Detta innebär att man skulle kunna utföra bootstrap på även detta skattningsförfarande!! Detta brukar kallas "double bootstrap", där man alltså utgående

från de B bootstrap-stickproven medelst bootstrap bias-korrigerar för varje enskilt bootstrap-stickprov.

Med bootstrap-stickproven $\boldsymbol{x}^{*1}, \boldsymbol{x}^{*2}, \cdots, \boldsymbol{x}^{*B}$ skaffar vi oss för stickprov b alltså först skattningen $\widehat{\theta}(\boldsymbol{x}^{*b})$ som vi sen med hjälp av bootstrap försöker bias-korrigera. Alltså skall vi utifrån detta stickprov $\boldsymbol{x}^{*b}$ utföra en bootstrap och väntevärdes-korrigera $\widehat{\theta}(\boldsymbol{x}^{*b})$ till $\bar{\theta}(\boldsymbol{x}^{*b}) = 2\widehat{\theta}(\boldsymbol{x}^{*b}) - \widehat{\theta}^{*b}(\cdot)$ där $\widehat{\theta}^{*b}(\cdot)$ är aritmetiska medelvärdet av skattningarna i bootstrap-stickproven framlottade ur $\boldsymbol{x}^{*b}$.

En skattning av $\bar{\theta}$:s medelfel får vi sedan genom

$$\widehat{se}_{boot,B}(\bar{\theta}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\bar{\theta}(\boldsymbol{x}^{*b}) - \bar{\theta}(\cdot))^2}$$

där $\bar{\theta}(\cdot) = \sum_1^B \bar{\theta}(\boldsymbol{x}^{*b})/B$.

Detta förfarande blir extremt beräkningsintensivt och är antagligen inte särskilt realistiskt. Med $B = 1000$ och om vi gör 500 bootstrap i varje bootstrap-stickprov tvingas vi alltså beräkna skattningen 500000 gånger.

8.3 Bias-skattning för median

Redan när vi studerar en så enkel skattning som medianen får vi ett systematiskt fel som med traditionell analys är mycket svårt att hantera utan att lägga på mycket restriktiva antaganden om den bakomliggande fördelningen F som genererat observationerna.

Exempel 8.4 *Skattning av median*
Följande exempel är hämtat från Urban Hjort:"Computer Intensive Statistical Methods".

Vi har observerat elva livslängder

5700 36300 12400 28000 19300 21500 12900 4100 91400 7600 1600

Medianen θ i fördelningen F som genererat livslängderna skattas med den 6:e i storleksordning av observationerna dvs $\widehat{\theta} = x_{(6)} = 12900$ som ju utgör plug-in-skattningen av medianen θ.

Vi gör en bootstrap av våra data $B = 1000$ ggr, skattar medianen med den mittersta observationen i vardera av dessa stickprov och erhåller medelvärdet av dessa till $\widehat{\theta}^*(\cdot) = 14843$ och vidare erhåller vi medelfelsskattningen 5737.

I Matlab görs detta med koden där vektorn `x` innehåller våra 11 observationer

```
boot=bootstrp(1000,'median',x);
bootest=mean(boot)
seboot=std(boot)
```

dvs vi fick för $E_{\widehat{F}}(\widehat{\theta}(\boldsymbol{X}^*))$ approximationen `bootest`=14843 och för $D_{\widehat{F}}(\widehat{\theta}(\boldsymbol{X}^*))$ approximationen $\widehat{se}_{boot,1000}$=`seboot`=5737.

Vår bias-skattning blir då $\widehat{\theta}^*(\cdot) - \widehat{\theta} = 14843 - 12900 = 1943$ och detta ger den bias-korrigerade skattningen $\bar{\theta} = 12900 - 1943 = 10957$. □

8.3.1 Skattning av medianen med medelvärdet

Det kan verka underligt att skatta medianen med $\bar{x}$. Detta är ju inte plug-in-skattningen (den är ju medianen av empiriska fördelningen dvs 12900), men om fördelningen är symmetrisk kring $E(X)$ så är medianen=$E(X)$ och eftersom man vet att $\bar{x}$ är optimal t ex om fördelningen är normalfördelning, så är det (i alla fall i den situationen) bättre att skatta medianen θ med $\bar{x}$ än med medianen av observationerna.

Vi vill nu använda $\bar{x}$ som skattning av medianen. Om våra data skulle komma från en exponentialfördelning så skulle ju en listigare skattning av medianen vara $\bar{x}\ln 2$ eftersom medianen=$E(X)\ln 2$ för just exponentialfördelningen. Detta konstaterande förutsätter dock att vi har gjort detta detaljerade fördelningsantagande. Vår nya skattning $\bar{x}$ av medianen skulle alltså (t ex i exponentialfördelningsfallet) ha en kraftig bias, men vi kommer att se att biaskorrigeringen faktiskt automatiskt kommer att kompensera för detta!

Vår nya skattning av medianen θ är nu alltså $\tau(\boldsymbol{x}) = \bar{x} = 21891$. Vi gör nu en bootstrap av våra data och beräknar $\bar{x}^*$ i dessa bootstrap-stickprov och erhåller därvid deras genomsnitt till 22608.

Egentligen är denna bootstrap onödig eftersom vi kan räkna exakt på just $\bar{x}$ och skulle alltså få bootstrap-väntevärdet $E_{\widehat{F}}(\tau(\boldsymbol{X}^*))$=21891 om vi tagit den teoretiska bootstrap-fördelningen, men för realismen i exemplet så använder vi oss av värdet 22608.

Vi kan nu göra bias-skattningen och erhåller

$$\widehat{bias}_{\widehat{F}} = E_{\widehat{F}}(\tau(\boldsymbol{X}^*)) - t(\widehat{F}) = 22608 - 12900 = 9708$$

dvs en mycket kraftig bias. Notera att andra termen är plug-in-skattningen av medianen och inte $\widehat{\theta} = \bar{x}$! Detta förfarande förutsätter att vi har tillgång till en plug-in-skattning, men sådana är ofta lätta att hitta.

Alltså skall vi korrigera vår skattning $\tau(\boldsymbol{x}) = \bar{x} = 21891$ med denna bias och erhåller då $\bar{\theta} = 21891 - 9708 = 12183$ dvs ungefär samma resultat som då vi använde medianen som skattning.

Om vi hade låtit bli att utföra bootstrap-genereringen när vi analyserade $\bar{x}$ och i stället använt den teoretiska bootstrap-skattningen hade vi fått $E_{\widehat{F}}(\tau(\boldsymbol{X}^*)) =$ 21891 och alltså fått bias-skattningen 21891-12900=8991. Detta hade i sin tur gett den bias-korrigerade skattningen $\bar{\theta} = 21891 - 8991 = 12900$ dvs precis plug-in-skattningen av medianen!

Detta är naturligtvis ingen tillfällighet utan beror på att när $\tau(\boldsymbol{x}^*)$ är centrerad kring $\tau(\boldsymbol{x})$ dvs att $E_{\widehat{F}}(\tau(\boldsymbol{X}^*)) = \tau(\boldsymbol{x})$ så får vi den bias-korrigerade skattningen till

$$\bar{\theta} = \tau(\boldsymbol{x}) - \widehat{bias}_{\widehat{F}} = \tau(\boldsymbol{x}) - \left(E_{\widehat{F}}(\tau(\boldsymbol{X}^*)) - t(\widehat{F})\right) = t(\widehat{F})$$

dvs plug-in-skattningen av θ.

Detta att vi med hjälp av bias-korrigering av $\bar{x}$ kommer fram till plug-in-skattningen kan tyckas vara lite futtigt speciellt som den resulterande plug-in-skattningen, dvs medianen, själv har en viss bias. Avsikten är snarare att visa att en skattning med extremt stor bias på ett automatiskt sätt korrigeras till en skattning med avsevärt mindre bias. Samma mekanism fungerar för en vettig skattning med liten bias, dvs vi lyckas i stort sett få bort bias genom rent mekaniska operationer.

8.3.2 Exakt kalkyl för medianen

Medianen av observationerna har alltså en viss bias som skattning av medianen i den bakomliggande fördelningen. Eftersom det (vad vi vet) inte finns någon explicit formel för bias hos medianen måste vi studera något specialfall om vi vill kunna se hur bra bias-korrigeringen i verkligheten är.

Antag (för att vi ska kunna räkna på medianen) att våra data består av 11 värden $x_1, x_2, \cdots, x_{11}$ från en exponentialfördelning Exp(θ). Eftersom θ är en ren skalfaktor kan vi anta att $\theta = 1$.

Man kan visa att $X_{(i)} - X_{(i-1)}$ är Exp($1/(12-i)$), $i = 1, 2, \cdots, 11$ med $X_{(0)} = 0$. Detta eftersom minimum av exponential-fördelade variabler är exponentialfördelad med en intensitet som är summan av de ingående variablernas intensiteter. Vidare utnyttjar man att exponentialfördelningen saknar minne så att

$$P(X > s + t | X > s) = P(X > t) \text{ för alla } s, t \geq 0.$$

Denna egenskap karaktäriserar faktiskt exponentialfördelningen.

Alltså får vi

$$E(X_{(i)}) = \sum_{j=1}^{i} \frac{1}{12-j}.$$

Som tidigare låter vi $X_{(i)}$ betyda den i:te ordningsvariabeln, dvs den i:te i storleksordning av $X_1, X_2, \cdots, X_n$.

Vår skattning $\widehat{\theta} = x_{(6)}$ =medianen i vårt stickprov om 11 data har väntevärde

$$E(X_{(6)}) = \frac{1}{11} + \frac{1}{10} + \frac{1}{9} + \frac{1}{8} + \frac{1}{7} + \frac{1}{6} \approx 0.7365$$

medan den sanna medianen i Exp(1)-fördelningen är $\ln 2 \approx 0.6931$. Vi vill nu biaskorrigera vår medianskattning $x_{(6)}$ och ska då göra en bootstrap av data.

Vi gör en analys med hjälp av den teoretiska bootstrap-fördelningen eftersom vi ju kunde räkna exakt på just medianen och får då med $p_i = P(X^*_{(6)} = x_{(i)}),\ i = 1, 2, \cdots, 11$ enligt avsnitt 7.3.1 på sidan 88 att

$$E_{\widehat{F}}(X^*_{(6)}) = \sum_{i=1}^{11} p_i x_{(i)}.$$

där alltså dessa blir $p_1 = p_{11} = 0.0002$, $p_2 = p_{10} = 0.0070$, $p_3 = p_9 = 0.0440$, $p_4 = p_8 = 0.1215$, $p_5 = p_7 = 0.2059$, $p_6 = 0.2427$. Att vi tar fram detta väntevärde i bootstrap-fördelningen med hjälp av den sanna fördelningen är egentligen mest av praktiska skäl – vi skulle kunna få fram väntevärdet $E_{\widehat{F}}(X^*_{(6)})$ hur bra som helst med hjälp av simulering.

Bootstrapskattningen $\widehat{bias}_{\widehat{F}}$ av bias blir

$$\widehat{bias}_{\widehat{F}} = \sum_{i=1}^{11} p_i x_{(i)} - x_{(6)}$$

och den biaskorrigerade skattningen $\bar{\theta}$ blir

$$\bar{\theta} = \widehat{\theta} - \widehat{bias}_{\widehat{F}} = x_{(6)} - \widehat{bias}_{\widehat{F}} = 2x_{(6)} - \sum_{i=1}^{11} p_i x_{(i)}.$$

Notera hur $\bar{\theta}$ blir en explicit funktion av våra ursprungliga observationer $x_1, \cdots, x_n$, dvs är en punktskattning som alltså oavsett bakomliggande fördelning bör ha mindre systematiskt fel än att ta medianen i stickprovet som ju utgjorde plug-in-skattningen.

$\bar{\theta}$ ger något konkret numeriskt för numeriska observationer men man kan undra över vad väntevärdet $E_F(\bar{\theta})$ blir, dvs vad det blir för biaskorrigerad skattning i genomsnitt över alla tänkbara datauppsättningar. Detta svarar mot att vi analyserar stickprovsvariabeln

$$\bar{\theta}(X_1, X_2, \cdots, X_n) = 2X_{(6)} - \sum_{i=1}^{11} p_i X_{(i)}$$

som hör till punktskattningen

$$\bar{\theta} = x_{(6)} - \widehat{bias}_{\widehat{F}} = 2x_{(6)} - \sum_{i=1}^{11} p_i x_{(i)}.$$

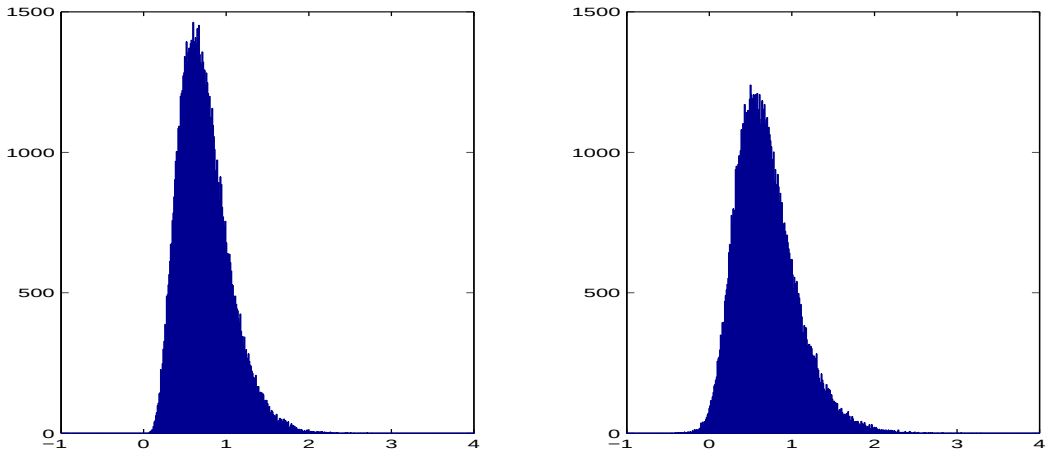
Vi får

$$E_F(\bar{\theta}) = E_F\left(\widehat{\theta} - \widehat{bias}_{\widehat{F}}\right) = E_F(X_{(6)}) - E_F\left(\widehat{bias}_{\widehat{F}}\right) = 2E_F(X_{(6)}) - \sum_{i=1}^{11} p_i E_F(X_{(i)}).$$

Koefficienterna p_i härledde vi tidigare. Man erhåller då att $E_F(\bar{\theta}) = 0.6918$ dvs ytterst nära det sanna värdet $\ln 2 \approx 0.6931$. Den biaskorrigerade skattningen blir i genomsnitt nästan helt väntevärdesriktig. Vi kommer alltså i genomsnitt att korrigera den ursprungliga skattningen (som ju har väntevärde 0.7365) till ett värde ytterst nära det sanna värdet 0.6931.

Att bias-korrigeringen ger ett så bra resultat (åtminstone i genomsnitt) kan man tycka är särdeles uppseendeväckande. Vi har ju i på inget sätt utnyttjat någon kunskap om att data kom från en exponentialfördelning och att alltså parametern i verkligheten var $\ln 2$ utan vi kom fram till bias-korrigeringen enbart genom flitigt framlottande från våra 11 observationer!

Dock kan man påpeka att vi får betala ett pris för att ha bias-korrigerat skattningen, nämligen att den bias-korrigerade skattningen i allmänhet har större varians än den ursprungliga skattningen.



Figur 8.1: Fördelning för $X_{(6)}$ till vänster och för den bias-korrigerade $\bar{\theta}$ till höger

Det är lite komplicerat att beräkna $V(\bar{\theta})$ exakt, men naturligtvis går det bra att simulera. Följande Matlab-program beräknar först p_i:na, genererar 100000 stickprov om 11 vardera från Exp(1)-fördelningen, sorterar varje stickprov, beräknar $\bar{\theta}$ samt dennas medelfel. Vidare beräknas medelfelet för den ursprungliga skattningen $x_{(6)}$. Notera att `sort`, `std` och `median` gör en kolonn-vis kalkyl då argumentet är en matris.

```
p=pexact(11);
x=exprnd(1,11,100000);
x=sort(x);
thetabar=2*median(x)-p*x;
medelfel=std(thetabar)
se=std(median(x));
```

Man får $D(\bar{\theta}) = 0.3703$ medan $D(X_{(6)}) = 0.3074$. Detta sista resultat hade man också kunna få analytiskt som $\sum_{i=1}^{6} 1/(12-i)^2 \approx 0.3073$. Vi ser alltså att man genom att biaskorrigera får ett större medelfel i skattningen. Utseendet på de två fördelningarna framgår av figur 8.1. Notera att den biaskorrigerade faktiskt kan bli negativ!

8.4 Exempel: Poisson-fördelning

Antag att vi har observationerna $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ från oberoende Po(θ)-fördelade variabler $X_1, X_2, \cdots, X_n$. Vi vill skatta $P(X = 0) = e^{-\theta}$ ur våra observationer. Vi använder oss av plug-in-skattningen $\widehat{\theta}(\boldsymbol{x}) = e^{-\bar{x}}$.

Vi har t ex $n = 5$ och observationerna $0, 3, 3, 1, 2$ dvs $\bar{x} = 1.8$ och skattningen blir $e^{-1.8} = 0.1653$. Vi vill nu bias-korrigera denna skattning. Vi noterar att

$$bias_F = E(e^{-\bar{X}}) - e^{-\theta}.$$

och detta kan vi faktiskt beräkna eftersom $Y = \sum_1^n X_i$ är Po($n\theta$). Att konkret beräkna detta är i princip samma sak som att beräkna sannolikhetsgenererande funktionen (ibland kallad z-transformen) för Y. Vi får

$$E(e^{-\bar{X}}) = E(\exp(-\frac{1}{n}\sum_{i=1}^{n} X_i)) = \sum_{k=0}^{\infty} e^{-k/n} P(Y = k) = \sum_{k=0}^{\infty} e^{-k/n} \frac{(n\theta)^k}{k!} e^{-n\theta} =$$

$$= \sum_{k=0}^{\infty} \frac{(n\theta e^{-1/n})^k}{k!} e^{-n\theta} = \exp(-n\theta)\exp(n\theta e^{-1/n}) = \exp(n\theta(e^{-1/n} - 1))$$

och alltså gäller att

$$bias_F = \exp(n\theta(e^{-1/n} - 1)) - e^{-\theta}.$$

$bias_F$ beror av θ och n där i alla fall n är känt. I mitt ovanstående numeriska exempel var i verkligheten $\theta = 2$ så

$$bias_F = \exp(10(e^{-1/5} - 1)) - e^{-2} \approx 0.0279.$$

Om vi nu går över till bootstrap och studerar $\widehat{bias}_{\widehat{F}}$ dvs bootstrap-skattningen av bias så har vi

$$\widehat{bias}_{\widehat{F}} = E_{\widehat{F}}(e^{-\bar{X}^*}) - e^{-\bar{x}} = e^{-\bar{x}} \left(E_{\widehat{F}}(e^{-(\bar{X}^* - \bar{x})}) - 1 \right).$$

Som vanligt kan vi inte beräkna detta exakt, men vi kan ju som alltid simulera värdet. Matlab-koden blir

```
boot=bootstrp(1000,'sum',x);
expboot=exp(-boot/5);
mean(expboot)-exp(-mean(x))   % gav 0.0240
```

och vi fick alltså skattningen 0.0240 av $\widehat{bias}_{\widehat{F}}$ att jämföra med det "sanna" värdet 0.0279. Bias-korrektionen verkar alltså vara rätt bra. Notera att denna korrektion kunde göras i Matlab utan något annat än flitig lottning av våra 5 observationer.

Ovanstående kan kanske räcka som numerisk illustration till att det är något substantiellt vi utför när vi med bootstrap skattar bias. Vi vill dock kunna göra en lite mer allmän jämförelse av den exakta bias-korrigeringen i denna situation och den som görs med bootstrap. I syfte att göra detta, tar vi fram andra approximationer av $bias_F$ och $\widehat{bias}_{\widehat{F}}$ genom att Taylorutveckla uttrycken.

Vi vill ha ett approximativt uttryck för $bias_F$ med allmänt θ och n och gör därför en Taylor-utveckling av $e^{-1/n} - 1$ och erhåller då om n är "stort"

$$bias_F = e^{-\theta}\left(\exp(n\theta(e^{-1/n}-1)+\theta)-1\right) \approx$$

$$\approx e^{-\theta}\left(\exp(n\theta(-\frac{1}{n}+\frac{1}{2n^2})+\theta)-1\right) \approx e^{-\theta}\left(\exp(\frac{\theta}{2n})-1\right) \approx e^{-\theta}\frac{\theta}{2n}$$

som numeriskt med $\theta = 2$ och $n = 5$ ger 0.0271, dvs rätt litet approximationsfel även för detta rätt lilla n.

Vi får med Taylor-serien $e^{-x} = \sum_0^\infty (-1)^k x^k/k!$ att

$$\widehat{bias}_{\widehat{F}} = e^{-\bar{x}}\left(\sum_{k=0}^{\infty}(-1)^k \frac{E_{\widehat{F}}\left((\bar{X}^* - \bar{x})^k\right)}{k!} - 1\right).$$

Vi ser att termen för $k = 0$ försvinner pga av den sista 1:an och termen för $k = 1$ försvinner eftersom $E_{\widehat{F}}(\bar{X}^*) = \bar{x}$. Vi har vidare för $k = 2$ termen

$$E_{\widehat{F}}\left((\bar{X}^* - \bar{x})^2\right) = V_{\widehat{F}}(\bar{X}^*) = \frac{1}{n^2}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

där vi använt tidigare resultat om $V_{\widehat{F}}(\bar{X}^*)$. Om vi bortser från termerna med $k \geq 3$ får vi

$$\widehat{bias}_{\widehat{F}} \approx e^{-\bar{x}}\frac{\sum_1^n (x_i - \bar{x})^2}{2n^2} \approx 0.0225$$

där det numeriska värdet erhållits med våra 5 observationer. Täljaren är (om vi delar med n) en skattning av $\theta = V(X)$ och detta betyder att vi approximativt har

$$\widehat{bias}_{\widehat{F}} \approx e^{-\theta}\frac{\theta}{2n}$$

om vi ersätter $\bar{x}$ med θ. Man ser alltså att vi approximativt får den sanna bias-korrektionen $bias_F$ som ju var approximativt $e^{-\theta}\theta/2n$.

Kapitel 9

Jackknife-skattningar

9.1 Inledning

Jackknife är en teknik som innebär att man ur en datauppsättning om n data skaffar sig n datauppsättningar bestående av $n-1$ data vardera genom att systematiskt plocka bort precis en observation i taget. Detta ger en möjlighet att skatta bias (systematiska felet) och medelfelet för en skattning $\widehat{\theta}$. Man kan med hjälp av medelfelet konstruera approximativa konfidensintervall om man dessutom antar att skattningen är approximativt normalfördelad, som ger ett approximativt 95%-igt konfidensintervall av typen

$$\text{skattning} \pm 1.9600 \text{ medelfel.}$$

Vi kommer i kapitel 11 att visa att medelfelsskattningen med jackknife ger samma resultat som medelfelsskattningen med hjälp av bootstrap av en linearisering av skattningen och att bias-skattningen med jackknife svarar mot bootstrap av en "kvadratisk" approximation av skattningen.

9.2 Jackknife-stickprov

Med Jackknife skaffar vi oss nya stickprov genom att systematiskt plocka bort precis en observation i taget. Detta betyder att om vårt ursprungliga stickprov består av $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ så blir det i:te Jackknife-stickprovet

$$\boldsymbol{x}_{(i)} = (x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n), \quad i = 1, 2, \cdots, n.$$

Den i:te Jackknife-skattningen $\widehat{\theta}_{(i)}$ av $\widehat{\theta} = \tau(\boldsymbol{x})$ blir då

$$\widehat{\theta}_{(i)} = \tau(\boldsymbol{x}_{(i)})$$

Notera att vi alltså ur vårt ursprungliga stickprov av storlek n tillverkar n st nya fiktiva stickprov vardera av storlek $n-1$.

Egentligen förutsätter vi här en sorts "kontinuitet" av $\widehat{\theta}$ när vi byter stickprovsstorlek från n till $n-1$. Medianen är här lite extra besvärlig eftersom den har olika definition för jämna och udda n.

För plug-in-skattningar $\widehat{\theta} = t(\widehat{F})$ blir $\widehat{\theta}_{(i)} = t(\widehat{F}_{(i)})$ där $\widehat{F}_{(i)}$ är fördelningen som lägger massan $1/(n-1)$ i vardera av de $n-1$ punkterna i $\boldsymbol{x}_{(i)}$, dvs alla data utom x_i.

9.3 Bias-skattning med Jackknife

Definition 9.1 *Jackknife-skattning av systematiskt fel (bias)*
Jackknife-skattningen av det systematiska felet är

$$\widehat{bias}_{jack} = (n-1)(\widehat{\theta}_{(\cdot)} - \widehat{\theta})$$

där

$$\widehat{\theta}_{(\cdot)} = \frac{1}{n}\sum_{i=1}^{n}\widehat{\theta}_{(i)}.$$

□

En fördel med denna bias-skattning är ju att den endast kräver n st beräkningar av skattningen, medan man för Bootstrap ofta behöver välja B mycket stor för att bias-skattningen skall bli tillförlitlig. En nackdel med Jackknife är att den bara fungerar för plug-in-skattningar och att den dessutom kräver att skattningen är en "snäll" funktion. T ex fungerar Jackknife inte för medianen.

9.3.1 Bakgrund till bias-skattningen

Bakgrunden till definitionen av $\widehat{bias}_{jack}$ är att många skattningar har en serieutveckling

$$E(\widehat{\theta}) = \theta + \frac{a_1(F)}{n} + \frac{a_2(F)}{n^2} + \cdots,$$

dvs att

$$\text{bias}_F = \frac{a_1(F)}{n} + O(\frac{1}{n^2}).$$

Notera att jackknife-stickproven består av $n-1$ observationer vardera så

$$E(\widehat{\theta}_{(i)}) = \theta + \frac{a_1(F)}{n-1} + \frac{a_2(F)}{(n-1)^2} + \cdots$$

Om vi nu studerar

$$E(\widehat{\theta}_{(\cdot)}) = E(\frac{1}{n}\sum_{i=1}^{n}\widehat{\theta}_{(i)}) = \frac{1}{n}\sum_{i=1}^{n}E(\widehat{\theta}_{(i)}) =$$

$$= \theta + \frac{a_1(F)}{n-1} + \frac{a_2(F)}{(n-1)^2} + \cdots$$

som ger

$$E\left((n-1)(\widehat{\theta}_{(\cdot)}-\widehat{\theta})\right)=\frac{a_1(F)}{n}+O(\frac{1}{n^2}).$$

får vi alltså

$$\widehat{bias}_{jack}=(n-1)(\widehat{\theta}_{(\cdot)}-\widehat{\theta})\approx\frac{a_1(F)}{n}\approx \text{bias}_F(\widehat{\theta},\theta)$$

Detta kan liknas vid vad som i Numeriska metoder kallas Aitken-extrapolation. Man har fått ner bias till storleksordningen $1/n^2$ i stället för $1/n$ som den var från början.

Exempel 9.1 *Tillämpning på LSAT/GPA:*
Om man skattar bias med jackknife för LSAT/GPA-data i exempel 5.5 får man bias-skattningen -0.064, dvs den ursprungliga skattningen av korrelationskoefficienten ρ 0.7764 kan bias-korrigeras till 0.7828.

Med bootstrap får man ett medelvärde av bootstrapskattningarna (10000 st) till $\widehat{\theta}^*(\cdot)=0.7680$ och detta ger bias-skattningen

$$\widehat{\theta}^*(\cdot)-\widehat{\theta}=0.7680-0.7764=-0.0084$$

och den bias-korrigerade skattningen blir alltså 0.7764-(-0.0084)=0.7848

Notera att jackknife- och bootstrap-skattningarna var mindre än $\widehat{\theta}=0.7764$ och att vi därför tror att $\widehat{\theta}$ var mindre än θ och korrigeringen alltså innebar en ökning. □

9.4 Bias-justering av varians-skattningen

Sats 9.1 Om man bias-korrigerar plug-in-skattningen av σ^2 dvs

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

med hjälp av $\widehat{bias}_{jack}$ så får man den "väntevärdeskorrigerade" skattningen

$$s^2=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2$$

Bevis:

Vi har

$$\widehat{\theta}=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$$

och vidare

$$\widehat{\theta}_{(i)}=\frac{1}{n-1}\sum_{j\neq i}^{n}\left(x_j-\frac{1}{n-1}\sum_{k\neq i}x_k\right)^2$$

som ger

$$(n-1)\widehat{\theta}_{(i)} = \sum_{j\neq i}^{n} x_j^2 - \frac{1}{n-1}\sum_{j\neq i}^{n} x_j \sum_{k\neq i}^{n} x_k =$$

$$= \sum_{j=1}^{n} x_j^2 - x_i^2 - \frac{1}{n-1}\left(\sum_{j=1}^{n} x_j - x_i\right)\left(\sum_{k=1}^{n} x_k - x_i\right) =$$

$$= \sum_{j=1}^{n} x_j^2 - x_i^2 - \frac{1}{n-1}\left(\sum_{j=1}^{n} x_j\right)^2 + \frac{2}{n-1}x_i\sum_{j=1}^{n} x_j - \frac{1}{n-1}x_i^2$$

Detta ger

$$(n-1)\sum_{i=1}^{n}\widehat{\theta}_{(i)} =$$

$$= n\sum_{j=1}^{n} x_j^2 - \sum_{i=1}^{n} x_i^2 - \frac{n}{n-1}\left(\sum_{j=1}^{n} x_j\right)^2 + \frac{2}{n-1}\left(\sum_{j=1}^{n} x_j\right)^2 - \frac{1}{n-1}\sum_{i=1}^{n} x_i^2 =$$

$$= \frac{n(n-2)}{n-1}\sum_{i=1}^{n} x_i^2 - \frac{n-2}{n-1}\left(\sum_{i=1}^{n} x_i\right)^2 = \frac{n(n-2)}{n-1}\left(\sum_{i=1}^{n}(x_i-\bar{x})^2\right)$$

som alltså ger

$$(n-1)\widehat{\theta}_{(\cdot)} = \frac{n-2}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2.$$

Eftersom

$$(n-1)\widehat{\theta} = \frac{n-1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$$

ger detta

$$\widehat{bias}_{jack} = (n-1)(\widehat{\theta}_{(\cdot)} - \widehat{\theta}) =$$

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

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

Detta ger den biaskorrigerade skattningen

$$\bar{\theta} = \widehat{\theta} - \widehat{bias}_{jack} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2 + \frac{1}{n(n-1)}\sum_{i=1}^{n}(x_i-\bar{x})^2 =$$

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

vilket är precis den "väntevärdeskorrigerade" σ^2-skattningen. □

Man kan faktiskt säga att definitionen av $\widehat{bias}_{jack} = (n-1)(\widehat{\theta}_{(\cdot)} - \widehat{\theta})$ gjorts så att man skulle få detta resultat.

9.5 Jackknife-skattning av medelfel

Definition 9.2 *Jackknife-skattning av medelfel*
Jackknife-skattningen av medelfelet är

$$\widehat{se}_{jack} = \sqrt{\frac{n-1}{n}\sum_{i=1}^{n}\left(\widehat{\theta}_{(i)} - \widehat{\theta}_{(\cdot)}\right)^2}$$

där alltså för $i = 1, 2, \cdots, n$

$$\widehat{\theta}_{(i)} = \widehat{\theta}(\boldsymbol{x}_{(i)}) = \widehat{\theta}(x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$$

dvs skattningen med data nr i bortplockat.

Vidare är $\widehat{\theta}_{(\cdot)} = \sum_{i=1}^{n}\widehat{\theta}_{(i)}/n$, dvs dessas aritmetiska medelvärde. □

Man kan notera att jämfört med bootstrap-skattningen av medelfelet har vi här en faktor $(n-1)/n \approx 1$ i stället för $1/(B-1)$ och detta beror på att vi i Jackknife "stör" vårt ursprungliga stickprov väsentligt mindre än vad vi gör med bootstrap-metodiken.

Att man valt just faktorn $(n-1)/n$ och inte 1 beror på att man vill att det skall stämma precis för fallet $\widehat{\theta} = \bar{x}$. I detta fall ger ovanstående uttryck (visas nedan)

$$\widehat{se}_{jack} = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{s}{\sqrt{n}}$$

dvs för $\widehat{\theta} = \bar{x}$ ger $\widehat{se}_{jack}$ "rätt" medelfel.

En av fördelarna med jackknife jämfört med bootstrap är att man bara behöver räkna ut skattningen för n fiktiva datauppsättningar i stället för B st ($B \to \infty$) som i bootstrap. Nackdelen är att man egentligen bara får en medelfelsskattning för $\widehat{\theta}$, medan bootstrap ger information om hela fördelningen för $\widehat{\theta}$. Man tvingas därför med jackknife att anta approximativ normalfördelning om man vill konstruera ett konfidensintervall för θ, t ex $\widehat{\theta} \pm 1.96\widehat{se}_{jack}$.

En stor nackdel med jackknife jämfört med bootstrap är att den kräver "snällare" $\widehat{\theta}$ – t ex fungerar inte jackknife för medianen. Vi kommer att se att jackknife i princip svarar mot bootstrap av en "lineariserad" approximation av $\widehat{\theta}$. Om alltså $\widehat{\theta}$ ej är approximativt linjär fungerar jackknife dåligt.

9.6 Pseudo-värden

Ett alternativt sätt att se Jackknife är att införa pseudo-värdena

$$\widetilde{\theta}_i = n\widehat{\theta} - (n-1)\widehat{\theta}_{(i)}, \quad i = 1, 2, \cdots, n$$

I specialfallet $\widehat{\theta} = \bar{x}$ är $\widetilde{\theta}_i = x_i$ och alltså är

$$\widetilde{\theta}_{(\cdot)} = \sum_{i=1}^{n} \widetilde{\theta}_i / n = \sum_{i=1}^{n} x_i / n = \bar{x}.$$

Detta inses ur kalkylen

$$\widetilde{\theta}_i = n\bar{x} - (n-1)\widehat{\theta}_{(i)} = \sum_{j=1}^{n} x_j - (n-1)\frac{1}{n-1}\sum_{j \neq i} x_j = x_i$$

Man kan för en allmän skattning $\widehat{\theta}$ notera att man kan uttrycka $\widehat{se}_{jack}$ med hjälp av pseudovärdena eftersom

$$\widehat{\theta}_{(i)} = \frac{n}{n-1}\widehat{\theta} - \frac{\widetilde{\theta}_i}{n-1}, \quad \text{som ger} \quad \widehat{\theta}_{(\cdot)} = \frac{n}{n-1}\widehat{\theta} - \frac{\widetilde{\theta}_{(\cdot)}}{n-1}$$

och alltså

$$\widehat{se}^2_{jack} = \frac{n-1}{n}\sum_{i=1}^{n}(\widehat{\theta}_{(i)} - \widehat{\theta}_{(\cdot)})^2 =$$

$$= \frac{n-1}{n}\sum_{i=1}^{n}\left(\frac{n}{n-1}\widehat{\theta} - \frac{\widetilde{\theta}_i}{n-1} - \frac{n}{n-1}\widehat{\theta} + \frac{\widetilde{\theta}_{(\cdot)}}{n-1}\right)^2 =$$

$$= \frac{1}{n(n-1)}\sum_{i=1}^{n}(\widetilde{\theta}_{(\cdot)} - \widetilde{\theta}_i)^2$$

Uttrycket för $\widehat{se}_{jack}$ kan skrivas som

$$\widehat{se}_{jack} = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(\widetilde{\theta}_i - \widetilde{\theta}_{(\cdot)})^2}$$

där $\widetilde{\theta}_{(\cdot)} = \sum_{i=1}^{n}\widetilde{\theta}_i/n$ som liknar den vanliga standardavvikelseskattningen $s/\sqrt{n}$. Idén är att pseudo-variablerna ska fungera ungefär som om de var oberoende variabler. P g a att pseudo-variablerna i specialfallet $\widehat{\theta} = \bar{x}$ överensstämmer med x_i stämmer det precis i detta specialfall. Alltså gäller följande sats som säger att för $\bar{x}$ ger jackknife "rätt" medelfel.

Sats 9.2 För $\widehat{\theta}(\boldsymbol{x}) = \bar{x}$ gäller att

$$\widehat{se}_{jack} = \frac{s}{\sqrt{n}}.$$

□

Jackknife fungerar som ovan nämnts inte för t ex medianen. När man plockar bort ett enda data så flyttar ju sig medianen bara ett snäpp åt vänster eller höger, vilket gör att variabiliteten blir alldeles fel i allmänhet. Man kan visa att $\widehat{se}_{jack}$ inte ens är en konsistent skattning av medelfelet för $\widehat{\theta}$ i detta specialfall.

Exempel 9.2 *LSAT/GP-data*
Med data på LSAT/GPA från exemeplet i avsnitt 5.5 får man med bootstrap $\widehat{se}_{boot,10000} = 0.1326$ medan jackknife ger $\widehat{se}_{jack} = 0.1425$. □

9.7 Medelfel: Jackknife och bootstrap

Detta avsnitt kopplar ihop $\widehat{se}_{jack}$ och $\widehat{se}_{boot}$ och visar att $\widehat{se}_{jack}$ kan ses som resultatet av bootstrap av en linearisering av $\widehat{\theta}$.

9.7.1 Jackknife för linjära skattningar

Antag att $\widehat{\theta} = \mu + \sum_{i=1}^{n} \alpha(x_i)/n$ och att vi känner $\widehat{\theta}$ för varje jackknife-stickprov $\boldsymbol{x}_{(i)}$ dvs

$$\widehat{\theta}_{(i)} = \widehat{\theta}(\boldsymbol{x}_{(i)}), \quad , i = 1, 2, \cdots, n$$

där

$$\widehat{\theta}_{(i)} = \mu + \frac{1}{n-1} \sum_{j \neq i} \alpha(x_j).$$

Vi vill nu uttrycka $\alpha(x_i)$ i $\widehat{\theta}(\boldsymbol{x}_{(i)})$:na.
Vi har

$$(n-1)\widehat{\theta}(\boldsymbol{x}_{(i)}) = (n-1)\mu + \sum_{j \neq i}^{n} \alpha(x_j) = (n-1)\mu + \sum_{j=1}^{n} \alpha(x_j) - \alpha(x_i)$$

och vi får alltså

$$(n-1) \sum_{i=1}^{n} \widehat{\theta}(\boldsymbol{x}_{(i)}) = (n-1)n\mu + n \sum_{i=1}^{n} \alpha(x_i) - \sum_{i=1}^{n} \alpha(x_i)$$

som ger

$$\sum_{i=1}^{n} \widehat{\theta}(\boldsymbol{x}_{(i)}) = n\mu + \sum_{i=1}^{n} \alpha(x_i)$$

dvs $\sum_{i=1}^{n} \alpha(x_i) = \sum_{i=1}^{n} \widehat{\theta}(\boldsymbol{x}_{(i)}) - n\mu$. Detta ger

$$\alpha(x_i) = (n-1)\mu + \left(\sum_{j=1}^{n} \widehat{\theta}(\boldsymbol{x}_{(j)}) - n\mu \right) - (n-1)\widehat{\theta}(\boldsymbol{x}_{(i)}) =$$

$$= \sum_{j=1}^{n} \widehat{\theta}(\boldsymbol{x}_{(j)}) - (n-1)\widehat{\theta}(\boldsymbol{x}_{(i)}) - \mu$$

Sats 9.3 *Samband mellan jackknife och bootstrap för linjära skattningar*
Antag att $\widehat{\theta}$ är en linjär skattning, dvs kan skrivas som $\widehat{\theta} = \mu + \sum_{i=1}^{n} \alpha(x_i)/n$

a) Den teoretiska bootstrapskattningen av medelfelet är

$$\widehat{se}_{boot} = \sqrt{\frac{1}{n^2}\sum_{i=1}^{n}(\alpha(x_i) - \bar{\alpha})^2}$$

där $\bar{\alpha} = \sum_{i=1}^{n} \alpha(x_i)/n$.

b) Jackknifeskattningen av medelfel är

$$\widehat{se}_{jack} = \sqrt{\frac{1}{(n-1)n}\sum_{i=1}^{n}(\alpha(x_i) - \bar{\alpha})^2}$$

och de överensstämmer alltså så när som på den eviga faktorn $\sqrt{(n-1)/n}$.

Bevis:

a)

Vi har

$$\widehat{\theta}^* = \mu + \frac{1}{n}\sum_{i=1}^{n}\alpha(X_i^*)$$

och får alltså

$$\widehat{se}^2_{boot} = V_{\widehat{F}}(\widehat{\theta}^*) = V_{\widehat{F}}\left(\frac{1}{n}\sum_{i=1}^{n}\alpha(X_i^*)\right) =$$

$$= \frac{1}{n^2}V_{\widehat{F}}\left(\sum_{i=1}^{n}\alpha(X_i^*)\right) = \frac{1}{n^2}\sum_{i=1}^{n}V_{\widehat{F}}(\alpha(X_i^*)) =$$

$$= \frac{1}{n}V_{\widehat{F}}(\alpha(X_1^*)) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n}\left(\alpha(x_i) - E_{\widehat{F}}(\alpha(X_1^*))\right)^2.$$

Men

$$E_{\widehat{F}}(\alpha(X_1^*)) = \frac{1}{n}\sum_{i=1}^{n}\alpha(x_i) = \bar{\alpha}$$

som ger det önskade resultatet.

b)

Vi har

$$\widehat{\theta}_{(i)} = \mu + \frac{1}{n-1}\sum_{j\neq i}\alpha(x_i)$$

och ser att pseudovärdena blir

$$\widetilde{\theta}_i = n\widehat{\theta} - (n-1)\widehat{\theta}_{(i)} =$$

$$= n\mu + \sum_{i=1}^{n} \alpha(x_i) - (n-1)\mu - \sum_{j \neq i} \alpha(x_j) = \mu + \alpha(x_i)$$

och alltså är

$$\widetilde{\theta}_{(\cdot)} = \frac{1}{n}\sum_{i=1}^{n} \widetilde{\theta}_i = \mu + \frac{1}{n}\sum_{j=1}^{n} \alpha(x_j) = \mu + \bar{\alpha}$$

som ger

$$\widehat{se}^2_{jack} = \frac{1}{n(n-1)} \sum_{i=1}^{n} (\widetilde{\theta}_i - \widetilde{\theta}_{(\cdot)})^2 =$$

$$= \frac{1}{n(n-1)} \sum_{i=1}^{n} (\mu + \alpha(x_i) - \mu - \bar{\alpha})^2 = \frac{1}{n(n-1)} \sum_{i=1}^{n} (\alpha(x_i) - \bar{\alpha})^2.$$

Alltså skiljer bara en faktor $(n-1)/n$ mellan $\widehat{se}^2_{boot}$ och $\widehat{se}^2_{jack}$. □

Sats 9.4 *Samband för allmänna skattningar*
Antag att $\widehat{\theta}$ är en icke-linjär skattning och approximera den med den linjära skattningen

$$\widehat{\theta}_{lin} = \mu + \frac{1}{n}\sum_{i=1}^{n} \alpha(x_i)$$

som tar samma värden $\widehat{\theta}(\boldsymbol{x}_{(i)})$ som $\widehat{\theta}$ för jackknife-stickproven $\boldsymbol{x}_{(i)}$, $i = 1, 2, \cdots, n$, dvs då x_i uteslutits ur data.

Då gäller att

$$\alpha(x_i) = \sum_{j=1}^{n} \widehat{\theta}(\boldsymbol{x}_{(j)}) - (n-1)\widehat{\theta}(\boldsymbol{x}_{(i)}) - \mu$$

Bevis:
Vi har alltså

$$\widehat{\theta}_{(i)} = \mu + \frac{1}{n-1}\sum_{j \neq i}^{n} \alpha(x_j)$$

och på samma sätt som i satsen ovan får vi

$$\alpha(x_i) = \sum_{j=1}^{n} \widehat{\theta}(\boldsymbol{x}_{(j)}) - (n-1)\widehat{\theta}(\boldsymbol{x}_{(i)}) - \mu$$

□

Sats 9.5 *Jackknifeskattningen av medelfelet $\widehat{se}_{jack}(\widehat{\theta})$ överensstämmer med den teoretiska bootstrap-skattningen av medelfelet för linjäriseringen $\widehat{\theta}_{lin}$ dvs $\widehat{se}_{boot}(\widehat{\theta}_{lin})$ förutom faktorn $\sqrt{(n-1)/n}$.*

Bevis:

Vi har enligt ovanstående satser att $\sum_{i=1}^{n} \widehat{\theta}(\boldsymbol{x}_{(i)}) = n\mu + \sum_{i=1}^{n} \alpha(x_i)$ dvs $\widehat{\theta}_{(\cdot)} = \mu + \bar{\alpha}$ som ger

$$\widehat{se}^2_{jack}(\widehat{\theta}) = \frac{n-1}{n} \sum_{i=1}^{n} (\widehat{\theta}(\boldsymbol{x}_{(i)}) - \widehat{\theta}_{(\cdot)})^2 =$$

$$= \frac{n-1}{n} \sum_{i=1}^{n} \left(\mu + \frac{1}{n-1} \sum_{j \neq i} \alpha(x_j) - \mu - \bar{\alpha} \right)^2 =$$

$$= \frac{n-1}{n} \sum_{i=1}^{n} \left(\frac{n}{n-1} \bar{\alpha} - \frac{\alpha(x_i)}{n-1} - \bar{\alpha} \right)^2 =$$

$$= \frac{n-1}{n} \sum_{i=1}^{n} (\bar{\alpha} - \alpha(x_i))^2 \frac{1}{(n-1)^2} =$$

$$= \frac{1}{n(n-1)} \sum_{i=1}^{n} (\alpha(x_i) - \bar{\alpha})^2 = \frac{n}{n-1} \widehat{se}^2_{boot}(\widehat{\theta}_{lin})$$

Slutsatsen är alltså att $\widehat{se}_{jack}$ ger samma resultat (så när som på faktorn $\sqrt{(n-1)/n}$) som om vi linjäriserar $\widehat{\theta}$ till $\widehat{\theta}_{lin}$ och sen bootstrappar denna linearisering. □

9.8 Modifikation av Jackknife

Man kan modifiera Jackknife genom att plocka bort mer än ett data i taget, t ex d st i taget. Detta kan göras på $\binom{n}{d}$ sätt och skattningen av medelfelet blir då

$$\sqrt{\frac{n-d}{d\binom{n}{d}} \sum_{s \in S} (\widehat{\theta}_{(s)} - \widehat{\theta}_{(\cdot)})^2}$$

där

$$\widehat{\theta}_{(\cdot)} = \frac{\sum_{s \in S} \widehat{\theta}_{(s)}}{\binom{n}{d}}.$$

där summorna går över S, dvs alla de $\binom{n}{d}$ delmängderna av storlek $n-d$ från $(x_1, x_2, \cdots, x_n)$.

Problemet är att $\binom{n}{d}$ växer dramatiskt med n och alltså måste man beräkna skattningen för ett stort antal datauppsättningar. Om man väljer d av storleksordning $\sqrt{n}$ fungerar denna modifikation av Jackknife även för medianen. Man kan undvika att beräkna skattningen för alla $\binom{n}{d}$ datauppsättningarna genom att välja ett urval av dem på måfå. Detta kommer vi dock inte alls att utveckla vidare.

Kapitel 10

Mer komplicerade datastrukturer

10.1 Inledning

I detta kapitel kommer vi att behandla mer komplicerade datastrukturer och visa hur bootstrap kan användas i dessa situationer. Bland de situationer vi kommer att behandla är
1) Flera oberoende stickprov (speciellt två oberoende stickprov)
2) Regressionsmodeller
3) Tidsserier
Det speciella med regressionsmodeller och flera oberoende stickprov är att de olika observationerna oftast har olika fördelningar, speciellt olika väntevärden. Komplikationen med tidserier är att observationerna typiskt har ett kraftigt beroende och att vi måste ta hänsyn till detta.

10.2 Allmänna sannolikhetsmodeller

Tidigare har vi betraktat ett stickprov $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ som genererat av en fördelning F, men nu ser vi det i stället som att vi har en sannolikhetsmodell P som genererat stickprovet. Tidigare har alltså sannolikhetsmodellen P specificerats av F och vi har därför inte behövt använda detta mer generella synsätt.

Sannolikhetsmodellen P kan t ex innehålla parametrar och "felfördelningar" och att skatta sannolikhetsmodellen kan t ex göras genom att dels skatta parametrarna och sedan med hjälp av dessa parameter-skattningar skatta "felfördelningen" med hjälp av residualerna. Den tidigare situationen där vi t ex skattat θ=väntevärdet av F-fördelningen med $\bar{x}$ skulle man kunna se som att sannolikhetsmodellen P består av två komponenter: dels θ och dels en "residualfördelning", dvs fördelningen för $X - \theta$.

En variant av bootstrap är att generera nya "störningar" (residualer) med hjälp av empiriska fördelningen för residualerna i modellen med de skattade

parametrarna. Detta kommer att illustreras med regressionsmodeller och med tidsserier. Vi börjar dock med en ännu enklare situation nämligen två oberoende stickprov.

10.3 Två oberoende stickprov

I situationen med två oberoende stickprov $\boldsymbol{z} = (z_1, z_2, \cdots, z_m)$ respektive $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$ så antar vi att dessa genereras av fördelningar F respektive G. Sannolikhetsmodellen kan då formuleras som $P = (F, G)$ och stickprovet $\boldsymbol{x} = (\boldsymbol{z}, \boldsymbol{y})$ kan betraktas som en $m + n$-dimensionell vektor där de första m koordinaterna genereras av F och de sista n med G. Vi skattar sannolikhetsmodellen $P = (F, G)$ med $\widehat{P} = (\widehat{F}, \widehat{G})$ där $\widehat{F}$ och $\widehat{G}$ är de empiriska fördelningarna för $\boldsymbol{z}$ respektive $\boldsymbol{y}$. Vårt bootstrap-stickprov $\boldsymbol{x}^*$ genereras på uppenbart sätt av $\widehat{P}$, dvs vi har $\boldsymbol{x}^* = (\boldsymbol{z}^*, \boldsymbol{y}^*)$ där $\boldsymbol{z}^*$ genereras av $\widehat{F}$ och $\boldsymbol{y}^*$ av $\widehat{G}$. Detta är ett invecklat sätt att säga att vi genererar bootstrap-stickprovet genom att lotta fram ett $\boldsymbol{z}$-stickprov genom dragning med återläggning från $\boldsymbol{z}$-värdena och ett $\boldsymbol{y}$-stickprov på samma sätt från $\boldsymbol{y}$-värdena.

Om nu parametern $\theta = T(F, G)$ så är plug-in-skattningen av denna $\widehat{\theta} = T(\widehat{F}, \widehat{G})$ där $\widehat{F}$ och $\widehat{G}$ är empiriska fördelningarna för $\boldsymbol{z}$ respektive $\boldsymbol{y}$. Låt Matlabfunktionen `estimate` beräkna skattningen av θ, dvs att

```
thetaest=estimate(z,y)
```

ger skattningen i variabeln `thetaest`.

För att generera bootstrap-stickprov av $\boldsymbol{z}$ respektive $\boldsymbol{y}$ kan man använda sig av följande "knep" där man använder sig av en "tom" skattningsfunktion:

```
boot_z=bootstrp(B,'',z);
boot_y=bootstrp(B,'',y);
boot=[];
for i=1:B,
        boot(i)=estimate(boot_z(i,:),boot_y(i,:));
end;
```

`boot` kommer då att innehålla skattningarna för vart och ett av de B bootstrap-stickproven. Om man är intresserad av systematiskt fel respektive medelfel för $\widehat{\theta}$ erhålls de som

```
bias_est=mean(boot)-estimate(z,y)
se_est=std(boot)
```

Om man har en skattningsfunktion `estimate` som tillåter matris-argument kan man slippa den sista "for"-loopen vilket är att föredra eftersom Matlab är förhållandevis långsam vad gäller "for"-loopar.

Exempel 10.1 *Jämförelse av fördelningar*
Om vi t ex är intresserade av att skatta $\theta = m_z - m_y$ dvs skillnaden i väntevärden mellan F- och G-fördelningarna så skattar vi lämpligtvis detta med $\widehat{\theta} = \bar{z} - \bar{y}$. Skattningen av medelfelet av denna skattning blir då

$$se_{\widehat{P}}(\widehat{\theta}^*) = \sqrt{V_{\widehat{P}}(\bar{z}^* - \bar{y}^*)}.$$

Just för medelvärdet kan vi ju beräkna detta analytiskt, men detta är lite otypiskt. I allmänhet kommer vi inte åt denna "ideala" medelfelsskattning, med vi kan ju alltid skatta den genom att generera ett stort antal bootstrap-stickprov $\boldsymbol{x}^{*b}, \quad b = 1, 2, \cdots, B$ för ett stort B (t ex $B = 1000$). Vi skattar $se_{\widehat{P}}(\widehat{\theta}^*)$ med

$$\widehat{se}_B = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\widehat{\theta}^{*b} - \widehat{\theta}^*(\cdot)\right)^2}$$

där som vanligt $\widehat{\theta}^*(\cdot) = \sum_{b=1}^{B}\widehat{\theta}^{*b}/B$. I ovanstående exempel tar vi alltså $\widehat{\theta}^{*b} = \bar{z}^{*b} - \bar{y}^{*b}$. □

Man kan notera att det på samma sätt skulle gå att skatta t ex skillnaden θ (eller kvoten τ) av medianerna i F- och G-fördelningarna genom att konstruera en lämplig skattning, t ex plug-in-skattningarna

$$\widehat{\theta} = \text{median}(z_1, \cdots, z_m) - \text{median}(y_1, \cdots, y_n) \text{ resp. } \widehat{\tau} = \frac{\text{median}(z_1, \cdots, z_m)}{\text{median}(y_1, \cdots, y_n)}$$

och sedan utföra bootstrap enligt ovanstående. I denna situation skulle det vara besvärligt att med analytiska metoder kunna få fram t ex medelfelsskattning, medan det med bootstrap är mycket enkelt. Vi genererar alltså helt enkelt B st (t ex $B = 1000$) stickprov $\boldsymbol{x}^{*b} = (\boldsymbol{z}^{*b}, \boldsymbol{y}^{*b})$, $b = 1, 2, \cdots, B$ vardera av storlek $m + n$ och beräknar den aktuella skattningen för vart och ett av dessa stickprov. För att få en medelfelsskattning av t ex $\widehat{\tau}$ beräknar vi

$$\widehat{se}_B = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\widehat{\tau}^{*b} - \widehat{\tau}^*(\cdot)\right)^2}$$

där alltså

$$\widehat{\tau}^{*b} = \frac{\text{median}(\boldsymbol{z}^{*b})}{\text{median}(\boldsymbol{y}^{*b})}$$

och

$$\widehat{\tau}^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B}\widehat{\tau}^{*b}$$

eller enklare uttryckt genom att beräkna standardavvikelsen i histogramet över de B skattningarna $\widehat{\tau}^{*1}, \widehat{\tau}^{*2}, \cdots \widehat{\tau}^{*B}$.

Med $\widehat{\theta}$ och $\widehat{\tau}$ enligt ovanstående kan vi "separera" bootstrap i de två stickproven och kombinera resultatet. Alltså kan man använda följande Matlab-kod för att erhålla bootstrap-fördelningarna för $\widehat{\theta}$ (i vektorn `boot_tet`) och för $\widehat{\tau}$ (i vektorn `boot_tau`):

```
boot_z=bootstrp(B,'median',z);
boot_y=bootstrp(B,'median',y);
boot_tet=boot_z-boot_y;
boot_tau=boot_z./boot_y;
```

Notera att i beräkningen av `boot_tau` utförs en koordinatvis division av vektornerna `boot_z` och `boot_y`. Vill vi sedan ha t ex medelfelet för $\widehat{\tau}$ erhålls det med `std(boot_tau)`.

Vi ser alltså att vi behöver kunna få en skattning $\widehat{P}$ av sannolikhetsmodellens P, men ofta går det lika lätt att göra som i ovanstående situation med två oberoende stickprov.

10.4 Regressionsmodeller

10.4.1 Inledning

I regressionsmodeller finns två distinkta sätt att generera bootstrap-stickprovet. Den ena skulle kunna kallas "Bootstrap av punkter" och den andra "Bootstrap av residualer".

Båda beskrivs nedan, men först lite allmänt om regressionsmodeller.

10.4.2 Om regressionsmodeller

Vi har en regressionsmodell där observationerna består av $(y_i, \boldsymbol{c}_i),\ i = 1, 2, \cdots, n$ där vi ser y_i som utfall av Y_i där

$$Y_i = \boldsymbol{c}_i\boldsymbol{\beta} + \varepsilon_i = \sum_{j=1}^{p} c_{ij}\beta_j + \varepsilon_i, \quad i = 1, 2, \cdots, n$$

och där vi betraktar $\boldsymbol{c}_i = (c_{i1}, c_{i2}, \cdots, c_{ip})$ som känd. $\boldsymbol{c}_i$ är att se som en kovariat som innehåller känd information om observation nr i. Vektorn $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_p)^T$ är en vektor av parametrar och det är dessa som skall skattas ur data.

Vi antar att feltermerna $\varepsilon_i, \quad i = 1, 2, \cdots, n$ är oberoende och kommer från en och samma fördelning F med väntevärde 0. I elementära böcker antas ju nästan alltid dessutom att ε_i är $N(0, \sigma^2)$.

Exempel 10.2 *Enkel linjär regression*
Som specialfall har vi det bekanta "Enkel linjär regression" $Y_i = a + bx_i + \varepsilon_i$ där med ovanstående formulering $\boldsymbol{c}_i = (1, x_i)$ och $\boldsymbol{\beta} = (a, b)^T$. Oftast brukar man formulera om denna modell genom att avcentrera med $\bar{x} = \sum_{i=1}^{n} x_i/n$ och får då $Y_i = \alpha + \beta(x_i - \bar{x}) + \varepsilon_i$ och har då $\boldsymbol{c}_i = (1, x_i - \bar{x})$ och $\boldsymbol{\beta} = (\alpha, \beta)^T$. □

Exempel 10.3 *Multipel linjär regression*
En vanligt förekommande modell är "Multipel linjär regression" där vi har för $i = 1, 2, \cdots, n$

$$Y_i = a + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_k x_{ik} + \varepsilon_i.$$

Detta svarar mot $\boldsymbol{c}_i = (1, x_{i1}, x_{i2}, \cdots, x_{ik})$ och $\boldsymbol{\beta} = (a, b_1, b_2, \cdots, b_k)^T$. $\boldsymbol{c}_i$ är alltså $p = (k+1)$-dimensionell. Det typiska är att $p < n$ ja kanske t o m att $p << n$. Notera att x_{ij}:na kan vara hur beroende av varandra som helst. Ett vanligt fall är polynom-regression där

$$Y_i = a + b_1 t_i + b_2 t_i^2 + \cdots + b_k t_i^k + \varepsilon_i$$

d v s man försöker anpassa ett k:te gradspolynom till data. □

Vi vill skatta parametervektorn $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_p)^T$ från observationerna

$$(y_1, \boldsymbol{c}_1), (y_2, \boldsymbol{c}_2), \cdots, (y_n, \boldsymbol{c}_n)$$

och för detta ändamål betraktar vi (om vi vill använda oss av Minsta kvadratmetoden)

$$Q(\boldsymbol{b}) = \sum_{i=1}^{n} (y_i - \boldsymbol{c}_i \boldsymbol{b})^2$$

och låter $\boldsymbol{\beta}$-skattningen $\widehat{\boldsymbol{\beta}}$ vara det $\boldsymbol{b}$-värde som minimerar $Q(\boldsymbol{b})$. Detta kan som bekant formuleras med linjär algebra om vi låter $\boldsymbol{C}$ vara den nxp-matris som har i-te raden lika med $\boldsymbol{c}_i$ (dvs $\boldsymbol{C}$ är design-matrisen). Vidare bildar vi vektorerna $\boldsymbol{Y} = (Y_1, Y_2, \cdots, Y_n)^T$ och $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^T$.

Man kan då skriva

$$Q(\boldsymbol{b}) = \sum_{i=1}^{n} (y_i - \boldsymbol{c}_i \boldsymbol{b})^2 = ||\boldsymbol{y} - \boldsymbol{C}\boldsymbol{b}||^2$$

där $||\cdot||$ är Euklidiska avståndet i R^n. Den med goda kunskaper i linjär algebra inser att den p-dimensionella vektor $\boldsymbol{b}$ som minimerar $Q(\boldsymbol{b})$ svarar mot en ortogonal projektion av den n-dimensionella vektorn $\boldsymbol{y}$ på det p-dimensionella underrum i R^n som spänns av kolonnerna i matrisen $\boldsymbol{C}$.

Regressionsmodellen kan skrivas

$$\boldsymbol{Y} = \boldsymbol{C}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

och skattningen $\widehat{\boldsymbol{\beta}}$ satisfierar ekvationen

$$\boldsymbol{C}^T \boldsymbol{C} \widehat{\boldsymbol{\beta}} = \boldsymbol{C}^T \boldsymbol{y}.$$

Om vi har en fullrangs-modell (dvs att $\boldsymbol{C}^T\boldsymbol{C}$ är inverterbar) får vi skattningen

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{C}^T \boldsymbol{C})^{-1} \boldsymbol{C}^T \boldsymbol{y}.$$

Man kan lätt visa att $\widehat{\boldsymbol{\beta}}$ får väntevärdes-vektor $\boldsymbol{\beta}$ och kovariansmatris $\sigma^2 (\boldsymbol{C}^T \boldsymbol{C})^{-1}$. Är slumpfelen dessutom normalfördelade blir $\boldsymbol{\beta}$ p-dimensionellt normalfördelad. Speciellt enkelt blir det ju om matrisen $\boldsymbol{C}^T\boldsymbol{C}$ är en diagonalmatris. Komponenterna i vektorn $\widehat{\boldsymbol{\beta}}$ blir då okorrelerade.

Exempel 10.4 *Enkel linjär regression*
I "Enkel linjär regressions"-fallet får man om man avcentrerar

$$\boldsymbol{C} = \begin{pmatrix} 1 & x_1 - \bar{x} \\ 1 & x_2 - \bar{x} \\ . & . \\ 1 & x_n - \bar{x} \end{pmatrix}$$

och vi får matrisen

$$\boldsymbol{C}^T\boldsymbol{C} = \begin{pmatrix} n & 0 \\ 0 & \sum_{i=1}^n (x_i - \bar{x})^2 \end{pmatrix}$$

och

$$(\boldsymbol{C}^T\boldsymbol{C})^{-1} = \begin{pmatrix} 1/n & 0 \\ 0 & \dfrac{1}{\sum_{i=1}^n (x_i - \bar{x})^2} \end{pmatrix}$$

vilket ger de sedvanliga skattningarna $\widehat{\alpha} = \bar{y}$ och

$$\widehat{\beta} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}.$$

Ett annat sätt att se detta är ju som nämnts att bilda

$$Q(\alpha, \beta) = \sum_{i=1}^n (y_i - \alpha - \beta(x_i - \bar{x}))^2$$

och välja $\widehat{\alpha}$ och $\widehat{\beta}$ som de (unika) värden som minimerar $Q(\alpha, \beta)$.

Vi kunde också ha använt ett annat mått på avvikelserna, t ex för enkel linjär regression

$$Q(\alpha, \beta) = \sum_{i=1}^n |y_i - \alpha - \beta(x_i - \bar{x})|$$

vilket innebär att vi utför median-regression. Detta gör skattningarna mindre känsliga för enstaka avvikande observationer. Dock kan problemet att minimera Q vara lite besvärligare än med kvadratisk förlustfunktion, dvs traditionell Minsta kvadrat-metod.

Ett annat val är

$$Q(\alpha, \beta) = \sum_{i=1}^n w_i (y_i - \alpha - \beta(x_i - \bar{x}))^2$$

där w_i är (positiva) vikter, som t ex kan väljas till $w_i = 1/V(\varepsilon_i)$ om vi har data med olika spridning, s k viktad minsta-kvadrat-skattning. □

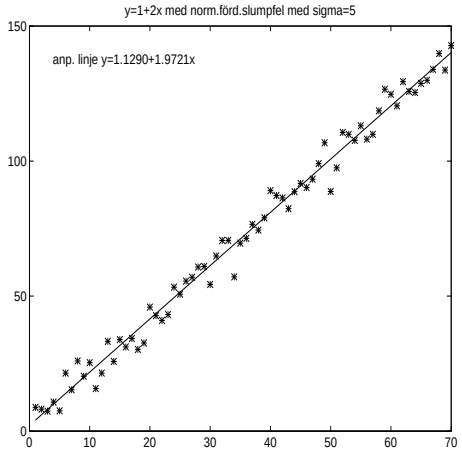
Naturligtvis kan median-regression eller en viktad minsta-kvadrat-metod även användas för multipel linjär regression.

10.4.3 Bootstrap av punkter

Vi skulle kunna göra bootstrap av punkter i samma anda som i kapitel 4. Detta innebär att vi ser de n observationerna $(y_i, \boldsymbol{c}_i)$ som kommande från en $p+1$-dimensionell fördelning och vi låter den empiriska fördelningen för $(y_i, \boldsymbol{c}_i)$, $i = 1, 2, \cdots, n$ vara bootstrap-fördelningen. Notera att detta innebär att vi anser att både "respons"-variabeln y och kovariaterna $\boldsymbol{c}_i$ är slumpmässiga.

Vi låter alltså vårt bootstrap-stickprov genereras genom att välja n st punkter bland $(y_i, \boldsymbol{c}_i)$, $\quad i = 1, 2, \cdots, n$ med återläggning. Detta betyder alltså att vi låter $(Y_i^*, \boldsymbol{c}_i^*)$ uppfylla $P((Y_i^*, \boldsymbol{c}_i^*) = (y_j, \boldsymbol{c}_j)) = 1/n$ för $j = 1, 2, \cdots, n$. Vi erhåller alltså den slumpmässiga design-matrisen $\boldsymbol{C}^*$. Vi skattar sen $\boldsymbol{\beta}$ genom att lösa normalekvationerna för dessa bootstrap-data genom $\widehat{\boldsymbol{\beta}^*} = (\boldsymbol{C}^{*T}\boldsymbol{C}^*)^{-1}\boldsymbol{C}^{*T}\boldsymbol{Y}^*$. Genom upprepad simulering erhålls hela fördelningen för $\widehat{\boldsymbol{\beta}}^*$ som vi alltså tror liknar fördelningen för $\widehat{\boldsymbol{\beta}}$. I och för sig gör vi bara B simuleringar men genom att ta B tillräckligt stort kan vi få fram fördelningen för $\widehat{\boldsymbol{\beta}}^*$ precis hur bra som helst. Vi kan därigenom skatta intressanta kvantiteter för $\widehat{\boldsymbol{\beta}}$ t ex $D(\widehat{\beta}_1)$ eller $C(\widehat{\beta}_1, \widehat{\beta}_2)$ genom att beräkna motsvarande kvantiteter för $\widehat{\boldsymbol{\beta}}^*$. Notera att komponenterna i $\widehat{\boldsymbol{\beta}}$ oftast är beroende, men vi har alltså samma beroende i $\widehat{\boldsymbol{\beta}}^*$ enligt bootstrap-hypotesen.

Exempel 10.5 *Enkel linjär regression*
Låt $Y_i = 1 + 2x_i + \varepsilon_i, \quad i = 1, 2, \cdots, 70$ med ε_i normalfördelade med $E(\varepsilon_i) = 0$ och $V(\varepsilon_i) = \sigma^2 = 25$ och $x_i = i, \quad i = 1, 2, \cdots, 70$. Alltså har vi regressionssambandet $y = a + bx$ med $a = 1, b = 2$. Se figur 10.1.

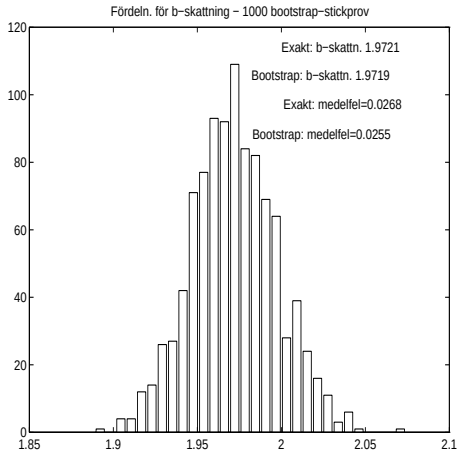


Figur 10.1: $y = 1 + 2x$ med $N(0, 5)$-fördelade slumpfel. Anpassad linje $y = 1.290 + 1.9721x$.

$\boldsymbol{\beta} = (a, b)^T$ har skattats med $\widehat{\boldsymbol{\beta}} = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{y}$ där

$$\boldsymbol{C} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ . & . \\ . & . \\ 1 & 69 \\ 1 & 70 \end{pmatrix}$$

Här används metoden "bootstrap av punkter" för att få bootstrap-fördelningen för $\widehat{\boldsymbol{\beta}}$ och då speciellt $\widehat{b}$. I figur 10.2 visas bootstrap-fördelningen för $\widehat{b}$.



Figur 10.2: Fördelning för b-skattning för 1000 bootstrap-stickprov. Exakt skattning=1.9721. Medelvärde av bootstrap-skattningarna=1.9719. Exakt medelfel=0.0268. Bootstrap-medelfel=0.0255. Sann standardavvikelse=0.0296

Medelfelsskattningen är nästan identisk för sedvanlig normalfördelnings-metodik och då man använt sig av bootstrap. $\widehat{b}$ är ju om man utnyttjar normalfördelningsantagandet

$$N\left(b, \frac{\sigma^2}{\sum_{i=1}^{70}(x_i - \bar{x})^2}\right) = N(b, 0.0296^2).$$

Fördelningen i figur 10.2 är alltså principiellt korrekt vad gäller fördelningsformen även om den är skiftad något från det "sanna" värdet 2 till $\widehat{b}$=1.9721. □

10.4.4 Bootstrap av residualer

Vi ser nu vår sannolikhetsmodell P som $P = (\boldsymbol{\beta}, F)$ där F är fördelningen för ε_i:na. Vi skattar P med $\widehat{P} = (\widehat{\boldsymbol{\beta}}, \widehat{F})$. Vi har redan sett hur man skattar $\widehat{\boldsymbol{\beta}}$, men hur får vi $\widehat{F}$? Vi kan använda $\widehat{\boldsymbol{\beta}}$ för att skatta residualerna $\widehat{\varepsilon_i} = y_i - \boldsymbol{c}_i\widehat{\boldsymbol{\beta}}, \quad i =$

$1, 2, \cdots, n$. Vi låter sen $\widehat{F}$ vara den empiriska fördelningen för dessa residualer $\widehat{\varepsilon}_i, \quad i = 1, 2, \cdots, n$.

Detta betyder alltså att vi först skattar $\widehat{\boldsymbol{\beta}}$, och sen väljer slumpfel enligt $\widehat{F}$, dvs lottar om nya slumpfel $\widehat{\varepsilon}^*$ med hjälp av dragning med återläggning från de observerade residualerna för den anpassade regressionslinjen. Vi får då med bootstrap de nya observationerna

$$y_i^* = \boldsymbol{c}_i\widehat{\boldsymbol{\beta}} + \widehat{\varepsilon}_i^*, \quad i = 1, 2, \cdots, n.$$

där $P(\widehat{\varepsilon}_i^* = \widehat{\varepsilon}_j) = 1/n$ för $j = 1, 2, \cdots, n$. Ur dessa observationer $\boldsymbol{y}^* = (y_1^*, y_2^*, \cdots, y_n^*)^T$ erhålls sen bootstrapskattningen $\widehat{\boldsymbol{\beta}}^*$ genom att lösa ekvationen

$$\widehat{\boldsymbol{\beta}}^* = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{y}^*$$

Detta förfarande upprepas sen B gånger och vi erhåller skattningar $\widehat{\beta}^{*1}, \widehat{\beta}^{*2}, \cdots, \widehat{\beta}^{*B}$ och fördelningen av dessa approximerar fördelningen för $\widehat{\beta}^*$ hur bra som helst om B "stort".

Just i denna situation är det lätt att få ett slutet uttryck på $V_{\widehat{F}}(\widehat{\boldsymbol{\beta}}^*)$. Vi har nämligen

$$V_{\widehat{F}}(\widehat{\boldsymbol{\beta}}^*) = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T V_{\widehat{F}}(\boldsymbol{Y}^*)\boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{C})^{-1}.$$

Notera att $V_{\widehat{F}}(\boldsymbol{Y}^*) = V_{\widehat{F}}(\boldsymbol{\varepsilon}^*)$ och komponenterna i $\boldsymbol{\varepsilon}^*$ är oberoende eftersom de valts genom dragning med återläggning. Alltså gäller att $V_{\widehat{F}}(\boldsymbol{\varepsilon}^*) = \widehat{\sigma}^2\boldsymbol{I}$ där $\boldsymbol{I}$ är en $n \times n$ identitetsmatris och

$$\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{a} - \widehat{b}x_i)^2.$$

Alltså ser vi att

$$V_{\widehat{F}}(\widehat{\boldsymbol{\beta}}^*) = \widehat{\sigma}^2(\boldsymbol{C}^T\boldsymbol{C})^{-1}.$$

Så när som på att man vanligtvis skattar σ^2 genom att dela med $n-p = n-2$ i stället för med n överensstämmer detta med skattningen i den vanliga linjära modellen.

10.4.5 "Bootstrap av punkter" i Matlab

Att utföra "bootstrap av punkter" i regressionsmodeller i Matlab innebär alltså att man ser hela vektorn $(y_i, \boldsymbol{c}_i)$ som sin i:te observation och det är alltså bland dessa man måste utföra bootstrap. Om man t ex använder sig av Matlab-funktionen `regress` från `Stats`-modulen som (bl a) skattar parametrarna i regressionsmodellen så måste man tänka på att även bootstrap ska utföras på designmatrisen $\boldsymbol{C}$.

Kommandot `help regress` ger utskriften

```
REGRESS Multiple linear regression using least squares.
   b = REGRESS(y,X) returns the vector of regression
   coefficients, B.
   Given the linear model: y = Xb,
   (X is an nxp matrix, y is the nx1 vector of observations.)
   [B,BINT,R,RINT,STATS] = REGRESS(y,X,alpha) uses the
   input ALPHA to calculate 100(1 - ALPHA) confidence
   intervals for B and the residual vector, R, in BINT
   and RINT respectively. The vector STATS contains
   the R-square statistic along with the F and p
   values for the regression.
```

Om man alltså vill göra bootstrap av punkter och har data lagrade i vektorn `y` och designmatrisen lagrad i `X` utförs bootstrap av

```
boot=bootstrp(B,'regress',y,X);
```

Matrisen `boot` innehåller då B rader och skattningarna av parametrarna finns i de olika kolumnerna.

Funktionen `regress` är dock rätt långsam och det kan därför löna sig att själv skriva en `m`-fil som beräknar skattningen, t ex

```
function b=reg(y,X)
b=inv(X'*X)*X'*y;
```

Vi får bootstrap-fördelningen för $\widehat{\boldsymbol{\beta}}$ i vektorn `boot` genom Matlab-koden

```
boot=bootstrp(B,'reg',y,X)
```

Rutinen `bootstrp` skapar då bootstrap-stickproven genom att på rätt sätt välja rader ur både `y` och `X` genom dragning med återläggning.

10.4.6 "Bootstrap av residualer" i Matlab

Om man å andra sidan vill utföra "bootstrap av residualer" måste man alltså först skatta parametrarna. Därefter skattas residualerna. Man genererar sedan nya observationer genom att för de ursprungliga kovariaterna störa den skattade modellen med bootstrappade residualer. Därefter skattas parametrarna i regressionsmodellen för dessa nya bootstrap-stickprov.

I Matlab kan koden se ut som följande där vi har den beroende variabeln lagrad i kolonnvektorn `y` och kovariaterna i matrisen X, där i:te raden innehåller i:te observationens kovariater. Man skulle kunna använda sig av Matlab-funktionen `regress` men i stället beräknar vi den ovan definierade funktionen `reg` som ger skattningen $\widehat{\theta}$ genom `b=reg(y,X)`:

```
b=reg(y,X);
residual=y-X*b;
boot_res=bootstrp(B,'',residual);
```

```
y_boot=[];
for i=1:B,
        y_boot(:,i)=X*b+boot_res(i,:)';
end ;
boot=[];
for j=1:B,
        boot=[boot ; (reg(y_boot(:,i),X)'];
end ;
```

Efter denna operation innehåller matrisen `boot` B rader och i i:te raden finns skattningarna för det i:te bootstrap-stickprovet i respektive kolonn. Man kan notera att man lite grann får passa sig vad gäller dimensioner, transponering etc så att allt passar ihop.

10.4.7 Jämförelse av metoderna

Vilken av dessa båda metoder är bäst?

Att göra "bootstrap av residualer" är mer modellberoende, dvs bygger på att vårt regressionssamband beskriver data och framför allt att residualerna är oberoende likafördelade. "Bootstrap av punkter" är mer modell-oberoende, men vi kan råka i svårigheter genom att vissa bootstrap-stickprov innehåller så få distinkta punkter att det blir omöjligt att skatta $\boldsymbol{\beta}$. Om vi gör "bootstrap av residualer" får samtliga data samma kovariater som de ursprungliga data hade, medan "bootstrap av punkter" innebär att vi faktiskt betraktar kovariaterna som slumpmässiga de också. Vilket som är det mest realistiska att anta beror på den specifika situationen. Vi har ju kanske systematiskt valt våra kovariater (x-variablerna) och det kan därför kännas stötande att se dessa som slumpmässiga.

Om man har någorlunda mycket data spelar det i praktiken ingen roll vilken av de två bootstrap-metoderna som man använder sig av. Man kan visa att metoderna ger samma resultat om antalet observationer $n \to \infty$.

10.4.8 Komplikationer

Någonting som ofta i litteraturen överhuvudtaget inte berörs angående metoden att göra bootstrap av residualer är att residualerna

1) inte är oberoende eftersom $\sum_{i=1}^{n} \widehat{\varepsilon}_i = 0$

2) residualerna inte är likafördelade.

Faktum är att residualerna i allmänhet minst för de punkter som ligger "längst ut" från centrum av x-värdena. Detta är ett lite paradoxalt faktum. Man skulle ju naivt tro att den skattade regressions-linjen var mest obestämd i utkanten och att alltså residualerna skulle vara störst där, men det är i stället så att den anpassade regressionslinjen tar mest hänsyn till punkter långt ut, och alltså försöker göra de residualerna minst på bekostnad av punkterna närmare

centrum av x-värdena. En korrekt analys av situationen borde egentligen ta hänsyn till detta.

I den traditionella regressionsmodellen $Y_i = \boldsymbol{c}_i\boldsymbol{\beta} + \varepsilon_i$ så är de skattade residualerna $\widehat{\boldsymbol{\varepsilon}} = (\widehat{\varepsilon}_1, \widehat{\varepsilon}_2, \cdots, \widehat{\varepsilon}_n)$

$$\widehat{\boldsymbol{\varepsilon}} = \boldsymbol{Y} - \boldsymbol{C}\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{I} - (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\right)\boldsymbol{Y}$$

vilket gör att de får kovariansmatrisen

$$\left(\boldsymbol{I} - (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\right) V(\boldsymbol{Y}) \left(\boldsymbol{I} - (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\right)^T = \sigma^2(\boldsymbol{I} - \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T)$$

och man borde därför ta residualerna från

$$\frac{\widehat{\varepsilon}_i}{\sqrt{(\boldsymbol{I} - \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T)_{ii}}}, \quad i = 1, 2, \cdots, n$$

eftersom denna vektor har väntevärdesvektor $\boldsymbol{0}$ och alla komponenterna har samma varians σ^2. Nämnaren (dvs $D(\widehat{\varepsilon}_i)/\sigma$) kompenserar alltså för att de olika skattade residualerna ej har samma fördelning. Dock är de fortfarande svagt beroende.

10.4.9 Median-regression

Observera att ovanstående metoder fungerar ungefär likadant även om vi inte skulle använda oss av Minsta Kvadratmetoden utan i stället t ex "Minsta Absolutbelopp-metoden". Vi tar då

$$Q(\boldsymbol{b}) = \sum_{i=1}^{m} |y_i - \boldsymbol{c}_i\boldsymbol{b}|$$

och låter $\widehat{\boldsymbol{\beta}}$ vara det $\boldsymbol{b}$-värde som minimerar $Q(\boldsymbol{b})$. Denna metod är mindre känslig för enstaka avvikande punkter. I Matlabkoden är enda modifieringen att skattningsrutinen `regress` måste bytas mot en som utför medianregressionen.

Notera dock hur svårt det skulle vara att (utan bootstrap) beräkna $V(\widehat{\beta})$ i denna situation. Med bootstrap spelar det ingen roll om vi använder oss av median-regression – medelfelsberäkningen är i princip likadan. Dock är det ju av visst intresse att skattningen skall gå att beräkna någorlunda snabbt.

10.4.10 Icke-linjär regression – olika spridningar

Vid icke-linjär regression dvs modellen

$$Y_i = g_i(\boldsymbol{\beta}, \boldsymbol{c}_i) + \varepsilon_i, \;\; i = 1, 2, \cdots, n$$

där funktionerna g_i är kända uttryck i parametern $\boldsymbol{\beta}$ och kovariaten $\boldsymbol{c}_i$ och där ε_i:n är oberoende likafördelade med $E(\varepsilon_i) = 0$ kan detta lätt utföras med

metoden "Bootstrap av residualer". Man skattar med någon rutin (t ex Minsta kvadratmetoden) $\boldsymbol{\beta}$ med $\widehat{\boldsymbol{\beta}}$. Därefter skattas residualerna med $\widehat{\epsilon}_i = y_i - g_i(\widehat{\boldsymbol{\beta}}, \boldsymbol{c}_i)$ för $i = 1, 2, \cdots, n$. Därefter genereras $Y_i^* = g_i(\widehat{\boldsymbol{\beta}}, \boldsymbol{c}_i) + \widehat{\varepsilon}_i^*$ för $i = 1, 2, \cdots, n$ där $\widehat{\varepsilon}_i^*$ dras med återläggning från $\widehat{\varepsilon}_j$, $j = 1, 2, \cdots, n$. Ur $\boldsymbol{y}^*$ skattas $\widehat{\boldsymbol{\beta}}^*$ med samma rutin som användes för att få $\widehat{\boldsymbol{\beta}}$. Genom att upprepa detta förfarande B gånger erhålls i princip fördelningen för $\widehat{\boldsymbol{\beta}}^*$.

Samma komplikation som tidigare uppstår här att de skattade residualerna ej är likafördelade och det är knepigt att på något lätt sätt åtgärda detta.

För icke-linjär regression är det ett orealistiskt antagande att störningarna ε_i skall vara likafördelade. Vill man tillåta dem att ha olika fördelningar, t ex ha olika varians, kan man inte använda empiriska fördelningen för de observerade residualerna som bootstrap-fördelning på samma sätt som i tidigare tillämpningar.

Ett förslag som framförts är följande konstruktion. Man skattar $\widehat{\boldsymbol{\beta}}$ och residualerna $\widehat{\varepsilon}_i$ för $i = 1, 2, \cdots, n$ som ovan. Man låter sen $\widehat{\varepsilon}_i^*$ vara $\pm\widehat{\varepsilon}_i$ med lika sannolikheter, dvs

$$P(\widehat{\varepsilon}_i^* = \widehat{\varepsilon}_i) = P(\widehat{\varepsilon}_i^* = -\widehat{\varepsilon}_i) = \frac{1}{2}$$

och fortsätter sedan som tidigare dvs låter $Y_i^* = g_i(\widehat{\beta}, \boldsymbol{c}_i) + \widehat{\varepsilon}_i^*$ för $i = 1, 2, \cdots, n$ och skattar $\widehat{\boldsymbol{\beta}}^*$ ur $\boldsymbol{Y}^*$.

10.5 Tidsserier

10.5.1 Allmänt om tidsserier

En tidsserie är sekvens observationer $(y_t;\ t \in T)$ som ses som utfall av $(Y_t;\ t \in T)$. Ett typiskt exempel på tidsmängden T är $\{1, 2, 3, \cdots n\}$.

Det typiska är att sannolikhetsmodellen för $(Y_t;\ t \in T)$ innebär ett mer eller mindre kraftigt beroende mellan de olika Y_t:na. För tidsserier finns ett stort antal modeller av typen $AR(p)$, $MA(q)$, $ARMA(p, q)$ etc och läsaren hänvisas till en kurs i tidserieanalys. Vi betraktar i fortsättningen bara $AR(1)$-processer och någon mån $AR(2)$-processer.

Precis som för regressionsmodeller finns för tidsserier två distinkta metoder att använda bootstrap. Den ena innebär att man utifrån en parametrisk modell för tidsserien (t ex autoregressiv process av ordning 2) skattar de ingående parametrarna och därigenom också får en skattning av slumpstörningarna (residualerna). Man genererar sedan nya "kopior" av tidsserien genom att utifrån modellen med de skattade parametrarna generera nya slumpstörningar genom dragning med återläggning från empiriska fördelningen av de skattade residualerna. Därigenom får man fiktiva upprepningar av tidsserien där man kan skatta parametrarna i modellen för tidsserien. Man inser att detta förfarande är kraftigt modellberoende eftersom de genererade "fiktiva" tidserierna genereras i en (så när som på parametrarna) helt specificerad tidsseriemodell.

Den andra metoden att generera en kopia av det ursprungliga försöket (dvs en ny tidsserie) innebär att man indelar tidsserien i överlappande tidsblock av viss storlek (t ex 5 tidssteg) och sedan drar från dessa block med återläggning och "klistrar" ihop dessa så att man erhåller en tidsserie. I denna simulerade tidsserie skattar man sedan parametern eller parametrarna. Detta förfarande upprepas sedan B gånger och ur den erhållna variabiliteten i parameterskattningen kan man skatta t ex medelfelet för parameterskattningen. Idén med att dra från dessa block är att man då bibehåller korttidsberoendet i tidsserien, men man inser att det finns en stor godtycklighet i valet av blockstorlek. Man vill gärna ha stor blockstorlek för att få med "långtidsberoendet" i tidsserien men då underskattar man variabiliteten. Om man å andra sidan väljer en för kort blockstorlek förutsätter man implicit ett snabbt avklingande beroende i tidsserien men får å andra sidan god variabilitet i de olika framlottade tidsserierna.

10.5.2 $AR(1)$-process

Vi har en observerad stationär tidsserie $(y_t,\ t = 1, 2, \cdots, n)$. Vi låter $\mu = E(Y_t)$ som alltså är lika för alla t och låter $Z_t = Y_t - \mu$, dvs vi avcentrerar så att $E(Z_t) = 0$. I och för sig är μ en okänd parameter men vi skattar den med $\bar{y}$ och bortser i fortsättningen från att denna avcentrering gjorts utan betraktar tidsserien $(Z_t,\ t = 1, 2, \cdots, n)$ där $Z_t = Y_t - \bar{y}$.

Vi antar alltså att Z_t:na utgör en första ordningens autoregressiv process, dvs att $Z_t = \beta Z_{t-1} + \varepsilon_t,\ t = 1, 2, \cdots, n$ där β är en okänd parameter (som för att det skall vara stabilt ska uppfylla $|\beta| < 1$). Vidare antar vi att "störningarna" ε_t är oberoende och kommer från en okänd fördelning F med väntevärde 0.

Man kan notera att för $t = 1$ ingår z_0 i $z_t = \beta z_{t-1} + \varepsilon_t$ där z_0 är obestämd, men om vi låter rekursionen starta från $t = 2$ försvinner detta problem.

Hur skall man då skatta β om vi tror på denna modell? Kurser i tidserieanalys anger en hel provkarta på metoder att skatta β t ex Yule-Walker-estimering, ML-metoden, Hannan-Rissanen-algoritmen, Burgs algoritm eller Minsta-kvadrat-metoden. Det karaktäristiska för bootstrap är att vi inte så mycket funderar över hur parametrar skattas utan mer ser skattningen som given.

Minsta kvadrat-metodik skulle t ex kunna innebära att vi bildar

$$Q(b) = \sum_{t=2}^{n} (z_t - b z_{t-1})^2$$

och låter $\widehat{\beta}$ = värdet på b som minimerar $Q(b)$.

Med Yule-Walker-metodik kan man skatta β med $\widehat{\beta} = \widehat{\gamma(1)}/\widehat{\gamma(0)}$ där

$$\widehat{\gamma(0)} = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2 \text{ och } \widehat{\gamma(1)} = \frac{1}{n}\sum_{i=1}^{n-1}(z_i - \bar{z})(z_{i+1} - \bar{z})$$

Oavsett vilken metod man använder sig av för att få fram skattningen av β är vi intresserade av t ex eventuellt systematiskt fel i skattningen, dess medelfel samt kanske hela fördelningen. Med hjälp av bootstrap är detta förhållandevis enkelt.

Vår slumpmekanism P som genererade tidsserien har två komponenter $P = (\beta, F)$ om vi betraktar μ som känd och lika med $\bar{y}$. Vi hade redan en skattning av β och behöver en skattning av $\widehat{F}$.

Om vi kände β skulle vi kunna bilda $\varepsilon_t = z_t - \beta z_{t-1}$ för $t = 2, 3, \cdots, n$ och vi skulle kunna skatta F med den empiriska fördelningen för ε_t:na. Vi känner inte β men betraktar de observerade residualerna $\widehat{\varepsilon}_t = z_t - \widehat{\beta} z_{t-1}, \quad t = 2, 3, \cdots, n$ dvs residualerna för den observerade tidsserien om β:s sanna värde vore $\widehat{\beta}$.

Vi låter nu $\widehat{F}$ vara den empiriska fördelningen för dessa $n-1$ st empiriska residualer, dvs vi lägger massan $1/(n-1)$ i vardera av $\widehat{\varepsilon}_t, \quad t = 2, 3, \cdots, n$. Dessa är kanske inte exakt (fast nästan) avcentrerade och man kan därför modifiera valet av $\widehat{F}$ genom att lägga massan $1/(n-1)$ i vardera av $\widehat{\varepsilon}_t - \bar{\varepsilon}$ där $\bar{\varepsilon} = \sum_2^n \widehat{\varepsilon}_t/(n-1)$.

Vår skattade bootstrap-modell $\widehat{P}$ definieras av $\widehat{P} = (\widehat{\beta}, \widehat{F})$ och vi kan generera bootstrap-tidsserien $(z_t^*, \quad t = 1, 2, \cdots n)$ genom rekursionen $z_1^* = y_1 - \bar{y}$ och sen $z_k^* = \widehat{\beta} z_{k-1}^* + \varepsilon_k^*, \quad k = 2, 3, \cdots, n$. I stället för att välja $z_1^* = y_1 - \bar{y}$ skulle man kunna välja $z_1^* = y_N - \bar{y}$ där vi låter N vara likformigt fördelad på $1, 2, \cdots, n$, dvs att låter den simulerade tidsserien starta slumpmässigt i en av de ursprungliga observationerna. I praktiken spelar detta val av start av rekursionen inte speciellt stor roll om den observerade tidsserien är någorlunda lång.

Om detta upprepas B gånger erhålls B tidsserier av längd n och i varje sådan tidserie skattar vi parametern (i detta fall β) med den metodik vi valt och erhåller på detta sätt B skattningar av β. Vi kan skatta det systematiska felet i skattningen $\widehat{\beta}$ genom att beräkna medelvärdet av bootstrap-skattningarna av β och från detta dra den ursprungliga skattningen β. Om skattningsrutinen ger en underskattning av β (dvs negativ bias) kommer den för vår bootstrap-tidsserie typiskt att ge en skattning som ligger under det β-värde som gäller för bootstrap-tidsserien, dvs $\widehat{\beta}$.

Vi kan också få en skattning $\widehat{se}_{boot,B}$ av medelfelet för $\widehat{\beta}$ ur de på detta sätt genererade B skattningarna genom att, som tidigare, beräkna spridningen i ett histogram över dessa B skattningar.

10.5.3 Hypotesprövning

På ungefär liknande sätt kan man utvidga modellen till en Autoregressiv modell av ordning 2, dvs där vi låter

$$Z_t = \beta_1 Z_{t-1} + \beta_2 Z_{k-2} + \varepsilon_t, \quad t = 1, 2, \cdots n$$

där $E(\varepsilon_t) = 0$ och ε_t:na är oberoende likafördelade med fördelningen F. Vår sannolikhetsmodell P består då av två komponenter (β_1, β_2) och F. Med någon

lämplig metodik (t ex Yule-Walker-metoden) skattar vi β_1 och β_2 med $\widehat{\beta}_1$ och $\widehat{\beta}_2$. Dessutom vill vi kunna skatta F med $\widehat{F}$=empiriska fördelningen för observerade residualer. Vi vill använda dessa två numeriska värden $\widehat{\beta}_1$ och $\widehat{\beta}_2$ samt $\widehat{F}$ för att generera en nya tidsserier.

Vi måste här starta rekursionen med z_1 och z_2 eftersom z_3 beror av dessa. Vi har alltså $n-2$ st residualer $\varepsilon_3, \varepsilon_4, \cdots, \varepsilon_n$ i resten av tidsserien och dessa $n-2$ residualer skattar vi t ex med $\widehat{\varepsilon}_t = z_t - \widehat{\beta}_1 z_{t-1} - \widehat{\beta}_2 z_{t-2}$ för $t = 3, 4, \cdots, n$. Dessa $n-2$ skattade residualer betraktar vi nu som oberoende likafördelade (eventuellt efter avcentrering med deras gemensamma medelvärde) och empiriska fördelningen för dem är vad vi använder för att skatta fördelningen F med (dvs fördelningen för ε_t). Vi kan nu generera fiktiva "kopior" av den ursprungliga processen genom att på analogt sätt som ovan generera tidsserien utgående från AR(2)-modellen. Vi får samma problem som för $AR(1)$-processen att vi behöver z_1^* och z_2^* för att kunna starta rekursionen. Man kan välja $z_1^* = y_1 - \bar{y}$ och $z_2^* = y_2 - \bar{y}$ men även andra val är tänkbara, t ex att låta z_1^*, z_2^* vara $y_N - \bar{y}, y_{N+1} - \bar{y}$ där N väljs på måfå bland $1, 2, \cdots, n-1$.

Vi erhåller alltså B "fiktiva" realiseringar av $AR(2)$-processen med parametrarna $\widehat{\beta}_1$ och $\widehat{\beta}_2$ och "störningsfördelning" $\widehat{F}$. Ur varje sådan realisering skattar vi β_1 och β_2 och vi erhåller skattningarna $(\widehat{\beta}_1^*, \widehat{\beta}_2^*)$ och dessas två-dimensionella fördelning är alltså bootstrap-fördelningen för $(\widehat{\beta}_1, \widehat{\beta}_2)$. Notera speciellt att de i allmänhet är beroende, men samma beroende finns i fördelningen för $(\widehat{\beta}_1^*, \widehat{\beta}_2^*)$.

Vi vill kanske testa $H_0 : \beta_2 = 0$, dvs att vi i verkligheten har en första ordningens autoregressiv modell. Detta kan göras genom att i grundmodellen (dvs AR(2)-processen) göra ett konfidensintervall för β_2 och se om 0 ligger i det intervallet eller ej. Eftersom vi i AR(2)-modellen kan skatta medelfelet av β_2 kan vi också göra approximativa konfidensintervall för β_2 och använda oss av konfidensmetoden för att testa $H_0; \beta_2 = 0$.

10.5.4 Bootstrap av tidsblock

Ett annat alternativ att använda bootstrap på tidsserien är att dela in tiden i (överlappande) block av lämplig längd och sen göra bootstrap av dessa block. Med detta förfarande uppnåår man effekten att ha ett beroende över korta tidsintervall, men ha oberoende för tider längre ifrån varandra. Man kan t ex välja att dela in tiden i sammanhängande block om 3 vilket ger $n-2$ st sammanhängande block. Man lottar fram sin bootstrap-tidsserie genom att dra med lika sannolikhet och med återläggning från dessa block om 3 och "klistrar" ihop dessa block.

Sen skattar man $\widehat{\beta}^*$ från den genererade tidsserien. Denna strategi är mindre modell-beroende än den tidigare beskrivna metoden. Den byggde ju helt på att vi ansatt en autoregressiv tidsserie som modell. Dock är ju valet av blockstorlek kopplat till en föreställning om graden av beroende i processen.

Kapitel 11

Geometrisk representation

11.1 Inledning

I detta kapitel ser vi bootstrap-stickprovet $\boldsymbol{X}^* = (X_1^*, X_2^*, \cdots, X_n^*)$ givet observationer $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ som genererad av multinomialfördelningen – en generalisering av binomialfördelningen. Lite allmänna resultat om multinomialfördelningen presenteras i nästa avsnitt. Med hjälp av dessa resultat kommer vi att kunna bevisa bootstrap-hypotesen som gränsvärdesresultat i specialfallet då fördelningen F som genererat det ursprungliga stickprovet bara kan anta ett ändligt antal värden.

Vidare kommer vi att kunna visa att jackknife-skattningen av medelfel i princip (så när som på en oväsentlig faktor $\sqrt{(n-1)/n}$) överensstämmer med bootstrap-skattningen av medelfel om skattningen $\widehat{\theta}$ är linjär dvs om

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = \mu + \frac{1}{n}\sum_{i=1}^{n} \alpha(x_i)$$

där μ är en konstant.

Vi kommer också att visa att skattningen av det systematiska felet (bias) med jackknife och bootstrap är i princip lika (så när som på en faktor $(n-1)/n$) om skattningen $\widehat{\theta}$ är kvadratisk dvs av typen

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = \mu + \frac{1}{n}\sum_{i=1}^{n} \alpha(x_i) + \frac{1}{n^2}\sum_{1 \le i < j \le n} \beta(x_i, x_j).$$

11.2 Multinomialfördelningen

Betrakta en urna med kulor med m olika "färger" $\boldsymbol{a} = (a_1, a_2, \cdots, a_m)$ där proportionen kulor med färgen a_i är f_i. Låt $\boldsymbol{f} = (f_1, f_2, \cdots, f_m)$ och att vektorn $\boldsymbol{f}$ alltså beskriver fördelningen av färger. Vi drar n kulor på måfå med återläggning och erhåller i detta urval Z_j kulor med färgen a_j för $j = 1, 2, \cdots n$. Vi bildar vektorn $\boldsymbol{Z} = (Z_1, Z_2, \cdots, Z_m)$ som alltså beskriver antalet kulor av de olika färgerna som erhållits. Notera att $n = Z_1 + Z_2 + \cdots Z_m$.

Man kan med klassisk sannolikhetsdefinition visa att $\boldsymbol{Z}$ har följande sannolikhetsfördelning:

Sats 11.1 *Multinomialfördelning*
Den m-dimensionella variabeln $\boldsymbol{Z}$ är Mult$(n, \boldsymbol{f})$ där $\sum_1^m f_i = 1$ dvs

$$P(Z_1 = z_1; Z_2 = z_2; \cdots ; Z_m = z_m) = \frac{n!}{z_1! z_2! \cdots z_m!} f_1^{z_1} f_2^{z_2} \cdots f_m^{z_m}$$

för $z_1 + z_2 + \cdots + z_m = n$. □

Bevis:
Man inser att $f_1^{z_1} f_2^{z_2} \cdots f_m^{z_m}$ är sannolikheten att få först z_1 st av färgen a_1, sen z_2 st av färgen a_2 och slutligen z_m st av färgen a_m och samma sannolikhet gäller för varje omordning av sekvensen erhållna färger. Låt antalet sätt att ordna om sekvensen betecknas med $g(z_1, z_2, \cdots , z_m)$. Man kan permutera de erhållna n kulorna på $n!$ sätt. Man kan få alla dessa permutationer genom att först välja färger på de n omgångarna och antalet sätt detta går att göra betecknade vi med $g(z_1, z_2, \cdots , z_m)$. Därefter kan vi för de z_i kulorna med färg a_i permutera dessa på $z_i!$ sätt. Alla sätt att sortera om inom färgerna kan kombineras med varandra dvs totalt finns $z_1! z_2! \cdots z_m!$ sätt att göra detta på. På detta sätt får vi alla de $n!$ permutationerna dvs vi har

$$n! = g(z_1, z_2, \cdots , z_m) z_1! z_2! \cdots z_m!$$

vilket ger $g(z_1, z_2, \cdots , z_m) = n!/(z_1! z_2!, \cdots , z_m!)$. Detta ger att $\boldsymbol{Z}$ har ovanstående fördelning, dvs är Mult$(n, \boldsymbol{f})$. □

I specialfallet $m = 2$, dvs att man bara har två "färger" på kulorna där Z_1=antalet i urvalet med färg a_1 så blir $Z_2 = n - Z_1$ antalet med färg a_2. Z_1 är i denna situation som bekant Bin(n, f_1). Notera att Z_2 är Bin(n, f_2), där $f_1 + f_2 = 1$ men att alltså Z_1 och $Z_2 = n - Z_1$ är beroende. Multinomialfördelningen är alltså en generalisering av binomial-fördelningen. Allmänt blir de olika Z_i:na beroende stokastiska variabler. Man inser också att Z_i:na blir negativt beroende eftersom om en viss av dem blivit "onormalt" stor blir antagligen de övriga "onormalt" små. Det negativa beroendet är svagt om m är stort.

11.2.1 Väntevärdesvektor och kovariansmatris

Härnäst avser vi att ta fram väntevärdesvektor och kovariansmatris för multinomialfördelningen och dessutom den asymptotiska fördelningen då $n \to \infty$ och gör därför följande konstruktion.

Låt $U_1, U_2, \cdots , U_n$ vara den erhållna sekvensen färger. Vi definierar

$$Z_{ij} = \begin{cases} 1 & \text{om } U_i = a_j \\ 0 & \text{annars} \end{cases}$$

dvs Z_{ij} indikerar om den i:te dragningen ger färg a_j eller inte.

Vi ser att $Z_{1j}, Z_{2j}, \cdots, Z_{nj}$ är oberoende likafördelade och att $P(Z_{ij} = 1) = f_j$ och $P(Z_{ij} = 0) = 1 - f_j$. Man ser att $Z_j = Z_{1j} + Z_{2j} + \cdots + Z_{nj}$ är antalet av färg a_j. Vidare inses att Z_j är Bin(n, f_j) och att alltså $E(Z_j) = nf_j$ och $C(Z_j, Z_j) = V(Z_j) = nf_j(1 - f_j)$.

De olika Z_j:na är beroende och vi får för $j \neq k$ att

$$C(Z_j, Z_k) = C\left(\sum_{r=1}^{n} Z_{rj}, \sum_{s=1}^{n} Z_{sk}\right) = \sum_{r=1}^{n}\sum_{s=1}^{n} C(Z_{rj}, Z_{sk}).$$

Z_{rj} och Z_{sk} är oberoende om $r \neq s$ eftersom de då rör olika dragningsomgångar. Alltså försvinner alla termer med $r \neq s$ och vi erhåller

$$C(Z_j, Z_k) = \sum_{i=1}^{n} C(Z_{ij}, Z_{ik}).$$

Vi får

$$C(Z_{ij}, Z_{ik}) = E(Z_{ij}Z_{ik}) - E(Z_{ij})E(Z_{ik}) = P(Z_{ij} = 1; Z_{ik} = 1) - f_j f_k = -f_j f_k$$

eftersom $Z_{ij} = 1$ och $Z_{ik} = 1$ inte kan inträffa samtidigt ty den i:te kulan kan inte både ha färg a_j och färg a_k – vi studerar ju fallet $j \neq k$. Alltså erhåller vi $C(Z_j, Z_k) = -nf_j f_k$.

Detta betyder att $\boldsymbol{Z}$ har väntevärdesvektor $n\boldsymbol{f}$ och kovariansmatris $n\boldsymbol{C}$ där $c_{ii} = f_i(1 - f_i)$ och $c_{ij} = -f_i f_j$ för $i \neq j$.

Vi skattar $\boldsymbol{f}$ med $\widehat{\boldsymbol{f}}_{obs} = \boldsymbol{z}/n$ som är ett utfall av $\widehat{\boldsymbol{f}} = \boldsymbol{Z}/n$ där $\boldsymbol{Z}$ är Mult$(n, \boldsymbol{f})$. Ur ovanstående inses att $\widehat{\boldsymbol{f}} = \boldsymbol{Z}/n$ har väntevärdesvektor $\boldsymbol{f}$ och kovariansmatris $\boldsymbol{C}_{\widehat{\boldsymbol{f}}} = \boldsymbol{C}/n$.

Man kan visa följande sats som är en generalisering av stora talens lag och Centrala gränsvärdessatsen

Sats 11.2 *Stora talens lag och CGS för Multinomialfördelning*
Om den m-dimensionella vektorn $\boldsymbol{Z}$ är Mult$(n, \boldsymbol{f})$ och $\widehat{\boldsymbol{f}} = \boldsymbol{Z}/n$ så gäller då $n \to \infty$ att

1. $\widehat{\boldsymbol{f}} \to \boldsymbol{f}$ med sannolikhet 1
2. $\sqrt{n}(\widehat{\boldsymbol{f}} - \boldsymbol{f})$ är asymptotiskt $N_m(\boldsymbol{0}, \boldsymbol{C})$

där $\boldsymbol{C}$ har $c_{ii} = f_i(1 - f_i)$ och $c_{ij} = -f_i f_j$ för $i \neq j$. □

11.3 Ändliga fallet – Bevis av Bootstrap-hypotesen

I detta avsnitt skall vi bevisa att den grundläggande bootstrap-hypotesen är sann i ett specialfall nämligen då den bakomliggande fördelningen för våra observationer bara har ett ändligt antal tänkbara värden. Vi utnyttjar beteckningar och resultat för föregående avsnitt om multinomialfördelningen.

Man har följande sats som visar att bootstrap-hypotesen är sann i denna situation då F bara kan anta ett ändligt antal värden.

Sats 11.3 *Bootstrap-hypotesen*
Det gäller då $n \to \infty$ att vi med sannolikhet 1 erhåller observationer $x_1, x_2, \cdots, x_n$ så att fördelningarna för

$$\sqrt{n}\left(\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta\right) \text{ och } \sqrt{n}\left(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) - \widehat{\theta}(x_1, x_2, \cdots, x_n)\right)$$

är asymptotiskt lika under följande villkor:

1. $X_1, X_2, \cdots, X_n$ kan bara anta ett ändligt antal värden $a_1, a_2, \cdots, a_m$ och $P(X_i = a_j) = f_j$ för $j = 1, 2, \cdots m$
2. $\widehat{\theta}$ är plug-in-skattningen av θ och med z_j=antalet x_i som blivit a_j och $\widehat{\boldsymbol{f}}_{obs} = \boldsymbol{z}/n$ där $\boldsymbol{z} = (z_1, z_2, \cdots, z_m)$ är $\widehat{\theta}(x_1, \cdots, x_n) = h(\widehat{\boldsymbol{f}}_{obs})$
3. Funktionen h är två gånger kontinuerligt deriverbar och med begränsade derivator. □

Skiss till bevis:

Låt $\boldsymbol{a} = (a_1, a_2, \cdots, a_m)$ vara de tänkbara värdena och anta att de har sannolikheter $\boldsymbol{f} = (f_1, f_2, \cdots, f_m)$ dvs att $X_1, X_2, \cdots, X_n$ är oberoende och alla har fördelningen

$$P(X_i = a_j) = f_j,\ j = 1, 2, \cdots, m,\ \ i = 1, 2, \cdots n$$

Givet observationerna $x_1, x_2, \cdots, x_n$ av $X_1, X_2, \cdots, X_n$ skattar vi $\boldsymbol{f}$ med $\widehat{\boldsymbol{f}}_{obs} = \boldsymbol{z}/n$ där $\boldsymbol{z} = (z_1, \cdots, z_m)$ och z_j = antalet x_i:n som blivit a_j. $\widehat{\boldsymbol{f}}_{obs}$ är ett utfall av $\widehat{\boldsymbol{f}} = \boldsymbol{Z}/n$ där $\boldsymbol{Z}$ är Mult$(n, \boldsymbol{f})$.

Parametern $\theta = T(F)$ kan betraktas som en funktion $h(\boldsymbol{f})$ av vektorn $\boldsymbol{f}$ eftersom F beskrivs av $\boldsymbol{f}$. Om vi skattar θ med plug-in-skattningen $\widehat{\theta}(x_1, x_2, \cdots, x_n)$ måste $\widehat{\theta}(x_1, x_2, \cdots, x_n) = h(\widehat{\boldsymbol{f}}_{obs})$ eftersom $\widehat{\boldsymbol{f}}_{obs}$ är empiriska fördelningen för $x_1, x_2, \cdots, x_n$.

Enligt satsen om multinomialfördelning gäller att $\widehat{\boldsymbol{f}} \to \boldsymbol{f}$ och att $\sqrt{n}(\widehat{\boldsymbol{f}} - \boldsymbol{f})$ är asymptotiskt $N_m(\boldsymbol{0}, \boldsymbol{C})$.

Vi genererar ett (stokastiskt) bootstrap-stickprov $X_1^*, X_2^*, \cdots, X_n^*$ genom icke-parametrisk bootstrap precis som tidigare, dvs att

$$P(X_i^* = x_j) = 1/n,\ j = 1, 2, \cdots, n,\ \ i = 1, 2, \cdots, n$$

där $X_1^*, X_2^*, \cdots, X_n^*$ är oberoende. Vi definierar $\boldsymbol{Z}^* = (Z_1^*, Z_2^* \cdots, Z_m^*)$ där Z_j^* =antalet av $X_1^*, X_2^*, \cdots, X_n^*$ som är $= a_j$ dvs komponenterna i $\boldsymbol{Z}^*$ anger hur bootstrap-stickprovet fördelar sig över de olika värdena $(a_1, a_2, \cdots, a_m)$. Man ser att $\boldsymbol{Z}^*$ är Mult$(n, \widehat{\boldsymbol{f}}_{obs})$ givet observationerna $x_1, x_2, \cdots, x_n$. Låt vidare vektorn $\widehat{\boldsymbol{f}}^* = \boldsymbol{Z}^*/n$.

Det gäller alltså att (givet $x_1, x_2, \cdots, x_n$ eller ekvivalent givet $\widehat{\boldsymbol{f}} = \widehat{\boldsymbol{f}}_{obs}$) att

$$E(\widehat{\boldsymbol{f}}^*) = \widehat{\boldsymbol{f}}_{obs}.$$

Kovariansmatrisen för $\widehat{\boldsymbol{f}}^*$ är

$$C_{\widehat{\boldsymbol{f}}^*} = \frac{\widehat{\boldsymbol{C}}_{obs}}{n}$$

där $\widehat{\boldsymbol{C}}_{obs}$ har $\widehat{c}_{ii} = \widehat{f}_{i,obs}(1 - \widehat{f}_{i,obs})$ och $\widehat{c}_{ij} = -\widehat{f}_{i,obs}\widehat{f}_{j,obs}$ för $i \neq j$.

Vi får då enligt CGS att givet $\widehat{\boldsymbol{f}} = \widehat{\boldsymbol{f}}_{obs}$ så gäller att $\sqrt{n}(\widehat{\boldsymbol{f}}^* - \widehat{\boldsymbol{f}}_{obs})$ är approximativt $N_m(\boldsymbol{0}, \widehat{\boldsymbol{C}})$. Eftersom $\widehat{\boldsymbol{f}} \to \boldsymbol{f}$ då $n \to \infty$ gäller att $\widehat{\boldsymbol{C}} \to \boldsymbol{C}$ med sannolikhet 1. Detta betyder att då $n \to \infty$ gäller att $\sqrt{n}(\widehat{\boldsymbol{f}}^* - \widehat{\boldsymbol{f}}_{obs})|\widehat{\boldsymbol{f}} = \widehat{\boldsymbol{f}}_{obs}$ och $\sqrt{n}(\widehat{\boldsymbol{f}} - \boldsymbol{f})$ asymptotiskt får samma fördelning.

Vidare antog vi att

$$\widehat{\theta}(X_1, X_2, \cdots, X_n) = h(Z_1/n, Z_2/n, \cdots, Z_m/n) = h(\widehat{\boldsymbol{f}}).$$

där $\theta = h(\boldsymbol{f})$ dvs att $\widehat{\theta}(x_1, x_2, \cdots, x_n)$ är plug-in-skattningen av θ.

Om nu h-funktionen är tillräckligt "snäll" (t ex två gånger kontinuerligt deriverbar och med begränsade derivator) så kommer fördelningen för

$$\sqrt{n}(h(\widehat{\boldsymbol{f}}^*) - h(\widehat{\boldsymbol{f}}))|\widehat{\boldsymbol{f}}$$

att konvergera mot fördelningen för

$$\sqrt{n}(h(\widehat{\boldsymbol{f}}) - h(\boldsymbol{f}))$$

då $n \to \infty$ för nästan alla $x_1, x_2, \cdots, x_n$ dvs bootstrap-hypotesen är sann i denna situation.

Detta inses ur en Taylorutveckling dvs

$$\widehat{\theta}(X_1, X_2, \cdots, X_n) = h(\widehat{\boldsymbol{f}}) = h(\boldsymbol{f}) + (\widehat{\boldsymbol{f}} - \boldsymbol{f})\frac{\partial h}{\partial \boldsymbol{u}}\bigg|_{\boldsymbol{u}=\boldsymbol{f}} + \text{ restterm}$$

och

$$\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) = h(\widehat{\boldsymbol{f}}^*) = h(\widehat{\boldsymbol{f}}) + (\widehat{\boldsymbol{f}}^* - \widehat{\boldsymbol{f}})\frac{\partial h}{\partial \boldsymbol{u}}\bigg|_{\boldsymbol{u}=\widehat{\boldsymbol{f}}} + \text{ restterm.}$$

Resttermerna är av typen

$$(\widehat{\boldsymbol{f}} - \boldsymbol{f})\frac{\partial^2 h}{\partial u_i \partial u_j}\bigg|_{\boldsymbol{u}=\boldsymbol{f}}(\widehat{\boldsymbol{f}} - \boldsymbol{f})^T$$

och dessa kommer att vara av storleksordning $1/n$ (dvs $O_P(1/n)$) medan $\widehat{\boldsymbol{f}} - \boldsymbol{f}$ är av storleksordning $1/\sqrt{n}$ och man kan visa att vi kan bortse från resttermerna då $n \to \infty$.

Eftersom $\widehat{\boldsymbol{f}} \to \boldsymbol{f}$ och h var kontinuerligt deriverbar gäller att

$$\left.\frac{\partial h}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\widehat{\boldsymbol{f}}} \to \left.\frac{\partial h}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\boldsymbol{f}}.$$

Eftersom alltså

$$\sqrt{n}\left(\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta\right) = \sqrt{n}\left(h(\widehat{\boldsymbol{f}}) - h(\boldsymbol{f})\right) =$$

$$\sqrt{n}(\widehat{\boldsymbol{f}} - \boldsymbol{f})\left.\frac{\partial h}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\boldsymbol{f}} + O_P(1/\sqrt{n})$$

och

$$\sqrt{n}\left(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) - \widehat{\theta}(x_1, x_2, \cdots, x_n)\right) = \sqrt{n}\left(h(\widehat{\boldsymbol{f}}^*) - h(\widehat{\boldsymbol{f}})\right) =$$

$$= \sqrt{n}((\widehat{\boldsymbol{f}}^* - \widehat{\boldsymbol{f}}))\left.\frac{\partial h}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\widehat{\boldsymbol{f}}} + O_P(1/\sqrt{n})$$

och $\sqrt{n}(\widehat{\boldsymbol{f}}^* - \widehat{\boldsymbol{f}})$ och $\sqrt{n}(\widehat{\boldsymbol{f}} - \boldsymbol{f})$ asymptotiskt får samma fördelning $N_m(\boldsymbol{0}, \boldsymbol{C})$ ser man att med sannolikhet 1 erhåller vi alltså en sekvens observationer $x_1, x_2, \cdots, x_n$ sådana att fördelningen för

$$\sqrt{n}\left(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) - \widehat{\theta}(x_1, x_2, \cdots, x_n)\right)$$

konvergerar mot fördelningen för

$$\sqrt{n}\left(\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta\right).$$

då $n \to \infty$ □

11.4 Allmänt om geometrisk representation

Resultatet i föregående avsnitt förutsatte att F bara kunde anta ett ändligt antal värden. Om F kan anta oändligt antal värden kan vi ändå för ett fixt n använda oss av multinomial-fördelningen vad gäller bootstrap. Vi kommer med hjälp av denna representation att kunna visa relationer mellan bootstrap och jackknife.

Tidigare har vi betraktat en skattning $\widehat{\theta}$ som en funktional $t(\widehat{F})$ av $\widehat{F}$ (den empiriska fördelningen för data). I det följande ser vi i stället skattningar som funktioner av sannolikhetsvektorer

$$\boldsymbol{P}^* = (P_1^*, P_2^*, \cdots, P_n^*)^T$$

dvs som uppfyller $0 \leq P_i^* \leq 1$ och $\sum_{i=1}^n P_i^* = 1$. Tolkningen av P_i^* är att P_i^* =andelen av stickprovet som x_i utgör. Det ursprungliga stickprovet svarar alltså mot vektorn $\boldsymbol{P}_0 = (1/n, 1/n, \cdots, 1/n)^T$.

Vi studerar i fortsättningen hur skattningar ändras när vi ändrar på dessa sannolikhetsvektorer. Vi betraktar $x_1, x_2, \cdots, x_n$ som fix och låter $\widehat{F}^* = \widehat{F}(\boldsymbol{P}^*)$ vara fördelningen som lägger massan P_i^* på x_i, $\quad i = 1, 2, \cdots, n$. Vi betraktar $\widehat{\theta}^*$ som en funktion av $\boldsymbol{P}^*$, $T(\boldsymbol{P}^*)$ genom

$$\widehat{\theta}^* = T(\boldsymbol{P}^*) = t(\widehat{F}(\boldsymbol{P}^*)).$$

Funktionalen $T(\boldsymbol{P}^*)$ kommer att spela huvudrollen i fortsättningen av detta avsnitt.

Man noterar som tidigare nämnts att den ursprungliga skattningen

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = t(\widehat{F}_n)$$

svarar mot vektorn

$$\boldsymbol{P}^0 = \left(\frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n}\right)^T$$

dvs att $\widehat{\theta}(x_1, x_2, \cdots, x_n) = T(\boldsymbol{P}_0)$ medan jackknife-stickprovet

$$\boldsymbol{x}_{(i)} = (x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$$

svarar mot vektorn

$$\boldsymbol{P}_{(i)} = \left(\frac{1}{n-1}, \frac{1}{n-1}, \cdots, \frac{1}{n-1}, 0, \frac{1}{n-1}, \cdots, \frac{1}{n-1}\right)^T$$

med 0:an i position i, $\quad i = 1, 2, \cdots, n$

så i:te jackknifeskattningen $\widehat{\theta}_{(i)} = T(\boldsymbol{P}_{(i)})$.

För att detta skall fungera skall $\widehat{\theta}$ vara invariant under permutationer av observationerna för i sådana fall ger vektorn $\boldsymbol{P}^*$ full information om stickprovet och vi kan betrakta $\widehat{\theta}^*$ som en funktion av $\boldsymbol{P}^*$. Detta är fallet om $\widehat{\theta}$ är en plug-in-skattning.

Man kan se $T(\boldsymbol{P}^*)$ som en yta över simplexen som spänns upp av enhetsvektorerna

$$\boldsymbol{e}_i = (0, 0, \cdots, 0, 1, 0, \cdots, 0)^T$$

med 1:an i i:te positionen. $\quad i = 1, 2, \cdots, n$

11.5 Bootstrap

Bootstrap-genereringen kan uttryckas på så vis att vi drar $n\boldsymbol{P}^*$ från en multinomialfördelning med n dragningar och sannolikheten $1/n$ för vardera elementet, dvs att $n\boldsymbol{P}^*$ är Mult$(n, \boldsymbol{P}_0)$.

$n\boldsymbol{P}^*$ har väntevärdesvektorn $(1/n, 1/n, \cdots, 1/n)^T = \boldsymbol{P}^0$. Enligt resulaten tidigare blir $V(nP_i^*) = n \cdot 1/n \cdot (1 - 1/n)$ och för $i \neq j$ har vi

$$C(nP_i^*, nP_j^*) = -n \cdot 1/n \cdot 1/n.$$

Detta kan sammanfattas som att $\boldsymbol{P}^*$ har väntevärdesvektorn $\boldsymbol{P}_0$ och kovariansmatrisen

$$\frac{\boldsymbol{I}}{n^2} - \frac{\boldsymbol{P}^0 \boldsymbol{P}^{0^T}}{n}$$

där $\boldsymbol{I}$ =identitetsmatrisen dvs den som har 1:or på diagonalen och 0:or i övrigt. Den slumpmässiga vektorn $\boldsymbol{P}^*$ med ovanstående fördelning ligger naturligtvis i simplexen eftersom $\sum_{i=1}^n P_i^* = 1$ och $0 \leq P_i^* \leq 1$ för alla i.

Alltså kan vi uttrycka bootstrapskattningen $\widehat{\theta}^*$ som

$$\widehat{\theta}^* = \widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*) = T(\boldsymbol{P}^*)$$

där $\boldsymbol{P}^*$ är en slumpmässig vektor där $n\boldsymbol{P}^*$ är Mult$(n, \boldsymbol{P}_0)$.

När vi skattar bootstrap-förväntan $E_{\widehat{F}}(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*))$ så bildar vi väntevärdet med avseende på denna fördelning och betecknar det med $E_*(T(\boldsymbol{P}^*))$. När vi skattar bootstrap-medelfelet $\widehat{se}_{boot}$ så bildar vi variansen med avseende på fördelningen för $\boldsymbol{P}^*$ och skriver detta som $V_*(T(\boldsymbol{P}^*))$.

11.6 Linjära skattningar

Definition 11.1 *Linjära skattningar*
En linjär skattning $T(\boldsymbol{P}^*)$ är linjär som funktion av vektorn $\boldsymbol{P}^*$ och har alltså formen

$$T(\boldsymbol{P}^*) = c_0 + \sum_{i=1}^n a_i P_i^* = c_0 + \boldsymbol{P}^* \boldsymbol{a}$$

där c_0 och $\boldsymbol{a} = (a_1, a_2, \cdots, a_n)$ ej beror av $\boldsymbol{P}^*$. □

Kommentar: c_0 och $\boldsymbol{a}$ beror i allmänthet av $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ men denna betraktas som fix. Vi ser T primärt som en funktion av "återsamplingsvektorn" $\boldsymbol{P}^*$.

En linjär skattning är alltså linjär i $\boldsymbol{P}^*$ när vi ser den som en funktion av "återsamplingsvektorerna" $\boldsymbol{P}^*$. Detta innebär också att om vi ser ovanstående som en funktion av $\boldsymbol{P}^*$ bildar $T(\boldsymbol{P}^*)$ för en linjär skattning ett hyperplan över simplexen där $\boldsymbol{P}^*$ är definierad.

Exempel 11.1 *Aritmetiskt medelvärde* $\bar{x}$
Om skattningen är $\widehat{\theta}(x_1, x_2, \cdots, x_n) = \bar{x}$ så har vi bootstrap-skattningen

$$\bar{x}^* = \sum_{i=1}^n P_i^* x_i$$

Vi har alltså $c_0 = 0$ och $a_i = x_i$ och $\bar{x}$ är alltså en linjär skattning. Man noterar att detta också kan skrivas som

$$\bar{x}^* = \sum_{i=1}^{n} P_i^* x_i = \bar{x} + \sum_{i=1}^{n} \left(P_i^* - \frac{1}{n} \right) (x_i - \bar{x})$$

dvs att vi skriver

$$T(\boldsymbol{P}^*) = \bar{x} + \sum_{i=1}^{n} (P_i^* - \frac{1}{n})(x_i - \bar{x}) = T(\boldsymbol{P}^0) + (\boldsymbol{P}^* - \boldsymbol{P}^0)^T \boldsymbol{U}$$

där $U_i = (x_i - \bar{x})$ och alltså $\sum_1^n U_i = 0$. □

Vi kan allmänt skriva om en linjär skattning $T(\boldsymbol{P}^*)$ på följande form i samma anda som för $\bar{x}$ i exemplet dvs att

$$T(\boldsymbol{P}^*) = T(\boldsymbol{P}^0) + (\boldsymbol{P}^* - \boldsymbol{P}^0)^T \boldsymbol{U}$$

där $\boldsymbol{P}^0 = (1/n, 1/n, \cdots, 1/n)$ och $\boldsymbol{U} = (U_1, U_2, \cdots, U_n)^T$ är en vektor som uppfyller $\sum_{i=1}^n U_i = 0$. Funktionen T ses ju primärt som en funktion av $\boldsymbol{P}^*$ men ovanstående definition av begreppet linjär skattning innebär också att $\widehat{\theta}(x_1, x_2, \cdots, x_n)$ (som vi ser som en funktion av $\boldsymbol{x}$) kan skrivas på formen

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = \mu + \frac{1}{n} \sum_{i=1}^{n} \alpha(x_i)$$

för en konstant μ och en funktion $\alpha(\cdot)$. Konstanten μ kan naturligtvis "bakas in" i funktionen $\alpha(\cdot)$. Det var detta vi i tidigare definierade som en linjär skattning. Detta inser man av följande resonemang.

Om vi sätter $\boldsymbol{P}^* = \boldsymbol{e}_i = (0, 0, \cdots, 0, 1, 0, \cdots, 0)^T$ med 1:an i i:e positionen (dvs att $\boldsymbol{e}_i$ är i:te enhetsvektorn) får vi $T(\boldsymbol{e}_i) = \widehat{\theta}(x_i, x_i, \cdots, x_i)$ och också $T(\boldsymbol{e}_i) = c_0 + a_i$. Alltså är $a_i = \widehat{\theta}(x_i, x_i, \cdots, x_i) - c_0$ och vi ser att vi skulle kunna låta $\alpha(x_i) = \widehat{\theta}(x_i, x_i, \cdots, x_i)$ och $\mu = 0$ för $i = 1, 2, \cdots, n$.

Om vi i stället sätter $\boldsymbol{P}^* = \boldsymbol{P}^0$ erhåller vi då

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = T(\boldsymbol{P}^0) = c_0 + \frac{1}{n} \sum_{i=1}^{n} a_i = \frac{1}{n} \sum_{i=1}^{n} \alpha(x_i)$$

och alltså är $\widehat{\theta}(x_1, \cdots, x_n)$ linjär enligt båda typerna av definition.

Vi får också

$$T(\boldsymbol{P}^*) = \sum_{i=1}^{n} P_i^* \alpha(x_i) = \frac{1}{n} \sum_{i=1}^{n} \alpha(x_i) + \sum_{i=1}^{n} P_i^* \left(\alpha(x_i) - \frac{1}{n} \sum_{j=1}^{n} \alpha(x_j) \right) =$$

$$= T(\boldsymbol{P}^0) + \sum_{i=1}^{n} (P_i^* - \frac{1}{n}) \left(\alpha(x_i) - \frac{1}{n} \sum_{j=1}^{n} \alpha(x_j) \right) = T(\boldsymbol{P}^0) + (\boldsymbol{P}^* - \boldsymbol{P}^0)^T \boldsymbol{U}$$

där $\boldsymbol{U} = (U_1, U_2, \cdots, U_n)^T$ med $U_i = (\alpha(x_i) - \sum_{j=1}^n \alpha(x_j)/n)$ och alltså gäller att $\sum_i^n U_i = 0$.

Mot jackknife-stickprovet $\boldsymbol{x}_{(i)}$ svarar vektorn

$$\boldsymbol{P}_{(i)} = \frac{n}{n-1}\boldsymbol{P}^0 - \frac{1}{n-1}\boldsymbol{e}_i$$

och vi får lätt

$$\boldsymbol{P}_{(i)} - \boldsymbol{P}^0 = \frac{n}{n-1}\boldsymbol{P}^0 - \frac{1}{n-1}\boldsymbol{e}_i - \boldsymbol{P}^0 = \frac{1}{n-1}(\boldsymbol{P}^0 - \boldsymbol{e}_i)$$

Sats 11.4 *Medelfel: Samband mellan Jackknife och bootstrap*
Välj T_{LIN} linjär så att den antar värdena $T(\boldsymbol{P}_{(i)})$ för jackknife-stickproven $\boldsymbol{x}_{(i)}$ för $i = 1, 2, \cdots, n$.

T_{LIN} beskriver alltså det hyperplan som går genom punkterna $(\boldsymbol{P}_{(i)}, T(\boldsymbol{P}_{(i)}))$ för $i = 1, 2, \cdots, n$.

Då gäller att

$$\frac{n-1}{n}\widehat{se}^2_{jack} = V_*(T_{LIN}(\boldsymbol{P}^*))$$

dvs att jackknife-skattningen av medelfelet är (förutom faktorn $\sqrt{(n-1)/n}$) samma sak som bootstrap-skattningen av medelfelet för lineariseringen T_{LIN} av T. □

Bevis:

Vi klarade egentligen av detta i kapitel 9 men det blir mycket lättare att genomföra kalkylen när man har tillgång till den geometriska representationen.

Vi har ekvationssystemet

$$\widehat{\theta}_{(i)} = T_{LIN}(\boldsymbol{P}_{(i)}) = c_0 + (\boldsymbol{P}_{(i)} - \boldsymbol{P}^0)^T\boldsymbol{U} =$$

$$= c_0 + \frac{1}{n-1}(\boldsymbol{P}^0 - \boldsymbol{e}_i)^T\boldsymbol{U} = c_0 - \frac{U_i}{n-1}, \quad i = 1, 2, \cdots, n$$

eftersom $\boldsymbol{P}^{0T}\boldsymbol{U} = 0$. Genom summation över i erhåller man att $c_0 = \widehat{\theta}_{(\cdot)}$ och alltså är $U_i = (n-1)(\widehat{\theta}_{(\cdot)} - \widehat{\theta}_{(i)})$. Vi får då

$$V_*(T_{LIN}(\boldsymbol{P}^*)) = V_*\left((\boldsymbol{P}^* - \boldsymbol{P}^0)^T\boldsymbol{U}\right) = \boldsymbol{U}^T V_*(\boldsymbol{P}^* - \boldsymbol{P}^0)\boldsymbol{U} =$$

$$= \boldsymbol{U}^T\left(\frac{\boldsymbol{I}}{n^2} - \frac{\boldsymbol{P}^0(\boldsymbol{P}^0)^T}{n}\right)\boldsymbol{U} =$$

$$= (\text{ ty } (\boldsymbol{P}^0)^T\boldsymbol{U} = 0) = \frac{\boldsymbol{U}^T\boldsymbol{U}}{n^2} - 0 = \frac{1}{n^2}\boldsymbol{U}^T\boldsymbol{U} =$$

$$= \frac{(n-1)^2}{n^2}\sum_{i=1}^n(\widehat{\theta}_{(i)} - \widehat{\theta}_{(\cdot)})^2 = \frac{n-1}{n}\widehat{se}^2_{jack}$$

Detta redde vi egentligen ut i kapitel 9! □

11.7 Andra jackknife-approximationer

I ovanstående resultat anpassade vi ett hyperplan $T_{LIN}(\boldsymbol{P}^*)$ så att det stämde för jackknife-stickproven $\boldsymbol{x}_{(i)}$ eller uttryckt annorlunda så att det stämde för punkterna $\boldsymbol{P}_{(i)}$, men en annan (och troligen bättre) approximation vore att approximera $T(\boldsymbol{P}^*)$ genom att ta tangentplanet i punkten $\boldsymbol{P}^0$ som svarar mot den ursprungliga skattningen $\widehat{\theta}$. Detta tangentplan har formen

$$T_{tan}(\boldsymbol{P}^*) = T(\boldsymbol{P}^0) + (\boldsymbol{P}^* - \boldsymbol{P}^0)\boldsymbol{U}$$

med $\boldsymbol{U} = (U_1, U_2, \cdots, U_n)^T$ där

$$U_i = \lim_{\epsilon \to 0} \frac{T\left(\boldsymbol{P}^0 + \epsilon(\boldsymbol{e}_i - \boldsymbol{P}^0)\right) - T(\boldsymbol{P}^0)}{\epsilon}, \quad i = 1, 2, \cdots, n$$

där som tidigare $\boldsymbol{e}_i$ är enhetsvektorn i i-riktningen dvs $(0, 0, \cdots, 0, 1, 0 \cdots, 0)^T$ med 1:an i position i.

Dessa U_i är de empiriska influens-värdena. Detta ger variansskattningen

$$\widehat{se}^2_{IJ} = \frac{1}{n^2} \sum_{i=1}^{n} U_i^2.$$

Detta kallas den infinitesimala jackknife-skattningen. Man kan uppfatta det som att vi stör vektorn $\boldsymbol{P}^0 = (1/n, 1/n, \cdots, 1/n)^T$ genom att lägga till massan ϵ till data x_i och kompensera genom att plocka bort massan ϵ/n från de n datapunkterna. U_i:na anger hur detta förändrar skattningen då $\epsilon \to 0$, dvs då störningen blir infinitesimal. Dessa kan uppfattas som ett slags derivator i i-te koordinatriktningen.

11.8 Serieutvecklingar

Man har ofta en serieutveckling av skattningen $\widehat{\theta} = T(\boldsymbol{P}^0) = t(\widehat{F})$ av parametern $\theta = t(F)$ av typen

$$t(\widehat{F}) = t(F) + \frac{1}{n} \sum_{i=1}^{n} U(x_i, F) + \text{restterm}$$

där resttermen är av storleksordning $1/n^2$. Detta utgör ett slags Taylorutveckling av funktionalen t.

Storheten $U(x_i, F)$ kallas en influensfunktion och definieras av

$$U(x, F) = \lim_{\epsilon \to 0} \frac{t\left((1-\epsilon)F + \epsilon\delta_x\right) - t(F)}{\epsilon}$$

där δ_x är en punktmassa i punkten x. Detta betyder att $U(x, F)$ är ett slags derivata för funktionalen t. Vi kan se det som att vi har stört fördelningen

F genom att lägga en punktmassa av storleken ϵ i punkten x och skalat ner resten av fördelningen med faktorn $(1-\epsilon)$.

Detta betyder att om vi kan försumma resttermen i serieutvecklingen ovan så får vi

$$V_F\left(t(\widehat{F})\right) \approx V_F\left(\theta + \frac{1}{n}\sum_{i=1}^{n} U(X_i, F)\right) =$$

$$= \frac{1}{n} V_F\left(U(X,F)\right) = \frac{1}{n} E_F\left(U(X,F)^2\right)$$

eftersom $E_F\left(U(X,F)\right) = 0$.

Man kan också notera att serieutvecklingen innebär att vi approximerat $t(\widehat{F}) = \widehat{\theta}$ med en linjär skattning.

De olika medelfelsskattningarna innebär nu att man gör olika typer av approximationer av $U(x,F)$. T ex så innebär jackknife-skattningen att vi tar $\epsilon = -1/(n-1)$ medan den infinitesimala jackknife-skattningen innebär att vi tar $U(x,\widehat{F})$ i stället för $U(x,F)$.

11.8.1 Postiv jackknife

Om man i approximationen av influensfunktionen i stället tar $\epsilon = 1/(n+1)$ får man "positiv jackknife", vilket svarar mot att man som det i :te av de n jackknife-stickproven tar

$$(x_1, x_2, \cdots, x_{i-1}, x_i, x_i, x_{i+1}, \cdots, x_n),\ i = 1, 2, \cdots, n$$

dvs tar den i :te observationen "dubbelt". Denna skattning har dock rätt dåliga egenskaper eftersom den "blåser upp" eventuellt avvikande observationer på ett överdrivet sätt.

För övrigt kan nämnas att detta är intimt förknippat med Gauss' approximationsformler (som ju också innebär en linearisering av en funktion av stokastiska variabler). Man kan visa att en s k icke-parametrisk variant av Gauss' approximation ger samma resultat som infinitesimal jackknife.

11.9 Bias-skattningar

Det finns en liknande relation mellan jackknife- och bootstrap-skattningarna av bias som det fanns för medelfelsskattningarna.

Om skattningen är linjär och är en plug-in-skattning så har den bias=0 och skattningen för bias är 0 både för bootstrap och jackknife vilket framgår av följande sats.

Sats 11.5 *Bias för linjära plug-in-skattningar*

Om $\widehat{\theta}$ är en linjär plug-in-skattning av θ där

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = \frac{1}{n}\sum_{i=1}^{n}\alpha(x_i) = \int_{-\infty}^{\infty}\alpha(x)d\widehat{F}(x) \text{ och } \theta = \int_{-\infty}^{\infty}\alpha(x)dF(x)$$

saknar den bias och bias-skattningarna är 0 både för bootstrap och jackknife.□

Bevis: Vi får

$$bias_F(\widehat{\theta}, \theta) = E_F(\widehat{\theta}(X_1, X_2, \cdots, X_n)) - \theta =$$

$$= \frac{1}{n}\sum_{i=1}^{n}E_F(\alpha(X_i)) - \int_{-\infty}^{\infty}\alpha(x)dF(x) =$$

$$= E_F(\alpha(X_1)) - \int_{-\infty}^{\infty}\alpha(x)dF(x) = 0$$

och alltså är $bias_F = 0$.

Vi får vidare för bootstrap-skattningen $\widehat{bias}_{boot}$

$$\widehat{bias}_{boot} = E_{\widehat{F}}(\widehat{\theta}(X_1^*, X_2^*, \cdots, X_n^*)) - \widehat{\theta}(x_1, x_2, \cdots, x_n) =$$

$$= E_{\widehat{F}}(\frac{1}{n}\sum_{i=1}^{n}\alpha(X_i^*)) - \frac{1}{n}\sum_{i=1}^{n}\alpha(x_i) =$$

$$= E_{\widehat{F}}(\alpha(X_1^*)) - \frac{1}{n}\sum_{i=1}^{n}\alpha(x_i) = \frac{1}{n}\sum_{i=1}^{n}\alpha(x_i) - \frac{1}{n}\sum_{i=1}^{n}\alpha(x_i) = 0.$$

Som alternativ kan vi använda representationen av $T(\boldsymbol{P}^*)$ och erhåller (där E_* avser väntevärdesbildning med avseende på multinomialfördelningen $\text{Mult}(n, \boldsymbol{P}^0)$)

$$\widehat{bias}_{boot} = E_*(T(\boldsymbol{P}^*)) = E_*\left(T(\boldsymbol{P}^0) + (\boldsymbol{P}^* - \boldsymbol{P}^0)^T\boldsymbol{U}\right) = T(\boldsymbol{P}^0)$$

eftersom $E_*(\boldsymbol{P}^*) = \boldsymbol{P}^0$.

För jackknife gäller att $\widehat{bias}_{jack} = (n-1)(\widehat{\theta}_{(\cdot)} - \widehat{\theta})$ där $\widehat{\theta}_{(\cdot)} = \sum_1^n \widehat{\theta}_{(i)}/n$ där $\widehat{\theta}_{(i)} = T(\boldsymbol{P}_{(i)})$ är skattningen i i:te jackknife-stickprovet. Vi får alltså

$$\widehat{bias}_{jack} = (n-1)\left(\frac{1}{n}\sum_{i=1}^{n}T(\boldsymbol{P}_{(i)}) - T(\boldsymbol{P}^0)\right).$$

Vidare har vi att $\boldsymbol{P}_{(i)} - \boldsymbol{P}^0 = (\boldsymbol{P}^0 - \boldsymbol{e}_i)/(n-1)$ och får alltså

$$T(\boldsymbol{P}_{(i)}) = T(\boldsymbol{P}^0) + (\boldsymbol{P}_{(i)} - \boldsymbol{P}^0)^T\boldsymbol{U} = T(\boldsymbol{P}^0) + \frac{1}{n-1}(\boldsymbol{P}^0 - \boldsymbol{e}_i)^T\boldsymbol{U} =$$

$$= T(\boldsymbol{P}^0) + \frac{1}{n-1}(\boldsymbol{P}^0)^T\boldsymbol{U} - \frac{U_i}{(n-1)} = T(\boldsymbol{P}^0) - \frac{U_i}{n-1}$$

eftersom $(\boldsymbol{P}^0)^T\boldsymbol{U} = (U_1 + U_2 + \cdots + U_n)/n = 0$. Detta ger

$$\widehat{\theta}_{(\cdot)} = \frac{1}{n}\sum_{i=1}^{n} T(\boldsymbol{P}_{(i)}) = T(\boldsymbol{P}^0)\frac{1}{n(n-1)}\sum_{i=1}^{n} U_i = T(\boldsymbol{P}^0).$$

Alltså blir $\widehat{bias}_{jack} = (n-1)(\widehat{\theta}_{(\cdot)} - T(\boldsymbol{P}^0)) = 0$. □ .

Eftersom jackknife- och bootstrap-skattningarna av bias blir = 0 för linjära skattningarna så måste man gå över till kvadratiska skattningar.

Definition 11.2 *Kvadratiska skattningar*
$T(\boldsymbol{P}^*)$ är en kvadratisk skattning om

$$T(\boldsymbol{P}^*) = c_0 + \sum_{i=1} a_i P_i^* + \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij} P_i^* P_j^*$$

dvs ett andragrads-polynom i P_i^*:na. □

Man kan skriva en kvadratisk skattning i "avcentrerad" form som

$$T(\boldsymbol{P}^*) = T(\boldsymbol{P}^0) + (\boldsymbol{P}^* - \boldsymbol{P}^0)^T\boldsymbol{U} + \frac{1}{2}(\boldsymbol{P}^* - \boldsymbol{P}^0)^T\boldsymbol{V}(\boldsymbol{P}^* - \boldsymbol{P}^0)$$

där $\boldsymbol{U} = (U_1, U_2, \cdots, U_n)^T$ uppfyller $\sum_{i=1}^n U_i = 0$ och $\boldsymbol{V}$ är en nxn-matris som uppfyller $\sum_i V_{ij} = \sum_j V_{ij} = 0$ för alla i, j.

På motsvarande sätt som för linjära skattningar betyder det att en kvadratisk skattning kan skrivas som

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) = \mu + \frac{1}{n}\sum_{i=1}^{n}\alpha(x_i) + \frac{1}{n^2}\sum_{1\le i<j\le n}\beta(x_i, x_j)$$

Exempel 11.2 *σ^2-skattning*
Plug-in-skattningen

$$\widehat{\sigma}^2(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

av σ^2 är en kvadratisk skattning eftersom

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{j=1}^{n} x_j)^2\right) =$$

$$= \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{j=1}^{n} x_j^2 + 2\sum_{1\le i<j\le n} x_i x_j)\right) =$$

$$= \frac{1}{n}\sum_{i=1}^{n} x_i^2(1 - \frac{1}{n}) + \frac{1}{n^2}\sum_{1\le i<j\le n}(-2x_i x_j)$$

dvs vi kan ta $\mu = 0$, $\alpha(x_i) = x_i^2(1 - 1/n)$ och $\beta(x_i, x_j) = -2x_i x_j$. □

Sats 11.6 *Bias: Samband mellan jackknife och bootstrap*
Låt $T_{QUAD}(\boldsymbol{P}^*)$ vara en kvadratisk skattning av ovanstående typ som överensstämmer med $T(\boldsymbol{P}^*)$ för jackknife-stickproven $\boldsymbol{x}_{(i)}$ dvs går genom punkterna $\left(\boldsymbol{P}_{(i)}, T(\boldsymbol{P}_{(i)})\right)$ för $i = 1, 2, \cdots, n$ och dessutom för det ursprungliga stickprovet $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ dvs går genom punkten $(\boldsymbol{P}^0, T(\boldsymbol{P}^0))$. Vi approximerar alltså T med den kvadratiska skattningen T_{QUAD}.

Då gäller att

$$\frac{n-1}{n}\widehat{bias}_{jack} = E_*\left(T_{QUAD}(\boldsymbol{P}^*)\right) - \widehat{\theta} = \widehat{bias}_{boot}(T_{QUAD})$$

dvs jackknife-skattningen $\widehat{bias}_{jack}$ av bias överensstämmer med den teoretiska bootstrap-skattning av bias som skulle erhållas med approximationen T_{QUAD} av T (förutom faktorn $(n-1)/n$). □

Kan alltså T väl approximeras med en kvadratisk funktional T_{QUAD} så överensstämmer jackknife- och bootstrap-skattningarna av bias med varandra (så när som på faktorn $(n-1)/n$).

Man kan i detta sammanhang t ex tänka på vad som var fallet för σ^2-skattningen (plug-in-skattningen)

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

där bias-korrektionen för jackknife gav den väntevärdesriktiga skattningen. För bootstrap stämde det "nästan".

Man kan också jämföra detta med att medelfelsskattningarna överensstämmer om T kan approximeras med en linjär funktional T_{LIN}.

Bevis:

Vi har alltså $T(\boldsymbol{P}^0) = T_{QUAD}(\boldsymbol{P}^0) = \widehat{\theta}$ och

$$\widehat{\theta}_{(i)} = T(\boldsymbol{P}_{(i)}) = T_{QUAD}(\boldsymbol{P}^0) + (\boldsymbol{P}_{(i)} - \boldsymbol{P}^0)^T\boldsymbol{U} + \frac{1}{2}(\boldsymbol{P}_{(i)} - \boldsymbol{P}^0)^T\boldsymbol{V}(\boldsymbol{P}_{(i)} - \boldsymbol{P}^0)$$

Vi får då med hjälp av det tidigare resultatet

$$(\boldsymbol{P}_{(i)} - \boldsymbol{P}^0) = (\boldsymbol{P}^0 - \boldsymbol{e}_i)/(n-1)$$

att

$$\widehat{\theta}_{(i)} = T(\boldsymbol{P}^0) + \frac{1}{n-1}(\boldsymbol{P}^0 - \boldsymbol{e}_i)^T\boldsymbol{U} + \frac{1}{2}\frac{1}{(n-1)^2}(\boldsymbol{P}^0 - \boldsymbol{e}_i)^T\boldsymbol{V}(\boldsymbol{P}^0 - \boldsymbol{e}_i) =$$

$$= T(\boldsymbol{P}^0) - \frac{U_i}{n-1} + \frac{1}{2(n-1)^2}\left(\boldsymbol{P}^{0T}\boldsymbol{V}\boldsymbol{P}^0 - \boldsymbol{P}^{0T}\boldsymbol{V}\boldsymbol{e}_i - \boldsymbol{e}_i{}^T\boldsymbol{V}\boldsymbol{P}^0 + \boldsymbol{e}_i{}^T\boldsymbol{V}\boldsymbol{e}_i\right) =$$

$$= T(\boldsymbol{P}^0) - \frac{U_i}{n-1} + \frac{1}{2(n-1)^2}V_{ii} = \widehat{\theta} - \frac{U_i}{n-1} + \frac{1}{2(n-1)^2}V_{ii}$$

där vi flera gånger använt oss av att $\sum_i V_{ij} = 0$ och $\sum_j V_{ij} = 0$. Detta ger efter summation över i att

$$\sum_{i=1}^{n} \widehat{\theta}_{(i)} = n\widehat{\theta} + \frac{1}{2(n-1)^2} \sum_{i=1}^{n} V_{ii}.$$

Detta ger

$$\widehat{bias}_{jack} = (n-1)(\widehat{\theta}_{(\cdot)} - \widehat{\theta}) = (n-1)\left(\widehat{\theta} + \frac{1}{2n(n-1)^2}\sum_{i=1}^{n} V_{ii} - \widehat{\theta}\right) =$$

$$= \frac{1}{2n(n-1)} \sum_{i=1}^{n} V_{ii}$$

Å andra sidan får vi

$$\widehat{bias}_{\text{boot}} = E_*\left(T_{QUAD}(\boldsymbol{P}^*)\right) - T_{QUAD}(\boldsymbol{P}^0) =$$

$$= E_*\left((\boldsymbol{P}^* - \boldsymbol{P}^0)^T\boldsymbol{U} + \frac{1}{2}(\boldsymbol{P}^* - \boldsymbol{P}^0)^T\boldsymbol{V}(\boldsymbol{P}^* - \boldsymbol{P}^0)\right) =$$

$$= E_*\left((\boldsymbol{P}^* - \boldsymbol{P}^0)^T\right)\boldsymbol{U} + \frac{1}{2}E_*\left((\boldsymbol{P}^* - \boldsymbol{P}^0)^T\boldsymbol{V}(\boldsymbol{P}^* - \boldsymbol{P}^0)\right) =$$

$$= 0 + \frac{1}{2}E_*\left((\boldsymbol{P}^* - \boldsymbol{P}^0)^T\boldsymbol{V}(\boldsymbol{P}^* - \boldsymbol{P}^0)\right).$$

Vi vet att $\boldsymbol{P}^*$ har väntevärdesvektorn $\boldsymbol{P}^0$ och kovariansmatrisen

$$\frac{\boldsymbol{I}}{n^2} - \frac{(\boldsymbol{P}^0)^T\boldsymbol{P}^0}{n}$$

Om $\boldsymbol{Y}$ har väntevärdesvektor $\boldsymbol{\mu}$ och kovariansmatrisen $\boldsymbol{\Sigma}$ så gäller

$$E(\boldsymbol{Y}^T\boldsymbol{A}\boldsymbol{Y}) = \boldsymbol{\mu}^T\boldsymbol{\Sigma}\boldsymbol{\mu} + \text{trace}(\boldsymbol{\Sigma}\boldsymbol{A})$$

som tillämpat på $\boldsymbol{Y} = \boldsymbol{P}^* - \boldsymbol{P}^0$ ger

$$E_*\left(T_{QUAD}(\boldsymbol{P}^*)\right) - T_{QUAD}(\boldsymbol{P}^0) = \frac{1}{2}\text{trace}\left(\frac{\boldsymbol{I}}{n^2} - \frac{(\boldsymbol{P}^0)^T\boldsymbol{P}^0}{n}\right)\boldsymbol{V} =$$

$$= \frac{1}{2n^2}\sum_{i=1}^{n} V_{ii}$$

Alltså ser vi att detta är (så när som på faktorn $(n-1)/n$) är samma sak som $\widehat{bias}_{jack}$ enligt ovan. □

11.10 En förbättrad bias-skattning

Man kan säga att den nedan beskrivna förbättringen innebär att man kompenserar för att de olika observationerna i $\boldsymbol{x}$ förekommer olika ofta i bootstrap-stickprovet p g a slumpvariationerna i genereringen av bootstrap-stickprovet.

Vi ser fortfarande Bootstrap-skattningarna som en funktion av "återsamplingsvektorn" $\boldsymbol{P}^* = (P_1^*, P_2^*, \cdots, P_n^*)^T$ där P_j^* = andelen som x_j utgör av bootstrap-stickprovet.

Den teoretiska bias-skattningen $\widehat{bias}_{boot} = E(\widehat{\theta}^*) - \widehat{\theta} = E_*(T(\boldsymbol{P}^*)) - T(\boldsymbol{P}^0)$.

Ibland går det att explicit uttrycka en skattning som funktion av återsamplingsvektorn $\boldsymbol{P}^*$.

Exempel 11.3 *Kvotskattning*
Om vi har två-dimensionella data (y_i, z_i), $i = 1, 2, \cdots, n$ och vill skatta $\theta = E(Y)/E(Z)$ så är plug-in-skattningen $\widehat{\theta} = \bar{y}/\bar{z}$. En sådan "kvot-skattning" är typiskt inte väntevärdesriktig. Vi får

$$\widehat{\theta} = \frac{\bar{y}}{\bar{z}} = \frac{\sum_{i=1}^n y_i/n}{\sum_{i=1}^n z_i/n} = \frac{\sum_{i=1}^n P_i^0 y_i}{\sum_{i=1}^n P_i^0 z_i}$$

och alltså

$$\widehat{\theta}^* = \frac{\bar{y}^*}{\bar{z}^*} = \frac{\sum_{i=1}^n P_i^* y_i}{\sum_{i=1}^n P_i^* z_i}$$

□

Mot våra bootstrap-stickprov $\boldsymbol{x}^{*1}, \boldsymbol{x}^{*2}, \cdots, \boldsymbol{x}^{*B}$ svarar återsamplingsvektorer $\boldsymbol{P}^{*1}$, $\boldsymbol{P}^{*2}$, $\cdots$, $\boldsymbol{P}^{*B}$ och vi låter $\bar{\boldsymbol{P}}^*$ vara medelvärdet av dessa, dvs

$$\bar{\boldsymbol{P}}^* = \frac{1}{B}\sum_{b=1}^B \boldsymbol{P}^{*b}$$

Vår tidigare bias-skattning baserad på de B genererade bootstrap-stickproven kan skrivas

$$\widehat{bias}_B = \widehat{\theta}^*(\cdot) - \widehat{\theta}(x_1, x_2, \cdots, x_n) = \widehat{\theta}^*(\cdot) - T(\boldsymbol{P}^0) = \frac{1}{B}\sum_{b=1}^B T(\boldsymbol{P}^{*b}) - T(\boldsymbol{P}^0).$$

En förbättrad bias-skattning är

$$\overline{bias}_B = \widehat{\theta}^*(\cdot) - T(\bar{\boldsymbol{P}}^*) = \frac{1}{B}\sum_{b=1}^B T(\boldsymbol{P}^{*b}) - T\left(\frac{1}{B}\sum_{b=1}^B \boldsymbol{P}^{*b}\right)$$

Kapitel 12

Konfidensintervall – pivotmetoden

12.1 Inledning och översikt

I detta kapitel och kapitel 13 och 14 behandlas tre olika metoder att konstruera konfidensintervall med hjälp av bootstrap-metoder.

1) Pivot-metoden

2) Enkelt percentil-intervall

3) BC_a-metoden – ett förbättrat percentil-intervall

Det råder rätt stor oenighet om vilken av framför allt metod 1) och 3) som är bäst. Generellt kan man säga att Pivot-metoden konstruerar konfidensintervall i enlighet med t-intervallen för väntevärde i en normalfördelning, dvs man konstruerar en variabel av pivot-typ t ex

$$T = \frac{\bar{X} - \theta}{\frac{s(X_1, X_2, \cdots, X_n)}{\sqrt{n}}}$$

som för normalfördelade observationer $X_1, X_2, \cdots, X_n$ är $t(n-1)$-fördelad och utför en bootstrap av denna kvantitet.

Percentil-intervallen tar i stället sin utgångspunkt i det faktum att det ofta finns en s k variansstabiliserande transformation i analogi med Fisher-transformationen för skattningen av korrelationskoefficienten.

Vi återkommer senare med en mer fyllig diskussion om fördelar respektive nackdelar med de olika metoderna.

12.2 Pivot-baserade konfidensintervall

För ett stort antal skattningar $\widehat{\theta}$ gäller att då stickprovsstorleken $n \to \infty$ blir $\widehat{\theta}$ alltmer normalfördelad. Detta gör att

$$\frac{\widehat{\theta} - \theta}{\widehat{se}} \approx N(0,1)$$

och om vi låter z_α lösa ekvationen $\Phi(z_\alpha) = \alpha$ ser vi att

$$P_F\left(z_{\alpha/2} \leq \frac{\widehat{\theta} - \theta}{\widehat{se}} \leq z_{1-\alpha/2}\right) \approx 1 - \alpha$$

som ju kan omformas till

$$P_F\left(\widehat{\theta} - z_{1-\alpha/2}\widehat{se} \leq \theta \leq \widehat{\theta} - z_{\alpha/2}\widehat{se}\right) \approx 1 - \alpha$$

och av detta följer att $(\widehat{\theta}_{\text{undre}}, \widehat{\theta}_{\text{övre}}) = (\widehat{\theta} - z_{1-\alpha/2}\widehat{se}, \widehat{\theta} - z_{\alpha/2}\widehat{se})$ utgör ett konfidensintervall för θ med den approximativa konfidensgraden $1 - \alpha$.

Man kan för övrigt notera att det är högra gränsen $z_{1-\alpha/2}$ i Normalfördelningen som bestämmer vänstra gränsen i konfidensintervallet och tvärtom. P g a symmetrin i normalfördelningen ser man att $z_{1-\alpha/2} = -z_{\alpha/2}$ och med beteckningar från grundkursen att $z_{\alpha/2} = -\lambda_{\alpha/2}$ samt $z_{1-\alpha/2} = \lambda_{\alpha/2}$.

Om man hade haft ett fixt och känt värde se för $D(\widehat{\theta})$ skulle vi ha haft $P_\theta(\theta < \widehat{\theta}_{\text{undre}}) = \alpha/2$ och $P_\theta(\theta > \widehat{\theta}_{\text{övre}}) = \alpha/2$. Om detta är uppfyllt kallar vi konfidensintervallet symmetriskt.

12.2.1 Hypotesprövning – konfidensmetoden

Konfidensintervall hänger ju intimt samman med hypotesprövning, i den meningen att om $\theta = \widehat{\theta}_{\text{undre}}$ så gäller att $P_\theta(\widetilde{\theta} \geq \widehat{\theta}_{obs}) = \alpha/2$ där $\widehat{\theta}_{obs}$ är det observerade värdet av skattningen (dvs det numeriska värdet) och $\widetilde{\theta}$ är $\text{N}(\widehat{\theta}_{\text{undre}}, se^2)$. På motsvarande sätt är $P_\theta(\widetilde{\theta} \geq \widehat{\theta}_{obs}) < \alpha/2$ om $\theta < \widehat{\theta}_{\text{undre}}$. Detta betyder att sannolikheten att få ett så stort värde som $\widehat{\theta}_{obs}$ av skattningen är mindre än $\alpha/2$ om θ ligger till vänster om konfidensintervallet. På motsvarande sätt är värden större än $\widehat{\theta}_{\text{övre}}$ på θ osannolika med tanke på vårt observerade värde på $\widehat{\theta}_{obs}$.

12.2.2 Pivot-variabler

I ovanstående situation utgör

$$Z = \frac{\widehat{\theta} - \theta}{\widehat{se}}$$

en approximativ s k pivot-variabel, dvs fördelningen beror ej av θ.

Med det traditionella (i allmänhet orealistiska) antagandet att våra data kommer från en $N(\theta, \sigma^2)$-fördelning där σ^2 är okänt brukar man ju i stället konstruera konfidensintervall för $\theta = E(X)$ som skattas med $\widehat{\theta} = \bar{x}$ utifrån pivotvariabeln

$$T = \frac{\widehat{\theta} - \theta}{\frac{s}{\sqrt{n}}} = \frac{\widehat{\theta}(X_1, X_2, \cdots, X_n) - \theta}{\frac{s(X_1, X_2, \cdots, X_n)}{\sqrt{n}}}$$

som är $t(n-1)$-fördelad när $\widehat{\theta}(X_1, \cdots, X_n) = \bar{X}$. Här är som vanligt

$$s = s(X_1, X_2, \cdots, X_n) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

Vi får alltså konfidensintervallet

$$\left(\widehat{\theta}_{\text{undre}}(x_1, x_2, \cdots, x_n), \widehat{\theta}_{\text{övre}}(x_1, x_2, \cdots, x_n)\right) =$$

$$= \left(\widehat{\theta} - z_{1-\alpha/2}(n-1)\frac{s}{\sqrt{n}}, \widehat{\theta} - z_{\alpha/2}(n-1)\frac{s}{\sqrt{n}}\right) =$$

$$= \Bigg(\widehat{\theta}(x_1, x_2, \cdots, x_n) - z_{1-\alpha/2}(n-1)\frac{s(x_1, x_2, \cdots, x_n)}{\sqrt{n}},$$

$$\widehat{\theta}(x_1, x_2, \cdots, x_n) - z_{\alpha/2}(n-1)\frac{s(x_1, x_2, \cdots, x_n)}{\sqrt{n}}\Bigg)$$

där percentilerna $z_{\alpha/2}(n-1)$ och $z_{1-\alpha/2}(n-1)$ hämtas från $t(n-1)$-fördelningen i stället för från normalfördelningen $N(0,1)$ som ovan. Detta beror på att $P(z_{\alpha/2}(n-1) < T < z_{1-\alpha/2}(n-1)) = 1 - \alpha$.

Som alltid skall konfidensintervallet

$$\left(\widehat{\theta}_{\text{undre}}(x_1, x_2, \cdots, x_n), \widehat{\theta}_{\text{övre}}(x_1, x_2, \cdots, x_n)\right)$$

uppfattas som ett utfall av det slumpmässiga intervallet

$$\left(\widehat{\theta}_{\text{undre}}(X_1, X_2, \cdots, X_n), \widehat{\theta}_{\text{övre}}(X_1, X_2, \cdots, X_n)\right)$$

och konfidensgraden $1-\alpha$ är ett mått på säkerheten i metoden och inte egentligen en utsaga om det konkreta numeriska intervall vi får med våra data. Det betyder att

$$P\left(\theta \in \left(\widehat{\theta}_{\text{undre}}(X_1, X_2, \cdots, X_n), \widehat{\theta}_{\text{övre}}(X_1, X_2, \cdots, X_n)\right)\right) = 1 - \alpha$$

(för alla värden på θ) som ju är en utsaga om sannolikheten att det slumpmässiga intervallet innehåller parametern θ.

12.3 Bootstrap av pivot-variabler

Man kan säga att den nedan beskrivna bootstrap-metodiken enligt pivotmetoden efterliknar denna konstruktion av konfidensintervall, men där man med hjälp av bootstrap-metodik konstruerar en motsvarighet till "t-fördelningstabellen" som är skräddarsydd för just den datauppsättning man fått. Vad man får som konfidensintervall med pivot-metoden är precis samma intervall som man skulle få om pivot-variabelns fördelning var känd men med den modifikationen att percentilerna skattas med hjälp av bootstrap.

Detta innebär att man som approximativ pivotvariabel tar

$$\frac{\widehat{\theta}-\theta}{\widehat{\tau}}$$

för en lämpligt vald skattning $\widehat{\tau}=\widehat{se}$ av $D(\widehat{\theta})$. Vad detta innebär är att vi antar att

$$P\left(\frac{\widehat{\theta}-\theta}{\widehat{\tau}}\leq z\right)=P\left(\frac{\widehat{\theta}(X_1,X_2,\cdots,X_n)-\theta}{\widehat{\tau}(X_1,X_2,\cdots,X_n)}\leq z\right)=G(z)$$

för någon funktion G. Kände vi denna funktion så skulle vi välja tal $z_{\alpha/2}$ och $z_{1-\alpha/2}$ sådana att $G(z_{\alpha/2})=\alpha/2$ och $G(z_{1-\alpha/2})=1-\alpha/2$ och konstruera konfidensintervallet

$$(\widehat{\theta}-z_{1-\alpha/2}\widehat{\tau},\widehat{\theta}-z_{\alpha/2}\widehat{\tau})=$$

$$=\Big(\widehat{\theta}(x_1,x_2,\cdots,x_n)-z_{1-\alpha/2}\widehat{\tau}(x_1,x_2,\cdots,x_n),$$

$$\widehat{\theta}(x_1,x_2,\cdots,x_n)-z_{\alpha/2}\widehat{\tau}(x_1,x_2,\cdots,x_n)\Big)$$

precis som i konstruktionen av t-intervallet ovan.

I situationen med ett stickprov från $N(\theta,\sigma^2)$-fördelningen och med $\widehat{\theta}=\bar{X}$ och

$$\widehat{\tau}=\frac{s}{\sqrt{n}}=\sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(X_i-\bar{X})^2}$$

är $G(\cdot)$ helt enkelt $t(n-1)$-fördelningens fördelningsfunktion.

Vi känner i allmänhet inte funktionen G och därmed inte heller $z_{\alpha/2}$ och $z_{1-\alpha/2}$. Dock kan vi erhålla skattningar av dessa genom bootstrap-metodik genom att vi ser $G(z)$ som en (komplicerad) funktional $S(F)$ av F. Plug-in-skattningen av $S(F)$ är $S(\widehat{F})$. Detta betyder att vi skattar $G(z)$ med

$$G^*(z)=S(\widehat{F})=P\left(\frac{\widehat{\theta}^*-\widehat{\theta}}{\widehat{\tau}^*}\leq z\right)=$$

$$=P\left(\frac{\widehat{\theta}(X_1^*,X_2^*,\cdots,X_n^*)-\widehat{\theta}(x_1,x_2,\cdots,x_n)}{\widehat{\tau}(X_1^*,X_2^*,\cdots,X_n^*)}\leq z\right)$$

där $\widehat{\theta}^*$ är bootstrap-varianten av $\widehat{\theta}$ och på samma sätt $\widehat{\tau}^*$ är motsvarigheten till $\widehat{\tau}$ fast applicerad på bootstrap-variablerna $\boldsymbol{X}^*$ i stället för på det ursprungliga variablerna $\boldsymbol{X}$.

$G^*(z)$ kan ju i allmänhet inte beräknas exakt, men vi får en skattning av den genom att skaffa oss B st bootstrapstickprov $\boldsymbol{x}^{*1}, \boldsymbol{x}^{*2}, \cdots, \boldsymbol{x}^{*B}$. Återigen är det så att $G^*(z)$ i princip kan beräknas hur noga som helst genom att låta $B \to \infty$.

Vi låter $\widehat{\theta}^*(b)$ och $\widehat{\tau}^*(b)$ vara skattningarna ur stickprovet $\boldsymbol{x}^{b*}$, dvs vi låter $\widehat{\theta}^*(b) = \widehat{\theta}(\boldsymbol{x}^{*b})$ och $\widehat{\tau}^*(b) = \widehat{\tau}(\boldsymbol{x}^{*b})$ för $b = 1, 2, \cdots, B$.

Om vi låter

$$Z^*(b) = \frac{\widehat{\theta}^*(b) - \widehat{\theta}}{\widehat{\tau}^*(b)}$$

så skattar vi $G^*(z)$ med den empiriska fördelningen för de B värden på Z^* som erhållits med bootstrap, dvs vi skattar $G^*(z)$ med $G^*_B(z)$ där

$$G^*_B(z) = \frac{\text{Antal } Z^*(b) \leq z}{B}$$

och alltså får vi t ex $z^*_{\alpha/2}$-skattningen genom att lösa $G^*_B(z^*_{\alpha/2,B}) = \alpha/2$, dvs vi löser ut $z^*_{\alpha/2,B}$ ur ekvationen

$$\alpha/2 = \frac{\text{Antal } Z^*(b) \leq z^*_{\alpha/2,B}}{B}$$

Vårt konfidensintervall $(\widehat{\theta} - z_{1-\alpha}\widehat{\tau}, \widehat{\theta} - z_\alpha\widehat{\tau})$ skattas alltså med

$$(\widehat{\theta} - z^*_{1-\alpha,B}\widehat{\tau}, \widehat{\theta} - z^*_{\alpha,B}\widehat{\tau}).$$

I praktiken tar vi $z^*_{\alpha/2,B} = k$:te i storleksordning av $Z^*(1), Z^*(2), \cdots, Z^*(B)$ där $k = [(B+1)\alpha/2]$ och $z_{1-\alpha/2,B}$ som nr $[(B+1)(1-\alpha/2)]$ i storleksordning.

Exempelvis tar vi med $\alpha = 0.10$ och $B = 999$ $k = 50$, dvs storheten $z^*_{0.05}$ approximeras med den 50:e i storleksordning av $Z^*(1), Z^*(2), \cdots, Z^*(999)$ och $z^*_{0.95}$ med den 950:e.

12.3.1 Sammanfattning

Man simulerar alltså fördelningen för $(\widehat{\theta}(\boldsymbol{X}^*) - \widehat{\theta}(\boldsymbol{x}))/\widehat{\tau}(\boldsymbol{X}^*)$ och skattar percentilerna ur denna fördelning. Dessa percentiler utgör skattningar av motsvarande percentiler i fördelningen för $(\widehat{\theta}(\boldsymbol{X}) - \theta)/\widehat{\tau}(\boldsymbol{X})$. Det är just dessa percentiler som ingår i konfidensintervallet

$$\left(\widehat{\theta}(\boldsymbol{x}) - z_{1-\alpha/2}\widehat{\tau}(\boldsymbol{x}), \widehat{\theta}(\boldsymbol{x}) - z_{\alpha/2}\widehat{\tau}(\boldsymbol{x})\right).$$

Pivot-metoden innebär alltså att vi skattar percentilerna i pivot-variabelns fördelning. Det är detta som avses med att pivot-metoden beräknar en t-fördelning skräddarsydd för just det stickprov som vi har erhållit. I övrigt ser konfidensintervallet ut precis som i normalfördelningsfallet.

12.3.2 Pivot-metoden i Matlab

I Matlab skriver man först en `m`-fil som ger pivot-variabeln och avsikten är att `bootstrp` skall skicka bootstrap-stickproven till denna rutin. Man får då tänka sig lite för så att rutinen fungerar på rätt sätt.

Exempel 12.1 *$(\bar{X}-\theta)/(s/\sqrt{n})$ som pivot-variabel*
Antag att vi har ett stickprov $x_1, x_2, \cdots, x_n$ lagrat i vektorn `x` och vill skatta θ med $\bar{x}$ och vill studera pivot-variabeln

$$T = \frac{\bar{X}-\theta}{\frac{s(X_1, X_2, \cdots, X_n)}{\sqrt{n}}}$$

där som vanligt

$$s(X_1, X_2, \cdots, X_n) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i-\bar{X})^2}.$$

Om data kommer från en $N(\theta, \sigma^2)$-fördelning är alltså T en $t(n-1)$-fördelad variabel. Vi vill alltså göra bootstrap av T och får

$$T^* = \frac{\bar{X}^*-\bar{x}}{\frac{s(X_1^*, X_2^*, \cdots, X_n^*)}{\sqrt{n}}}$$

och skriver `m`-filen `pivot` enligt följande

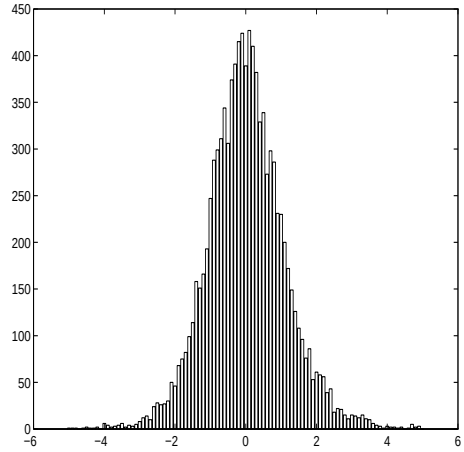
```
function t=pivot(x,xbar)
n=length(x);
t=(mean(x)-xbar)/(std(x)/sqrt(n));
```

där vi alltså skickar med `xbar=mean(x)` som en parameter till `pivot`. Detta värde `xbar`=$\bar{x}$ ska ju räknas ut i det ursprungliga stickprovet och kan alltså inte beräknas i `m`-filen `pivot` som i rutinen `bootstrp` kommar att operera på de framlottade bootstrap-stickprovet $(x_1^*, x_2^*, \cdots, x_n^*)$. Operationen `mean(x)` i `pivot` kommer alltså att beräkna $\bar{x}^*$. Ännu bättre vore kanske att skicka med även storleken `n` av vektorn `x` så att man inte behöver ta reda på storleken i varje bootstrap-stickprov. I just detta fall kan man faktiskt avvakta med multiplikationen med $\sqrt{n}$ till efter man skapat vektorn `boot`. Allmänt går inte detta.

Vi skapar filen

```
function t=pivot(x,xbar,n)
t=(mean(x)-xbar)/(std(x)/sqrt(n));
```

En simulering av fördelningen för T^* får då Matlab-koden



Figur 12.1: Simulerad fördelning för $(\bar{x}^* - \bar{x})/(s/\sqrt{10})$ för 10 $N(1, 2^2)$-fördelade.

```
xbar=mean(x);
n=length(x);
boot=bootstrp(B,'pivot',x,xbar,n) ;
```

och vi får sen percentiler genom `prctile` men man kan också sortera `boot` och ta rätt element i den sorterade vektorn. Om vi söker 5%:s och 95%-percentilen och valt $B = 999$ tar vi 50:de och 950:de i storleksordning från `boot`. Med `prctile` blir koden `[lower upper]=prctile(boot,[5 95])`. I den följande Matlab-utskriften genereras alltså först 10 st $N(1, 2^2)$-fördelade observationer varefter fördelningen för T^* simuleras. Den sanna fördelningen blir alltså en $t(9)$-fördelning.

```
 x=normrnd(1,2,10,1);
 xbar=mean(x)      % gav 0.7950
 n=max(size(x))    % gav 10
 boot=bootstrp(9999,'pivot',x,xbar,n);
 p=[1 2.5 5 95 97.5 99]';
 perc=prctile(boot,p)
perc =
   -2.6421   -2.1111   -1.7117    1.9612    2.4484    3.4028
 p=p/100;
 a=tinv(p,9)
a =
   -2.8214   -2.2622   -1.8331    1.8331    2.2622    2.8214
 hist(boot,100)
```

som ger figur 12.1. Man ser också hur bootstrap skattar de sanna värdena på ett antal percentiler i följande tabell.

Percentil	0.1	0.025	0.05	0.95	0.975	0.99
$t(9)$	-2.8214	-2.2622	-1.8331	1.8331	2.2622	2.8214
Bootstrap	-2.6421	-2.1111	-1.7117	1.9612	2.4484	3.4028

□

12.4 Problem med pivot-metoden

Efron är inte speciellt förtjust i pivot-baserade konfidensintervall utan hans favorit är de modifierade percentilintervallen som beskrivs i kapitel 14. Hans huvudinvändningar mot de pivot-baserade intervallen är

1) Det kan vara svårt att skaffa den nödvändiga medelfelsskattningen i nämnaren av pivot-variabeln.

2) De inte är transformationsbevarande

3) De behöver inte uppfylla krav på begränsningar av parameterns värde av typen $\theta \geq 0$ eller $-1 \leq \theta \leq 1$.

12.4.1 Hur få medelfelsskattning?

Vi behöver en medelfelsskattning i nämnaren i

$$Z^*(b) = \frac{\widehat{\theta}^*(b) - \widehat{\theta}}{\widehat{\tau}^*(b)}$$

som kan beräknas för bootstrap-stickprovet. Dessutom måste $\widehat{\tau}$ vara så konstruerad att vi verkligen får en (approximativ) pivot-variabel, dvs att fördelningen för $(\widehat{\theta} - \theta)/\widehat{\tau}$ ej beror av några parametrar.

Om man inte lätt kan konstruera en sådan medelfelsskattning så är en möjlighet att få fram den att utföra en bootstrap (eller jackknife) av varje enskilt bootstrap-stickprov. Även om medelfelsskattningar inte kräver så många bootstrap-stickprov ($B = 25$ räcker ofta) så måste denna operation då göras för vart och ett av de kanske 1000 bootstrap-stickproven. För små stickprovsstorlekar blir resultatet ibland rätt erratiskt. Vi skall ju utifrån ett bootstrapstickprov som kanske innehåller bara några få distinkta värden utföra ytterligare en bootstrap som kanske ytterligare "glesar ut" stickprovet.

Just för $\widehat{\theta} = \bar{x}$ finns en naturlig medelfelsskattning nämligen $s/\sqrt{n}$. På liknande sätt finns naturliga medelfelsskattningar i stuationen med två oberoende stickprov $x_1, x_2, \cdots, x_n$ och $y_1, y_2, \cdots, y_m$ där vi försöker skatta skillnaden i väntevärden mellan de bakomliggande fördelningarna. Om observationerna kommer från $\mathrm{N}(\theta_1, \sigma_1^2)$- respektive $\mathrm{N}(\theta_2, \sigma_2^2)$-fördelade variabler $X_1, X_2, \cdots, X_n$ respektive $Y_1, Y_2, \cdots, Y_m$. Då skattar vi ju θ_1 med $\bar{x}$, θ_2 med $\bar{y}$, σ_1 med s_x där

$$s_x^2 = s_x^2(\boldsymbol{x}) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

och på samma sätt σ_2 med s_y där

$$s_y^2 = s_y^2(\boldsymbol{y}) = \frac{1}{m-1}\sum_{i=1}^{m}(y_i - \bar{y})^2.$$

Då borde

$$T = \frac{\bar{X} - \bar{Y} - (\theta_1 - \theta_2)}{\sqrt{\frac{s_x^2(\boldsymbol{X})}{n} + \frac{s_y^2(\boldsymbol{Y})}{m}}}$$

vara en approximativ pivot-variabel och det är denna vi försöka göra bootstrap av. Detta betyder att vi med bootstrap-simulering försöker få fram fördelningen för

$$T^* = \frac{\bar{X}^* - \bar{Y}^* - (\bar{x} - \bar{y})}{\sqrt{\frac{(s_x^*)^2}{n} + \frac{(s_y^*)^2}{m}}}.$$

I ovanstående uttryck skall alltså även medelfelsskattningen i nämnaren beräknas för bootstrap-stickprovet.

För andra skattningar är det väsentligt mer komplicerat att få fram en medelfelsskattning. Detta gäller t ex för korrelations-skattningen

$$\widehat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}.$$

12.4.2 Ej transformationsbevarande

Den andra invändningen Efron har är att pivot-intervallen inte är transformationsbevarande. Det spelar alltså roll om vi försöker göra ett konfidensintervall för parametern θ eller för $\psi = m(\theta)$ där $m(\cdot)$ är en monoton transformation av parametern. Att en metod är transformationsbevarande är ju en trevlig egenskap, eftersom parametriseringen ofta är rätt godtycklig. T ex borde det ju inte spela någon roll om vi betraktar θ eller $1/\theta$ som parameter – våra slutsatser från konfidensintervall vid t ex hypotesprövning borde vara den samma oavsett parametriseringen.

12.4.3 Restriktioner på parametern

Som exempel på den tredje invändningen kan man ta skattningen av korrelationskoefficienten ρ. Där kan ett pivot-baserat intervall lätt hamna utanför intervallet $[-1, 1]$. Vi stötte på samma fenomen vid användningen av CGS på livslängdsdata där konfidensintervall kunde ha ha gränser < 0, vilket är otillfredsställande.

12.4.4 Variansstabilisering

Ofta finns en "variansstabiliserande" transformation i stil med

$$\psi = \frac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right)$$

för korrelationskoefficienten. Om vi försöker göra ett konfidensintervall för ψ så vet vi att

$$\widehat{\psi} - \psi \approx N\left(0, \frac{1}{n-3}\right)$$

så problemet med skevhet försvinner i stort sett om vi väljer denna transformation av parametern ρ. I och för sig gäller detta resultat egentligen bara för två-dimensionellt normalfördelade data, men den gäller även för en större klass av två-dimensionella fördelningar.

Följande utgör ett alternativ att uppnå denna variansstabilisering i praktiken: Låt X ha väntevärdet $E(X) = \theta$ och standardavvikelsen $s(\theta)$. Om vi betraktar $g(X)$ så gäller enligt Gauss' approximationsformler att

$$V(g(X)) \approx (g'(\theta))^2 V(X)$$

och alltså gäller att om vi tar $g(x)$ till en primitiv funktion till $1/s(x)$ t ex

$$g(x) = \int_0^x \frac{1}{s(u)} du$$

att $V(g(X)) \approx$ konstant eftersom $V(X) = s^2(\theta)$.

Detta betyder att vi kan hitta en approximativt variansstabiliserande transformation genom att ta $g(x)$ som en primitiv funktion till $1/s(x)$. Nu är ju funktionen $s(x)$ inte känd, men den kan naturligtvis approximeras.

Ett förslag som Efron ger för att hitta en approximation till g är följande algoritm:

1) Generera B_1 st bootstrap-stickprov $\boldsymbol{x}^{*b}$, $b = 1, 2, \cdots, B_1$

2) Gör B_2 st bootstrappningar av vardera av dessa B_1 och skatta därigenom $\widehat{se}(\widehat{\theta}^*(b))$ för vart och ett av de B_1 bootstrap-stickproven.

3) Anpassa en kurva till talparen $\left(\widehat{\theta}^*(b), \widehat{se}(\widehat{\theta}^*(b))\right)$ dvs konstruera en skattning av $s(u) = se(\widehat{\theta}|\theta = u)$.

4) Beräkna genom integration (kanske numerisk)

$$g(x) = \int_0^x \frac{1}{s(u)} du$$

5) Generera B_3 st nya bootstrap-stickprov och gör konfidensinterval för $\psi = g(\theta)$, dvs tag $\widehat{\psi} - \psi$ som pivot-variabel. Alltså studerar vi fördelningen för $\widehat{\psi}^* - \widehat{\psi}$ och konstruerar ur denna ett konfidensintervall för ψ.

6) Transformera detta intervall genom g^{-1} tillbaka till θ.

Man inser att detta program är rätt besvärligt att genomföra och motivera i en konkret situation. Pivot-metoden lämpar sig alltså bäst i situationer där det finns en "naturlig" medelfelsskattning.

Man kan dock helt bortse från nämnaren och i stället använda $\widehat{\theta} - \theta$ som pivot-variabel. Detta förfarande ger en korrekt korrigering för t ex bristande väntevärdesriktighet och för skevhet i fördelningen för $\widehat{\theta}$ och är nog att föredra framför de enkla percentil-intervallen som beskrivs i kapitel 13. Dock har detta förfarande sämre egenskaper än BC_a-metoden (beskriven i kapitel 14) som korrigerar de enkla percentil-intervallen och som beskrivs i kapitel 13.

12.5 Förenklad pivot-variabel

Om vi baserar vårt konfidensintervall på $\widehat{\theta} - \theta$ och approximerar dennas fördelning med bootstrap-fördelningen för $\widehat{\theta}^* - \widehat{\theta}$ så får vi intervallet

$$[2\widehat{\theta} - \widehat{G}^{-1}(1-\alpha/2), 2\widehat{\theta} - \widehat{G}^{-1}(\alpha/2)]$$

där $\widehat{G}(z) = P(\widehat{\theta}^* \leq z)$ dvs bootstrapfördelningen. Denna skattas direkt ur våra bootstrap-skattningar och även percentilerna erhålls lätt.

Detta inses ur följande resonemang:

Om vi kallar $P(\widehat{\theta} - \theta \leq z) = H(z)$ så har vi

$$1 - \alpha = P\left(H^{-1}(\alpha/2) \leq \widehat{\theta} - \theta \leq H^{-1}(1-\alpha/2)\right) =$$

$$= P\left(\widehat{\theta} - H^{-1}(1-\alpha/2) \leq \theta \leq \widehat{\theta} - H^{-1}(\alpha/2)\right)$$

och konfidensintervallet skall alltså vara

$$[\widehat{\theta} - H^{-1}(1-\alpha/2), \widehat{\theta} - H^{-1}(\alpha/2)].$$

Vi låter vidare $P(\widehat{\theta}^* - \widehat{\theta} \leq z) = \widehat{H}(z)$ och approximerar $H(z)$ med $\widehat{H}(z)$ och alltså approximerar vi även percentilerna med varandra. Vi har då

$$\widehat{H}(z) = P(\widehat{\theta}^* - \widehat{\theta} \leq z) = P(\widehat{\theta}^* \leq \widehat{\theta} + z) = \widehat{G}(\widehat{\theta} + z)$$

där $\widehat{G}(u) = P(\widehat{\theta}^* \leq u)$ är fördelningen för bootstrap-skattningen $\widehat{\theta}^*$ som vi ju kan approximera med våra framlottade bootstrap-stickprov.

Vi får då eftersom $\widehat{H}(z) = \widehat{G}(\widehat{\theta} + z)$ att om $z = \widehat{H}^{-1}(\alpha/2)$ så är

$$\widehat{G}^{-1}(\alpha/2) = \widehat{\theta} + z = \widehat{\theta} + \widehat{H}^{-1}(\alpha/2)$$

dvs att $\widehat{H}^{-1}(\alpha/2) = \widehat{G}^{-1}(\alpha/2) - \widehat{\theta}$ och på samma sätt

$$\widehat{H}^{-1}(1-\alpha/2) = \widehat{G}^{-1}(1-\alpha/2) - \widehat{\theta}$$

som insatt i intervallet ovan ger $[2\widehat{\theta} - \widehat{G}^{-1}(1-\alpha/2), 2\widehat{\theta} - \widehat{G}^{-1}(\alpha/2)]$ då vi approximerar $H^{-1}(\alpha/2)$ med $\widehat{H}^{-1}(\alpha/2)$ och $H^{-1}(1-\alpha/2)$ med $\widehat{H}^{-1}(1-\alpha/2)$.

12.6 Andra pivot-variabler

Vi antar att data kommer från en skalinvariant familj, dvs vi antar att X_i/θ har en fördelning som ej beror av θ. Ett sådant exempel stötte vi på i exemplet med livslängderna där vi då kunde frigöra oss från antagandet att observationerna kom från en exponentialfördelning genom att anta att $P(X_i \leq x) = H(x/\theta)$ för någon fördelningsfunktion H. Exempelvis skulle man kunna ha

$$H(z) = 1 - e^{-z^a}, \ z \geq 0$$

dvs en Weibull-fördelning med formparameter $a > 0$. exponentialfördelningen svarar mot $a = 1$.

Då vi försöker skatta θ med $\bar{x}$ ser vi att $\bar{X}/\theta$ utgör en pivot-variabel och vi gör alltså bootstrap av denna, dvs av $\bar{X}^*/\bar{x}$. Återigen behöver vi inte beräkna dennas exakta fördelning utan vi simulerar den. Vi genererar alltså B stickprov $\boldsymbol{x}^{1*}, \boldsymbol{x}^{2*}, \cdots, \boldsymbol{x}^{B*}$ vardera bestående av n observationer erhållna med hjälp av dragning med återläggning från $x_1, x_2, \cdots, x_n$. I vardera av dessa B stickprov beräknar vi det aritmetiska medelvärdet och erhåller då medelvärdena $\bar{x}^{1*}, \bar{x}^{2*}, \cdots, \bar{x}^{B*}$ och bildar

$$t^{b*} = \frac{\bar{x}^{b*}}{\bar{x}}, \ b = 1, 2, \cdots, B.$$

Den observerade fördelningen för dessa $t^{1*}, t^{2*}, \cdots, t^{B*}$ kallar vi G_B^* samt tar $\alpha/2-$ respektive $(1 - \alpha/2)$-percentilerna $z^*_{\alpha/2,B}$ respektive $z^*_{1-\alpha/2,B}$ i denna fördelning genom att i princip lösa ekvationerna

$$G_B^*(z^*_{\alpha/2,B}) = \alpha/2 \text{ och } G_B^*(z^*_{1-\alpha/2,B}) = 1 - \alpha/2.$$

I praktiken alltså genom att ta den $[(B+1)\alpha/2]$-te och $[(B+1)(1-\alpha/2)]$-te i storleksordning av $t^{*1}, t^{*2}, \cdots, t^{*B}$. T ex med $B = 999$ och $\alpha = 0.10$ tar vi den 50:de och 950:e i storleksordning av de 999 värdena.

När vi så erhållit dessa percentiler använder vi dem för att skatta de sanna percentilerna $z_{\alpha/2}$ och $z_{1-\alpha/2}$ i fördelningen för $\bar{X}/\theta$.

Slutligen får vi då konfidensintervallet

$$\left(\frac{\bar{x}}{z^*_{1-\alpha/2,B}}, \frac{\bar{x}}{z^*_{\alpha/2,B}} \right)$$

helt i analogi med hur vi beräknade konfidensintervallet i avsnitt 3.6.3 på sidan 34 då vi visste att observationerna kom från Exp(θ)-fördelningen. Det var också detta som gjordes i kapitel 6 på sidan 75. Där undersökte vi också rätt ingående hur denna typ av intervall fungerade empiriskt.

Kapitel 13

Konfidensintervall -percentilmetoden

Vi har ett bootstrap-stickprov $\boldsymbol{X}^* = (X_1^*, X_2^*, \cdots, X_n^*)$ och bootstrap-variabeln $\widehat{\theta}^* = \widehat{\theta}(\boldsymbol{X}^*)$. Om vi låter $\widehat{G}$ vara fördelningsfunktion för $\widehat{\theta}^*$ så definieras percentil-intervallet med konfidensgrad $1-\alpha$ som

$$[\widehat{\theta}_{\%,\text{undre}}, \widehat{\theta}_{\%,\text{övre}}] = [\widehat{G}^{-1}(\alpha/2), \widehat{G}^{-1}(1-\alpha/2)].$$

Om vi låter $\widehat{G}(\widehat{\theta}^{*(\beta)}) = \beta$ så kan intervallet också skrivas

$$[\widehat{\theta}_{\%,\text{undre}}, \widehat{\theta}_{\%,\text{övre}}] = [\widehat{\theta}^{*(\alpha/2)}, \widehat{\theta}^{*(1-\alpha/2)}]$$

eftersom $\widehat{G}^{-1}(\alpha/2) = \widehat{\theta}^{*(\alpha/2)}$ eller ekvivalent $\widehat{G}(\widehat{\theta}^{*(\alpha/2)}) = \alpha/2$.

Detta innebär alltså att man som konfidensgränser tar $\alpha/2$ respektive $(1-\alpha/2)$-percentilerna i bootstrap-fördelningen, dvs fördelningen för $\widehat{\theta}^*$. Detta avser den "ideala" bootstrap-fördelningen, men i praktiken skattas ju dessa percentiler genom att vi gör B st bootstrap-stickprov $\boldsymbol{x}^{*1}, \boldsymbol{x}^{*2}, \cdots, \boldsymbol{x}^{*B}$, beräknar motsvarande skattningar $\widehat{\theta}^*(1), \widehat{\theta}^*(2), \cdots, \widehat{\theta}^*(B)$ och skattar $\alpha/2$- respektive $1-\alpha/2$-percentilerna ur detta material.

Alltså blir det approximativa percentil-intervallet med konfidensgrad $1-\alpha$

$$[\widehat{\theta}_B^{*(\alpha/2)}, \widehat{\theta}_B^{*(1-\alpha/2)}]$$

där index B indikerar att vi fått percentilerna ur de B bootstrap-stickproven och inte ur den verkliga bootstrap-fördelningen.

Vad Efron argumenterar för är att detta intervall (som förfinas i kapitel 14 för att ta hänsyn till bias och skevhet) har bra egenskaper. Speciellt är det transformationsinvariant dvs inget konstigt händer med intervallet om vi i stället för att studera parametern θ tar en monoton transformation av den $\psi = m(\theta)$, t ex θ^2 eller $1/\theta$ utan intervallgränserna transformeras enligt funktionen m. Eftersom vi jobbar med plug-in-skattningar så blir bootstrap-fördelningen för $\widehat{\psi}^*$ bara en transformation av fördelningsfunktion $\widehat{G}$ för $\widehat{\theta}^*$.

13.1 Transformations-egenskaper

Sats 13.1 *Percentil-intervall är transformationsbevarande*
Om $\psi = m(\theta)$ för en monoton (växande) transformation m så blir percentil-intervallet för ψ helt enkelt percentil-intervallet för θ transformerat med $m(\cdot)$-funktionen dvs

$$[\widehat{\psi}_{\%,\text{undre}}, \widehat{\psi}_{\%,\text{övre}}] = [m(\widehat{\theta}_{\%,\text{undre}}), m(\widehat{\theta}_{\%,\text{övre}})]$$

dvs percentil-intervallet är transformationsbevarande. □

Bevis:
Notera först att om $\widehat{\theta}$ är en plug-in-skattning så är plug-in-skattningen av $\psi = m(\theta)$ då $\widehat{\psi} = m(\widehat{\theta})$. Detta betyder att $\widehat{\psi}^* = m(\widehat{\theta}^*)$. Om vi låter $\widehat{G}(z) = P(\widehat{\theta}^* \leq z)$ och $\widehat{H}(z) = P(\widehat{\psi}^* \leq z)$ så ser vi att

$$\widehat{H}(z) = P(\widehat{\psi}^* \leq z) = P(m(\widehat{\theta}^*) \leq z) =$$

$$= P(\widehat{\theta}^* \leq m^{-1}(z)) = \widehat{G}(m^{-1}(z)).$$

Då ser vi att $\alpha/2 = \widehat{H}(\widehat{H}^{-1}(\alpha/2))$ innebär att $\widehat{G}\left(m^{-1}(\widehat{H}^{-1}(\alpha/2))\right) = \alpha/2$ dvs att $m^{-1}(\widehat{H}^{-1}(\alpha/2)) = \widehat{G}^{-1}(\alpha/2)$ som ju ger att

$$\widehat{H}^{-1}(\alpha/2) = m(\widehat{G}^{-1}(\alpha/2))$$

och eftersom

$$[\widehat{H}^{-1}(\alpha/2), \widehat{H}^{-1}(1-\alpha/2)]$$

är percentil-intervallet för ψ följer resultatet. Skulle m vara en monotont avtagande funktion kastas gränserna om på naturligt sätt. □

13.1.1 Percentil-interval-lemmat

Detta betyder att vi kan bevisa Percentil-intervall-lemmat:

Sats 13.2 *Percentil-intervall-lemmat*
Om $\widehat{\psi} = m(\widehat{\theta})$ är en variansstabiliserande transformation som gör att $\widehat{\psi}$ är $N(\psi, c^2)$ för något c så blir percentil-intervallet för θ baserat på $\widehat{\theta}$

$$[m^{-1}(\widehat{\psi} - z_{1-\alpha/2}c), m^{-1}(\widehat{\psi} - z_{\alpha/2}c)]$$

Alltså betyder det att vi får percentil-intervallet för $\theta = m^{-1}(\psi)$ genom att applicera m^{-1} på det traditionella intervallet för ψ, dvs intervallgränserna transformeras på naturligt sätt. □

Bevis:
Vi skall alltså visa att om $\widehat{G}(z) = P(\widehat{\theta}^* \leq z)$ så är

$$\widehat{G}^{-1}(\alpha/2) = m^{-1}(\widehat{\psi} - z_{1-\alpha/2}c) \text{ och } \widehat{G}^{-1}(1-\alpha/2) = m^{-1}(\widehat{\psi} - z_{\alpha/2}c)$$

eftersom dessa utgör percentil-intervallgränserna för θ.

Vi har

$$\alpha/2 = P\left(\frac{\widehat{\psi} - \psi}{c} \leq z_{\alpha/2}\right)$$

eftersom $(\widehat{\psi} - \psi)/c$ är $\mathrm{N}(0,1)$. Men vi tror att enligt den grundläggande bootstrap-idén har $(\widehat{\psi}^* - \widehat{\psi})/c$ samma fördelning dvs att

$$\alpha/2 = P\left(\frac{\widehat{\psi}^* - \widehat{\psi}}{c} \leq z_{\alpha/2}\right) = P(\widehat{\psi}^* \leq \widehat{\psi} + cz_{\alpha/2}) =$$

$$= P(m(\widehat{\theta}^*) \leq \widehat{\psi} + cz_{\alpha/2}) = P\left(\widehat{\theta}^* \leq m^{-1}(\widehat{\psi} + cz_{\alpha/2})\right) =$$

$$= \widehat{G}\left(m^{-1}(\widehat{\psi} + cz_{\alpha/2})\right).$$

Men detta argument måste då vara just $\widehat{G}^{-1}(\alpha/2)$ som ger

$$\widehat{G}^{-1}(\alpha/2) = m^{-1}(\widehat{\psi} + cz_{\alpha/2}) = m^{-1}(\widehat{\psi} - cz_{1-\alpha/2})$$

eftersom $z_{\alpha/2} = -z_{1-\alpha/2}$. Den andra halvan visas på likartat sätt.

Detta betyder att det approximativa percentil-intervallet är transformationsbevarande, dvs att gränserna transformeras med funktionen m.

Om vi har $\widehat{\theta}^*(1), \widehat{\theta}^*(2), \cdots, \widehat{\theta}^*(B)$ så blir

$$\widehat{\psi}^*(1), \widehat{\psi}^*(2), \cdots, \widehat{\psi}^*(B) = m(\widehat{\theta}^*(1)), m(\widehat{\theta}^*(2)), \cdots, m(\widehat{\theta}^*(B))$$

så vi ser att percentilerna uppfyller $\widehat{\psi}_B^{*(\alpha/2)} = m(\widehat{\theta}_B^{*(\alpha/2)})$ och $\widehat{\psi}_B^{*(1-\alpha/2)} = m(\widehat{\theta}_B^{*(1-\alpha/2)})$ om m är växande. Fallet då m är avtagande behandlas helt analogt. Notera att detta innebär att vi inte behöver veta transformationen $m(\cdot)$ som variansstabiliserar eller känna värdet på c. Vad vi gör är att ta $\alpha/2$ respektive $1-\alpha/2$-percentilerna i fördelningen för $\widehat{\theta}^*$ som vi kan approximera med motsvarande storheter ur våra simuleringar. Det centrala är att det existerar en variansstabiliserande transformation som ger upphov till en symmetrisk fördelning. □

13.2 Pivot-baserade eller percentil-interval?

Om fördelningen för bootstrap-fördelningen är skev t ex uppåt så förlängs de percentil-baserade uppåt, medan de pivot-baserade intervallen förlängs nedåt. Detta är föremål för en intensiv debatt i bootstrap-kretsar. Man kan t ex

studera artikeln "Theoretical comparison of bootstrap confidence intervals" av Peter Hall i Annals of Statistics Vol.16, No 3 – September 1988 som följs av en diskussion med inlägg bl a av Efron och där tonen är ovanligt hätsk för att vara en vetenskaplig diskussion.

En fördel som de pivot-baserade metoderna har är att de automatiskt kompenserar för bristande väntevärdesriktighet i den ursprungliga skattningen $\widehat{\theta}$ genom att man approximerar fördelningen för $\widehat{\theta}-\theta$ med bootstrap-fördelningen $\widehat{\theta}^* - \widehat{\theta}$. Skattningen av bias för $\widehat{\theta}$ erhålls alltså som $\widehat{\theta}^*(\cdot) - \widehat{\theta}$= medelvärdet av bootstrapskattningarna – den ursprungliga skattningen. Percentilmetoden klarar inte alls av en bias i skattningen, eftersom om $\widehat{\theta}$ i genomsnitt överskattar θ så kommer fördelningen för $\widehat{\theta}^*$ att vara förskjuten uppåt i förhållande till $\widehat{\theta}$ och alltså förskjuts percentilerna i fördelningen för $\widehat{\theta}^*$ uppåt och hela intervallet förskjuts uppåt.

På samma sätt verkar pivot-metoden behandla en skevhet i fördelningen för $\widehat{\theta}$ på ett naturligare sätt. Är fördelningen skev uppåt så betyder ju det att ett förhållandevis stort antal utfall av skattningen kommer att ge för stora värden i förhållande till θ. Eftersom fördelningen för $\widehat{\theta}^*$ approximerar fördelningen för $\widehat{\theta}$ så verkar det i denna situation naturligt att dra ut konfidensintervallet nedåt eftersom vår skattning ju tenderat att bli för stor. Percentil-intervallet förlänger i stället intervallet uppåt.

I kapitel 14 modifierar vi percentil-intervallen (den s k BC_a-metoden) på ett sånt sätt att man i princip försöker justera percentilintervallet vad gäller bias och skevhet. Peter Hall och andra tycker detta är ett bakvänt sätt att lösa problemet med percentil-intervallet och säger:

"... and using the percentile method critical points [is equivalent to] looking up the wrong tables backwards. Bias-corrected methods use adjusted probability levels to correct some of the error incurred by looking up wrong tables backwards."

Kapitel 14

Konfidensintervall -BC_a-metoden

14.1 Inledning

I detta kapitel inför vi korrektioner av de percentil-intervall som infördes i kapitel 13. Dessa klarar ju inte av om skattningen ej är väntevärdesriktig och dessutom blir de "bakfram" om fördelningen är skev. Percentil-intervallet tar ju övre konfidensgränsen för parametern med utgångspunkt från övre delen av fördelningen av $\widehat{\theta}^*$. Detta är ju bakvänt i jämförelse med hur ett pivot-baserat intervall fungerar om man som pivot-variabel tar $\widehat{\theta} - \theta$.

Man kan säga att idén är att justera konfidensgränserna på ett sådant sätt att man tar hänsyn till eventuell bias och eventuell skevhet. Man tar alltså kanske inte 5%- respektive 95%-percentilerna från fördelningen av $\widehat{\theta}^*$ utan korrigerar dessa valda percentilpunkter på ett sådant sätt att man korrigerar för bias och skevhet.

Detta förfarande kan uppfattas som lite bakvänt, och har som sagt väckt mycket motstånd från de kretsar som föredrar pivot-baserade konfidensintervall.

14.2 BC_a-metoden

BC_a-metoden är Bias-Correcting och accelererande och är Efrons favoritmetod.

Enligt percentilmetoden får vi konfidensintervallet med konfidensgrad $1 - \alpha$ genom att ta intervallet

$$(\widehat{\theta}_{\text{undre}}, \widehat{\theta}_{\text{övre}}) = (\widehat{\theta}^{*(\alpha/2)}, \widehat{\theta}^{*(1-\alpha/2)})$$

dvs $\alpha/2$- respektive $1 - \alpha/2$-punkterna i bootstrapfördelningen för $\widehat{\theta}$. Detta intervall tar inte hänsyn till om $\widehat{\theta}$ är väntevärdesriktig eller ej. Dessutom blir intervallet skevt åt "fel håll". Detta innebär att om fördelningen för $\widehat{\theta}^*$ (som vi tror approximerar fördelningen för $\widehat{\theta}$) är skev åt höger så kommer intervallet

att bli skevt åt höger. Detta i kontrast till pivot-metoden som i stället skulle göra intervallet skevt åt vänster. Detta beroende på att vi i pivot-metoden skulle använda oss av pivotvariabeln $\widehat{\theta} - \theta$ och när vi "löser ut" θ så blir detta intervall skevt åt vänster.

BC_a-metoden kan uppfattas som ett sätt att

1) kompensera för den bristande väntevärdesriktigheten

2) Kompensera för skevheten i fördelningen

så att slutresultatet liknar det som uppnås med pivot-metoden.

Detta uppnås genom att man använder sig av fördelningen för $\widehat{\theta}^*$ (dvs bootstrap-fördelningen) men inte tar percentilerna $\alpha/2$ respektive $1-\alpha/2$ utan man korrigerar dessa. Det är detta förfarande som av Peter Hall och andra karaktäriserats som att Efron använder sig av "wrong tables backwards".

I BC_a-metoden väljer man intervallet $(\widehat{\theta}^{*(\alpha_1)}, \widehat{\theta}^{*(\alpha_2)})$ där man väljer percentilerna α_1 och α_2 listigt på följande sätt:

$$\alpha_1 = \Phi\left(\widehat{\Delta}_0 + \frac{\widehat{\Delta}_0 + z_{\alpha/2}}{1 - \widehat{a}(\widehat{\Delta}_0 + z_{\alpha/2})}\right)$$

och

$$\alpha_2 = \Phi\left(\widehat{\Delta}_0 + \frac{\widehat{\Delta}_0 + z_{1-\alpha/2}}{1 - \widehat{a}(\widehat{\Delta}_0 + z_{1-\alpha/2})}\right)$$

där z_β är β-percentilen i $N(0,1)$-fördelningen dvs lösningen till ekvationen $\Phi(z_\beta) = \beta$ och

$$\widehat{a} = \text{den skattade "accelerationskonstantanten"}$$

och

$$\widehat{\Delta}_0 = \text{den skattade biaskorrektionen}$$

Dessa beräknas genom

$$\widehat{\Delta}_0 = \Phi^{-1}\left(\frac{\text{antalet } \widehat{\theta}^*(b) \leq \widehat{\theta}}{B}\right)$$

respektive (det finns alternativ)

$$\widehat{a} = \frac{\sum_{i=1}^n (\widehat{\theta}_{(\cdot)} - \widehat{\theta}_{(i)})^3}{6\left(\sum_{i=1}^n (\widehat{\theta}_{(\cdot)} - \widehat{\theta}_{(i)})^2\right)^{3/2}}$$

där $\widehat{\theta}_{(i)}$ är jackknife-skattningen av θ dvs

$$\widehat{\theta}_{(i)} = \widehat{\theta}(x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$$

dvs skattningen av θ där vi uteslutit data nr i, $\quad i = 1, 2, \cdots, n$ och där $\widehat{\theta}_{(\cdot)} = \frac{1}{n}\sum_{i=1}^n \widehat{\theta}_{(i)}$.

Detta ser komplicerat ut men är i fallet med ett stickprov förhållandevis lätt att beräkna och nedan syns en enkel m-fil i Matlab som producerar BC_a-intervall.

Om man har en mer komplicerad datastruktur som t ex två oberoende stickprov blir det värre.

14.2.1 Matlab-kod för BC_a-intervall

Följande m-fil beräknar BC_a-intervallet.

```
function [lower,upper,estimate]=bca(nboot,x,fun,low,up)
% Calculates BCA-intervals
% [lower,upper,estimate]=bca(nboot,x,fun,low,up)
% Parametrar:
% nboot=number of bootstraps
% x=datavector
% fun=estimation-function
% low=lower confidence degree in percent
% up=upper confidence degree in percent
% Output:
% lower=lower confidence limit
% upper=upper confidence limit
% estimate=point estimate

% Point estimate
estimate=feval(fun,x);
% Bootstrap-generation
bootstat=bootstrp(nboot,fun,x);
% Calculation of zhat
b=length(bootstat);
antal=sum(bootstat<=estimate);
zhat=norminv(antal/b);
% Calculation of ahat
[nrow,ncol]=size(x);
if nrow<ncol
        x=x';
        nrow=ncol ;
end
for j=1:nrow
        if j==1
                values=[2:nrow];
        elseif j==nrow
                values=[1:nrow-1];
        else
                values=[1:j-1, j+1:nrow];
        end
```

```
        s(j,:)=feval(fun,x(values,:));
end
mn=mean(s);
[b c]=size(s);
d=s-ones(b,1)*mn;
sum2=sum(d.^2);
sum3=sum(d.^3);
ahat=-sum3./(6*(sum2.^(1.5)));
% Calculation of adjusted confidence levels
n=norminv(low/100);
alfa1=100*normcdf(zhat+(zhat+n)/(1-ahat*(zhat+n)));
n=norminv(up/100);
alfa2=100*normcdf(zhat+(zhat+n)/(1-ahat*(zhat+n)));
% Calculation of confidence limits
lower=prctile(bootstat,alfa1);
upper=prctile(bootstat,alfa2);
```

Om vi använder denna rutin för våra 10 livslängdsdata och vill göra konfidensintervall för θ =väntevärdet blir Matlab-koden:

```
[lower upper estimate]=bca(10000,x,'mean',5,95)
```

där vi kostat på mig 10000 bootstrap-stickprov som tog några minuters exekvering. Man erhöll `lower=3.54` och `upper=10.35` att jämföra med "facit" 3.85 respektive 11.15 enligt en analys där vi använde kunskapen om att data kom från en exponential-fördelning. Med pivot-metoden erhöll vi 4.05 respektive 11.19 och med en enkel CGS-approximation 2.68 respektive 9.42.

14.3 Motivering till definitionen

Bakgrunden till (den skattade) biaskorrektionen $\widehat{\Delta}_0$ är att man antar att det finns en transformation g som gör att $g(\widehat{\theta}) - g(\theta) + \Delta_0$ är $N(0,1)$ för någon konstant Δ_0.

Bakgrunden till (den skattade) accelerationskonstanten $\widehat{a}$ är att man antar att

$$U = \frac{g(\widehat{\theta}) - g(\theta)}{1 + ag(\theta)} + \Delta_0 \text{ är } N(0,1)$$

för någon transformation g och konstanter a respektive Δ_0. Man har här i tankarna att det skall finnas någon approximativt variansstabiliserande transformation g – i stil med Fishers transformation

$$h(\rho) = \frac{1}{2}\ln\left(\frac{1+\rho}{1-\rho}\right)$$

för korrelationskoefficienten ρ.

Att hitta en sådan variansstabiliserande transformation är ju knepigt i en konkret komplicerad situation. Det fiffiga med BC_a-metoden är att man inte behöver hitta den, utan det räcker med att den existerar. BC_a-metoden hittar den i princip "automatiskt"!

Om man i stället för parameten θ studerar den (åtminstone approximativt variansstabiliserade) $\psi = g(\theta)$ så skulle alltså vi studera

$$\frac{\widehat{\psi} - \psi}{D(\widehat{\psi})}$$

där man antar att $D(\widehat{\psi}) \approx 1 + a\psi = 1 + ag(\theta)$, dvs man har en ungefär linjärt beroende av ψ för $D(\widehat{\psi})$. Accelerationskonstanten a svarar då mot den linjära termen i denna "serieutveckling" av $D(\widehat{\psi})$. $a = 0$ skulle svara mot en perfekt variansstabiliserande transformation.

Man antar då (som alltid) att samma relation gäller för bootstrapfördelningen, dvs att

$$U^* = \frac{g(\widehat{\theta}^*) - g(\widehat{\theta})}{1 + ag(\widehat{\theta})} + \Delta_0 \text{ är } N(0,1).$$

Då gäller att om $\alpha/2 = P(U^* \leq z_{\alpha/2})$ så får vi om vi "löser ut" $\widehat{\theta}^*$ att

$$\alpha/2 = P\left(\widehat{\theta}^* \leq g^{-1}\left(g(\widehat{\theta}) + (z_{\alpha/2} - \Delta_0)(1 + ag(\widehat{\theta}))\right)\right).$$

Detta betyder att om vi tar β-percentilen $\xi^*_{(\beta)}$ från fördelningen för $\widehat{\theta}^*$ eller med andra ord $\beta = P(\widehat{\theta}^* \leq \xi^*_{(\beta)})$ så är

$$\xi^*_{(\beta)} = g^{-1}\left(g(\widehat{\theta}) + (z_\beta - \Delta_0)(1 + ag(\widehat{\theta}))\right)$$

och högerledet kan alltså skattas ur bootstrap-fördelningen eftersom vänsterledet erhålls ur bootstrap-fördelningen.

Om vi nu löser ut θ ur relationen

$$1 - \alpha = P(z_{\alpha/2} \leq U \leq z_{1-\alpha/2})$$

så får vi för den undre gränsen

$$1 - \alpha/2 = P(U \geq z_{\alpha/2}) = P\left(\frac{g(\widehat{\theta}) - g(\theta)}{1 + ag(\theta)} + \Delta_0 \geq z_{\alpha/2}\right) =$$

$$P\left(\theta \leq g^{-1}\left(g(\widehat{\theta}) + \frac{\Delta_0 - z_{\alpha/2}}{1 - a(\Delta_0 - z_{\alpha/2})}(1 + ag(\widehat{\theta}))\right)\right).$$

Vi kan alltså identifiera detta (som ger den övre konfidensgränsen för θ) med percentilen från bootstrap-fördelningen om vi där väljer en percentil α_2 så att

$$\frac{\Delta_0 - z_{\alpha/2}}{1 - a(\Delta_0 - z_{\alpha/2})} = z_{\alpha_2} - \Delta_0$$

för då får vi övre gränsen $\xi^*_{(\alpha_2)}$ i konfidensintervallet för θ. Men detta ger

$$z_{\alpha_2} = \Delta_0 + \frac{\Delta_0 - z_{\alpha/2}}{1 - a(\Delta_0 - z_{\alpha/2})}$$

som ger

$$\alpha_2 = \Phi\left(\Delta_0 + \frac{\Delta_0 - z_{\alpha/2}}{1 - a(\Delta_0 - z_{\alpha/2})}\right) = \Phi\left(\Delta_0 + \frac{\Delta_0 + z_{1-\alpha/2}}{1 - a(\Delta_0 + z_{1-\alpha/2})}\right)$$

eftersom $z_{\alpha/2} = -z_{1-\alpha/2}$.

På motsvarande sätt ser vi att vi får den undre gränsen i konfidensintervallet för θ genom att välja α_1-percentilen i bootstrap-fördelningen där

$$\alpha_1 = \Phi\left(\Delta_0 + \frac{\Delta_0 + z_{\alpha/2}}{1 - a(\Delta_0 + z_{\alpha/2})}\right).$$

Ovanstående kalkyl förutsätter att Δ_0 och a skulle vara kända vilket de ju naturligtvis inte är. Dessa måste alltså skattas och för Δ_0 är detta rätt lätt. Vi har ju nämligen

$$P(\widehat{\theta}^* \leq \widehat{\theta}) = P\left(g(\widehat{\theta}^*) \leq g(\widehat{\theta})\right) = P\left(g(\widehat{\theta}^*) - g(\widehat{\theta}) \leq 0\right) =$$

$$= P\left(\frac{g(\widehat{\theta}^*) - g(\widehat{\theta})}{1 + ag(\widehat{\theta})} \leq 0\right) =$$

$$= P\left(\frac{g(\widehat{\theta}^*) - g(\widehat{\theta})}{1 + ag(\widehat{\theta})} + \Delta_0 \leq \Delta_0\right) = P(U^* \leq \Delta_0) \approx \Phi(\Delta_0)$$

eftersom U^* är approximativt $N(0,1)$ eftersom vi antagit att U är $N(0,1)$. Detta ger att vi kan skatta Δ_0 med

$$\widehat{\Delta}_0 = \Phi^{-1}\left(P(\widehat{\theta}^* \leq \widehat{\theta})\right)$$

och $P(\widehat{\theta}^* \leq \widehat{\theta})$ kan ju skattas med $\dfrac{\text{antalet } \widehat{\theta}^*(b) \leq \widehat{\theta}}{B}$ för något stort B.

Skattningen av accelerationsfaktorn är mer besvärlig, men man kan uppfatta det som att vi försöker skatta skevheten i fördelningen. Skattningen med hjälp av jackknife-skattningarna kan uppfattas som att vi skattar

$$\text{tredjemomentet}/(\text{andramomentet})^{3/2}$$

som ingår i en Edgeworth-utveckling av fördelningen. Detta innebär att vi tar med en term utöver den rena normalapproximationen i Edgeworth-utvecklingen och detta förbättrar approximationen.

14.4 Komplikationer

Som nämnts tidigare blir det lite mer komplicerat om man inte bara har ett stickprov, för då måste man skatta skevheten med en två-dimensionell motsvarighet.

Motsvarigheten till BC_a-metoden vid två oberoende stickprov

$$z_1, z_2, \cdots, z_n$$

$$y_1, y_2, \cdots, y_m$$

blir

$$\widehat{a} = \frac{\sum_{i=1}^{n} U_{z,i}^3/n^3 + \sum_{i=1}^{m} U_{y,i}^3/m^3}{6\left(\sum_{i=1}^{n} U_{z,i}^2/n^2 + \sum_{i=1}^{m} U_{y,i}^2/m^2\right)^{3/2}}$$

där $U_{z,i} = (n-1)(\widehat{\theta}_{z,(\cdot)} - \widehat{\theta}_{z,(i)})$ där $\widehat{\theta}_{z,(i)}$=jackknife-skattningen av θ när man utelämnat data $z_i, \quad i = 1, 2, \cdots, n$ och $\widehat{\theta}_{z,(\cdot)}$ = medelvärdet av dessa, dvs

$$\widehat{\theta}_{z,(\cdot)} = \frac{1}{n}\sum_{i=1}^{n} \widehat{\theta}_{z,(i)}$$

och på motsvarande sätt för $U_{y,i}$. Genast har det blivit ganska komplicerat med BC_a-metoden, trots att denna situation är förhållandevis enkel och vanlig.

14.5 Diskussion och jämförelse

Den stora fördelen med BC_a-metoden är att man slipper fundera på vilken pivot-variabel som kan vara lämplig och, framför allt, att man slipper fundera ut den spridningsskattning som ofta dyker upp i pivot-metoden. Ofta har man ju som pivot-variabel $(\widehat{\theta} - \theta)/\widehat{\tau}(\widehat{\theta})$ där $\widehat{\tau}(\widehat{\theta})$ är en skattning av medelfelet för $\widehat{\theta}$. Denna spridningsskattning kan, förutom att den är knepig att fundera ut, vara tidsödande att beräkna. Man kan ju t ex tvingas att använda sig av bootstrap-metodik för att skatta spridningen i nämnaren av $(\widehat{\theta}^* - \widehat{\theta})/\widehat{\tau}(\widehat{\theta}^*)$. Detta innebär i så fall att man måste utföra en bootstrap för varje enskilt bootstrap-stickprov, vilket kan vara mycket tidsödande om skattningen är komplicerad att beräkna. En annan möjlighet är att skatta medelfelet med hjälp av jackknife för varje enskilt bootstrap-stickprov, men då uppstår bekymret att jackknife inte fungerar för alla intressanta skattningar, t ex medianer. Dock kan man undra om BC_a-metoden ger ett korrekt resultat i dessa fall eftersom accelerationskonstanten ju faktiskt skattas med hjälp av jackknife-metodik.

En mycket stor fördel med BC_a-metoden är att den är transformationsbevarande, dvs om vi i stället för att skatta θ tar och skattar en funktion av θ, t ex $\sqrt{\theta}$ eller $1/\theta$, så transformeras konfidensintervallet på motsvarande sätt. Vi får alltså med BC_a-metoden i princip samma intervall oavsett vilken funktion av parametern som vi valt.

Vidare kan man visa att (under lämpliga förutsättningar) så är BC_a-intervallen "second-order accurate" dvs att

$$P(\theta < \widehat{\theta}_{\text{undre}}) \approx \alpha + \frac{c'}{n} \text{ och } P(\theta > \widehat{\theta}_{\text{övre}}) \approx \alpha + \frac{c"}{n}.$$

Detta står i kontrast mot de enkla percentilintervallen och pivot-intervall baserade på $\widehat{\theta} - \theta$ som pivot-variabel, som bara är "first-order accurateS dvs som uppfyller

$$P(\theta < \widehat{\theta}_{\text{undre}}) \approx \alpha + \frac{c'}{\sqrt{n}} \text{ och } P(\theta > \widehat{\theta}_{\text{övre}}) \approx \alpha + \frac{c"}{\sqrt{n}}$$

Pivot-intervall är "second-order accurate" om man tar $(\widehat{\theta} - \theta)/s(\widehat{\theta})$ som pivot-variabel, men de är inte transformationsbevarande.

Man kan också fråga sig hur generell modellen som ligger bakom BC_a-intervallen är.

Frågan om man skall använda sig av Efrons BC_a-intervall eller intervallen enligt pivot-metoden är omdebatterad och kontroversiell.

Man skulle kunna säga att valet beror på om man tror att intervall bör bygga på variansstabiliserande transformationer eller om man tror de bör kopieras på t-metoden.

Kapitel 15

Bayesianska metoder

15.1 Översikt

Vi har en storhet (parameter) θ som vi vill t ex skatta, ge konfidensintervall för eller utföra en hypotesprövning om. Denna parameter har i tidigare kurser betraktats som fix men okänd. I Bayesiansk statistik så uppfattas θ i stället som ett utfall av en stokastisk variabel Θ som har en fördelning (den s k a-priori-fördelningen – a-priori=i förväg) beskriven av t ex en sannolikhetsfunktion eller en täthetsfunktion. Denna a-priori-fördelning kan t ex avspegla tidigare skattningar av θ, men kan också avspegla en subjektiv uppfattning om vilka värde på θ som är mest sannolika. A-priori-fördelningen kan också avspegla en genuin osäkerhet om parameterns värde genom att vi låter den ha en likformig fördelning.

Exempel 15.1 *Pumpars felintensitet*
Vi har ett förråd med pumpar som alla har konstant felintensitet. men där denna varierar mellan de olika pumparna. Vi väljer nu en pump på måfå och till denna pump hör en "sann" felintensitet λ. (Vi kallar parametern λ i stället för θ av bekvämlighetsskäl.)

Denna felintensitet λ ser vi som ett framlottat värde från en stokastisk felintensitet Λ. A-priori-fördelningen för Λ beskriver då hur felintensiteten varierar mellan de olika pumparna i förrådet. Utfallet λ av denna stokastiska felintensitet beror alltså av vilken pump vi råkat få. När vi sedan tittar på tiden till ett fel på pumpen så har denna tid en Exp$(1/\lambda)$-fördelning. Man kan uppfatta det som en lottning i två steg. Först lottas en felintensitet λ fram genom att vi råkar välja en viss pump, och sen blir det en slumpmässighet i tiden till första felet beskriven av exponentialfördelningen som har just denna felintensitet λ.□

15.2 Likelihood-funktion

Traditionellt formuleras en statistisk modell för n observationer $x_1, x_2, \cdots x_n$ som att x_i:na är ett utfall av (oftast oberoende) stokastiska variabler X_1, X_2,

$\cdots$, X_n vars simultana fördelning beror av en parameter θ dvs t ex att man anger

$$p_{X_1,X_2,\cdots X_n}(x_1,x_2,\cdots,x_n;\theta) \quad (x_1,x_2,\cdots,x_n)\in Z^n$$

eller

$$f_{X_1,X_2,\cdots,X_n}(x_1,x_2,\cdots x_n;\theta) \quad (x_1,x_2,\ldots,x_n)\in R^n$$

beroende på om vi har diskreta eller kontinuerliga mätdata.

Dessa sannolikheter (tätheter) att få de data vi fått har vi i tidigare kurser sett som en funktion av θ, den s k likelihoodfunktionen $L(x_1,x_2,\cdots,x_n;\theta)$.

Maximum Likelihoodmetoden (ML-metoden) innebär att vi som skattning $\widehat{\theta}$ av parametern θ tagit det värde på θ som maximerat likelihoodfunktionen $L(x_1,x_2,\cdots,x_n;\theta)$.

En Bayesian uppfattar i stället denna likelihoodfunktion (dvs sannolikhet eller täthet) som en betingad sannolikhet (täthet) där man betingat på att $\Theta=\theta$. Man vill nu "vända på" denna betingning och i stället få fördelningen för Θ givet de observationer $x_1,x_2,\cdots,x_n$ vi fått. Man kanske minns från grundkursen att Bayes' sats var ett verktyg för att "vända på" betingningar. Man kan uppfatta detta som att man vill uppdatera a-priori-fördelningen för Θ med de observationer $x_1,x_2,\cdots,x_n$ vi fått och på så vis erhålla fördelningen för Θ givet de observationer $x_1,x_2,\cdots,x_n$ vi fått.

Om man skulle renodla skillnaden mellan en Bayesian och en icke-Bayesian (frekventist) så ser Bayesianen data (x-n) som fixa medan parametern är slumpmässig, medan icke-Bayesianen ser parametern som fix medan data ses som utfall av stokastiska variabler, dvs som slumpmässiga.

15.3 Bayes' sats

Sats 15.1 *Bayes' sats*
Om $\cup_{i=1}^{\infty}H_i=\Omega$ och $H_i\cap H_j=\emptyset,\ i\neq j$, dvs att H_i:na delar upp utfallsrummet Ω i disjunkta delar så gäller att

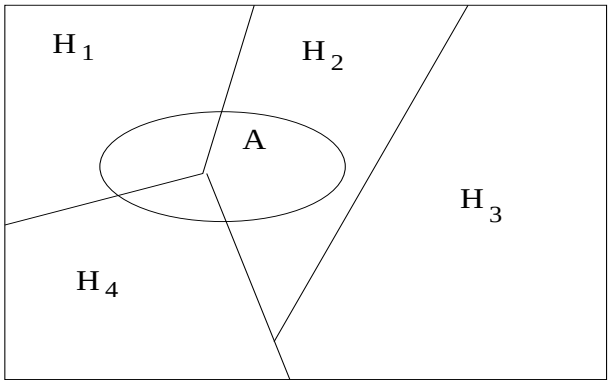
$$P(H_{i_0}|A)=\frac{P(H_{i_0}\cap A)}{P(A)}=\frac{P(A|H_{i_0})P(H_{i_0})}{P(A)}=\frac{P(A|H_{i_0})P(H_{i_0})}{\sum_{i=1}^{\infty}P(A|H_i)P(H_i)}$$

som kan anses som en formel som hjälper till att "vända på" betingningar. Jämför figur 15.1. □

Denna sats visades redan tidigt i grundkursen men är här av avgörande betydelse. Bayesiansk statistik bygger helt och hållet på användningen av Bayes' sats.

Om vi låter $A=\{X=x\}$ och $H_y=\{Y=y\}$ (notera att då är $\cup_{y=0}^{\infty}H_y=\Omega$ och att de är disjunkta) så får vi (jämför $P(A|B)=P(A\cap B)/P(B)$ eller $P(A\cap B)=P(A|B)P(B)$)

$$P(Y=y|X=x)=p_{Y|X=x}(y)=\frac{p_{X,Y}(x,y)}{p_X(x)}=\frac{p_{X|Y=y}(x)p_Y(y)}{p_X(x)}=$$



Figur 15.1: Uppdelning av utfallsrummet Ω i disjunkta H_i:n

$$= \frac{p_{X|Y=y}(x)p_Y(y)}{\sum_{k=0}^{\infty} p_{X|Y=k}(x)p_Y(k)}.$$

Denna "diskreta" formel har "kontinuerliga" motsvarigheter där man på lämpliga ställen byter sannolikhetsfunktion mot täthetsfunktion. Vi har t ex (om både X och Y är kontinuerliga)

$$f_{Y|X=x}(y) = \frac{f_{X,Y}(x,y)}{f_X(x)} = \frac{f_{X|Y=y}(x)f_Y(y)}{f_X(x)} = \frac{f_{X|Y=y}(x)f_Y(y)}{\int_{-\infty}^{\infty} f_{X|Y=u}(x)f_Y(u)du}.$$

eller (om Y är kontinuerlig och X är diskret)

$$f_{Y|X=x}(y) = \frac{p_{X|Y=y}(x)f_Y(y)}{p_X(x)} = \frac{p_{X|Y=y}(x)f_Y(y)}{\int_{-\infty}^{\infty} p_{X|Y=u}(x)f_Y(u)du}.$$

eller (om Y är diskret och X är kontinuerlig)

$$p_{Y|X=x}(y) = \frac{f_{X|Y=y}(x)p_Y(y)}{f_X(x)} = \frac{f_{X|Y=y}(x)p_Y(y)}{\sum_{u=0}^{\infty} f_{X|Y=u}(x)p_Y(u)du}.$$

Dessutom kan man göra flerdimensionella varianter av ovanstående uttryck där vi alltså betraktar X och/eller Y som flerdimensionella. Vi kommer här att låta Y stå för Θ och X stå för $X_1, X_2, \cdots, X_n$ (om vi har mer än en observation).

15.4 Några exempel på likelihoodfunktioner

Vi ger här ett antal elementära (och förhoppningsvis bekanta) exempel på likelihoodfunktioner.

Exempel 15.2 *Binomialfördelning*
Vi har ett utfall x av X som vi anser vara $\text{Bin}(n,\theta)$, dvs att

$$L(x;\theta) = p_{X|\Theta=\theta}(x) = \binom{n}{x}\theta^x(1-\theta)^{n-x}, \quad x = 0,1,2,\cdots,n, \quad 0 \le \theta \le 1.$$

Här har vi alltså bara ett enda mätdata x. □

Exempel 15.3 *Poissonfördelning*
Vi har observationer $x_1, x_2, \cdots x_n$ som är utfall av oberoende Po(θ)-fördelade variabler
$X_1, X_2, \cdots X_n$ dvs

$$L(x_1, x_2, \cdots, x_n; \theta) = p_{X_1, X_2, \cdots, X_n | \Theta=\theta}(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} p_{X_i}(x_i; \theta) =$$

$$= \prod_{i=1}^{n} \frac{\theta^{x_i}}{x_i!} \exp(-\theta) = \frac{\theta^{x_1+x_2+\cdots+x_n}}{x_1! x_2! \cdots x_n!} \exp(-n\theta)$$

□

Exempel 15.4 *Normalfördelning*
Vi har observationer $x_1, x_2, \cdots x_n$ som är utfall av oberoende stokastiska variabler $X_1, X_2, \cdots, X_n$ som alla är N(m, σ^2), dvs med $\theta = (m, \sigma^2)$ har vi

$$L(x_1, x_2, \cdots, x_n; m, \sigma^2) = L(x_1, x_2, \cdots, x_n; \theta) =$$

$$= f_{X_1, X_2, \cdots, X_n | \Theta=\theta}(x_1, x_2, \cdots x_n) = \prod_{i=1}^{n} f_{X_i}(x_i; m, \sigma) =$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i - m)^2}{2\sigma^2}\right) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - m)^2\right)$$

□

15.5 A-posteriori-fördelning

Vi ser nu likelihoodfunktionen som en betingad sannolikhet (täthet) för mätdata $x_1, x_2, \cdots, x_n$ givet $\Theta = \theta$. Med hjälp av Bayes' sats skaffar vänder vi på betingningen och erhåller fördelningen för Θ givet observationerna. Denna fördelning för Θ kallas a-posteriori-fördelningen (a-posteriori = i efterhand). Man kan uppfatta det som om man med hjälp av Bayes' sats uppdaterar a-priori-fördelningen med den information som mätdata ger.

15.6 Några exempel

15.6.1 Binomial-fördelning

Exempel 15.5 *Binomialfördelning – tvåpunktsfördelning för* Θ
Låt X vara Bin($5, \theta$) där θ i sin tur är ett utfall av en stokastisk variabel Θ där $P(\Theta = 0.2) = 0.4$ och $P(\Theta = 0.5) = 0.6$. Vi har alltså en lottning i två steg – dels en lottning om vilken sannolikhet Θ vi har och sen en lottning om vad X blir. Vi kan se det som att X är Bin($5, 0.2$) med sannolikhet 0.4 och att X är Bin($5, 0.5$) med sannolikhet 0.6. Vi har a-priori-fördelningen som lägger

massan 0.4 respektive 0.6 på värdena 0.2 respektive 0.5 för Θ. Detta betyder att vi har den simultana fördelningen

$$p_{X,\Theta}(x,\theta) = P(X = x; \Theta = \theta) = P(X = x|\Theta = \theta)P(\Theta = \theta)$$

som

$$p_{X,\Theta}(x, 0.2) = \binom{5}{x} 0.2^x (1 - 0.2)^{5-x} \cdot 0.4, \quad x = 0, 1, 2, 3, 4, 5$$

och

$$p_{X,\Theta}(x, 0.5) = \binom{5}{x} 0.5^x (1 - 0.5)^{5-x} \cdot 0.6, \quad x = 0, 1, 2, 3, 4, 5$$

Om man nu observerar ett utfall x av X, t ex $X = 4$ så är man intresserad av fördelningen för Θ givet denna observation av $X = 4$. Denna fördelning för parametern Θ givet observationen är a-posteriori-fördelningen för Θ givet $X = 4$. Notera att likelihoodfunktionen anger sannolikheten för observationer givet utfallen av parametern Θ, medan a-posteriorifördelningen anger fördelningen för parametern givet observationen. Denna "vändning" av betingningen är precis vad Bayes' sats tillhandahåller. Vi får då (med $X = 4$) att

$$P(\Theta = 0.2|X = 4) =$$

$$= \frac{P(X = 4|\Theta = 0.2)P(\Theta = 0.2)}{P(X = 4|\Theta = 0.2)P(\Theta = 0.2) + P(X = 4|\Theta = 0.5)P(\Theta = 0.5)} =$$

$$= \frac{\binom{5}{4} 0.2^4 (1 - 0.2)^{5-4} \cdot 0.4}{\binom{5}{4} 0.2^4 (1 - 0.2)^{5-4} \cdot 0.4 + \binom{5}{4} 0.5^4 (1 - 0.5)^{5-4} \cdot 0.6} = 0.0266$$

och $P(\Theta = 0.5|X = 4) = 0.9734$, dvs vår a-prioriuppfattning om sannolikheterna för 0.2 och 0.5 som parametervärden förändras från 0.4 resp 0.6 till att bli 0.0266 resp 0.9734 då vi observerar $X = 4$, som ju tyder på att Θ nog är ganska stort. □

Exempel 15.6 *Binomialfördelning – allmän diskret fördelning för* Θ
På motsvarande sätt skulle det fungera om vi hade en (diskret) a-priorifördelning för Θ på värden $\theta_1, \theta_2, \theta_3, \cdots$ beskriven av sannolikhetsfördelningen $P(\Theta = \theta_i), \quad i = 1, 2, \cdots$. Om vi sen har att X är $\text{Bin}(n, \theta_i)$ så betyder det alltså att X med sannolikhet $P(\Theta = \theta_i)$ är $\text{Bin}(n, \theta_i)$, dvs att $P(X = x|\Theta = \theta_i) = \binom{n}{x} \theta_i^x (1 - \theta_i)^{n-x}$ - notera detta är helt enkelt likelihoodfunktionen då $X = x$ för $\Theta = \theta_i$. Om vi nu observerar $X = x$ så ger detta för Θ a-posteriorifördelningen

$$P(\Theta = \theta_{i_0}|X = x) = \frac{P(X = x|\Theta = \theta_{i_0})P(\Theta = \theta_{i_0})}{\sum_{i=1}^{\infty} P(X = x|\Theta = \theta_i)P(\Theta = \theta_i)}.$$

Detta går naturligtvis utmärkt att räkna ut i en konkret situation med t ex Matlab. Man genererar en vektor `theta` som innehåller de olika θ_i-värdena och en vektor `ptheta` som innehåller $P(\Theta = \theta_i)$ på gittret definierat av `theta` samt en vektor `pxtheta` som innehåller värdet av $\binom{n}{x} \theta_i^x (1 - \theta_i)^{n-x}$ för de olika θ_i:na (och det observerade värdet x) samt skaffar sig vektorn

```
post=pxtheta.*ptheta/sum(pxtheta.*ptheta)
```

som då kommer att innehålla a-posteriori-sannolikheterna för de olika θ_i-na i `theta`. Notera att operationen `.*` utför multiplikationen av de två vektorna koordinatvis. Vektorn `ptheta`:s i-te komponent behöver bara vara proportionell mot $P(\Theta = \theta_i)$ och vektorn `ptheta` alltså inte behöver summera sig till 1 i den numeriska kalkylen eftersom de förekommer i både täljare och nämnare!□

Exempel 15.7 *Binomialfördelning – kontinuerlig fördelning för* Θ
På samma sätt skulle en täthet $f_\Theta(\theta), \quad 0 \leq \theta \leq 1$ som a-prioritäthet för Θ ge upphov till a-posteriori-tätheten

$$f_{\Theta|X=x}(\theta) = \frac{P(X = x|\Theta = \theta)f_\Theta(\theta)}{p_X(x)} = \frac{P(X = x|\Theta = \theta)f_\Theta(\theta)}{\int_0^1 P(X = x|\Theta = u)f_\Theta(u)du} =$$

$$= \frac{\binom{n}{x}\theta^x(1-\theta)^{n-x}f_\Theta(\theta)}{\int_0^1 \binom{n}{x}u^x(1-u)^{n-x}f_\Theta(u)du}$$

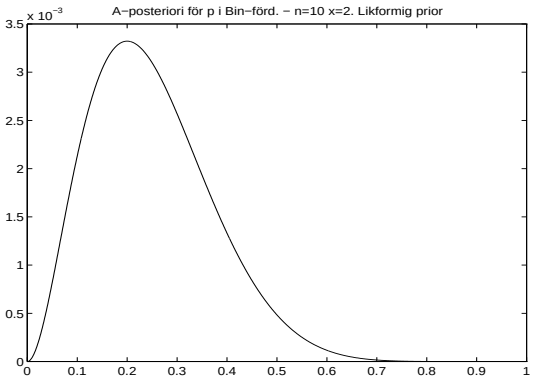
Nämnaren i detta uttryck, alltså

$$p_X(x) = P(X = x) = \int_0^1 P(X = x|\Theta = u)f_\Theta(u)du$$

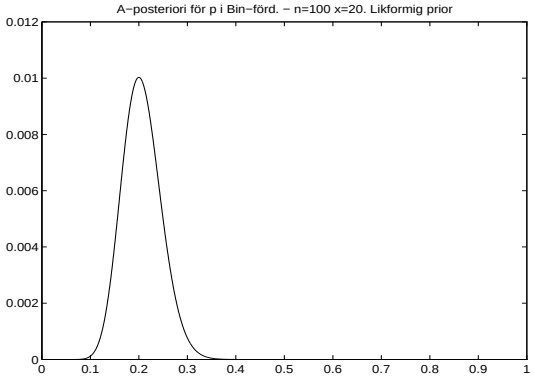
är ofta rätt besvärlig att räkna ut för en given täthet, dvs för en given f_Θ-funktion. Därför väljer man gärna denna täthet på ett sådant sätt att integralen går lätt att räkna ut. Speciellt går det bra om man låter Θ ha en s k Beta-fördelning, som gör att a-posteriori-fördelningen också blir en (annan) Beta-fördelning. Mer om detta i samband med s k konjugerade fördelningar som beskrivs i avsnitt 15.8 på sid 188.

Observera dock det skulle gå bra numeriskt att utföra denna integration med hjälp av Matlab genom att generera en "diskretiserad" variant av täthetsfunktionen f_Θ. Om man alltså inte är ute efter ett slutet analytiskt uttryck för a-posteriori-fördelningen utan nöjer sig med en numeriskt given funktion är intergrationsbekymret inte så stort. Man approximerar tätheten med en diskret fördelning och applicerar sedan (t ex i Matlab) Bayes' sats för diskret fördelning hos Θ. Ett annat alternativ är att utföra en numerisk integration.

Som exempel kan man ta situationen att x är ett utfall av X som är $\text{Bin}(n, \theta)$, dvs $p_{X|\Theta=\theta}(x) = \binom{n}{x}\theta^x(1-\theta)^{n-x}$. Figurerna 15.2 och 15.3 visar a-posteriori-fördelningarna för Θ om $n = 10$, $x = 2$ respektive $n = 100$, $x = 20$ då man haft likformig fördelning på $[0, 1]$ för Θ, dvs $f_\Theta(\theta) = 1$, $0 \leq \theta \leq 1$. Som jämförelse kan man även i figur 15.4 se hur liten skillnaden blir om man som a-priori-fördelning för Θ haft en linjärt växande täthet, dvs $f_\Theta(\theta) = 2\theta$, $0 \leq \theta \leq 1$. □



Figur 15.2: A-posteriori-fördelning för p i binomialfördelning $\text{Bin}(10, p)$ när $x = 2$ observerats. Likformig a-priori-fördelning.



Figur 15.3: A-posteriori-fördelning för p i binomialfördelning $\text{Bin}(100, p)$ när $x = 20$ observerats. Likformig a-priori-fördelning.

15.7 Binomialfördelning

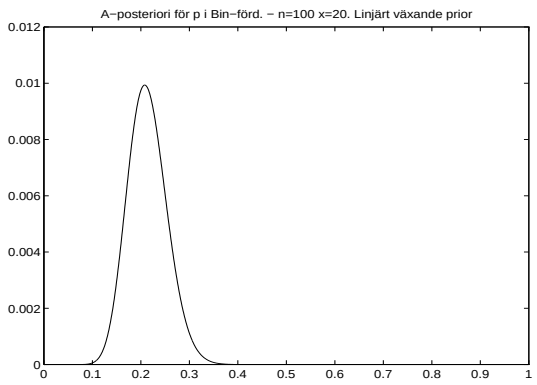
Om man vill räkna exakt i Binomialfördelningssituationen blir det lite besvärligare. Med rätt val av a-priori-fördelning kan det hela bli lite enklare.

Exempel 15.8 *Likformig fördelning*
Om man tar $f_\Theta(\theta) = 1, \quad 0 \leq \theta \leq 1$ får man (efter enkla kalkyler) att

$$f_{\Theta|X=x}(\theta) = \frac{(n+1)!}{x!(n-x)!}\theta^x(1-\theta)^{n-x} \text{ för } 0 \leq \theta \leq 1$$

en s k Beta$(x+1, n-x+1)$-fördelning. □

Detta utgör ett exempel på en lämpligt vald a-priori-fördelning som gör a-posteriori-fördelningen enkel. Vi inför en klass av fördelningar som visar sig vara lämplig klass av a-priori-fördelningar för Binomialfördelningssituationen.



Figur 15.4: A-posteriori-fördelning för p i binomialfördelning $\text{Bin}(100, p)$ när $x = 20$ observerats. Linjärt växande a-priori-fördelning.

15.7.1 Beta-fördelning

Definition 15.1 *En stokastisk variabel* Θ *säges ha Beta*(r, s)*-fördelning om*

$$f_\Theta(u) = \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} u^{r-1}(1-u)^{s-1} \text{ för } 0 \le u \le 1$$

där $r, s > 0$*. Beta*(r, s)*-fördelningen har väntevärdet* $r/(r+s)$*.*

Här är $\Gamma(r) = \int_0^\infty t^{r-1}e^{-t}dt$ – speciellt är $\Gamma(n) = (n-1)!$ (som erhålls med hjälp av partiell integration). Γ-funktionen kan ses som en generalisering av "fakultet" till även icke-heltal. Notera dock att Γ-funktionen är "förskjuten" en enhet i förhållande till fakultet. Den beräknas i Matlab med funktionen `gamma`.

T ex är Beta(1,1)-fördelningen samma sak som likformig fördelning på intervallet $[0, 1]$.

Eftersom man har totalmassan 1 i de olika Beta-fördelningarna ser man att

$$\int_0^1 u^{r-1}(1-u)^{s-1}du = \frac{\Gamma(r)\Gamma(s)}{\Gamma(r+s)}$$

eller mer egentligt att koefficienten i Beta-fördelningens täthet valts så att totalmassan är 1. Av detta följer också att Beta(r, s)-fördelningen har väntevärdet $r/(r+s)$.

15.7.2 Beta-fördelning som a-priori-fördelning

Om nu vi för $\text{Bin}(n, \theta)$-fördelade mätdata låter θ vara ett utfall av Θ som är Beta(a, b), dvs vi låter Θ ha a-priori-fördelningen Beta(a, b) så får vi a-posteriori-fördelningen

$$f_{\Theta|X=x}(\theta) = \frac{p_{X|\Theta=\theta}(x)f_\theta(\theta)}{p_X(x)} = \frac{p_{X|\Theta=\theta}(x)f_\theta(\theta)}{\int_0^1 p_{X|\Theta=u}(x)f_\Theta(u)du}$$

Vi ser att

$$p_X(x) = \int_0^1 p_{X|\Theta=u}(x) f_\Theta(u) du =$$

$$= \int_0^1 \binom{n}{x} u^x (1-u)^{n-x} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} u^{a-1}(1-u)^{b-1} du =$$

$$= \binom{n}{x} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^1 u^{x+a-1}(1-u)^{n-x+b-1} du$$

Den sista integralen ser man är

$$\frac{\Gamma(x+a)\Gamma(n-x+b)}{\Gamma(x+a+n-x+b)} = \frac{\Gamma(x+a)\Gamma(n-x+b)}{\Gamma(a+n+b)}$$

som ger att

$$p_X(x) = \binom{n}{x} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \cdot \frac{\Gamma(x+a)\Gamma(n-x+b)}{\Gamma(a+n+b)}$$

Detta ger då att

$$f_{\Theta|X=x}(\theta) = \frac{\Gamma(a+n+b)}{\Gamma(x+a)\Gamma(n-x+b)} \cdot \theta^{x+a-1}(1-\theta)^{n-x+b-1}$$

dvs att a-posteriori-fördelningen för Θ (givet $X = x$) är Beta$(x+a, n-x+b)$. Speciellt ser vi att om $a = b = 1$ (dvs att vi har en likformig a-priori-fördelning på $[0, 1]$ så blir a-posteriori-fördelningen en Beta$(x+1, n-x+1)$-fördelning.

Faktum är att dessa (rätt besvärliga) integrationer kan undvikas om man noterar att i

$$f_{\Theta|X=x}(\theta) = \frac{p_{X|\Theta=\theta}(x) f_\theta(\theta)}{p_X(x)}$$

utgör nämnaren bara är en normering så att $f_{\Theta|X=x}(\theta)$ verkligen blir en sannolikhetstäthet, dvs har totalmassa (totalintegral)=1.

Vi noterar att

$$f_{\Theta|X=x}(\theta) = \frac{p_{X|\Theta=\theta}(x) f_\theta(\theta)}{p_X(x)} =$$

$$= \frac{\binom{n}{x}\theta^x(1-\theta)^{n-x}\Gamma(a+b)\theta^{a-1}(1-\theta)^{b-1}}{\Gamma(a)\Gamma(b)p_X(x)} \propto \theta^{x+a-1}(1-\theta)^{n-x+b-1}.$$

där vi använder symbolen $\propto$ för "proportionell mot". Som funktion av θ är detta samma uttryck som för Beta$(x+a, n-x+b)$-fördelningen och alltså måste normeringen vara sådan att vi får just Beta$(x+a, n-x+b)$-fördelningen.

Denna typ av resonemang är mycket praktiskt och fungerar ju eftersom vi bara är intresserade av beroendet av θ eftersom vi är ute efter a-posteriori-fördelningen för Θ dvs att få ett funktionellt uttryck i θ. Allt övrigt i uttrycket är enbart att betrakta som en normering eftersom det är en sannolikhetsfördelning (sedd som funktion av θ) vi är ute efter!

15.8 Allmänt om konjugerade fördelningar

Vad man ofta vill göra är att välja a-priori-fördelningen som en lämpligt vald parametrisk familj som gör att a-posteriori-fördelningen tillhör samma parametriska familj.

Ovan såg vi att Beta-fördelningarna var sådana att om man hade en Beta(a, b)-fördelning som a-priori-fördelning för Θ i Binomialfördelningssituationen, där X var Bin(n, Θ), så blev a-posteriori-fördelningen, då vi observerat $X = x$ en Beta$(x + a, n - x + b)$-fördelning.

15.8.1 Definition av konjugerad fördelning

Definition 15.2 *Konjugerad fördelning*
Låt familjen av a-priori-fördelningar $f_\Theta(\theta) = f_\Theta(\theta; \boldsymbol{\alpha})$ bero av en parameter $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_p)$. Antag att a-posteriori-fördelningen $f_{\Theta|\boldsymbol{X}=\boldsymbol{x}}$ tillhör samma parametriska familj, dvs att

$$f_{\Theta|\boldsymbol{X}=\boldsymbol{x}}(\theta) = f_\Theta(\theta; \boldsymbol{\beta})$$

för ett annat värde på parametern

$$\boldsymbol{\beta} = \boldsymbol{\beta}(\boldsymbol{\alpha}, \boldsymbol{x}) = (\beta_1, \beta_2, \cdots, \beta_p).$$

Då säges familjen $f_\Theta(\theta, \boldsymbol{\alpha})$ av tänkbara a-priori-fördelningar vara konjugerad till familjen av fördelningar $F_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{\theta})$ för vektorn $\boldsymbol{X}$ □

Det fiffiga med konjugerade fördelningar är alltså att a-priori-fördelningen och a-posteriori-fördelningen kan sammanfattas med hjälp av $\boldsymbol{\alpha}$ respektive $\boldsymbol{\beta}$ där alltså $\boldsymbol{\beta}$ typiskt beror av våra observationer $\boldsymbol{x}$ och $\boldsymbol{\alpha}$.

15.8.2 Exponentialfördelning

Om vi låter Λ ha en $\Gamma(a, b)$-fördelning och låter $X|\Lambda = \lambda$ vara Exp$(1/\lambda)$ dvs att X har "felintensiteten" λ där λ är ett utfall av Λ som är $\Gamma(a, b)$ så kommer vi nedan att se att $\Lambda|X = x$ kommer att ha en $\Gamma(a + 1, b + x)$-fördelning. Detta betyder alltså att *Gamma*-fördelningarna är konjugerade till familjen av exponentialfördelningar. Vi har nämligen när Λ är $\Gamma(a, b)$

$$f_\Lambda(\lambda) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda}$$

och (eftersom $X|\Lambda = \lambda$ var Exp$(1/\lambda)$)

$$f_{X|\Lambda=\lambda}(x) = \lambda e^{-\lambda x}.$$

Detta ger att

$$f_{\Lambda|X=x}(\lambda) = \frac{f_{X|\Lambda=\lambda}(x) f_\Lambda(\lambda)}{f_X(x)}$$

(vi håller bara reda på λ-beroendet!)

$$f_{\Lambda|X=x}(\lambda) \propto \lambda e^{-\lambda x} \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda} \propto$$

$$\propto \lambda^a e^{-(b+x)\lambda} = \lambda^{(a+1)-1} e^{-(b+x)\lambda}$$

som (sett som funktion av λ) alltså säger att $\Lambda|X = x$ måste vara $\Gamma(a+1, b+x)$.

Om vi alltså har a-priori-fördelningen $\Gamma(a, b)$ för Λ och observerar $X = x$ så blir $\Lambda|X = x$ en $\Gamma(a+1, b+x)$-fördelning.

15.8.3 Normalfördelning

Antag att vi har X_i som är $\mathrm{N}(m, \sigma_0^2)$ med σ_0 känd. Om m är ett utfall av M som har en a-priori-fördelning som är $\mathrm{N}(\mu, s^2)$ så får vi

$$f_{M|\boldsymbol{X}=\boldsymbol{x}}(m) \propto L(m; \boldsymbol{x}) f_M(m) = \left(\prod_{i=1}^{n} f_{X_i}(x_i|M=m) \right) f_M(m) \propto$$

$$\propto \exp\left(-\frac{\sum_{i=1}^{n}(x_i - m)^2}{2\sigma_0^2} \right) \exp\left(-\frac{(m-\mu)^2}{2s^2} \right)$$

$$\propto \exp\left(-\frac{1}{2} \left(\frac{n}{\sigma_0^2} + \frac{1}{s^2} \right) \left(m - \frac{n\bar{x}/\sigma_0^2 + \mu/s^2}{n/\sigma_0^2 + 1/s^2} \right)^2 \right)$$

genom triviala med tidsödande kvadratkompletteringar. Detta uttryck kan, som funktion av m, identifieras med tätheten för en normalfördelning bortsett från proportionalitetsfaktorer (som inte innehåller m) nämligen

$$N\left(\frac{n\bar{x}/\sigma_0^2 + \mu/s^2}{n/\sigma_0^2 + 1/s^2}, \frac{1}{n/\sigma_0^2 + 1/s^2} \right)$$

och detta måste alltså vara a-posteriori-fördelningen för M dvs fördelningen för

$$(M|\boldsymbol{X} = \boldsymbol{x}) = (M|X_1 = x_1; X_2 = x_2; \cdots; X_n = x_n)$$

Slutsatsen är alltså att i denna situation är normalfördelningarna sin egen konjugerade fördelning.

Detta harvande med konjugerade fördelningar kan ha ett visst intresse för sig, men utgör också en motivering för hur praktisk Markov Chain Monte Carlo-simulering är. I sådan simulering slipper man alla dessa komplicerade kalkyler rörande konjugerade fördelningar utan man simulerar fram a-posteriori-fördelningen med förvånansvärd enkelhet. För att få tillbörlig respekt för hur praktiskt detta är har vi därför studerat hur komplicerat det är att reda ut detta med konjugerade fördelningar.

15.8.4 A-posteriori-fördelning som ny a-priori

När man fått en observation och kombinerar denna med a-priori-fördelningen och därvid erhåller en a-posteriori-fördelning så kan ju tänkas göra ytterligare en observation. Om man då som a-priori-fördelning för denna andra observation tar a-posteriori-fördelningen efter första observationen så skulle man erhålla en ny a-posteriori-fördelning. Denna blir precis samma som om man med den ursprungliga a-priori-fördelningen gjort två oberoende observationer och då beräknat a-posteriori-fördelningen givet båda observationerna. I denna mening kan man alltså se a-posteriori-fördelningen som en successiv uppdatering av a-priori-fördelningen med de erhållna observationerna.

Sats 15.2 Låt observationerna $x_1, x_2, \cdots, x_n$ vara oberoende likafördelade givet $\Theta = \theta$. Vi har a-priori-fördelningen $f_\Theta(\theta)$ för Θ. Man får samma a-posteriori-fördelning med följande två betraktelsesätt:

1) Vi beräknar a-posteriori-fördelningen för Θ givet första observationen x_1 och använder denna a-posteriori-fördelning som a-priori-fördelning för Θ då data nr 2 x_2 insamlas. A-posteriori-fördelningen givet detta x_2 används sedan som a-priori-fördelningen vid analys med hjälp av x_3, osv ända till sista observationen x_n.

2) Vi beräknar a-posteriori-fördelningen givet hela observationsvektorn $x_1, x_2, \cdots, x_n$.

Bevis:
Låt $f_1(\theta|x_1)$ vara a-posteriori-fördelningen efter att vi observerat x_1 och $f_2(\theta|x_2)$ vara a-posteriori-fördelningen efter man observerat x_2 och har $f_1(\theta|x_1)$ som a-priori-fördelning. P s s låter vi $f_i(\theta|x_i)$ vara a-posteriori-fördelningen efter att x_i observerats för $i = 1, 2, \cdots, n$ och vi har $f_{i-1}(\theta|x_{i-1})$ som a-priori-fördelning. Vi får då

$$f_1(\theta|x_1) \propto f_\Theta(\theta) f_{X_1}(x_1|\theta)$$

och

$$f_2(\theta|x_2) \propto f_1(\theta|x_1) f_{X_2}(x_2|\theta) \propto f_\Theta(\theta) f_{X_1}(x_1|\theta) f_{X_2}(x_2|\theta)$$

och på samma sätt erhålls

$$f_n(\theta|x_n) \propto f_{n-1}(\theta|x_{n-1}) f_{X_n}(x_n|\theta) \propto \cdots \propto f_\Theta(\theta) f_{X_1}(x_1|\theta) f_{X_2}(x_2|\theta) \cdots f_{X_n}(x_n|\theta).$$

Å andra sidan gäller att

$$f_\Theta(\theta|x_1, x_2, \cdots, x_n) \propto f_\Theta(\theta) f_{X_1, X_2, \cdots, X_n}(x_1, x_2, \cdots, x_n|\theta) =$$

$$= \text{oberoendet} = f_\Theta(\theta) \prod_{i=1}^{n} f_{X_i}(x_i|\theta)$$

och alltså ger båda metoderna samma resultat. Faktum är att i beviset behövs inte ens antagandet att X_i är likafördelade, men däremot måste de vara betingat oberoende givet θ. □

15.9 Bayesiansk inferens

För en Bayesian uttrycker a-posteriori-fördelningen all information om vår uppfattning om Θ när vi tagit hänsyn till våra mätdata och a-priori-fördelningen. Man kan även sammanfatta denna fördelning med t ex väntevärdet, dvs $E(\Theta|X = x)$ eller medianen eller det värde (moden) som ger maximum i tätheten för a-posteriori-fördelningen. Oftast används just det betingade väntevärdet $E(\Theta|X = x)$, dvs väntevärdet i a-posteriori-fördelningen och denna skattning brukar kallas Bayes-skattningen $\widehat{\theta}_{\text{Bayes}}$. Man kan f ö notera att om man har en likformig fördelning (dvs $f_\Theta(\theta)$ =konstant) som a-priori-fördelning så är det just Maximum-Likelihood-skattningen som maximerar a-posteriori-tätheten! Den vanliga skattningsteorin som bygger på Maximum-Likelihoodmetoden är specialfallet att ha likformig a-priori-fördelning och skatta med det värde som maximerar a-posteriori-tätheten. Klassisk teori kan därför ses som ett specialfall av Bayesiansk metodik.

Om man i binomialfördelningssituationen har en $Beta(r, s)$-fördelning som a-priori-fördelning för Θ och observerar utfallet $X = x$ av $Bin(n, \theta)$ så såg vi att a-posteriori-fördelningen för Θ blev en $Beta(x + r, n - x + s)$-fördelning. Denna a-posteriori-fördelning har ett väntevärde som är $(r + x)/(r + s + n)$ och detta utgör då Bayes-skattningen $\widehat{\theta}_{\text{Bayes}}$ av Θ. Man kan för övrigt notera att

$$\widehat{\theta}_{\text{Bayes}} = \frac{r+x}{r+s+n} = \frac{n}{r+s+n} \cdot \frac{x}{n} + \frac{r+s}{r+s+n} \cdot \frac{r}{r+s}$$

där x/n är den traditionella ML-skattningen av θ och $r/(r+s)$ är väntevärdet av a-priori-fördelningen $Beta(r, s)$. Bayes-skattningen av θ är alltså ett viktat medelvärde av ML-skattningen och a-priori-skattningen (dvs väntevärdet) där vikterna är $n/(r+s+n)$ respektive $(r+s)/(r+s+n)$. Man ser att om bara n är stort läggs nästan all vikt på skattningen som härrör sig ur data (dvs x/n).

15.9.1 Bayesianska konfidensintervall

Man kan också göra Bayesianska konfidensintervall (credibility intervals) för parametern genom att ta punkter i a-posteriori-fördelningen mellan vilka det ligger t ex 95% av massan. Det finns även andra alternativ som att välja de regioner där a-posteriori-tätheten är stor och att avpassa området så att totala sannolikhetsmassan blir t ex 95%. Detta kan innebära att "konfidensintervallet" blir uppdelat i flera disjunkta delar om a-posteriori-fördelningen inte är unimodal (dvs har mer än en topp).

Det numeriska intervallet $(a(\boldsymbol{x}), b(\boldsymbol{x})) = (a(x_1, x_2, \cdots, x_n), b(x_1, x_2, \cdots, x_n))$ är alltså ett konfidensintervall för Θ med konfidensgrad $1 - \alpha$ om

$$P(a(\boldsymbol{x}) < \Theta < b(\boldsymbol{x})|\boldsymbol{X} = \boldsymbol{x}) = \int_{a(\boldsymbol{x})}^{b(\boldsymbol{x})} f_\Theta(\theta|\boldsymbol{X} = \boldsymbol{x})d\theta = 1 - \alpha$$

dvs i ord just två punkter $a(\boldsymbol{x})$ och $b(\boldsymbol{x})$ mellan vilka det finns massan $1 - \alpha$ i a-posteriori-fördelningen för Θ.

Exempel 15.9 *Bayesianskt konfidensintervall för normalfördelning*
Vi såg tidigare att om X_i är $N(m, \sigma_0^2)$ där σ_0 är känd och m var ett utfall av M och vi har $N(\mu, s^2)$ som a-priori-fördelning för M så blev a-posteriori-fördelningen för M givet observationerna $x_1, x_2, \cdots, x_n$

$$N\left(\frac{n\bar{x}/\sigma_0^2 + \mu/s^2}{n/\sigma_0^2 + 1/s^2}, \frac{1}{n/\sigma_0^2 + 1/s^2}\right).$$

Om vi låter $s \to \infty$ vilket svarar mot att vi väljer likformig fördelning för m som a-priori-fördelning ser vi att a-posteriori-fördelningen för M blir $N(\bar{x}, \sigma_0^2/n)$ och det 95%-iga Bayesianska konfidensintervallet för m blir $\bar{x} \pm \lambda_{0.025}\sigma_0/\sqrt{n}$ dvs det helt sedvanliga intervallet. Tolkningen av detta intervall är alltså att mellan de två gränserna finns 95% av massan i a-posteriori-fördelningen, dvs i överensstämmelse med den "naiva" (och felaktiga) tolkningen av konfidensintervall som ett intervall som med sannolikheten 95% innehåller parametern.□

15.9.2 Prediktiv fördelning

A-posteriori-fördelningen är ju ett sätt att beskriva fördelningen för parametern Θ då man gjort observationerna $\boldsymbol{x}$. Ett praktiskt sätt att utnyttja denna a-posteriori-fördelning är att studera fördelningen för en ny observation X_0. Man definierar den prediktiva fördelningen som

$$f_{X_0}(x_0|\boldsymbol{X} = \boldsymbol{x}) = \int_\Omega f_{X_0}(x_0|\Theta = \theta) f_\Theta(\theta|\boldsymbol{X} = \boldsymbol{x}) d\theta.$$

Denna integral innebär alltså att vi som fördelning för X_0 tar en hopviktning av tätheten för X_0 för olika θ-värden med hänsyn till hur sannolika dessa θ-värden är enligt a-posteriori-fördelningen.

Exempel 15.10 Låt N vara $\text{Po}(\lambda)$ då $\Lambda = \lambda$ där Λ är $\Gamma(a, b)$. Vi observerar $N = n$. Detta betyder alltså att

$$f_\Lambda(\lambda) = \frac{\lambda^{a-1} b^a}{\Gamma(a)} e^{-b\lambda},\ \lambda > 0$$

och

$$P(N = n|\Lambda = \lambda) = \frac{\lambda^n}{n!} e^{-\lambda},\ n = 0, 1, 2, \cdots$$

Man får då att a-posteriori-fördelningen för Λ (dvs givet $N = n$) blir (där vi bara håller reda på λ-beroendet)

$$f_\Lambda(\lambda|N = n) \propto f_\Lambda(\lambda) P(N = n|\Lambda = \lambda) \propto$$
$$\propto \lambda^{a-1} e^{-b\lambda} \cdot \lambda^n e^{-\lambda} = \lambda^{n+a-1} e^{-\lambda(b+1)}$$

vilket alltså innebär att $\Lambda|N = n$ är $\Gamma(n + a, b + 1)$-fördelad dvs att

$$f_\Lambda(\lambda|N = n) = \frac{\lambda^{n+a-1}(b+1)^{n+a}}{\Gamma(n+a)} e^{-(b+1)\lambda},\ \lambda > 0.$$

Vi har förresten därigenom i förbigående visat att Γ-fördelningarna utgör konjugerade fördelningar till Poisson-fördelningen.

Vi får då den prediktiva fördelningen för en ny observation N_0 till

$$P(N_0 = n_0|N = n) = \int_0^\infty P(N_0 = n_0|\Lambda = \lambda) f_\Lambda(\lambda|N = n) d\lambda =$$

$$= \int_0^\infty \frac{\lambda^{n_0}}{n_0!} e^{-\lambda} \cdot \frac{\lambda^{n+a-1}(b+1)^{n+a}}{\Gamma(n+a)} e^{-(b+1)\lambda} d\lambda.$$

Detta ger efter en del enkla räkningar att

$$P(N_0 = n_0|N = n) = \frac{(b+1)^{n+a}}{\Gamma(n+a)} \cdot \frac{1}{n_0!} \cdot \frac{\Gamma(n_0+n+a)}{(b+2)^{n_0+n+a}},\ n_0 = 0, 1, 2, \cdots .$$

Man kan notera att detta fungerar även om $N = 0$, dvs vi kan få fram den prediktiva fördelningen även då ingen händelse observerats. Vi får för $n = 0$ att

$$P(N_0 = n_0|N = 0) = \frac{(b+1)^a}{\Gamma(a)} \cdot \frac{1}{n_0!} \cdot \frac{\Gamma(n_0+a)}{(b+2)^{n_0+a}}$$

som för $a = 1$ och $b = 1$ ger

$$P(N_0 = n_0|N = 0) = \frac{2}{3^{n_0+1}},\ n_0 = 0, 1, 2, \cdots$$

dvs t ex $P(N_0 = 0|N = 0) = 2/3$ och $P(N_0 = 1|N = 0) = 2/9$. Detta betyder att $N_0|N = 0$ i detta fall är Geo(2/3)-fördelningen.

Man kan jämföra detta med de problem en icke-Bayesian skulle råka i om han observerar $N = 0$ och alltså skattar λ med $\widehat{\lambda} = 0$. Ingen tror väl att andra värden än 0 är omöjliga?!

15.10 Hur väljs a-priori-fördelning?

Detta att man som Bayesian måste välja en a-priori-fördelning är både en svaghet och en styrka för de Bayesianska metoderna. Oftast känns valet av a-priori-fördelning ganska godtyckligt, men, glädjande nog, spelar valet förvånansvärt liten roll i de flesta situationer. Det kan också uppfattas som en styrka att man kan väga in ”förutfattade meningar” om vilka parametervärden som är troligast i sin a-priori-fördelning.

Oftast har valet av a-priori-fördelning dock mindre styrts av vad man haft anledning att tro om parametern än av att kalkylerna skall bli hanterliga – speciellt har Bayesianer känt sig bundna av att hålla sig till konjugerade fördelningar. Som vi kommer att se i kapitel 17 om Markov Chain Monte Carlo-simulering bortfaller detta behov av speciella val av a-priori-fördelningar. Vi kommer i stället att kunna erhålla resultat genom simulering.

15.10.1 Icke-informativa a-priori-fördelningar

En icke-informativ a-priori-fördelning för Θ är i princip en likformig fördelning för Θ. Om de tänkbara värdena för Θ är ett ändligt intervall uppstår inga matematiska problem. I binomialfördelningssituationen är alltså likformig fördelning på intervallet $(0, 1)$ en sådan icke-informativ a-priori-fördelning. Skulle de tänkbara värdena för Θ vara ett oändligt intervall kan matematiska problem uppstå eftersom man inte kan definiera en likformig sannolikhetsfördelning på ett oändligt intervall. I och för sig behöver inte a-priori-fördelningen vara normerad så ibland fungerar matematiskt en likformig "fördelning" (oändlig totalmassa) på ett oändligt intervall men man kan stöta på problem i form av att även a-posteriori-fördelningen inte går att normera.

Ibland kan man "fuska" och i stället ta likformig fördelning på ett stort (men ändligt) intervall $(0, M)$ där M valts så stor att alla "realistiska" värden på Θ finns i intervallet. Om man t ex tror att realistiska värden på felintensiteten Λ är ca 1 kan man välja a-priori-fördelning som likformig fördelning på intervallet (0,1000). Om man inte räknar exakt utan har tänkt sig att utföra en numerisk kalkyl i t ex Matlab så kommer man ändå att tvingas inskränka sig till ett ändligt intervall. En annan vanlig teknik är att välja a-priori-fördelningen som en sannolikhetsfördelning som är "nästan konstant" – t ex kan man välja $N(0, 1000)$-fördelningen eller $\Gamma(10^{-3}, 10^{-3})$-fördelningen beroende på om man vill ha "nästan" likformig fördelning på hela x-axeln eller på positiva x-axeln.

Hela begreppet icke-informativ a-priori-fördelning är dock rätt skumt eftersom de inte är invarianta under omparametriseringar. Om man t ex har likformig fördelning för standardavvikelsen σ eller för variansen σ^2 så kommer detta att spela en viss roll. På samma sätt spelar det roll om vi använder oss av felintensitet λ eller medellivslängd $m = 1/\lambda$ som den parameter vi studerar. Båda dessa val av parametrisering kan ju anses vara "naturliga".

En annan vanlig invändning från Bayesianer om godtyckligheten i a-priori-fördelningar är att en icke-Bayesian som baserar sin inferens på t ex Maximum Likelihood-metoden "egentligen" är en Bayesian som har likformig fördelning på parameter-rummet som a-priori-fördelning. Att detta i någon mån är sant inser man av att om man har konstant a-priori-fördelning så blir a-posteriori-fördelningen direkt proportionell mot likelihood-funktionen. Att använda sig av ML-skattningen, dvs att maximera likelihoodfunktionen är inget annat än att som Bayesian använda moden (dvs maximum) i a-posteriori-fördelningen som skattning.

Kapitel 16

Modellval – korsvalidering

16.1 Inledning och översikt

Ett centralt problem i matematisk statistik är att utgående från observationer konstruera en statistisk modell M för hur dessa har uppkommit. Speciellt är vi intresserade av att kunna välja den "bästa" av ett antal föreslagna modeller $M_1, M_2, \cdots, M_m$. Modellval (Model selection) handlar om just detta centrala problem. Korsvalidering är en viktig teknik som på ett förhållandevis bra sätt utför detta modellval. Andra metoder som AIC (Aikaike Information Criterion) och BIC (Bayesian Information Criterion) kommer också att i någon mån beröras.

En statistisk modell M innebär att vi ser våra observationer $\boldsymbol{y} = (y_1, \cdots, y_n)^T$ som utfall av $\boldsymbol{Y} = (Y_1, Y_2, \cdots, Y_n)^T$ som har en n-dimensionell fördelning och i denna fördelning ingår ett antal parametrar $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_p)^T$. Storleken p av denna parametervektor kallar vi modellstorleken.

Genomgående antar vi att

$$Y_i = g_i(\boldsymbol{\theta}) + \epsilon_i, \ i = 1, 2, \cdots, n$$

där $E(\epsilon_i) = 0$ och $V(\epsilon_i) = \sigma^2$ och att ε_i:na är oberoende. g_i:na är kända explicita funktioner. Modellstorleken p hänför sig alltså till väntevärdesdelen. Med modell avser vi alltså en situation som ovanstående, där alltså $\boldsymbol{\theta}$ liksom σ^2 och fördelningen för ϵ_i:na ej är specificerade.

Givet en sådan modell M konstruerar vi en numerisk skattning $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}(\boldsymbol{y}, M)$. Vi kallar då $g_i(\widehat{\boldsymbol{\theta}})$ för en prediktor av y_i och betcknar den med $\widehat{y}_i$. Ofta konstruerar vi också en skattning av σ^2.

Exempel 16.1 *Allmänna linjära modellen*
Teorin för allmänna linjära modellen är ett exempel på hur vi givet en viss modell M kan skatta parametrarna i denna.

I allmänna linjära modellen anser vi att

$$\boldsymbol{Y} = A\boldsymbol{\theta} + \boldsymbol{\epsilon},$$

där $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \cdots, \epsilon_n)^T$ och ϵ_i:na är oberoende och har $E(\varepsilon_i) = 0$ och $V(\varepsilon_i) = \sigma^2$. Ofta antar vi dessutom att de är $\mathrm{N}(0, \sigma^2),\ i = 1, 2, \cdots, n$. Vi antar alltså att g_i-funktionerna är kända linjära uttryck av parametrarna.

Matrisen A kallas designmatrisen och är känd – t ex kan den innehålla kovariater (förklaringsvariabler) $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{ip})$ hörande till observationen $y_i,\ i = 1, 2, \cdots, n$.

Som skattning av $\boldsymbol{\theta}$ tar man

$$\widehat{\boldsymbol{\theta}} = (A^T A)^{-1} A^T \boldsymbol{y}$$

och som prediktorer tar man

$$\widehat{\boldsymbol{y}} = A\widehat{\boldsymbol{\theta}}.$$

Som ett konkret exempel på linjära modeller kan man ta regressionsmodeller.

Exempel 16.2 *Regressionsmodeller*
I regressionsmodeller antar vi att

$$Y_j = \theta_1 x_{j1} + \theta_2 x_{j2} + \theta_3 x_{j2} + \cdots + \theta_p x_{jp} + \epsilon_j,\ j = 1, 2, \cdots, n.$$

I allmänhet tar man $x_{j1} = 1$ dvs låter θ_1 vara "interceptet". Vi har då designmatrisen

$$A = \begin{pmatrix} 1 & x_{12} & x_{13} & \cdots & x_{1p} \\ 1 & x_{22} & x_{23} & \cdots & x_{2p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{n2} & x_{n3} & \cdots & x_{np} \end{pmatrix}$$

Prediktorn $\widehat{y}_j$ är då vad det skattade regressionsplanet förutsäger om y_j. □

16.2 Flera alternativa modeller

Ett centralt problem är att ställa olika alternativa modeller $M_1, M_2, \cdots, M_m$ mot varandra och dels

1) Välja en modell $\widehat{M}$

2) Skatta parametrarna i denna valda modell.

Vi har här två motstridiga intressen nämligen

1) Den valda modellen skall väl beskriva de erhållna data

2) Modellen skall vara så enkel som möjligt, dvs innehålla så få parametrar som möjligt.

Dessa två mål är oftast i konflikt med varandra – vi kan erhålla god anpassning till priset av att välja en komplicerad modell med många parametrar. Modellval innebär att man gör denna avvägning på ett lämpligt sätt. I princip kommer vi att vilja att våra modeller på ett bra sätt "predikterar" framtida observationer.

16.2.1 Test av hypotes i allmänna linjära modellen

I den allmänna linjära modellen kan vi t ex ställa två sådana modeller M_1 och M_2 mot varandra om M_1 (hypotesmodellen) är ett specialfall av M_2 (grundmodellen), dvs att $M_1 \subset M_2$. Hypotesmodellen kan t ex innebära att vissa av parametrarna i grundmodellen är 0. I regressionsmodellen antar man t ex att vissa av förklaringsvariablerna (x-variablerna) saknas i hypotesmodellen, dvs att motsvarande θ-parametrar är 0 och ställer denna hypotesmodell mot den fullständiga modellen . Man brukar då säga att man har "nested models" dvs att den ena modellen är "innesluten" i (dvs ett specialfall av) den andra.

Hypotesmodellen M_1 kan t ex vara $H_0 : B\boldsymbol{\theta} = \mathbf{0}$ där B lägger $p - r$ linjärt oberoende restriktioner på $\boldsymbol{\theta}$-vektorn. Detta innebär att H_0 betyder att $\boldsymbol{\theta}$-vektorn har "verklig" dimension $r < p$. I det enklaste fallet kan det innebära att t ex $\theta_1 = 0$ och $\theta_2 = 0$ där vi alltså har $r = p - 2$ eftersom H_0 innebär två linjärt oberoende bivillkor på $\boldsymbol{\theta}$-vektorn.

Om nu $M_1 \subset M_2$ och M_1 har p_1 parametrar och M_2 har p_2 parametrar så studerar vi residualkvadratsummorna

$$n\widehat{\sigma}^2(M_1) = |\boldsymbol{y} - \widehat{\boldsymbol{y}}(M_1)|^2 = \sum_{i=1}^{n}(y_i - \widehat{y}_i(M_1))^2$$

och motsvarande storhet för grundmodellen M_2.

Enligt resultat från teorin för allmänna linjära modellen gäller att om ϵ_i:na är oberoende $\mathrm{N}(0, \sigma^2)$ att

1) $n(\widehat{\sigma}^2(M_1) - \widehat{\sigma}^2(M_2))/\sigma^2$ är $\chi^2(p_2 - p_1)$ om $H_0 : M_1$ är sann

2) $n\widehat{\sigma}^2(M_2)/\sigma^2$ är $\chi^2(n - p_2)$

3) de är oberoende.

Kvoterna av dessa (delade med frihetsgradsantalen $p_2 - p_1$ respektive $n - p_2$) har alltså en $F(p_2 - p_1, n - p_2)$-fördelning om M_1 är sann och detta kan användas för att testa hypotesen $H_0 : M_1$ mot $H_1 : M_2$.

16.3 "Non-nested models"

Vi vill dock inte begränsa oss till denna typ av "nested models", där alltså den ena är ett specialfall av den andra utan vi vill kunna välja bland mer allmänna typer av modeller.

Exempel 16.3 Hur skall vi jämföra modellerna M_1, M_2 och M_3 när vi har kovariaterna x_j och z_j till observation j

$$M_1 : Y_j = \theta_1 + \theta_2 x_j + \varepsilon_j$$

$$M_2 : Y_j = e^{\theta_1 + \theta_2 x_j} + \theta_3 z_j + \varepsilon_j$$

$$M_3 : Y_j = \theta_1 + \theta_2 x_j + \theta_3 x_j^2 + \theta_4 z_j + \varepsilon_j$$

□

Exempel 16.4 *Regressionsmodeller*
I en regressionsmodell med $p-1$ förklaringsvariabler samt ett intercept (dvs totalt p väntevärdesparametrar) vill vi kunna välja vilka av dessa som skall ingå i en optimalt vald modell. Med p parametrar finns 2^p tänkbara sådana regressionsmodeller (eller 2^{p-1} st om vi alltid vill ha med "interceptet"). Vissa av dessa utgör "specialfall" av varandra men generellt är de ej av ovanstående typ och F-test fungerar alltså inte för alla dessa jämförelser mellan modellerna. □

Hur skall man nu kunna jämföra modeller av helt olika typ och dessutom ofta med olika antal parametrar med varandra? Vi kommer som utgångspunkt för värderingen av en modell att ta dess förmåga att prediktera nya observationer. Vi har allmänhet inte några nya observationer att ta till – hade vi det skulle dessa naturligtvis ha använts redan från början – och vi måste alltså försöka efterlikna en situation med nya observationer. Korsvalidering är en sådan teknik.

Ett stort problem vid val av modell är att vi gärna vill välja en modell som ger god anpassning för de observationer vi fått, men egentligen vill vi använda modellen för framtida observationer. Kanske har vi skaffat oss livslängdsdata för en uppsättning pumpar. Idén med inferensen är ju dock att kunna förutsäga livslängden för en annan uppsättning pumpar som skall ingå i ett tekniskt system. De speciella pumpar vi undersökt och baserat inferensen på är ju bara intressanta som "representanter" för de pumpar vi sedan i verkligheten tänkt använda.

Ett annat problem som dyker upp i samband med modellval är att vi oftast använder de erhållna observationerna dels som underlag för inferensen och dels som "facit", dvs som det vi använder för att kontrollera om inferensen gett korrekt och/eller precist resultat. Ett sätt att undvika detta är att för en viss observation basera skattningen av denna (egentligen prediktionen av den) på en datamängd där observationen själv inte ingår! Korsvalidering innebär just detta, dvs att vi för var och en av våra observationer använder bara de övriga för inferensen för just den observationen.

16.4 Prediktionsförmåga

Vi vill kunna välja en viss modell $\widehat{M}$ bland $M_1, M_2, \cdots, M_m$ och vi kommer att låta vårt val avgöras av vilken modell som ger bäst prediktion av nya (tänkta) data $\boldsymbol{Y'} = (Y_1', Y_2', \cdots, Y_n')^T$ med samma fördelning som de ursprungliga variablerna $\boldsymbol{Y} = (Y_1, Y_2, \cdots, Y_n)^T$ som vi såg våra numeriska observationer $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^T$ som utfall av.

Vi skall i detta avsnitt i någon mån diskutera teoretiska mått på prediktionsförmåga där man ser sina observationer som utfall av stokastiska variabler

och i princip bildar medelvärde över denna tänkta uppsättning ursprungsdata och dessutom över den tänkta uppsättning av kommande observationer som man vill prediktera. I nästa avsnitt återvänder vi till den mer relevanta situationen där vi ser våra observationer som fixa och bara tar hänsyn till att de kommande observationerna tänkes vara slumpmässiga. Inte ens dessa tänkta kommande observationer har vi ju dock tillgång till utan vi måste kunna använda våra ursprungliga observationer både till att ge förutsägelser och använda dem till att utvärdera dessa förutsägelser.

I en viss modell M beräknar vi numeriska prediktorer $\widehat{y}_j,\ j = 1, 2, \cdots, n$ av observationerna $y_1, y_2, \cdots, y_n$. Detta sker ju i allmänhet genom att vi skattar parametrarna i modellen med $\widehat{\boldsymbol{\theta}}$ och prediktorerna är sedan explicita funktioner av dessa skattningar. I allmänna linjära modellen väljer man t ex $\widehat{\boldsymbol{y}} = A\widehat{\boldsymbol{\theta}}$ som prediktorer där $\widehat{\boldsymbol{\theta}} = (A^TA)^{-1}A^T\boldsymbol{y}$ dvs vi har

$$\widehat{\boldsymbol{y}} = A(A^TA)^{-1}A^T\boldsymbol{y}$$

I verkligheten innebär minsta kvadratmetoden just att vi som skattning $\widehat{\boldsymbol{\theta}}$ valt den vektor $\boldsymbol{\theta}$ som minimerar $|\boldsymbol{y} - A\boldsymbol{\theta}|^2$, dvs som minimerar

$$RSS = |\boldsymbol{y} - \widehat{\boldsymbol{y}}|^2 = (y_1 - \widehat{y}_1)^2 + (y_2 - \widehat{y}_2)^2 + \cdots + (y_n - \widehat{y}_n)^2$$

där RSS står för Residual Sum of Squares (residualkvadratsumman).

Geometriskt betyder det att vi projicerar ner den n-dimensionella vektorn $\boldsymbol{y}$ på det p-dimensionella underrum som spänns av kolonnerna i A-matrisen. Detta rum är ju precis det man kan få genom att bilda $A\boldsymbol{\theta}$. $A\boldsymbol{\theta}$ är ju just en linjärkombination av kolonnvektorerna i matrisen A.

Matrisen $\Pi = A(A^TA)^{-1}A^T$ utför just denna projektion. Man ser t ex att Π är en symmetrisk matris som uppfyller $\Pi^2 = \Pi$ och sådana matriser är just projektioner på underrum.

Vi får då också de skattade residualerna $\widehat{\boldsymbol{r}} = \boldsymbol{y} - \widehat{\boldsymbol{y}}$. Dessa kan också skrivas som

$$\widehat{\boldsymbol{r}} = (I - A(A^TA)^{-1}A^T)\boldsymbol{y} = (I - \Pi)\boldsymbol{y}.$$

Man ser att även $I - \Pi$ är en projektion – i detta fallet på det $(n - p)$-dimensionella underrum i R^n som är ortogonalt mot det p-dimensionella underrum som spänns av kolonnerna i A-matrisen. Vi har här hela tiden underförstått att A-matrisen har full rang, dvs att A^TA varit inverterbar. Samma resonemang som ovan rörande projektioner fungerar även om vi inte har en fullrangsmodell, men då är vissa av θ_i:na "onödiga" eftersom de kan uttryckas i de övriga.

Vid test av $H_0 : M_1$ sann mot $H_1 : M_2$ sann där $M_1 \subset M_2$ har vi projektioner Π_1 och Π_2 hörande till modell M_1 respektive M_2. Det är dessa projektioner som projicerar $\boldsymbol{y}$ på prediktorerna i respektive modell. Att $M_1 \subset M_2$ innebär att $\Pi_1\Pi_2 = \Pi_2\Pi_1 = \Pi_1$. Vi har

$$I = (I - \Pi_2) + (\Pi_2 - \Pi_1) + \Pi_1.$$

En projektion Π är en symmetrisk matris som uppfyller $\Pi^2 = \Pi$. Av detta följer direkt att egenvärdena uppfyller $\lambda^2 = \lambda$, dvs att egenvärden är antingen 0 eller 1. Om man diagonaliserar en projektion, dvs hittar en ortogonal matris U (som alltså uppfyller $UU^T = U^TU = I$) sådan att $\Pi = U^TDU$ där D är en diagonalmatris så står egenvärdena till Π på diagonalen. Om Π projicerar på ett p-dimensionellt underrum av R^n så är alltså p st egenvärden 1 och resten ($n - p$ st) 0. Man visar ganska lätt att $I - \Pi_2$, $\Pi_2 - \Pi_1$ och Π_1 alla är projektioner (kvadrera dem) och att de dessutom projicerar på ortogonala underrum (multiplicera dem med varandra) med dimensioner $n - p_2$, $p_2 - p_1$ respektive p_1. Man kan också finna en enda ortogonal transformation som diagonaliserar alla tre projektionerna samtidigt på så sätt att de resulterande diagonalmatriserna har "icke-överlappande" 1:or. Om $\boldsymbol{\epsilon}$ består av n oberoende $\mathrm{N}(0, \sigma^2)$ så gäller att $U\boldsymbol{\epsilon}$ också gör det om U är en ortogonal transformation.

Alltså gäller att

$$\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^T(I - \Pi_2)\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^T(\Pi_2 - \Pi_1)\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^T(\Pi_1)\boldsymbol{\epsilon}.$$

där vänsterledet är $\sigma^2\chi(n)$ och termerna i högerledet är oberoende $\sigma^2\chi(n-p_2)$, $\sigma^2\chi(p_2 - p_1)$ och $\sigma^2\chi(p_1)$. Detta utgör den s k Cochrans sats som motiverar förekomsten av F-fördelningen vid test av M_1 mot M_2.

16.4.1 Teoretiskt mått på prediktionsförmåga

För att mäta hur väl $\widehat{Y}$ predikterar Y' väljer vi en förlustfunktion $Loss(Y', \widehat{Y})$ som mäter "kostnaden" för att prediktera Y' med $\widehat{Y}$. Denna kostnad är ju stokastisk och därför betraktar vi den förväntade förlusten $E(Loss(Y', \widehat{Y}))$ som ett mått på prediktionens kvalitet som vi kallar prediktionsfelet.

Ofta väljer man en kvadratisk förlustfunktion dvs $Loss(Y', \widehat{Y}) = (Y' - \widehat{Y})^2$ och mäter medelvärdet av det kvadratiska prediktionsfelet dvs tar

$$Q_{pred} = E(Loss(Y', \widehat{Y})) = E\left((Y' - \widehat{Y})^2\right).$$

När vi har n observationer $\boldsymbol{y}$ använder vi i stället

$$Q_{pred} = \frac{1}{n}\sum_{i=1}^{n} E\left((Y_i' - \widehat{Y}_i)^2\right)$$

som teoretiskt mått på prediktionsförmågan, dvs ett medelvärde av de kvadrerade förväntade prediktionsfelen för de enskilda observationerna. Detta förutsätter att vi ser alla observationerna som lika osäkra och likvärdiga.

Exempel 16.5 *Kända väntevärden*
Man kan notera att om vi kände $E(Y_i),\ i = 1, 2, \cdots, n$ skulle man med $\widehat{\boldsymbol{Y}} = (E(Y_1), E(Y_2), \cdots, E(Y_n))^T$ erhålla ($\boldsymbol{Y}'$ och $\boldsymbol{Y}$ har ju samma fördelning)

$$Q_{pred} = \frac{1}{n}\sum_{i=1}^{n} V(Y_i')$$

dvs om alla observationerna har varians σ^2 får vi $Q_{pred} = \sigma^2$. Detta avspeglar att om vi känner väntevärdena för våra observationer i den "sanna" modellen så är prediktionsfelet just σ^2. I allmänhet känner vi inte $E(Y_i)$ för $i = 1, 2, \cdots, n$ utan måste skatta dessa. □

Exempel 16.6 *Maximalt antal parametrar* Om man använder Y_i själv som prediktor dvs tar $\widehat{Y}_i = Y_i$ så får vi det teoretiska prediktionsfelet

$$Q_{pred} = \frac{1}{n}\sum_{i=1}^{n} E\left((Y_i' - \widehat{Y}_i)^2\right) = E\left((Y_1' - Y_1)^2\right) =$$

$$= E\left(((Y_1' - E(Y_1)) + (Y_1 - E(Y_1))^2\right) = \text{oberoendet} = V(Y_1') + V(Y_1) = 2\sigma^2.$$

Att använda $\widehat{Y}_i = Y_i$ innebär i princip att vi har n parametrar i vår modell, dvs en för varje observation. Detta ger ju "perfekt" anpassning för våra ursprungliga observationer, men genom att vi använder prediktionsfelet som mått på om modellen är bra ser vi att detta kriterium "straffar" ett sådant överparametriserande. □

Med vettigare prediktioner kan vi få ett mindre prediktionsfel än ovanstående (groteskt) överparametriserade förfarande i exempel 16.6. Detta ger en indikation på att användning av prediktionsförmåga är ett vettigt sätt att utvärdera en modell.

Exempel 16.7 *Oberoende likafördelade observationer*
Om vi t ex har modellen att Y_i:na är oberoende likafördelade med $E(Y_i) = \theta$ så kan vi använda prediktorn $\widehat{Y}_i = \bar{Y}$ och erhåller prediktionsfelet

$$Q_{pred} = \frac{1}{n}\sum_{i=1}^{n} E\left((Y_i' - \bar{Y})^2\right) = E\left((Y_1' - \bar{Y})^2\right) = \text{oberoendet} =$$

$$= V(Y_1') + V(\bar{Y}) = \sigma^2 + \frac{\sigma^2}{n} = (1 + \frac{1}{n})\sigma^2$$

dvs ett väsentligt mindre prediktionsfel än med den överparametriserade modellen ovan där vi tog $\widehat{Y}_i = Y_i$.

Faktum är att man t o m kan erhålla mindre prediktionsfel än $(1 + 1/n)\sigma^2$ genom att välja en prediktor som inte är väntevärdesriktig. Detta har samband med "Steins paradox" och så kallade "shrinkage estimatorer".

Vi kan t ex ta prediktorn $\widehat{Y}_i = a\bar{Y}$ och välja a så att förväntade prediktionsfelet minimeras. För vissa val av $a < 1$ kan man faktiskt uppnå att få ett mindre förväntat prediktionsfel än om man använder $\bar{Y}$. □

16.4.2 Andra förlustfunktioner

Man kan även använda andra förlustfunktioner t ex $Loss(Y, \widehat{Y}) = |Y - \widehat{Y}|$ men detta kommer vi inte alls att göra trots att den ofta har bättre statistiska egenskaper än kvadratisk förlustfunktion. Den är dock mycket mer komplicerad att använda rent matematiskt.

Däremot kommer vi att använda oss av

$$Loss(\boldsymbol{Y'}, \widehat{\boldsymbol{Y}}) = -\frac{1}{n} \ln L(\widehat{\boldsymbol{\theta}}|\boldsymbol{Y'})$$

där L är likelihoodfunktionen för $\boldsymbol{Y'}$ baserad på de prediktioner som erhållits ur de ursprungliga variablerna (via $\widehat{\boldsymbol{\theta}}$ som ju är en funktion av $\boldsymbol{Y}$). Vi mäter alltså prediktionsförmågan genom att studera hur sannolika de nya observationerna är på basis av de förutsägelser som gjorts med hjälp av de ursprungliga observationerna.

Man kan notera att för oberoende observationer med kontinuerliga fördelningar med tätheter f_{Y_i} betyder detta att vi har

$$\ln L(\boldsymbol{\theta}|\boldsymbol{y}) = \sum_{i=1}^{n} \ln f_{Y_i}(y_i|\boldsymbol{\theta})$$

och att alltså vi får

$$Q_{pred} = -\frac{1}{n} \sum_{i=1}^{n} E\left(\ln f_{Y_i}(Y_i'|\widehat{\boldsymbol{\theta}})\right)$$

Notera här att $\widehat{\boldsymbol{\theta}}$ är stickprovsvariabeln dvs $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}(\boldsymbol{Y})$. Vi skall alltså se det som att Q_{pred} skall beräknas över två tänkta framlottningar dvs för

1) $\boldsymbol{Y}$ (som i sig innefattar slumpmässigheten i våra observationer) och ger upphov till stickprovsvariabeln $\widehat{\boldsymbol{\theta}}$

2) $\boldsymbol{Y'}$ som innefattar slumpmässigheten i de nya observationer vi skulle få om hela försöket upprepades.

Exempel 16.8 *Allmänna linjära modellen*
Med modellen $\boldsymbol{Y} = A\boldsymbol{\theta} + \boldsymbol{\epsilon}$ där $\boldsymbol{\epsilon}$ består av oberoende $\mathrm{N}(0, \sigma^2)$ erhåller vi

$$L(\boldsymbol{\theta}|\boldsymbol{Y'}) = f_{\boldsymbol{Y}}(\boldsymbol{Y'}|\boldsymbol{\theta}) = \frac{1}{(\sqrt{2\pi})^n \sigma^n} \exp\left(-\frac{\sum_{i=1}^{n}(Y_i' - E(Y_i))^2}{2\sigma^2}\right).$$

Vi får alltså

$$\ln L(\widehat{\boldsymbol{\theta}}|\boldsymbol{Y'}) = -n\ln(\sqrt{2\pi}) - \frac{n}{2}\ln(\widehat{\sigma}^2) - \frac{\sum_{i=1}^{n}(Y_i' - \widehat{Y}_i)^2}{2\widehat{\sigma}^2}$$

med

$$\widehat{\sigma}^2 = \frac{\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2}{n}$$

(dvs ML-skattningen av σ^2).

Vi erhåller

$$Q_{pred} = \ln(\sqrt{2\pi}) + \frac{1}{2}E\left(\ln(\widehat{\sigma}^2)\right) + E\left(\frac{\sum_{i=1}^{n}(Y_i' - \widehat{Y}_i)^2}{2n\widehat{\sigma}^2}\right).$$

□

Exempel 16.9 *Exponentialfördelade data*
Om Y_i är Exp(θ) för $i = 1, 2, \cdots, n$, dvs att

$$f_{Y_i}(u) = \frac{1}{\theta}e^{-u/\theta},\ u > 0$$

och skattar θ med $\widehat{\theta} = \bar{y}$ så är alltså prediktorn $\widehat{Y}_i = \bar{Y}$.

Med hjälp av $\bar{Y}$ försöker vi alltså prediktera en ny uppsättning observationer $\boldsymbol{Y'} = (Y_1', Y_2', \cdots, Y_n')^T$ från Exp(θ)-fördelningen

Vi har här oberoende observationer med Exp(θ)-fördelningens täthetsfunktion dvs de nya observationerna har logaritmerade tätheten

$$\ln f_{Y_i'}(u|\theta) = -\ln(\theta) - u/\theta$$

och vid prediktion av Y_i' använder vi oss av $\widehat{Y}_i = \bar{Y}$. Vi får alltså måttet på förlusten till

$$Loss(\boldsymbol{Y'}, \widehat{\boldsymbol{Y}}) = -\frac{1}{n}\sum_{i=1}^{n}\ln f_{Y_i}(Y_i'|\bar{Y}) = \ln(\bar{Y}) + \bar{Y}'/\bar{Y}$$

och måttet på prediktionsförmågan Q_{pred} blir förväntade värdet av denna stokastiska förlust, dvs (notera att $\boldsymbol{Y'}$ och $\boldsymbol{Y}$ är oberoende och att $E(\bar{Y}') = \theta$)

$$Q_{pred} = E(Loss(\boldsymbol{Y'}, \widehat{\boldsymbol{Y}})) = E(\ln(\bar{Y})) + \theta E(\frac{1}{\bar{Y}}).$$

Detta uttryck är besvärligt att beräkna (och är kanske inte ens ändligt) men detta kommer inte att bekymra oss. Problemen med att det eventuellt är oändligt kan ses som ett resultat av att vi bildat medelvärde över en tänkt uppsättning ursprungsdata, där vissa sådana uppsättningar kan ställa till matematiska problem vid beräkningen av Q_{pred}. I praktiken måste vi ju skatta Q_{pred} på basis av våra observationer och då försvinner en del av dessa matematiska problem. □

16.5 Skattade mått på prediktionsförmågan

Vi såg i föregående avsnitt att man kan ta fram teoretiska mått på prediktionsförmågan hos en modell, men dels var kalkylen inte alldeles lätt och dels innehåller svaret typiskt okända storheter. I exemplet med exponentialfördelade data ingick θ både direkt och indirekt (genom att $E(\ln(\bar{Y}))$ och $E(1/\bar{Y})$ beror av θ). I allmänhet har vi dessutom ingen möjlighet att utföra motsvarigheten till kalkylerna ovan. Hur kan vi använda detta kriterium – vi har ju ingen möjlighet att generera nya data!? En traditionell metod är att använda de observationer vi redan fått och studera skillnaden mellan dessa och de prediktioner som modellen ger på basis av samtliga observationer. I princip använder man sig av

$$\widehat{Q}_{pred} = \frac{|\boldsymbol{y} - \widehat{\boldsymbol{y}}|^2}{n} = \frac{\sum_{i=1}^{n} \widehat{r}_i^2}{n} = \widehat{\sigma}^2$$

där $\widehat{r}_1, \widehat{r}_2, \cdots, \widehat{r}_n$ är residualerna dvs $\widehat{r}_i = y_i - \widehat{y}_i$.

I allmänna linjära modellen tar vi

$$\widehat{Q}_{pred} = \frac{|\boldsymbol{y} - A\widehat{\boldsymbol{\theta}}|^2}{n} = \frac{\sum_{i=1}^{n} \widehat{r}_i^2}{n} = \widehat{\sigma}^2$$

där $\widehat{\boldsymbol{y}} = A\widehat{\boldsymbol{\theta}} = A(A^TA)^{-1}A^T\boldsymbol{y}$.

Som bekant är denna (sedd som skattning av σ^2) inte väntevärdesriktig i den linjära modellen och det traditionella sättet att försöka korrigera detta är att modifiera den så att den blir väntevärdesriktig genom att välja

$$\widehat{Q}'_{pred} = s^2 = \frac{1}{n-p}\sum_{i=1}^{n} \widehat{r}_i^2 = \frac{1}{n-p}\sum_{i=1}^{n} (y_i - \widehat{y}_i)^2$$

där p=antalet parametrar i modellen.

Man kan här notera att inte ens $\widehat{Q}'_{pred}$ är en väntevärdesriktig skattning av prediktionsfelet utan detta skulle i stället vara $\widehat{Q}_{pred} = E((Y' - \widehat{y})^2)$ eller snarare, om man vill göra detta för de n observationer vi fått,

$$\widehat{Q}_{pred} = \frac{1}{n}\sum_{i=1}^{n} E((Y'_i - \widehat{y}_i)^2)$$

där Y'_i:na är variabler som har samma fördelning som Y_i :na som vi såg våra data som framlottade av.

För just den allmänna linjära modellen visar vi nedan att

$$\widehat{Q}_{pred} = s^2(1 + \frac{p}{n}) = \frac{\sum_{i=1}^{n}(y_j - \widehat{y}_j)^2}{n} \cdot \frac{n+p}{n-p}$$

är en väntevärdesriktig skattning av prediktionsfelet dvs att $E(\widehat{Q}_{pred}) = Q_{pred}$.

Detta inses genom att vi ser att $\boldsymbol{Y} = A\boldsymbol{\theta} + \boldsymbol{\epsilon}$ och att $\widehat{\boldsymbol{\theta}} = (A^TA)^{-1}A^T\boldsymbol{Y}$ dvs att

$$\widehat{\boldsymbol{Y}} = A(A^TA)^{-1}A^T\boldsymbol{Y} = A\boldsymbol{\theta} + A(A^TA)^{-1}A^T\boldsymbol{\epsilon}.$$

Om vi nu skulle generera nya observationer $\boldsymbol{Y'}$ med samma designmatris A så skulle vi erhålla $\boldsymbol{Y'} = A\boldsymbol{\theta} + \boldsymbol{\nu}$ där $\boldsymbol{\nu}$ är nya oberoende slumpstörningar som alltså är oberoende $\mathrm{N}(0, \sigma^2)$. Alltså skulle vi få det förväntade prediktionsfelet

$$Q_{pred} = \frac{1}{n}E\left(|\boldsymbol{Y'} - \widehat{\boldsymbol{Y}}|^2\right) = \frac{1}{n}E\left(|A\boldsymbol{\theta} + \boldsymbol{\nu} - A\boldsymbol{\theta} - A(A^TA)^{-1}A^T\boldsymbol{\epsilon}|^2\right).$$

P g a att $\boldsymbol{\nu}$ och $\boldsymbol{\epsilon}$ är oberoende ger detta (Pythagoras sats)

$$Q_{pred} = \frac{1}{n}\left(E(|\boldsymbol{\nu}|^2) + E(|A(A^TA)^{-1}A^T\boldsymbol{\epsilon}|^2)\right) = \frac{1}{n}(n\sigma^2 + p\sigma^2) = \sigma^2(1 + \frac{p}{n})$$

där vi använt att $A(A^TA)^{-1}A^T$ är en projektion på det p-dimensionella underrum som spänns av kolonnerna i A.

Detta betyder alltså att man i den allmänna linjära modellen kan skatta prediktionsfelet väntevärdesriktigt genom att ta

$$\widehat{Q}_{pred} = s^2(1 + \frac{p}{n}) = \frac{1 + \frac{p}{n}}{(n-p)}\sum_{i=1}^{n}(y_j - \widehat{y}_j)^2.$$

Eftersom (åtminstone om p/n är litet)

$$\frac{1 + \frac{p}{n}}{n-p} \approx \frac{1}{1 - \frac{p}{n}} \cdot \frac{1}{n-p} = \frac{n}{(n-p)^2} = \frac{n}{n^2 - 2pn + p^2} \approx \frac{1}{n-2p}$$

så ser vi att

$$\widehat{Q}_{pred} \approx \frac{\sum_{i=1}^{n}(y_j - \widehat{y}_j)^2}{n-2p}.$$

Kriteriet att minimera prediktionsfelet innebär att vi alltså bör välja den modell som ger minsta värdet på $\widehat{Q}_{pred}$. Man kan här notera att nämnaren $n-2p$ ger lite större "straff" för större modeller (dvs med större p) än den traditionella "väntevärdesjusterade" skattningen s^2 som har divisor $(n-p)$. Detta ger alltså en möjlighet att i den allmänna linjära modellen jämföra prediktionsförmågan hos olika modeller oavsett om dessa är specialfall av varandra eller ej.

Man skulle kunna uttrycka det som att vi tar $s^2 = RSS/(n-p)$ som ett sätt att göra s^2 väntevärdesriktig som skattning av σ^2, men eftersom vi vill skatta $Q_{pred} = \sigma^2(1 + p/n)$ är faktorn $(1 + p/n)$ (approximativt att ta $RSS/(n-2p)$) ett sätt att väntevärdesjustera för att det är förväntat prediktionsfel vi försöker skatta.

16.6 Korsvalidering

Ett typiskt fenomen vid skattning är att genom att vi använder (bl a) observationen y_i för att erhålla prediktorn $\widehat{y}_i$ så erhålles en överdrivet god anpassning för y_i (ibland kallat "over-fitting"). Att väntevärdesjustera ML-skattningen $\widehat{\sigma}^2 = RSS/n$ genom att ta $s^2 = RSS/(n-p)$ kan ses som ett försök att kompensera för detta.

Korsvalidering innebär att man undviker "over-fitting" genom att låta prediktorn $\widehat{y}_i$ bero av alla observationer utom just y_i. Detta betyder att vi kommer att använda datamängden $ES_i = (y_1, y_2, \cdots, y_{i-1}, y_{i+1}, \cdots, y_n)$ för att erhålla prediktorn $\widehat{y}_i$. Det innebär att vi får beräkna en skattning $\widehat{\boldsymbol{\theta}}_{(-i)}$ baserad på ES_i dvs samtliga data utom just y_i. Denna parameterskattning ger upphov till prediktorn $\widehat{y}_{(-i)}$ av y_i. Korsvaliderings-skattningen av prediktionsfelet är då

$$\widehat{Q}_{CV} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_{(-i)})^2 = \frac{PRESS}{n}$$

där $PRESS$ står för "Predictive Residual Error Sum of Squares". Man kan notera att man för att beräkna $\widehat{Q}_{CV}$ måste beräkna n st parameter-skattningar $\widehat{\boldsymbol{\theta}}_{(-i)},\ i = 1, 2, \cdots, n$ för att erhålla prediktorerna $\widehat{y}_{(-i)},\ i = 1, 2, \cdots, n$. Detta innebär n gånger det arbete som åtgår då man skattar traditionellt med hjälp av samtliga data och detta motiverar att korsvalidering kan betecknas som en datorintensiv metod.

En av de stora fördelarna med korsvalidering som metod är att den (förutom att ha bra statistiska egenskaper vad gäller att skatta prediktionsfelet för en fix given modell) dessutom är helt "automatisk". För en viss modell kan man se skattningsförfarandet (och därmed prediktionsförfarandet) som en svart låda som utifrån observationer genererar skattningar av parametrar (och därmed också prediktioner av observationerna). Egentligen behöver man inte ha någon som helst kunskap om hur prediktionerna genereras utifrån data utan det centrala är bara att det finns en algoritm som gör detta. I detta avseende liknar korsvalidering bootstrap-metodiken. Å andra sidan liknar korsvalidering jackknife genom att man i varje skattning utelämnar precis en observation.

16.6.1 Korsvalidering för allmänna linjära modellen

I detta avsnitt visar vi att en användning av korsvalidering i den allmänna linjära modellen (asymptotiskt) ger upphov till en väntevärdesriktig skattning av prediktionsfelet, dvs att $E(\widehat{Q}_{CV}) \approx Q_{pred}$ där Q_{pred} är det sanna teoretiska prediktionsfelet $\sigma^2(1 + p/n)$ där p är antalet parametrar i modellen.

Sats 16.1 *Det gäller att*

$$y_i - \widehat{y}_{(-i)} = \frac{y_i - \widehat{y}_i}{1 - h_{ii}}$$

där h_{ii} är i:te diagonalelementet i matrisen $H = A^T(A^TA)^{-1}A^T$ dvs matrisen som ger prediktorerna $\widehat{\boldsymbol{y}} = H\boldsymbol{y}$.
Beviset utlämnas men bygger på Sherman-Morrison-Woodburys formel

$$(B + UV^T)^{-1} = B^{-1} - B^{-1}U(I + V^TB^{-1}U)^{-1}V^TB^{-1}$$

där B är en inverterbar $p \times p$-matris och U och V är $p \times k$ (se t ex Golub & van Loan "Matrix Computations" sid 50). □ □

Vi lägger på tilläggsvillkoret att $\max_i h_{ii} \to 0$ då $n \to \infty$ dvs i princip att ingen prediktor $\widehat{y}_i$ skall ha stor varians om vi har många observationer.

$h_{ii}\sigma^2$ är faktiskt $V(\widehat{Y}_i)$. Detta inses ur följande resonemang:

1) $\widehat{\boldsymbol{Y}} = H\boldsymbol{Y}$

2) $\boldsymbol{Y}$ har kovariansmatrisen $\sigma^2 I$

3) $H\boldsymbol{Y}$ har enligt momentsatsen kovariansmatrisen $HV(\boldsymbol{Y})H^T = \sigma^2 HH^T$

4) H är en projektion dvs $H^T = H$ och $H^2 = H$

Alltså gäller att $V(\widehat{Y}_i) = \sigma^2 h_{ii}$.

Vi har då

$$\widehat{Q}_{CV} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_{(-i)})^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \widehat{y}_i)^2}{(1-h_{ii})^2}.$$

Eftersom $1/(1-h_{ii})^2 \approx 1 + 2h_{ii}$ då h_{ii} är litet erhåller vi (med $\widehat{r}_i = y_i - \widehat{y}_i$) och $\widehat{\sigma}^2 = \sum_{i=1}^{n}\widehat{r}_i^2/n$)

$$\widehat{Q}_{CV} \approx \frac{1}{n}\sum_{i=1}^{n}\widehat{r}_i^2 + \frac{2}{n}\sum_{i=1}^{n}h_{ii}\widehat{r}_i^2 = \widehat{\sigma}^2 + \frac{2}{n}\sum_{i=1}^{n}h_{ii}\sigma^2 + \frac{2}{n}\sum_{i=1}^{n}h_{ii}(\widehat{r}_i^2 - \sigma^2).$$

Vi har $\widehat{\sigma}^2 = (n-p)s^2/n$. Vidare gäller $\sum_1^n h_{ii} = \text{Trace}(H) = p$ eftersom $H = A(A^TA)^{-1}A^T$ är en projektion på det p-dimensionella rummet som spänns av kolonnerna i A vilket redan användes i kalkylen i avsnitt 16.5. Den sista termen i uttrycket kan visas $\to 0$ då $n \to \infty$ eftersom $\max(h_{ii}) \to 0$ och $\widehat{r}_i^2 \to \sigma^2$. Vi får alltså

$$\widehat{Q}_{CV} \approx \widehat{\sigma}^2 + \frac{2p}{n}\sigma^2 + \text{lite krafs} = s^2\frac{n-p}{n} + \frac{2p}{n}\sigma^2 + \text{lite krafs} \approx s^2(1 + \frac{p}{n}) \approx$$

$$\approx \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{n-2p}$$

vilket var precis den skattning vi kom fram till tidigare. Alltså ger (asymptotiskt då $n \to \infty$) korsvalideringen i allmänna linjära modellen samma skattning av prediktionsfelet som divisionen med $(n-2p)$ gav.

x_1	x_2	x_3	x_4	y
7	26	6	60	78.5
1	29	15	52	74.3
11	56	8	20	104.3
11	31	8	47	87.6
7	52	6	33	95.9
11	55	9	22	109.2
3	71	17	6	102.7
1	31	22	44	72.5
2	54	18	22	93.1
21	47	4	26	115.9
1	40	23	34	83.8
11	66	9	12	113.3
10	68	8	12	109.4

Tabell 16.1: Prognostiska variabler x_1, x_2, x_3, x_4 och responsvariabel y

Exempel 16.10 *Multipel regression – exempel från Urban Hjorths bok*
Vi har förklaringsvariabler och observationer enligt tabell 16.1:
Vi har här alltså $n = 13$ och maximalt $p = 5$ (inklusive intercept). Den fullständiga modellen är alltså

$$Y_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \epsilon_i$$

med $\boldsymbol{\theta} = (\alpha, \beta_1, \beta_2, \beta_3, \beta_4)^T$ och det finns $2^4 = 16$ modeller som innehåller interceptet α. Följande tabell 16.2 ger för olika val av uppsättningar av x-variablerna värdet av $\widehat{Q}_{CV}$ och för jämförelse $s^2 = RSS/(n-p)$. Totalt 15 modeller innehåller någon x-variabel.
Man ser ur tabell 16.2 för det första att s^2 är väsentligt mindre än $\widehat{Q}_{CV}$ för samtliga modeller, vilket avspeglar att $\widehat{Q}_{CV}$ skattar prediktorerna utan att använda respektive data. Approximativt gäller också att

$$\widehat{Q}_{CV} \approx s^2(1 + p/n) = s^2(1 + p/13).$$

T ex gäller för modellen med x_1 och x_4 att $s^2 = 7.48$ och vi får $7.48(1+3/13) = 9.21$ medan $\widehat{Q}_{CV} = 9.33$ - att approximationen är så dålig beror ju att vi har rätt få data i förhållande till antalet parametrar.

Man ser också att den "bästa" modellen, dvs den som har minst $\widehat{Q}_{CV}$ är modellen med x_1, x_2 och x_4 och om man skattar med hjälp av samtliga data erhålls det skattade regressionsplanet

$$y = 71.6 + 1.45x_1 + 0.42x_2 - 0.24x_4$$

och det är alltså för denna modell vi har skattningen $\widehat{Q}_{CV} = 6.57$ av prediktionsfelet. □

x_1	x_2	x_3	x_4	$\widehat{Q}_{CV}$	s^2
*	*	*	*	8.49	5.99
*	*	*		6.92	5.35
*	*		*	6.57	5.33
*		*	*	7.27	5.65
	*	*	*	11.30	8.20
*	*			7.22	5.79
*		*		170.62	122.71
*			*	9.33	7.48
		*	*	22.62	17.57
	*	*		53.48	41.54
	*		*	112.45	86.89
*				130.74	115.06
	*			92.47	82.39
		*		201.26	176.31
			*	91.86	80.35

Tabell 16.2: Skattning av prediktionsfel med korsvalidering samt variansskattning för samtliga 15 modeller med någon x-variabel samt intercept. $*$ betecknar vilka variabler som ingår i respektive modell.

I ovanstående exempel undersökte vi samtliga regressionsmodeller men om antalet tänkbara förklaringsvariabler (dvs $p-1$) är stort kan detta bli oöverstigligt eftersom det finns 2^{p-1} sådana modeller. Man kan då välja ett stegvist förfarande så att man t ex startar med den fullständiga modellen och sen plockar bort en förklaringsvariabel i taget (lämpligen den med minst förklaringsvärde). Detta brukar kallas "Backward Elimination". Man får då en uppsättning regressionsmodeller som kan jämföras med korsvalidering. Ett annat alternativ är "Forward selection" där man startar med en modell med bara ett intercept och sedan successivt inkluderar de förklaringsvariabler som har störst förklaringsvärde. Denna typ av stegvisa förfaranden är dock otillfredsställande eftersom man kan missa en modell som är utmärkt genom att de första stegen går i fel riktning. Med datorteknikens utveckling är dessa stegvisa metoder mer att betrakta som en halvmesyr – man väljer ju att bara undersöka en liten del av de tänkbara modellerna.

16.7 Model selection bias

Vi kommer dock härnäst att se att ovanstående förfaranden (t ex korsvalidering alternativt att studera $RSS/(n-2p)$) som alltså ger en väntevärdesriktig skattning av prediktionsfelet för varje enskild modell ändå kommer att underskatta

prediktionsfelet för den valda modellen just därför att den valts på basis av att den minimerade prediktionsfelet! Att det är på det viset kan man inse genom följande resonemang: Genom att slumpen kan göra att det skattade prediktionsfelet för en viss modell blir "abnormt" litet kommer just denna modell att väljas. Om man då baserar skattningen av prediktionsfelet genom att använda denna modell kommer det att tendera att bli för litet.

Egentligen är essensen av ovanstående resonemang att

$$E(\min_i(X_i)) \leq \min_i(E(X_i)).$$

Vi gör ett formellt bevis, men först ett enkelt hjälpresultat.

Sats 16.2 *Om $X \geq 0$ och $P(X = 0) < 1$ så gäller att $E(X) > 0$*

Bevis: Om $X \geq 0$ och $E(X) = 0$ måste $P(X = 0) = 1$. □

Denna sats ger följande viktiga (men aningen deprimerande) resultat:

Sats 16.3 *Bias i Model-val*
Låt μ_i vara väntevärdet för prediktionsfelet för modell M_i, dvs att

$$\mu_i = E(\widehat{Q}_{M_i}).$$

Om vi nu väljer den modell $\widehat{M}$ som minimerar prediktionsfelet dvs väljer $\widehat{M}$ så att $\widehat{Q}_{\widehat{M}} = \min_i(\widehat{Q}_{M_i})$ så gäller att

$$E(\widehat{Q}_{\widehat{M}}) < \min_i \left(E(\widehat{Q}_{M_i}) \right) \leq \mu_{\widehat{M}}$$

dvs förväntade prediktionsfelet för den valda modellen ger mindre förväntat prediktionsfel än samtliga modeller inklusive den valda!

Bevis:
Vi låter $X_i = \widehat{Q}_{M_i} - \widehat{Q}_{\widehat{M}} \geq 0$. Hjälpresultatet ger då att

$$E(X_i) = E(\widehat{Q}_{M_i}) - E(\widehat{Q}_{\widehat{M}}) > 0$$

om inte $\widehat{Q}_{\widehat{M}} = \widehat{Q}_{M_i}$ med sannolikhet 1, villket skulle svara mot att vi alltid skulle välja en viss modell M_i oavsett hur våra observationer än ser ut!

Alltså gäller att $E(\widehat{Q}_{\widehat{M}}) < E(\widehat{Q}_{M_i})$ för alla i alltså även $E(\widehat{Q}_{\widehat{M}}) < \mu_{\widehat{M}}$, dvs att det det förväntade prediktionsfelet för den valda modellen t o m underskattar det förväntade prediktionsfelet för just denna modell om den nu är den sanna modellen. Även för den sanna modellen kan det finnas skenbart bättre alternativ som kommer att väljas om data råkar bli sådana att de passar något av dessa alternativ. □

Med risk för övertydlighet och upprepning: En intuitiv förklaring till denna "Model selection"-effekt är att även om vi antar ett en viss modell är korrekt, så finns vissa tänkbara "konstiga" utfall av observationerna som faktiskt skulle göra att en annan modell bättre förklarar data än den "sanna" modellen. Om vi får sådana "konstiga" observationer skulle vi ju välja den (felaktiga) modell som skenbart förklarar dessa, dvs en annan model än den "sanna". Om nu sannolikheten att få sådana "konstiga" observationer är > 0 kommer det skattade prediktionsfelet att bli mindre än för den sanna modellen. För de "konstiga" observationerna väljer vi en annan modell än den "sanna" och lyckas därför få mindre förväntat prediktionsfel.

Denna "model selection bias" fungerar oavsett hur vi försöker skatta prediktionsförmågan om detta görs för varje enskild modell för sig. I avsnitt 16.10 på sidan 217 skall vi se att det finns en metod CMV (Cross Model Validation) som undviker denna effekt genom att man inte utvärderar varje enskild modell för sig utan hela modelvalsförfarandet.

16.8 AIC – Akaike Information Criterion

16.8.1 Allmänt om "penalized" likelihood

Ett alternativt angreppssätt på problemet att vi använder oss av observationerna både för att beräkna (skatta) prediktorerna och för utvärdering av dessa är att "straffa" modeller med många parametrar genom att som kriterium för "optimalitet" av en modell med p parametrar och n observationer ta $\widehat{Q}_{pred} + g(p, n)$. Ett vanligt sådant val är att ta

$$\widehat{Q}_{AIC} = \frac{1}{n}\Big(-\ln L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}) + p\Big) = -\frac{1}{n}\ln L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}) + \frac{p}{n}$$

kallat AIC (Aikaike Information Criterion)) respektive

$$\widehat{Q}_{BIC} = -\frac{1}{n}\ln L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}) + \frac{p\ln(n)}{2n}$$

kallat BIC (Bayesian Information Criterion) som alltså använder likelihoodfunktionen för de observationer vi fått, beräknad i modellen med parameterskattningen $\widehat{\boldsymbol{\theta}}$ som erhålls med samtliga observationer. Den extra termen p/n respektive $p\ln(n)/(2n)$ kan ses som ett sätt att "straffa" modeller med många parametrar, dvs med för stort p. Dessutom ser man att BIC (åtminstone då $n > e^2$ dvs $n \geq 9$) straffar modeller med många parametrar (p stor) hårdare än AIC. BIC undersöks i nästa avsnitt.

Vi kommer att se att AIC i princip innebär att vi justerar $-\ln L$ med hänsyn till att vi använder samma data för att skatta parametrarna och för att utvärdera prediktorn. Denna korrektion innebär att $\widehat{Q}_{AIC}$ kommer att bli en (approximativt) väntevärdesriktig skattning av prediktionsfelet när vi har $-\ln L$ som förlustfunktion. Detta kommer att göras troligt för den allmänna linjära

modellen, men resultatet gäller mer allmänt än så. Notera att detta att försöka minimera med förlust-funktionen $-\ln L$ för (tänkta) nya data är samma sak som att försöka maximera (förväntade) likelihooden L för de tänkta nya observationerna.

Dock kommer vi också att se att AIC inte kommer att ge oss den "sanna" modellen ens då $n \to \infty$ dvs att sannolikheten att den valda modellen $\widehat{M}$ enligt AIC kommer inte att uppfylla att $P(\widehat{M} \neq M_{sann}) \to 0$ då $n \to \infty$ där M_{sann} är den sanna modellen. Alltså är inte AIC ett "konsistent" modellvalskriterium.

BIC kan motiveras med ett Bayesiansk baserat resonemang och kommer också att (asymptotiskt då $n \to \infty$) att välja den "sanna" modellen M_{sann} med en sannolikhet som $\to 1$. BIC är alltså ett konsistent modellvalskriterium.

16.8.2 AIC

Med

$$\widehat{Q}_{AIC} = -\frac{1}{n}\ln L + \frac{p}{n} = -\frac{1}{n}\ln L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}) + \frac{p}{n} = -\frac{1}{n}\ln f_{\boldsymbol{Y}}(\boldsymbol{y}|\widehat{\boldsymbol{\theta}}) + \frac{p}{n}$$

erhåller vi för den allmänna linjära modellen där $\boldsymbol{Y} = A\boldsymbol{\theta} + \boldsymbol{\epsilon}$ med $\boldsymbol{\epsilon}$ bestående av n oberoende $\mathrm{N}(0, \sigma^2)$ att vi skattar prediktorerna $\widehat{\boldsymbol{y}}$ med

$$\widehat{\boldsymbol{y}} = A(A^TA)^{-1}A^T\boldsymbol{y} = A\widehat{\boldsymbol{\theta}}$$

och σ^2 med ML-skattningen

$$\widehat{\sigma}^2 = \frac{RSS}{n} = \frac{|\boldsymbol{y} - \widehat{\boldsymbol{y}}|^2}{n} = \frac{\sum_{i=1}^n (y_i - \widehat{y}_i)^2}{n}.$$

Vi har alltså

$$f_{\boldsymbol{Y}}(\boldsymbol{y}|\widehat{\boldsymbol{\theta}}) = \prod_{i=1}^n \left(\frac{1}{\sqrt{2\pi}\widehat{\sigma}} \exp(-\frac{(y_i - \widehat{y}_i)^2}{2\widehat{\sigma}^2}) \right) =$$

$$= \frac{1}{(2\pi)^{n/2}}\widehat{\sigma}^{-n} \exp\left(-\frac{\sum_{i=1}^n (y_i - \widehat{y}_i)^2}{2\widehat{\sigma}^2} \right)$$

och detta ger

$$\widehat{Q}_{AIC} = -\frac{1}{n}\ln f_{\boldsymbol{Y}}(\boldsymbol{y}|\widehat{\boldsymbol{\theta}}) + \frac{p}{n} = \ln(\sqrt{2\pi}) + \ln(\widehat{\sigma}) + \frac{\sum_{i=1}^n (y_i - \widehat{y}_i)^2}{2n\widehat{\sigma}^2} + \frac{p}{n} =$$

$$= \ln(\sqrt{2\pi}) + \frac{1}{2}\ln(\widehat{\sigma}^2) + \frac{1}{2} + \frac{p}{n}.$$

Å andra sidan gäller att om vi studerar likelihoodfunktionen för en ny uppsättning observationer $\boldsymbol{Y}'$ med samma designmatris A som $\boldsymbol{Y}$ så ser vi att denna har den (stokastiska) likelihoodfunktionen

$$L(\widehat{\boldsymbol{\theta}}|\boldsymbol{Y}') = f_{\boldsymbol{Y}}(\boldsymbol{Y}'|\widehat{\boldsymbol{\theta}}).$$

Alltså har vi som ovan

$$-\frac{1}{n}\ln L(\widehat{\boldsymbol{\theta}}|\boldsymbol{Y}') = \ln(\sqrt{2\pi}) + \frac{1}{2}\ln(\widehat{\sigma}^2) + \frac{|\boldsymbol{Y}' - \widehat{\boldsymbol{Y}}|^2}{2n\widehat{\sigma}^2}$$

som ger väntevärdet (dvs det teoretiska värdet på förlusten Q_{pred} "medelvärdat" över $\boldsymbol{Y}$ och $\boldsymbol{Y}'$)

$$Q_{pred} = E\left(-\frac{1}{n}\ln L(\widehat{\boldsymbol{\theta}}|\boldsymbol{Y}')\right) =$$

$$= \ln(\sqrt{2\pi}) + \frac{1}{2}E(\ln(\widehat{\sigma}^2)) + \frac{1}{2}E\left(\frac{|\boldsymbol{Y}' - \widehat{\boldsymbol{Y}}|^2}{n\sigma^2}\cdot\frac{\sigma^2}{\widehat{\sigma}^2}\right).$$

Men (som visats tidigare) gäller p g a oberoendet mellan $\boldsymbol{Y}'$ och $\boldsymbol{Y}$ att

$$E(|\boldsymbol{Y}' - \widehat{\boldsymbol{Y}}|^2) = E(|\boldsymbol{Y}' - E(\boldsymbol{Y})|^2) + E(|\widehat{\boldsymbol{Y}} - E(\boldsymbol{Y})|^2) = n\sigma^2 + p\sigma^2$$

dvs att

$$E\left(\frac{|\boldsymbol{Y}' - \widehat{\boldsymbol{Y}}|^2}{n\sigma^2}\right) = (1 + \frac{p}{n}).$$

Vidare gäller ju att

$$E\left(\frac{\sigma^2}{\widehat{\sigma}^2}\right) \approx \frac{1}{E(\widehat{\sigma^2}/\sigma^2)} = \frac{1}{(n-p)/n} = \frac{1}{1 - \dfrac{p}{n}} \approx 1 + \frac{p}{n}$$

som ger den teoretiska förväntade förlusten

$$Q_{pred} \approx \ln(\sqrt{2\pi}) + \frac{1}{2}E(\ln(\widehat{\sigma}^2)) + \frac{1}{2}\cdot(1 + \frac{p}{n})\cdot\frac{1}{1 - \dfrac{p}{n}} \approx$$

$$\approx \ln(\sqrt{2\pi}) + \frac{1}{2}E(\ln(\widehat{\sigma}^2)) + \frac{1}{2} + \frac{p}{n}.$$

Detta betyder alltså att vi har $E(\widehat{Q}_{AIC}) \approx Q_{pred}$ dvs att AIC på ett (approximativt) väntevärdesriktigt sätt skattar prediktionsfelet (mätt med likelihoodfunktionen som förlustfunktion).

16.8.3 Varianter av AIC

Vi har (med Taylor-utveckling) att

$$\ln(\widehat{\sigma}^2) \approx \ln(\sigma^2) + \frac{\widehat{\sigma}^2 - \sigma^2}{\sigma^2} = \ln(\sigma^2) + \frac{\widehat{\sigma}^2}{\sigma^2} - 1$$

och alltså gäller att

$$\widehat{Q}_{AIC} = \ln(\sqrt{2\pi}) + \frac{1}{2}\ln(\widehat{\sigma}^2) + \frac{1}{2} + \frac{p}{n} \approx$$

$$\approx \frac{1}{2}\ln(2\pi) + \frac{1}{2}\left(\ln(\sigma^2) + \frac{\widehat{\sigma}^2}{\sigma^2} - 1\right) + \frac{1}{2} + \frac{p}{n} =$$

$$= \frac{1}{2}\ln(2\pi) + \frac{1}{2}\ln(\sigma^2) + \frac{1}{2\sigma^2}\left(\widehat{\sigma}^2 + \frac{2p\sigma^2}{n}\right).$$

Eftersom man erhåller samma minimerande modell vare sig man minimerar Q_{AIC} eller $g(Q_{AIC})$ för en växande funktion g så inser man att vi kan välja att minimera

$$C_p = \widehat{\sigma}^2 + \frac{2p\sigma^2}{n} = \frac{RSS}{n} + \frac{2p\sigma^2}{n}$$

som brukar kallas C_p-statistikan. Nu är ju inte σ^2 känd, men ofta ersätter man den med en skattning av σ^2 beräknad från en "stor" grundmodell. T ex kan man för multipel-regressionsmodeller hämta skattningen från modellen med alla x-variabler medan RSS alltså beräknas för den aktuella (mindre) modellen.

Ett annat val är att ersätta σ^2 med $s^2 = RSS/(n-p)$ och få

$$\widehat{Q}'_{pred} = \widehat{\sigma}^2 + \frac{2p\sigma^2}{n} = \frac{n-p}{n}s^2 + \frac{2p\sigma^2}{n} \approx s^2(1 + \frac{p}{n})$$

dvs samma storhet som dök upp i samband med korsvalidering. Detta svarar alltså (approximativt) mot att att man skall dela RSS med $n-2p$ i stället för med $n-p$ eftersom $\widehat{Q}'_{pred} \approx RSS/(n-2p)$. Man inser att det betyder att man får samma resultat med AIC och korsvalidering (åtminstone approximativt).

Man ser också att om vi tar $p = n$, dvs tar $\widehat{y}_i = y_i$, $i = 1, 2, \cdots, n$ (det som tidigare benämnts "grotesk överparametrisering" i exempel 16.6) så blir $C_p = 0 + 2\sigma^2$ som vi tidigare såg var det förväntade prediktionsfelet i denna situation.

AIC är ju rätt svårt att använda rent praktiskt i en komplicerad situation eftersom det oftast är besvärligt att skriva upp likelihoodfunktionen. Eftersom korsvalidering åtminstone för allmänna linjära modellen ger i princip samma resultat som AIC är det oftast mer praktiskt att använda korsvalidering. För C_p krävs att man har en skattning av σ^2 från en "stor" modell.

16.8.4 AIC är ej konsistent

Vi har hitills värderat modeller efter deras prediktionsförmåga. Ett naturligt alternativt sätt att undersöka modellvalskriterier är att undersöka om förfarandet (åtminstone asymptotiskt då antalet observationer $n \to \infty$) lyckas hitta den "sanna" modellen.

Det är ett beklagligt faktum att man med AIC faktiskt inte ens asymptotiskt väljer rätt modell i den mening att om man har två modeller $M_1 \subset M_2$ och M_1 är "sann" så gäller inte att $P(\widehat{M} \neq M_1) \to 0$ då $n \to \infty$ där n är antalet observationer. Om man alltså har en sann hypotesmodell M_1 med p_1 parametrar och ställer den mot en större grundmodell med p_2 parametrar så väljs alltså inte M_1 ens då antalet data $n \to \infty$.

Vi har ju (med hjälp av Taylor-utveckling av $\ln(x)$ kring a) att

$$\ln(\widehat{\sigma}^2) \approx \ln(\sigma^2) + \frac{(\widehat{\sigma}^2 - \sigma^2)}{\sigma^2}$$

och detta ger sannolikheten att välja fel modell M_2 när M_1 är sann till

$$\begin{aligned} P(\widehat{M} = M_2 | M_1 \text{ sann}) &= P\left(Q_{AIC}(M_2) < Q_{AIC}(M_1) | \, M_1 \text{ sann}\right) = \\ &= P\left(n(\ln(\widehat{\sigma}^2(M_1)) - \ln(\widehat{\sigma}^2(M_2))) > 2(p_1 - p_2) | M_1 \text{ sann}\right) \approx \\ &\approx P\left(\frac{n(\widehat{\sigma}^2(M_1) - \widehat{\sigma}^2(M_2))}{\sigma^2} > 2(p_2 - p_1) | M_1 \text{ sann}\right) = \\ &= P\left(\chi^2(p_2 - p_1) > 2(p_2 - p_1)\right). \end{aligned}$$

Tyvärr har alltså AIC en tendens att välja en för stor modell – t ex gäller för $p_2 - p_1 = 1$ att ovanstående sannolikhet blir 0.163 och detta är alltså sannolikheten att vi väljer den för stora grundmodellen M_2 i stället för den sanna hypotesmodellen M_1 även om $n \to \infty$.

16.9 BIC – Bayesian Information Criterion

Med Bayesian Information Criterion (BIC) använder man sig av

$$\widehat{Q}_{BIC} = -\frac{1}{n} \ln L(\widehat{\boldsymbol{\theta}} | \boldsymbol{y}) + \frac{p \ln(n)}{2n}$$

som mått på den skattade prediktionsförmågan.

16.9.1 Bayesiansk motivering för BIC

Man kan motivera valet av "straff" $p\ln(n)/2$ med ett Bayesianskt resonemang enligt följande:

Antag att de olika modellerna $M_1, M_2, \cdots, M_m$ har a-priori-sannolikheter $P(M_j)$, $j = 1, 2, \cdots, m$. Låt $\pi(\cdot|M_j)$ vara a-priori-fördelningen för $\boldsymbol{\theta}$ för modellen M_j.

Vi får då med Bayes' sats

$$P(M_j | \boldsymbol{y}) = \frac{f(\boldsymbol{y}|M_j) P(M_j)}{f(\boldsymbol{y})}.$$

Enligt lagen om total sannolikhet gäller

$$f(\boldsymbol{y}|M_j) = \int_{\Theta_j} f(\boldsymbol{y}|\boldsymbol{\theta}, M_j) \pi(\boldsymbol{\theta}|M_j) d\boldsymbol{\theta}$$

och vi erhåller

$$P(M_j|\boldsymbol{y}) = c_1 P(M_j) \int_{\Theta_j} f(\boldsymbol{y}|\boldsymbol{\theta}, M_j) \pi(\boldsymbol{\theta}|M_j) d\boldsymbol{\theta}$$

med

$$f(\boldsymbol{y}) = \sum_{j=1}^{m} f(\boldsymbol{y}|M_j)P(M_j)$$

och $c_1 = 1/f(\boldsymbol{y})$.

Om vi har den allmänna linjära modellen med känt σ^2 och med A_j= designmatrisen för modell M_j så får vi

$$f(\boldsymbol{y}|\boldsymbol{\theta}, M_j) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}|\boldsymbol{y} - A_j\boldsymbol{\theta}|^2\right) =$$

$$= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}|\boldsymbol{y} - A_j\widehat{\boldsymbol{\theta}}|^2\right) \exp\left(-\frac{1}{2\sigma^2}|A_j\widehat{\boldsymbol{\theta}} - A_j\boldsymbol{\theta}|^2\right) =$$

$$= L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}, M_j) \exp\left(-n(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})^T\boldsymbol{\Sigma}_j(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})\right)$$

med $\boldsymbol{\Sigma}_j = (A_j^T A_j)/(2n\sigma^2)$. Vi får då

$$P(M_j|\boldsymbol{y}) = c_1 P(M_j) L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}, M_j) \int_{\Theta_j} \exp\left(-n(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})^T\boldsymbol{\Sigma}_j(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})\right) \pi(\boldsymbol{\theta}|M_j)d\boldsymbol{\theta}.$$

Vi gör nu först variabelbytet $\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}} = \widetilde{\boldsymbol{\theta}}$ och erhåller

$$P(M_j|\boldsymbol{y}) = c_1 P(M_j) L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}, M_j) \int_{\Theta_j} \exp\left(-n\widetilde{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}_j\widetilde{\boldsymbol{\theta}}\right) \pi(\widetilde{\boldsymbol{\theta}} + \widehat{\boldsymbol{\theta}}|M_j)d\widetilde{\boldsymbol{\theta}}.$$

Vi gör sen variabelbytet $\widetilde{\boldsymbol{\theta}} = \boldsymbol{u}/\sqrt{n}$ och erhåller (notera att integralen sker över ett p_j-dimensionellt parameterrum där p_j är antalet parametrar i M_j och alltså $d\widetilde{\boldsymbol{\theta}} = d\boldsymbol{u}\, n^{-p_j/2}$)

$$P(M_j|\boldsymbol{y}) = c_1 P(M_j) L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}, M_j) \int_{\Theta_j} \exp\left(-\boldsymbol{u}^T\boldsymbol{\Sigma}_j\boldsymbol{u}\right) \pi(\boldsymbol{u}n^{-1/2} + \widehat{\boldsymbol{\theta}}|M_j)d\boldsymbol{u}\, n^{-p_j/2} \approx$$

$$\approx c_1 P(M_j) L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}, M_j)\pi(\widehat{\boldsymbol{\theta}}|M_j)n^{-p_j/2} \int_{\Theta_j} \exp(-\boldsymbol{u}^T\boldsymbol{\Sigma}_j\boldsymbol{u})d\boldsymbol{u}$$

Om vi tar $-\ln$ av detta uttryck får vi

$$-\ln(P(M_j|\boldsymbol{y})) \approx$$

$$\approx -\ln(L(\widehat{\boldsymbol{\theta}}|\boldsymbol{y}, M_j) + \frac{p_j}{2}\ln(n) - \ln\left(c_1 P(M_j)\pi(\widehat{\boldsymbol{\theta}}|M_j) \int_{\Theta_j} \exp(-\boldsymbol{u}^T\boldsymbol{\Sigma}_j\boldsymbol{u})d\boldsymbol{u}\right)$$

där den sista termen blir betydelselös då $n \to \infty$. Uttrycket övergår alltså i $n\widehat{Q}_{BIC}$.

Man kan uttrycka ovanstående på följande sätt: BIC väljer (asymptotiskt) den modell som har maximal a-posteriori-sannolikhet (givet observationerna) av $M_1, M_2, \cdots, M_m$.

16.9.2 BIC är konsistent

Med nästan precis samma kalkyl som vid undersökningen av AIC erhåller man för BIC att

$$P(\widehat{M} \neq M_1) = P\left(Q_{BIC}(M_2) < Q_{BIC}(M_1) | \, M_1 \text{ sann}\right) =$$

$$= P\left(n(\ln(\widehat{\sigma}^2(M_1)) - \ln(\widehat{\sigma}^2(M_2))) > (p_1 - p_2)\ln(n) | M_1 \text{ sann}\right) \approx$$

$$\approx P\left(\frac{n(\widehat{\sigma}^2(M_1) - \widehat{\sigma}^2(M_2))}{\sigma^2} > (p_2 - p_1)\ln(n) | M_1 \text{ sann}\right) =$$

$$= P\left(\chi^2(p_2 - p_1) > (p_2 - p_1)\ln(n)\right).$$

Då $n \to \infty$ gäller alltså att $P(\widehat{M} \neq M_1) \to 0$ då $n \to \infty$, dvs att BIC är konsistent.

16.10 CMV – Cross Model Validation

Vi såg tidigare att oavsett vilken metod vi än använde oss av för att skatta prediktionsfelet för en modell så får vi en underskattning av prediktionsfelet för den valda modellen just för att denna valdes med utgångspunkt i våra skattningar av prediktionsfelen. Vi skall här beskriva en teknik, CMV, som undviker detta problem genom att inte skatta prediktionsfelet för varje modell för sig utan bygger in hela modellvalsförfarandet i valet av modell.

Man gör detta genom att för varje modellstorlek (dvs varje p) göra följande:

1) För varje enskild observation y_j väljer vi att skatta parametrarna för samtliga modeller av storlek p och detta med hjälp av ES_j dvs samtliga observationer utom y_j. Vi väljer sedan för y_j den modell av storlek p som bäst predikterar y_j. Denna modell kallar vi $\widehat{M}(p, ES_j, y_j) = \widehat{M}(p, j)$ för att understryka att den (för ett visst val av p) beror av ES_j och observationen y_j. Prediktionen av y_j på basis av denna modell kallar vi $\widehat{y}_j(p)$, dvs vi har

$$\widehat{y}_j(p) = \widehat{y}_j(p, ES_j, y_j, \widehat{M}(p, j))$$

Man noterar att olika observationer y_j i allmänhet ger olika modeller $\widehat{M}(p, j)$ som bästa modell av storlek p.

2) Vi beräknar sedan

$$\widehat{Q}_{CMV}(p) = \frac{1}{n}\sum_{i=1}^{n}(y_j - \widehat{y}_j(p))^2.$$

Detta leder till ett värde $\widehat{Q}_{CMV}(p)$ för varje modellstorlek p och vi väljer sedan det p-värde $\widehat{p}$ som har minimerar $\widehat{Q}_{CMV}(p)$. I och med detta har vi valt modellstorlek $\widehat{p}$. Den slutgiltiga modell $\widehat{M}$ vi sedan väljer är den modell av storlek $\widehat{p}$ som är bäst i meningen att den har minst $\widehat{\sigma}^2$, dvs residualkvadratsumma. $\widehat{Q}_{CMV}(\widehat{p})$ är då skattningen av prediktionsfelet i denna modell.

Finessen med CMV är att vi inte skattar prediktionsfelet för en modell i taget utan snarare för en hel uppsättning modeller av storlek p och därigenom kan välja modellstorlek $\widehat{p}$. Notera att det är fullt tänkbart att den slutgiltiga modellen av storlek $\widehat{p}$ som slutligen väljs inte använts för alla (eller ens någon) av prediktionerna för de enskilda observationerna vid beräkningen av $\widehat{Q}_{CMV}(\widehat{p})$!

Kapitel 17

Markov Chain Monte Carlo – MCMC

17.1 Inledning

Som nämndes i kaptiel 2 om Monte Carlo simulering är det ett besvärligt problem att generera utfall av en allmän d-dimensionell fördelning med täthet $f_{\boldsymbol{X}}(\boldsymbol{x}),\ \boldsymbol{x} \in R^d$ alternativt med sannolikhetsfunktion $p_{\boldsymbol{X}}(\boldsymbol{x}),\ \boldsymbol{x} \in Z^d$.

Vid "vanlig" Monte Carlo-simulering genererade man oberoende utfall av den d-dimensionella variabeln $\boldsymbol{X} = (X_1, X_2, \cdots, X_d)$. Vid Markov Chain Monte Carlo simulering genererar man i stället beroende utfall $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \cdots$ som utgör en Markovkedja med d-dimensionellt tillståndsrum och med den åstundade fördelningen $f_{\boldsymbol{X}}(\boldsymbol{x})$ (alternativt $p_{\boldsymbol{X}}(\boldsymbol{x})$ om $\boldsymbol{X}$ är diskret) för $\boldsymbol{X}$ som stationär (och förhoppningsvis asymptotisk) fördelning.

En av de stora poängerna med detta är att man inte behöver ha normerat fördelningen. Vi vet alltså bara att

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{\int_{R^d} \pi(\boldsymbol{u}) d\boldsymbol{u}} \text{ respektive } p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{\sum_{\boldsymbol{u} \in R^d} \pi(\boldsymbol{u})}$$

för en känd funktion $\pi(\boldsymbol{x})$ då $\boldsymbol{X}$ har kontinuerlig respektive diskret fördelning. Vi vet alltså bara att $f_{\boldsymbol{X}}(\boldsymbol{x}) \propto \pi(\boldsymbol{x})$ alternativt $p_{\boldsymbol{X}}(\boldsymbol{x}) \propto \pi(\boldsymbol{x})$

I kurser för Markov-kedjor brukar övergångsmekanismen vara specificerad och man undrar över

1) finns det någon stationär fördelning?

2) är den unik?

3) hur ser den ut?

4) utgör den asymptotisk fördelning, dvs om kedjans fördelning efter lång tid konvergerar mot den stationära fördelningen oavsett starttillstånd?

Här ställs vi inför det omvända problemet att givet en "målfördelning" skapa en Markovkedja som har denna målfördelning till stationär fördelning. Detta visar sig vara förvånansvärt lätt!

Det visar sig också ha stora tillämpningar inom

1) Statistisk fysik

2) Bildanalys

3) Optimering

4) Bayesiansk statistik.

Det är framför allt att man inte behöver ha målfördelningen normerad till en sannolikhetsfördelning som är så användbart i dessa tillämpningar.

17.2 Ergod-satser

Ofta är det lätt att visa att den sålunda genererade Markovkedjan är ergodisk, dvs oavsett startvärde konvergerar den mot den (unika) stationära fördelningen (dvs π normerad)

$$\frac{\pi(\boldsymbol{x})}{\int_{R^d} \pi(\boldsymbol{u}) d\boldsymbol{u}} \text{ alternativt } \frac{\pi(\boldsymbol{x})}{\sum_{\boldsymbol{u} \in R^d} \pi(\boldsymbol{u})}.$$

Detta betyder att man kan använda ergod-satser av typen

$$\lim_{B \to \infty} \frac{1}{B} \sum_{i=1}^{B} g(\boldsymbol{x}_i) = E_\pi \left(g(\boldsymbol{X}) \right) = \int_{\boldsymbol{x} \in R^n} g(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} = \frac{\int_{\boldsymbol{x} \in R^n} g(\boldsymbol{x}) \pi(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{x} \in R^n} \pi(\boldsymbol{x}) d\boldsymbol{x}}.$$

som alltså i princip säger att "ett tidsmedelvärde för en realisering konvergerar mot motsvarande väntevärde i den asymptotiska fördelningen" (med sannolikhet 1 precis som i Stora Talens Lag).

Har vi alltså bara en tillräckligt lång realisering (så att vi vet att denna har den asymptotiska fördelningen) kan vi beräkna väntevärden för godtyckliga funktioner av den d-dimensionella variabeln genom att beräkna motsvarande aritmetiska medelvärden över den erhållna realiseringen av Markovkedjan.

Med hjälp av denna typ av ergodsatser kan en massa mycket intressanta storheter rörande fördelningen beräknas. Även hela fördelningen, fördelningen för en viss komponent i vektorn eller någon aspekt av denna, t ex väntevärde, varians, median etc. kan erhållas. Man kan alltså erhålla precis samma typ av resultat som för oberoende simuleringar enligt exempel 2.4 på sidan 18. Om den simulerade Markovkedjan kan visas vara ergodisk kan vi vid tidsmedelvärden få precis samma typer av resultat som om vi haft oberoende simuleringar.

Eftersom vi kommer att starta simuleringen (Markovkedjan) i en godtycklig punkt vill vi att den skall vara ergodisk eftersom vi då med hjälp av en lång simulering av kedjan vet att den får sin stationära fördelning (dvs "målfördelningen"). Ergod-satsen innebär att vi då från denna enda realisering kan approximativt beräkna godtyckliga aspekter av målfördelningen precis som om vi haft en realisering bestående av oberoende värden.

För att visa att en Markov-kedja med ändligt diskret tillståndsrum E är ergodisk räcker det som bekant att visa att den

1) är irreducibel (dvs att alla tillstånd kommunicerar)

2) är aperiodisk (dvs har perioden 1)

Detta kommer oftast att vara näst intill trivialt att visa i konkreta situationer!

Om kedjan har oändligt diskret tillståndsrum uppstår traditionellt frågan om kedjan överhuvudtaget har någon stationär fördelning, men eftersom vi konstruerar kedjan så att den har målfördelningen till stationär fördelning bortfaller denna komplikation.

Om kedjan har ett kontinuerligt tillståndsrum kan det vara lite mer intrikat, men är oftast ingen stor svårighet. Dessutom bortfaller i detta fall i allmänhet möjligheten till periodiskt uppträdande eftersom övergångarna sker med en kontinuerlig fördelning.

17.3 Metropolis-Hastings algoritm

Det visar sig mycket enkelt att konstruera en Markovkedja med en föreskriven stationär fördelning π även om π är onormerad. Idén är att då aktuellt tillstånd är $\boldsymbol{x}$ generera förslag på nästa värde $\boldsymbol{y}$ (givet det aktuella värdet $\boldsymbol{x}$) med hjälp av en "förslagsfördelning"

$$q(\boldsymbol{x}, \boldsymbol{y}) = P\left(\text{föreslå } \boldsymbol{y} | \text{aktuellt tillstånd är } \boldsymbol{x}\right)$$

och sen välja att acceptera eller förkasta detta förslag $\boldsymbol{y}$ med en sannolikhet avpassad så att man får rätt stationär fördelning. Om förslaget $\boldsymbol{y}$ förkastas stannar kedjan kvar i det aktuella tillståndet även vid nästa tidpunkt.

Faktum är att "förslagsfördelningen" $q(\boldsymbol{x}, \boldsymbol{y})$ (förvånande nog) kan se ut nästan hur som helst, men naturligtvis kan den väljas mer eller mindre listigt. Valet kan kraftigt påverka konvergenshastigheten. Dessutom skall den väljas så att den resulterande kedjan blir irreducibel och aperiodisk.

Denna korrekt avpassade "acceptans"-sannolikhet

$$\alpha(\boldsymbol{x}, \boldsymbol{y}) = P(\text{acceptera förslaget } \boldsymbol{y} | \text{aktuellt tillstånd är } \boldsymbol{x})$$

kan beräknas ur

$$\alpha(\boldsymbol{x}, \boldsymbol{y}) = \min\left(1, \frac{\pi(\boldsymbol{y}) q(\boldsymbol{y}, \boldsymbol{x})}{\pi(\boldsymbol{x}) q(\boldsymbol{x}, \boldsymbol{y})}\right).$$

Denna sannolikhet innehåller π-fördelningen i både täljare och nämnare och alltså behöver inte denna π-fördelning vara normerad. I fortsättningen kommer vi att låta π stå för den normerade sannolikhetsfunktionen alternativt täthetsfunktionen. Notera dock att den faktiskt inte kommer att behöva vara normerad.

Ovanstående algoritm, som kallas Metropolis-Hastings algoritm, ger uppenbarligen upphov till en Markovkedja eftersom nästa tillstånd bara beror av aktuellt tillstånd och ej av hur aktuellt tillstånd uppnåtts, vilket innebär att den har Markov-egenskapen (minneslöshet). Den blir vidare tidshomogen ty dynamiken beror ej av tiden.

Vi vill ju simulera en fördelning beskriven av $\pi(\boldsymbol{x})$, $\boldsymbol{x} \in E$ där $E = R^d$. I princip kommer vi bara att betrakta kedjor med diskreta tillståndsrum men resultaten går förhållandevis lätt att generalisera till kontinuerliga tillståndsrum. F ö innebär den tekniska implementeringen av simuleringarna att vi i praktiken alltid egentligen arbetar med diskreta tillståndsrum p g a den oundvikliga numeriska avrundningen som måste ske i en dator.

Eftersom vi vill använda oss av Markovkedje-teknik kommer vi dessutom att vilja beskriva fördelningen π med hjälp av radvektorer $\boldsymbol{\pi} = (\pi(\boldsymbol{x}), \boldsymbol{x} \in E)$. Vi identifierar alltså radvektorn $\boldsymbol{\pi}$ med fördelningen $\pi(\boldsymbol{x})$, $\boldsymbol{x} \in E$. De olika komponenterna i $\boldsymbol{\pi}$ anger sannolikheterna att kedjan befinner sig i motsvarande tillstånd.

Kedjans dynamik beskrivs av en övergångsmatris $\boldsymbol{P}$ med element $P(\boldsymbol{x}, \boldsymbol{y})$ för $\boldsymbol{x}$ och $\boldsymbol{y}$ i E där

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{X}_{n+1} = \boldsymbol{y} | \boldsymbol{X}_n = \boldsymbol{x})$$

där vi alltså antar att kedjan är tidshomogen eftersom vi antar att högerledet inte beror av n.

Vi låter som i grundkursens avsnitt om Markovkedjor övergångsmatrisen i n steg vara

$$\boldsymbol{P}^{(n)}(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{X}_n = \boldsymbol{y} | \boldsymbol{X}_0 = \boldsymbol{x})$$

och fördelningen i steg n beskrivas av radvektorn $\boldsymbol{p}^{(n)} = (p_{\boldsymbol{x}}^{(n)}, \boldsymbol{x} \in E)$ där $p_{\boldsymbol{x}}^{(n)} = P(\boldsymbol{X}_n = \boldsymbol{x})$.

Vi har som bekant Chapman-Kolmogorovs ekvationer

$$\boldsymbol{P}^{(n+m)} = \boldsymbol{P}^n \boldsymbol{P}^m$$

och

$$\boldsymbol{p}^{(n)} = \boldsymbol{p}^{(0)} \boldsymbol{P}^n.$$

Om man alltså startar kedjan med fördelningen $\boldsymbol{p}$ har den fördelningen $\boldsymbol{pP}$ efter ett tidssteg.

Man kan notera att Metropolis-Hastings algoritm innebär att för $\boldsymbol{y} \neq \boldsymbol{x}$ är $P(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{x}, \boldsymbol{y})\alpha(\boldsymbol{x}, \boldsymbol{y})$. För matriselementen $P(\boldsymbol{x}, \boldsymbol{x})$ gäller att

$$P(\boldsymbol{x}, \boldsymbol{x}) = 1 - \sum_{\boldsymbol{y} \neq \boldsymbol{x}} P(\boldsymbol{x}, \boldsymbol{y})$$

ty radsumman i en övergångsmatris summerar sig ju till 1.

Man kan observera att om kedjan ligger kvar i tillståndet $\boldsymbol{x}$ kan det bero på att antingen förslaget $\boldsymbol{y}$ förkastats eller att q-fördelningen föreslagit just

$\boldsymbol{x}$ (som görs med sannolikhet $q(\boldsymbol{x},\boldsymbol{x})$) – detta förslag accepteras då eftersom $\alpha(\boldsymbol{x},\boldsymbol{x}) \equiv 1$.
Att Metropolis-Hastings algoritm ger upphov till en Markovkedja med den avsedda fördelningen som stationär fördelning beror på att med det angivna valet på "acceptanssannolikhet" blir den resulterande Markovkedjan en tidsreversibel Markovkedja

Definition 17.1 *Tidsreversibilitet*
En Markovkedja med övergångsmatris $\boldsymbol{P} = (P(\boldsymbol{x},\boldsymbol{y}),\ \boldsymbol{x},\boldsymbol{y} \in E)$ kallas tidsreversibel med avseende på fördelningen $\pi(\boldsymbol{x}),\ \boldsymbol{x} \in E$ om

$$\pi(\boldsymbol{x})P(\boldsymbol{x},\boldsymbol{y}) = \pi(\boldsymbol{y})P(\boldsymbol{y},\boldsymbol{x}) \text{ för alla } \boldsymbol{x} \text{ och } \boldsymbol{y} \in E.$$

□

Betydelsen av begreppet "tidsreversibilitet" framgår av följande sats som innebär att om vi kan skapa en kedja som är tidsreversibel med avseende på målfördelningen så har denna kedja målfördelningen som stationär fördelning.

Sats 17.1 En kedja som är tidsreversibel med avseende på en fördelning $\boldsymbol{\pi}$ har också $\boldsymbol{\pi}$ till stationär fördelning, vilket alltså innebär att $\boldsymbol{\pi} = \boldsymbol{\pi P}$. □

Bevis:
Vi koncentrerar oss på fallet att kedjan har diskret tillståndsrum, men samma typ av kalkyl (integraler i stället för summor) kan användas för att visa motsvarigheter då man har kontinuerligt tillståndsrum.

Satsen inses om man summerar villkoret för tidsreversibilitet över $\boldsymbol{x}$ eftersom man då erhåller för vänsterledet

$$\sum_{\boldsymbol{x}} \pi(\boldsymbol{x})P(\boldsymbol{x},\boldsymbol{y})$$

som är sannolikheten att ligga i $\boldsymbol{y}$ efter ett tidssteg i Markovkedjan då man startat den med fördelningen $\boldsymbol{\pi}$. Man erhåller helt enkelt $\boldsymbol{y}$-komponenten i $\boldsymbol{\pi P}$. (Jämför $\boldsymbol{p}^{(1)} = \boldsymbol{p}^{(0)}\boldsymbol{P}$).

Eftersom $\boldsymbol{P}$ är en övergångsmatris och alltså

$$\sum_{\boldsymbol{x}} P(\boldsymbol{y},\boldsymbol{x}) = 1$$

ger en summering över $\boldsymbol{x}$ av högerledet i villkoret för tidsreversibilitet

$$\sum_{\boldsymbol{x}} \pi(\boldsymbol{y})P(\boldsymbol{y},\boldsymbol{x}) = \pi(\boldsymbol{y}) \sum_{\boldsymbol{x}} P(\boldsymbol{y},\boldsymbol{x}) = \pi(\boldsymbol{y})$$

vilket alltså innebär att $\boldsymbol{\pi} = \boldsymbol{\pi P}$. Alltså är $\boldsymbol{\pi}$ fördelningen för kedjan efter ett tidssteg då den startats med fördelningen $\boldsymbol{\pi}$ och $\boldsymbol{\pi}$ är alltså en stationär fördelning till kedjan. □

Notera att existensen av den stationära fördelningen till kedjan är klar eftersom vi vet att $\boldsymbol{\pi}$ är en sannolikhetsfördelning!

Följande sats visar att acceptanssannolikheten α verkligen är avpassad så att den resulterande kedjan blir tidsreversibel med $\boldsymbol{\pi}$ som stationär fördelning.

Sats 17.2 *Med acceptanssannolikheten α enligt ovan är Markovkedjan tidsreversibel med $\boldsymbol{\pi}$ som stationär fördelning.*

Bevis:
Vi har för $\boldsymbol{y} \neq \boldsymbol{x}$

$$P(\boldsymbol{x},\boldsymbol{y}) = P(\text{att föreslå } \boldsymbol{y}|\text{start i } \boldsymbol{x})P(\text{acceptera förslaget}) = q(\boldsymbol{x},\boldsymbol{y})\alpha(\boldsymbol{x},\boldsymbol{y})$$

Vi får då att för $\boldsymbol{y} \neq \boldsymbol{x}$

$$\pi(\boldsymbol{x})P(\boldsymbol{x},\boldsymbol{y}) = \pi(\boldsymbol{x})q(\boldsymbol{x},\boldsymbol{y})\alpha(\boldsymbol{x},\boldsymbol{y}) = \pi(\boldsymbol{x})q(\boldsymbol{x},\boldsymbol{y})\min\left(\frac{\pi(\boldsymbol{y})q(\boldsymbol{y},\boldsymbol{x})}{\pi(\boldsymbol{x})q(\boldsymbol{x},\boldsymbol{y})},1\right) =$$

$$= \min\left(\pi(\boldsymbol{y})q(\boldsymbol{y},\boldsymbol{x}),\pi(\boldsymbol{x})q(\boldsymbol{x},\boldsymbol{y})\right)$$

och på samma sätt (byt $\boldsymbol{x}$ och $\boldsymbol{y}$)

$$\pi(\boldsymbol{y})P(\boldsymbol{y},\boldsymbol{x}) = \min\left(\pi(\boldsymbol{x})q(\boldsymbol{x},\boldsymbol{y}),\pi(\boldsymbol{y})q(\boldsymbol{y},\boldsymbol{x})\right)$$

och dessa är lika! Om $\boldsymbol{x} = \boldsymbol{y}$ är å andra sidan villkoret för tidsreversibilitet trivialt uppfyllt! □

Man kan tolka

$$\pi(\boldsymbol{x})P(\boldsymbol{x},\boldsymbol{y}) = P(\boldsymbol{X}_n = \boldsymbol{x})P(\boldsymbol{X}_{n+1} = \boldsymbol{y}|\boldsymbol{X}_n = \boldsymbol{x}) = P(\boldsymbol{X}_n = \boldsymbol{x}\,\&\,\boldsymbol{X}_{n+1} = \boldsymbol{y})$$

som ett "flöde" från $\boldsymbol{x}$ till $\boldsymbol{y}$ och på samma sätt är högerledet

$$\pi(\boldsymbol{y})P(\boldsymbol{y},\boldsymbol{x}) = P(\boldsymbol{X}_n = \boldsymbol{y})P(\boldsymbol{X}_{n+1} = \boldsymbol{x}|\boldsymbol{X}_n = \boldsymbol{y}) = P(\boldsymbol{X}_n = \boldsymbol{y}\,\&\,\boldsymbol{X}_{n+1} = \boldsymbol{x})$$

"flödet" från $\boldsymbol{y}$ till $\boldsymbol{x}$. Tidsreversibiliteten innebär alltså att dessa flöden balanserar varandra och det är därför naturligt att kedjan har $\boldsymbol{\pi}$ som stationär fördelning.

Man kan säga att acceptans-sannolikheten $\alpha(\boldsymbol{x},\boldsymbol{y})$ avpassats så att dessa flöden precis balanserar varandra.

17.3.1 Metropolis algoritm

Metropolis-Hastings algoritm har ett specialfall, den s k Metropolis-algoritmen, där $q(\boldsymbol{x},\boldsymbol{y}) = q(\boldsymbol{y},\boldsymbol{x})$ dvs att sannolikheten att föreslå $\boldsymbol{y}$ från $\boldsymbol{x}$ är samma som sannolikheten att föreslå $\boldsymbol{x}$ från $\boldsymbol{y}$. I så fall blir acceptanssannolikheten

$$\alpha(\boldsymbol{x},\boldsymbol{y}) = \min\left(1, \frac{\pi(\boldsymbol{y})}{\pi(\boldsymbol{x})}\right)$$

Om förslagsfördelningen $q(\boldsymbol{x},\boldsymbol{y})$ ej beror på aktuellt tillstånd $\boldsymbol{x}$ får man oberoende likafördelade förslag $\boldsymbol{y}$.

Ofta väljer man förslagsfördelningen $q(\boldsymbol{x},\boldsymbol{y}) = f(\boldsymbol{y}-\boldsymbol{x})$ för någon täthet (sannolikhetsfunktion) f. Man har då en rumshomogen slumpvandrings-fördelning som förslagsfördelning. Ofta kan man t ex låta f vara täthet för en d-dimensionell normalfördelning med $\mathbf{0}$ som väntevärdesvektor och $\sigma^2\boldsymbol{I}_d$ som kovariansmatris. Parametern σ mäter då hur "vilda" förslagen är, dvs hur långt från aktuellt tillstånd de ligger.

17.3.2 En-dimensionella fallet $d = 1$

Detta är egentligen inte ett speciellt intressant fall utan styrkan i tekniken är om $d > 1$, men för enkelhets skull börjar vi med detta fall.

Exempel 17.1 *Diskret ändlig Markovkedja*
Antag att vi vill simulera en fördelning $\boldsymbol{f} = (f_i; i = 1, 2, \cdots, m)$ på talen $1, 2, \cdots, m$ med hjälp av en förslagsfördelning given av en matris Q dvs där

$$q_{ij} = P(\text{föreslå } j|\text{aktuellt tillstånd } i).$$

Följande Matlab-funktion levererar en matris alpha med acceptanssannolikheterna och den resulterande Markovkedjans övergångsmatris P.

```
function [ P,alpha ]=pmatris(f,Q);
s=size(f);
n=s(2);
P=zeros(n,n);
for i=1:n,
 for j=1:n,
  alpha(i,j)=min(f(j)*Q(j,i)/max(f(i)*Q(i,j),eps),1);
   if j ~=i,
    P(i,j)=alpha(i,j)*Q(i,j);
   end,
 end,
end,
su=sum(P');
for i=1:n,
 P(i,i)=1-su(i);
end ;
```

Att det står max(f(i)*Q(i,j),eps) beror på att 0 i nämnaren i princip skall ge $+\infty$ vid divisionen.

Om t ex målfördelningen är $f = (1/6, 2/6, 3/6)$ och

$$Q = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}$$

blir

$$\text{alpha} = \begin{pmatrix} 1 & 1 & 1 \\ 3/4 & 1 & 1 \\ 1/2 & 2/3 & 1 \end{pmatrix}$$

och den resulterande Markovkedjan får övergångsmatrisen

$$P = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 5/12 & 1/3 \\ 1/6 & 2/9 & 11/18 \end{pmatrix}$$

och man ser att $f = fP$ dvs att f är en stationär fördelning till P och man inser också att denna kedja är ergodisk. □

Samma metodik fungerar även för kontinuerliga fördelningar men då är förslagsfördelningen en kontinuerlig fördelning och Markovkedjan har kontinuerligt tillståndsrum. Övergångsmekanismen kan ofta beskrivas av en s k övergångstäthet $p(x, y)$ som uppfyller

$$P(X_{n+1} \in A | X_n = x) = \int_A p(x, y) dy$$

dvs att $p(x, y)$ är den betingade tätheten för X_{n+1} givet att $X_n = x$.

Vi tolkar förslagsfördelningen $q(\boldsymbol{x}, \boldsymbol{y})$ som tätheten att föreslå $\boldsymbol{y}$ då aktuellt tillstånd är $\boldsymbol{x}$. Acceptanssannolikheten beräknas som tidigare genom

$$\alpha(\boldsymbol{x}, \boldsymbol{y}) = \min\left(1, \frac{\pi(\boldsymbol{y}) q(\boldsymbol{y}, \boldsymbol{x})}{\pi(\boldsymbol{x}) q(\boldsymbol{x}, \boldsymbol{y})}\right).$$

där π är proportionell mot måltätheten.

Exempel 17.2 *Simulering av N(0,1)-fördelningen*
Vi vill alltså simulera tätheten

$$\pi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

och tar som förslagsfördelning $R(x - a, x + a)$ för olika val av a, dvs vi har

$$q(x, y) = \begin{cases} \dfrac{1}{2a} & \text{om } |y - x| \leq a \\ 0 & \text{annars.} \end{cases}$$

Vi använder oss alltså av Metropolis-algoritmen eftersom $q(x, y) = q(y, x)$.

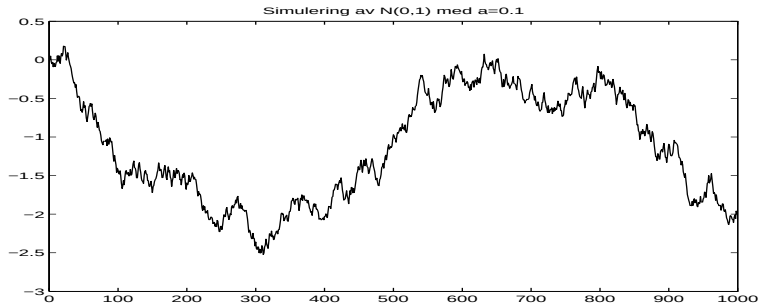
Vi får acceptanssannolikheten

$$\alpha(x, y) = \min\left(1, e^{(x^2-y^2)/2}\right) \text{ om } |y - x| \leq a$$

I Matlab blir detta

```
function x=n_a(antal,a,start);
x=start;
last=start;
for i=1:antal,
 y=last-a+rand(1)*2*a ;
 alfa=min([1,exp(0.5*(last^2-y^2))]);
 if rand(1)<alfa,
  x=[x y];
  last=y ;
 else,
  x=[x last];
 end,
end,
```

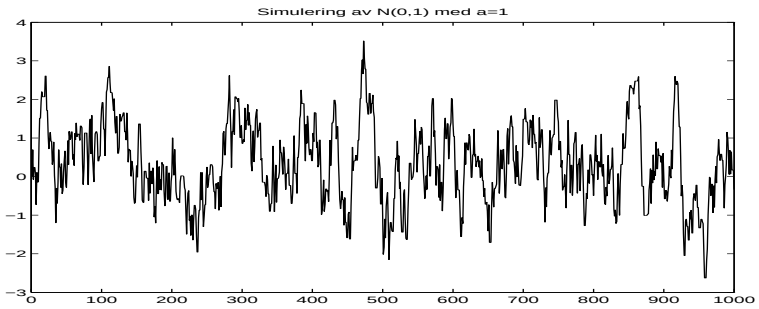
Om vi använder $a = 0.1$, 1 respektive 10 erhålls simuleringar enligt figur 17.1, 17.2 och 17.3.



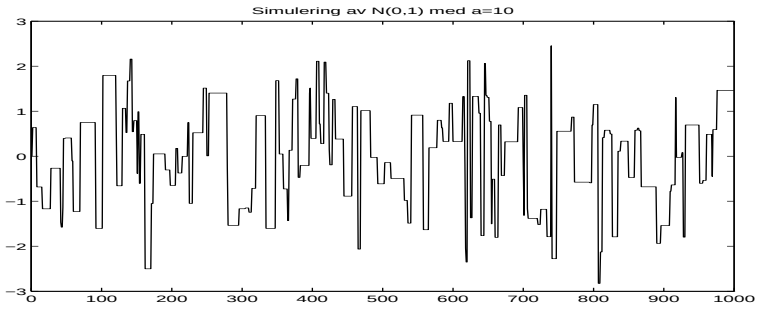
Figur 17.1: Simulering av $N(0, 1)$ med $a = 0.1$

Man kan notera att $a = 0.1$ innebär att de flesta förslagen ligger alltför nära det aktuella tillståndet och att man därför förflyttar sig alltför långsamt runt i tillståndsrummet. Nästan alla förslag accepteras. Problemet är att stora delar av tillståndsrummet inte besökts med den längd på simuleringen vi gjort. Man ser i figur 17.1 att kedjan inte på 1000 tidssteg gett något värde över 0.5. De tillstånd kedjan besökt avspeglar alltså inte målfördelningens utseende med den simuleringslängd vi valt.

Om å andra sidan $a = 10$ som i figur 17.3 är förslagen ofta rätt "vilda" men kommer å andra sidan då att i allmänhet inte accepteras och kedjan ligger



Figur 17.2: Simulering av $N(0,1)$ med $a = 1$



Figur 17.3: Simulering av $N(0,1)$ med $a = 10$

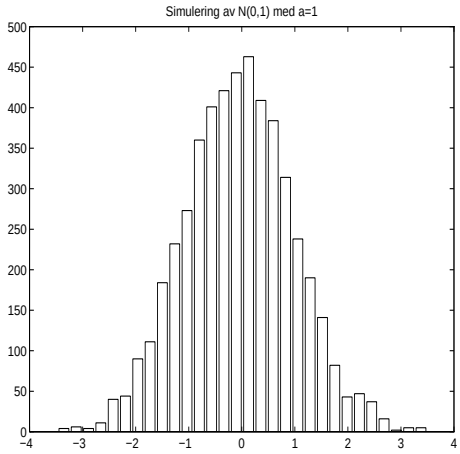
sålunda ofta kvar ett antal tidpunkter i samma tillstånd. Detta bidrar till att den resulterande Markovkedjan "blandar" sig rätt långsamt.

$a = 1$ är ett mellanläge som också innebär att man får i figur 17.2 en förhållandevis bra genomlöpning av tillståndsrummet och den resulterande fördelningen i figur 17.4) blir rätt nära den avsedda $N(0,1)$-fördelningen.
Man kan dock notera att med en tillräckligt lång simulering kommer alla ovanstående val av a att ge upphov till $N(0,1)$-fördelningen. Problemet är dock att vi ju bara gör en ändlig simulering och i en mer komplicerad situation där vi inte har "facit" kan det vara svårt att avgöra om alla delar av tillståndsrummet besökts i rätt proportion. □

Ovanstående exempel saknar praktiskt intresse, men avsikten var att illustrera grundläggande principer. I båda ovanstående exempel är de resulterande Markovkedjorna ergodiska och alltså får man oavsett starttillstånd rätt fördelning. Av $N(0,1)$-exemplet inser man också att det kan spela stor roll för konvergenshastigheten hur förslagsfördelningen ser ut.

17.3.3 Konvergens mot målfördelning

Det finns metoder att empiriskt undersöka om kedjan verkar ha svängt in i ett stationärt skede, dvs om simuleringen är så lång att alla delar av till-



Figur 17.4: Fördelning av simulerings-resultat av $N(0,1)$ med $a = 1$

ståndsrummet besöks i rätt proportion i enlighet med målfördelningen $\pi(\boldsymbol{x})$. Ett enkelt sätt är att starta flera simuleringar med olika startvärde och se om de ger upphov till skenbart olika fördelningar vilket är en varningssignal.

Detta förfarande kan formaliseras genom att vi genererar K simuleringar vardera av längd n som vi kan kalla $\boldsymbol{x}_i^k$, $i = 1, 2, \cdots, n$ där $k = 1, 2, \cdots, K$.

Vi skattar $E(g(\boldsymbol{X}))$ med aritmetiska medelvärdet av $g(\boldsymbol{x}_i^k)$:na i respektive simulering. Vi erhåller då K skattningar $\widehat{m}_k$, $k = 1, 2, \cdots, K$ genom

$$\widehat{m}_k = \frac{1}{n} \sum_{i=1}^{n} g(\boldsymbol{x}_i^k), \ k = 1, 2, \cdots, K$$

och bildar även

$$\widehat{m} = \frac{1}{K} \sum_{1}^{K} \widehat{m}_k.$$

Man bildar teststorheterna

$$B = \frac{n}{K-1} \sum_{k=1}^{K} (\widehat{m}_k - \widehat{m})^2$$

("mellan-kedje-variation") och

$$W = \frac{1}{K} \sum_{k=1}^{K} s_k^2 = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n-1} \sum_{i=1}^{n} (g(\boldsymbol{x}_i^k) - \widehat{m}_k)^2.$$

där s_k^2 mäter "inom-kedje-variationen" i kedja k.

I en analogi till varianskomponentmodeller (ensidig variansanalys med slumpmässig faktor) kombineras dessa sedan till Gelman-Rubin-teststorheten

$$\widehat{R} = \frac{\frac{n-1}{n}W + \frac{1}{n}B}{W}$$

som borde ligga nära 1 då kedjorna svängt in mot ett stationärt skede.

Vid ensidig variansanalys antar vi att $Y_{ki} = m_k + \epsilon_{ki}$ där m_k är $N(m, \sigma_m^2)$ och ϵ_{ij} är $N(0, \sigma^2)$. Teststorheten ovan svarar ungefär mot ett test av $H_0 : \sigma_m = 0$.

Den intresserade hänvisas till med djuplodande litteratur.

17.4 Koordinatvis uppdatering

Vår målfördelning

$$\pi_{X_1,X_2,\cdots,X_d}(x_1, x_2, \cdots, x_d) = \pi_{\boldsymbol{X}}(\boldsymbol{x})$$

är d-dimensionell och en intressant variant av Metropolis-Hastings algoritm är att uppdatera en koordinat i taget. Vid uppdatering av koordinat i är alltså alla de övriga konstanta och den enda förändringen som sker är ett byte av värde i koordinat i. Man har alltså vid uppdateringen av koordinat i en förslagsfördelning $q_i(\boldsymbol{x}, \boldsymbol{y})$ som är 0 om inte $\boldsymbol{y}$ överensstämmer med $\boldsymbol{x}$ för alla koordinater utom nr i. Detta betyder att förslagen från q_i lämnar alla koordinater utom nr i oförändrade. Till denna förslagsfördelning q_i hör en acceptanssannolikhet $\alpha_i(\boldsymbol{x}, \boldsymbol{y})$ definierad som ovan fast med utgångspunkt i q_i. Detta ger upphov till en övergångsmekanism beskriven av en matris $\boldsymbol{P}_i$. I allmänhet löper man systematiskt igenom alla koordinaterna innan man anser att man fått en ny observation i Markovkedjan. Ett steg i Markovkedjan svarar alltså mot en uppdatering av samtliga koordinater. Om vi låter $\boldsymbol{P}_i$ vara övergångsmatrisen vid uppdatering av koordinat i består ett steg i Markov-kedjan av att vi först applicerar $\boldsymbol{P}_1$ och sedan $\boldsymbol{P}_2$ osv t o m $\boldsymbol{P}_d$. Markov-kedjans övergångsmatris är alltså

$$\boldsymbol{P} = \boldsymbol{P}_1\boldsymbol{P}_2 \cdots \boldsymbol{P}_d.$$

Orsaken till att vi vill uppdatera alla koordinater innan vi anser oss ha fått ett nytt tillstånd är att kedjan annars inte skulle vara tidshomogen. En alternativ metod som ger en tidshomogen Markovkedja skulle vara att lotta fram en koordinat I (lika sannolikheter för alla d koordinater) och sen uppdatera denna med hjälp av motsvarande matris $\boldsymbol{P}_i$.

17.5 Gibbs-sampling

Ett mycket viktigt, intressant och praktiskt användbart specialfall av koordinatvis uppdatering enligt Metropolis-Hastings algoritm är Gibbs-sampling

som innebär ett mycket speciellt val av förslagsfördelning q_i vid uppdateringen av koordinat i. Man låter helt enkelt förslagsfördelningen q_i vara den betingade fördelningen för X_i givet alla de övriga koordinaterna.

Vi försöker alltså konstruera en d-dimensionell Markovkedja med målfördelningen

$$\pi_{X_1,X_2,\cdots,X_d}(x_1, x_2, \cdots, x_d) = \pi_{\boldsymbol{X}}(\boldsymbol{x})$$

som täthet (sannolikhetsfunktion). Vid Gibbs-sampling gör man alltså detta genom att uppdatera en koordinat i taget med den betingade fördelningen för denna komponent givet alla de övriga.

17.5.1 Två-dimensionella fallet

Vi tittar först på det två-dimensionella fallet:

Om vi vill simulera den två-dimensionella tätheten $f_{X,Y}$ startar vi med en godtycklig punkt (x_0, y_0). Vi uppdaterar sen x-koordinaten genom att ge förslaget (x_1, y_0) där X_1 har tätheten

$$f_{X|Y=y_0}(x) = \frac{f_{X,Y}(x, y_0)}{f_Y(y_0)}$$

Vi kommer strax att visa att acceptanssannolikheten är 1 för detta förslag. Kedjan flyttar sig alltså till (x_1, y_0) och vi övergår sedan till att uppdatera y-komponenten med betingade fördelningen för Y givet $X = x_1$ dvs med förslaget (x_1, y_1) där Y_1 har tätheten

$$f_{Y|X=x_1}(y) = \frac{f_{X,Y}(x_1, y)}{f_X(x_1)}.$$

Man har då uppdaterat båda komponenterna i vektorn och betraktar den erhållna punkten (x_1, y_1) som nästa observation av Markovkedjan.
Från (x_0, y_0) har vi erhållit (x_1, y_1) och vi genererar sedan en ny observation (x_2, y_2) genom att på samma sätt först uppdatera x-komponenten och sedan y-komponenten enligt figur 17.5.

Exempel 17.3 *Två-dimensionell normalfördelning*
Om vi har en två-dimensionell normalfördelning för X, Y

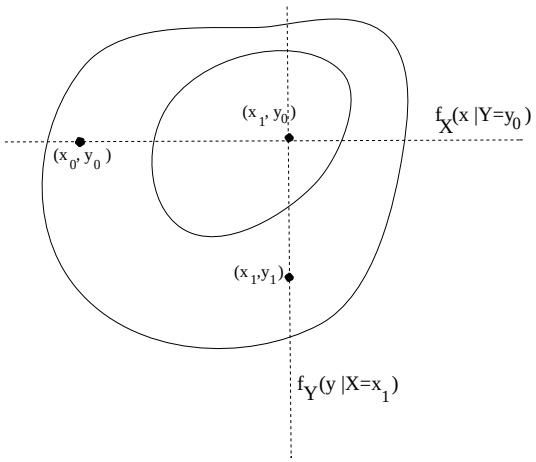
$$N_2\left(\begin{pmatrix} m_1 \\ m_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}\right)$$

vet man att den betingade fördelningen för $Y|X = x$ är

$$N\left(m_2 + \rho\frac{\sigma_2}{\sigma_1}(x - m_1), \sigma_2^2 - \rho^2\sigma_2^2\right)$$

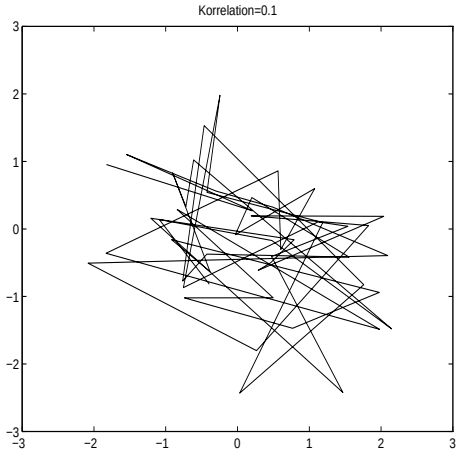
och att fördelningen för $X|Y = y$ på motsvarande sätt är

$$N\left(m_1 + \rho\frac{\sigma_1}{\sigma_2}(y - m_2), \sigma_1^2 - \rho^2\sigma_1^2\right)$$



Figur 17.5: Simulering av två-dimensionell fördelning en koordinat i taget.

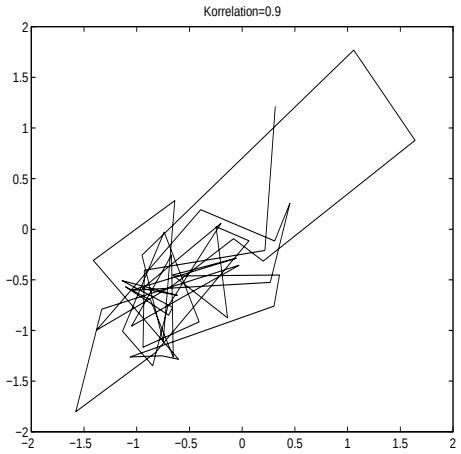
Om vi tar $m_1 = m_2 = 0$, $\sigma_1 = \sigma_2 = 1$ får vi de betingade fördelningarna $N(\rho x, 1-\rho^2)$ respektive $N(\rho y, 1-\rho^2)$ och dessa utgör alltså de fördelningar som vi använder oss av vid den successiva uppdateringen av respektive komponenter.



Figur 17.6: Simulering av 2-dimensionell normalfördelning med ρ=0.1.

Med $\rho = 0.1$ (dvs nästan oberoende) i figur 17.6 och $\rho = 0.9$ (dvs mycket kraftigt beroende) i figur 17.7 visas 50 uppdateringar.

Man noterar att genomlöpningen av tillståndsrummet ser bättre ut för $\rho = 0.1$ och detta innebär att man skall försöka göra de olika komponenterna så nära oberoende som möjligt. Detta kan ofta ske genom en listig omparametrisering. □



Figur 17.7: Simulering av 2-dimensionell normalfördelning med ρ=0.9.

17.5.2 d-dimensionella fallet

De förslag som lämnas av Gibbs-sampling accepteras alltid vilket följande sats visar.

Sats 17.3 *Betrakta Gibbs-sampling av en (för enkelhets skull) två-dimensionell variabel* (X_1, X_2) *där vi vill simulera fördelningen* $f_{X_1,X_2}(x_1, x_2),\ (x_1, x_2) \in R^2$. *Då vi uppdaterar koordinat nr 1 (dvs* X_1*) är alltså förslagsfördelningen*

$$q((x_1, x_2), (y_1, y_2)) = \begin{cases} f_{X_1|X_2=x_2}(y_1) & om\ y_2 = x_2 \\ 0 & om\ y_2 \neq x_2 \end{cases}$$

Acceptanssannolikheten i Metropolis-Hastings algoritm, dvs

$$\alpha((x_1, x_2), (y_1, y_2)) = \min\left(\frac{f_{X_1,X_2}(y_1, y_2)q((y_1, y_2), (x_1, x_2))}{f_{X_1,X_2}(x_1, x_2)q((x_1, x_2), (y_1, y_2))}, 1\right)$$

är 1, dvs att samtliga förslag accepteras.

Samma resultat gäller om vi uppdaterar komponent i i en d-dimensionell Markovkedja med den betingade fördelningen för i:te koordinaten givet de övriga.

Bevis: Vi har alltså

$$q((x_1, x_2), (y_1, y_2)) = \begin{cases} f_{X_1|X_2=x_2}(y_1) & \text{om } y_2 = x_2 \\ 0 & \text{om } y_2 \neq x_2 \end{cases}$$

Eftersom (x_1, x_2) ej kan föreslås från (y_1, y_2) om inte $x_2 = y_2$ så kan vi anta att $x_2 = y_2$ och erhåller för täljaren i $\alpha((x_1, x_2), (y_1, y_2))$

$$f_{X_1,X_2}(y_1, y_2)f_{X_1|X_2=y_2}(x_1) = f_{X_1,X_2}(y_1, y_2)\frac{f_{X_1,X_2}(x_1, y_2)}{f_{X_2}(y_2)} =$$

$$= \frac{f_{X_1,X_2}(y_1,y_2)}{f_{X_2}(y_2)} f_{X_1,X_2}(x_1,y_2) =$$

$$= f_{X_1|X_2=y_2}(y_1) f_{X_1,X_2}(x_1,y_2) = f_{X_1|X_2=x_2}(y_1) f_{X_1,X_2}(x_1,x_2)$$

där vi utnyttjade att $x_2 = y_2$ i sista ledet. Detta uttryck är precis det som förekommer i nämnaren i $\alpha((x_1,x_2),(y_1,y_2))$ så kvoten är 1 och alltså accepteras alla förslag.

Det generella resulatet följer om man överallt byter
X_1 mot X_i,
x_1 mot x_i,
y_1 mot y_i
X_2 mot $\boldsymbol{X_{-i}} = (X_1, X_2, \cdots, X_{i-1}, X_{i+1}, \cdots, X_n)$
x_2 mot $\boldsymbol{x_{-i}} = (x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$ och
y_2 mot $\boldsymbol{y_{-i}} = (y_1, y_2, \cdots, y_{i-1}, y_{i+1}, \cdots, y_n)$. □

Om man vill simulera en d-dimensionell fördelning $\pi_{\boldsymbol{X}}$ så krävs alltså att vi på ett någorlunda enkelt sätt kan beräkna fördelningen för (t ex)

$$X_1 | X_2 = x_2, X_3 = x_3, \cdots, X_d = x_d$$

eller mer allmänt för

$$X_i | X_1 = x_1, \cdots, X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \cdots, X_d = x_d$$

för olika i, $i = 1, 2, \cdots, d$.

Dessutom måste vi kunna någorlunda lätt kunna simulera dessa betingade fördelningar.

Med $\boldsymbol{X}_{-i} = (X_1, \cdots, X_{i-1}, X_{i+1}, \cdots, X_d)$ och $\boldsymbol{x}_{-i} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_d)$ kan ovanstående skrivas som

$$X_i | \boldsymbol{X}_{-i} = \boldsymbol{x}_{-i}.$$

och denna betingade fördelning måste vi alltså kunna uttrycka enkelt för att kunna uppdatera komponent i i vektorn med den korrekta betingade fördelningen. Vi kommer (lite slarvigt) att låta $\boldsymbol{x}$ och $(x_i, \boldsymbol{x}_{-i})$ stå för samma sak.

Förvånansvärt ofta kan man enkelt uttrycka dessa betingade fördelningar trots att det är i stort sett omöjligt att ange den d-dimensionella fördelningen.

Man uppdaterar successivt en koordinat i taget och efter att ha gått igenom alla koordinaterna har man en ny observation av Markovkedjan. Som nämndes tidigare är orsaken till att vi vill uppdatera alla koordinater innan vi anser oss ha fått en ny observation är att vi gärna vill konstruera en tidshomogen Markovkedja.

När vi uppdaterar koordinat i kommer resultatet att bibehålla värdet på de övriga, dvs på $\boldsymbol{x}_{-i}$. Detta betyder att denna uppdatering (beskriven av övergångsmatrisen $\boldsymbol{P}_i$) innebär att vi erhåller

$$P_i((x_i, \boldsymbol{x}_{-i}), (y_i, \boldsymbol{y}_{-i})) = 0$$

om $\boldsymbol{x}_{-i} \neq \boldsymbol{y}_{-i}$ medan för $\boldsymbol{x}_{-i} = \boldsymbol{y}_{-i}$ har vi matriselementet

$$P_i((x_i, \boldsymbol{x}_{-i}), (y_i, \boldsymbol{x}_{-i})) = \pi_{\boldsymbol{X}|\boldsymbol{X}_{-i}=\boldsymbol{x}_{-i}}((y_i, \boldsymbol{x}_{-i})) = \frac{\pi((y_i, \boldsymbol{x}_{-i}))}{\pi_{\boldsymbol{X}_{-i}}(\boldsymbol{x}_{-i})} =$$

$$= \frac{\pi((y_i, \boldsymbol{x}_{-i}))}{\sum_u \pi((u, \boldsymbol{x}_{-i}))}.$$

Ovanstående ger utseendet av övergångsmatrisen $\boldsymbol{P}_i$ som beskriver övergångarna då koordinat i uppdateras. Ett steg i Markovkedjan beskrivs alltså av $\boldsymbol{P} = \boldsymbol{P}_1 \cdot \boldsymbol{P}_2 \cdots \boldsymbol{P}_d$.

Sats 17.4 *Om man startar kedjan med fördelningen $\boldsymbol{\pi}$ och uppdaterar koordinat i med $\boldsymbol{P}_i$ enligt ovan så har resultatet också fördelningen $\boldsymbol{\pi}$, dvs denna utgör en stationär fördelning.*

Bevis:
När man uppdaterar Markovkedjan med $\boldsymbol{P}_i$ blir

$$P(\boldsymbol{X} = \boldsymbol{y}) = \sum_{\boldsymbol{x} \in E} \pi(\boldsymbol{x}) P_i(\boldsymbol{x}, \boldsymbol{y})$$

då vi startar kedjan med fördelningen $\boldsymbol{\pi} = (\pi(\boldsymbol{x}), \boldsymbol{x} \in E)$. Vi vill alltså visa att detta är just vad $\boldsymbol{\pi}$-fördelningens massa i $\boldsymbol{y}$, dvs är $\pi(\boldsymbol{y})$. Vi delar upp summeringen över $\boldsymbol{x} \in E$ i en summation över x_i och en över $\boldsymbol{x}_{-i}$. Vi erhåller (eftersom $P_i(\boldsymbol{x}, \boldsymbol{y}) = 0$ om inte $\boldsymbol{x}_{-i} = \boldsymbol{y}_{-i}$)

$$\sum_{x_i} \sum_{\boldsymbol{x}_{-i}} \pi(x_i, \boldsymbol{x}_{-i}) P_i((x_i, \boldsymbol{x}_{-i}), (y_i, \boldsymbol{y}_{-i})) =$$

$$= \sum_{x_i} \pi(x_i, \boldsymbol{x}_{-i}) P_i((x_i, \boldsymbol{y}_{-i}), (y_i, \boldsymbol{y}_{-i})) =$$

$$= \sum_{x_i} \pi(x_i, \boldsymbol{y}_{-i}) \frac{\pi((y_i, \boldsymbol{y}_{-i}))}{\sum_u \pi((u, \boldsymbol{y}_{-i}))} =$$

$$= \pi((y_i, \boldsymbol{y}_{-i})) \frac{\sum_{x_i} \pi(x_i, \boldsymbol{y}_{-i})}{\sum_u \pi(u, \boldsymbol{y}_{-i})} = \pi((y_i, \boldsymbol{y}_{-i})) = \pi(\boldsymbol{y})$$

vilket var vad vi ville visa. □

Man inser att eftersom en uppdatering av en viss koordinat lämnar fördelningen oförändrad så innebär en successiv uppdatering av samtliga koordinater (vilket utgör ett steg i Markovkedjan) att fördelningen $\boldsymbol{\pi}$ bevaras. Startar man alltså kedjan med "rätt" fördelning (dvs $\boldsymbol{\pi}$) bevaras denna, dvs den utgör en stationär fördelning för Markovkedjan. Kan vi bara dessutom visa att denna Markovkedja är irreducibel och aperiodisk har vi alltså en ergodisk kedja som har den stationära fördelningen som asymptotisk fördelning oavsett starttillstånd. Efter "tillräckligt lång" tid kommer kedjan alltså att "nästan" ha den stationära fördelningen och vi kan alltså generera slumptal med den avsedda fördelningen $\boldsymbol{\pi}$.

17.6 Gibbs-sampling: Ising-modell för spinn

Som ett icke-trivialt exempel på att det kan vara lätt att simulera en mycket komplicerad fördelning tar vi följande exempel med anknytning till Statistisk mekanik.

En enkel två-dimensionell modell för magnetiskt spinn är att vi har ett $n \times m$ gitter

$$S = \{(i,j) : i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, m\}.$$

I dessa nm punkter finns slumpmässiga spinn $X_{i,j}$ som kan anta värdena $+1$ och -1 som svarar mot spinn upp respektive spinn ner. Vi låter $\boldsymbol{X} = (X_{i,j},\ (i,j) \in S)$. Vi har då ett utfallsrum Ω för $\boldsymbol{X}$ med 2^{nm} tänkbara utfall.

Notera att även om gittret är så litet som 100×100 finns 2^{10000} tänkbara konfigurationer. Vi vill nu simulera ett sådant utfall. Problemet är att vi antar att det finns en (temperaturberoende) växelverkan mellan de olika gitterpunkterna och då framför allt mellan gitterpunkter som ligger nära varandra. Vi definierar en "omgivning" $N(i,j)$ till gitterpunkten (i,j) t ex

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

där (i,j) ligger i mitten och 1:or står för "omgivningspunkter". Ett annat tänkbart utseende på "omgivningen" skulle vara

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Växelverkan sker primärt endast mellan mittpunkten och dess grannar, men detta beroende "fortplantas" genom gittret till att bli ett beroende som kanske sträcker sig över stora områden. Ofta knyter man ihop vänster- och högerkanten, respektive över- och underkanten av S för att slippa randeffekter.

Vi definierar en energi

$$E(\boldsymbol{x}) = -J \sum_{(i,j)\in S} \sum_{(i',j')\in N(i,j)} x_{i,j} x_{i',j'}$$

där $J > 0$ kvantifierar växelverkans storlek. Detta innebär att energin är låg (stabilt tillstånd) om näraliggande gitterpunkter har samma spinn-riktning.

Vi antar nu för ett fixt värde på parametern $\beta > 0$ att

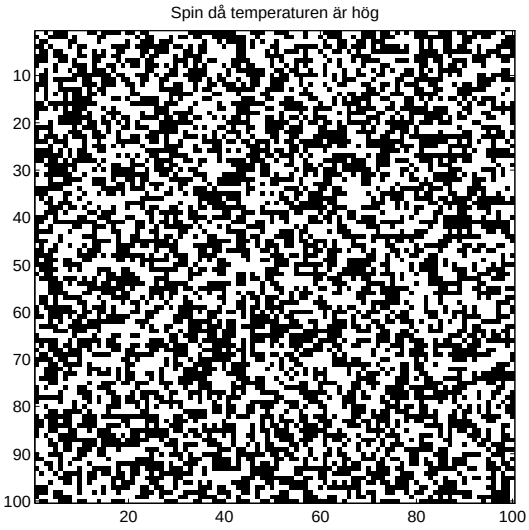
$$p_\beta(\boldsymbol{x}) = P(\boldsymbol{X} = \boldsymbol{x}) = \frac{e^{-\beta E(\boldsymbol{x})}}{Z}$$

där Z är en normeringskonstant dvs

$$Z = \sum_{\boldsymbol{y} \in \Omega} e^{-\beta E(\boldsymbol{y})}$$

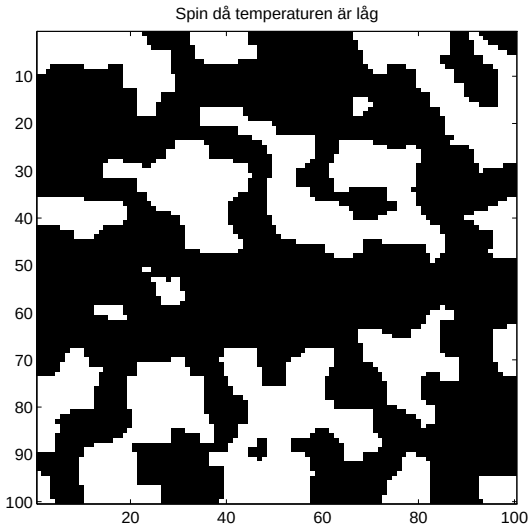
Den som kan lite statistisk mekanik ser att Z är "tillståndssumman" och att $p_\beta(\boldsymbol{x})$ utgör den s k kanoniska fördelningen eller Gibbsfördelningen. Parametern β svarar fysikaliskt mot 1/temperaturen. Systemet antas alltså vara placerat i ett stort omgivande system ("värmebad") med temperatur $1/\beta$ som det utbyter energi med.

Det är uppenbart att tillståndssumman Z är extremt besvärlig att beräkna, medan $E(\boldsymbol{x})$ är förhållandevis enkel. Z kan ju som i exemplet ovan kunna bestå av 2^{10000} termer!



Figur 17.8: Spin i Ising-modell då temperaturen är hög.

Man vill gärna simulera p_β-fördelningen och helst för ett tätt gitter av β-värden eftersom en hel del termodynamiskt intressanta storheter som t ex specifikt värme kan beräknas ur p_β-fördelningen (se avsnitt 17.10.2). Ett typiskt utseende av spinnen på gittret är att för låga β-värden (svarande mot höga temperaturer) är de olika gitterpunkternas spinn rätt oberoende av varandra (se figur 17.8), medan för höga β-värden (dvs låga temperaturer) har stora regioner av S samma spinn-riktning (se figur 17.9). Vid en viss kritisk temperatur sker ett dramatiskt omslag av uppträdandet – över den temperaturen är



Figur 17.9: Spin i Ising-modell då temperaturen är låg.

spinnen kaotiska, men under den temperaturen får stora regioner av S samma magnetisering, s k fasövergångar. Det finns alltså stort intresse av att kunna simulera uppträdandet av dessa spinn-system.
Hur kan man nu simulera sådana spinn, dvs hur får man ett utfall från p_β-fördelningen? Idén är att för fixt β konstruera en Markovkedja

$$\boldsymbol{X} = (X_{i,j};\, (i,j) \in S)$$

som har p_β till asymptotisk fördelning.

Vad man gör är att utifrån en godtycklig spinn-konfiguration successivt uppdatera de nm gitterpunkternas spinn med fördelningen för just det spinnet betingat av alla de övrigas aktuella värde!

Om vi låter $\boldsymbol{x}_{-(i,j)}$ beteckna spinnen i alla punkter i gittret S utom i punkten (i,j) så inser man att fördelningen för spinnet i (i,j) givet alla de övriga blir

$$P(X_{i,j} = 1 | \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)}) = (\text{Jfr } P(A|B) = P(A \cap B)/P(B)) =$$

$$= \frac{P(X_{i,j} = 1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)})}{P(\boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)})} = (\text{Jfr } P(A) = P(A \cap B) + P(A \cap B^*)) =$$

$$= \frac{P(X_{i,j} = 1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)})}{P(X_{i,j} = 1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)}) + P(X_{i,j} = -1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)})} =$$

$$= \frac{\exp\left(-\beta E(X_{i,j} = 1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)})\right)}{\exp\left(-\beta E(X_{i,j} = 1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)})\right) + \exp\left(-\beta E(X_{i,j} = -1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)})\right)} =$$

(förkorta med sista termen i nämnaren)

$$= \frac{\exp\left(-\beta\left(E(X_{i,j} = 1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)}) - E(X_{i,j} = -1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)})\right)\right)}{\exp\left(-\beta\left(E(X_{i,j} = 1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)}) - E(X_{i,j} = -1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)})\right)\right) + 1}.$$

Men uttrycket i exp-funktionerna blir

$$E(X_{i,j} = +1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)}) - E(X_{i,j} = -1; \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)}) =$$

$$= -J \sum_{(i',j') \in N(i,j)} (+1) x_{i',j'} + J \sum_{(i',j') \in N(i,j)} (-1) x_{i',j'} = -2J \sum_{(i',j') \in N(i,j)} x_{i',j'}$$

och detta sista blir $-2J\left(N_+(i,j) - N_-(i,j)\right)$ där $N_+(i,j)$=antalet spinn upp i $N(i,j)$ och $N_-(i,j)$ = antalet spinn ner i $N(i,j)$, dvs i omgivningen till (i,j).

Alltså får vi

$$P\left(X_{i,j} = +1 | \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)}\right) = \frac{\exp\left(2\beta J(N_+(i,j) - N_-(i,j))\right)}{\exp\left(2\beta J(N_+(i,j) - N_-(i,j))\right) + 1}$$

och

$$P\left(X_{i,j} = -1 | \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)}\right) = 1 - P\left(X_{i,j} = +1 | \boldsymbol{X}_{-(i,j)} = \boldsymbol{x}_{-(i,j)}\right) =$$

$$= \frac{1}{\exp\left(2\beta J(N_+(i,j) - N_-(i,j))\right) + 1}.$$

Betingade fördelningen för $X_{i,j}$ givet alla övriga spinn beror alltså enkelt av bara antalet spinn upp respektive ner bland närmaste grannarna, dvs de som de primärt växelverkar med.

Vi kan då "uppdatera" en spinn-punkt i taget där sannolikheten för spinn=+1 resp spinn=−1 bara beror på ovanstående sätt av spinnen bland de närmsta grannarna. Vi stegar alltså runt gittret en gitterpunkt i taget och uppdaterar den med alla andra fixa. När vi så vandrat runt gittret ett helt varv har vi fått en ny observation av $\boldsymbol{X}$. Detta utgör då en irreducibel aperiodisk Markovkedja som har p_β till stationär fördelning och vi kan således på detta sätt simulera p_β.

Detta att uppdatera en komponent i taget i $\boldsymbol{X}$ med den betingade fördelningen givet de övriga komponenterna är essensen i den s k Gibbs-algoritmen eller Gibbs-samplern.

17.7 Simulated annealing

Det exempel som ges nedan är ett exempel på användning av s k "simulated annealing" som teknik att minimera en komplicerad funktion. Namnet betyder ungefär "simulerad härdning (av metall)". Man vill med härdning få metallen i så lågt energitillstånd som möjligt genom successiv avsvalning från en hög temperatur. Det är ett exempel på användning av Metropolis-algoritmen.

Vi har en ändlig (men jättestor) mängd χ och en funktion $E : \chi \to R$ definierad på χ. Denna funktion är "någorlunda kontinuerlig" och vi vill hitta en minimipunkt $\widehat{x}$ till E, dvs vi vill lösa minimeringsproblemet $\min_{x \in \chi} E(x)$. Vi antar att detta minimum är unikt. Vidare antar vi att $E(x)$ är lätt att beräkna, men eftersom χ är så stort är det svårt att hitta en minimerande punkt $\widehat{x}$.

Vi inför nu en parameter $\beta > 0$ (som kommer att svara mot 1/temperatur) och definierar för varje β en sannolikhetsfördelning

$$p_\beta(x) = \frac{e^{-\beta E(x)}}{\sum_{y\in\chi} e^{-\beta E(y)}} \text{ för } x \in \chi$$

Återigen igenkänner vi från statistisk mekanik den kanoniska fördelningen, där $E(x)$ svarar mot energin hos systemet och p_β beskriver sannolikhetsfördelningen för tillståndet då systemet är placerat i ett stort omgivande system med temperaturen $1/\beta$. Nämnaren (som är enormt besvärlig att räkna ut) svarar då mot den s k "tillståndssumman". Vad man nu möjligen inser är att då $\beta \to \infty$ (dvs temperaturen$\to 0$) så kommer p_β-fördelningen att konvergera mot en punktmassa i $\widehat{x}$, dvs den punkt som minimerar "energin" $E(x)$. Vi har ju

$$p_\beta(x) = \frac{e^{-\beta E(x)}}{\sum_{y\in\chi} e^{-\beta E(y)}} = \frac{e^{-\beta(E(x)-E(\widehat{x}))}}{1+\sum_{y\neq\widehat{x}} e^{-\beta(E(y)-E(\widehat{x}))}}.$$

Om nu $\beta \to \infty$ så kommer alltså $p_\beta(\widehat{x}) \to 1$ medan $p_\beta(x) \to 0$ om $x \neq \widehat{x}$. Alltså har p_β nästan all massa koncentrerad i $\widehat{x}$ för stora β.

Vad vi nu gör är att (för ett visst β) konstruera en Markovkedja $X_\beta(n)$, $n = 0, 1, 2, \cdots$ som har p_β till stationär fördelning.

Vi kommer i princip att från en viss punkt välja att föreslå en "närliggande" punkt (detta kommer att vara "förslagsfördelningen") och detta förslag väljer vi att acceptera eller förkasta med en korrekt avpassad sannolikhet.

Vi inför därför ett begrepp "närmaste grannar" till punkten x i χ. Vi antar oftast att detta antal grannar $N(x)$ är detsamma för alla punkter x.

Detta att skapa "närmsta grannar" kan naturligtvis göras på en massa sätt och här ges två enkla exempel.

Exempel 17.4 *Binära vektorer*
Ett exempel på ovanstående situation är om $\chi = \{0,1\}^n$ dvs punkterna x i χ består av binära sekvenser (dvs (0-1)-sekvenser) av längd n, där t ex n=20 eller 30. χ består då av t ex $2^n = 2^{30} \approx 1000000000$ punkter. Som närmsta grannar tar vi de 30 punkter x' som skiljer sig från x i en enda position (en 0:a i stället för en 1:a eller tvärtom). □

Exempel 17.5 *Handelsresandeproblemet*
Ett klassiskt (och notoriskt svårt problem) är att givet ett antal punkter i planet konstruera ett polygontåg som passerar alla dessa punkter och återvänder till utgångspunkten och som är av minimal längd. Detta kan tolkas som att en handelsresande vill från sin hemstad besöka ett antal givna städer och välja en resrutt som leder tillbaka hem och som har minimal längd. χ består då av alla tänkbara rutter ($(n-1)!$ stycken rutter omfattar n städer inklusive hemstaden) och $E(x)$ är sammanlagda längden av rutten x och vi söker den rutt $\widehat{x}$ som minimerar sammanlagda reslängden $E(x)$. Detta minimeringsproblem kan alltså ges en approximativ lösning med hjälp av simulated annealing.

En viss resrutt genom de n städerna (inklusive hemstaden) kan representeras av en permutation av talen $1, 2, \cdots, n$ och "närmsta grannar" till denna rutt skulle då t ex vara vad man kan få genom att välja två av "länkarna" (i_1, j_1) och (i_2, j_2) i rutten och ändra rutten så att från den första länkens start i_1 resa direkt till den andra länkens start i_2, sen resa baklänges till den till den första länkens slut j_1 i enlighet med den ursprungliga resrutten, och sen resa från j_1 direkt till den andra länkens slut j_2 och sen vidare enligt den ursprungliga resrutten. □

Övergångsmekanismen vi väljer för Markovkedjan är nu följande: Om aktuellt tillstånd är x tar vi en av de $N(x)$ grannarna x' till x på måfå (lika sannolikhet). Detta utgör ett "förslag" på nästa tillstånd för Markovkedjan. Vi beräknar $\Delta E(x, x') = E(x') - E(x)$, dvs jämför x med x'. Om $\Delta E(x, x') \leq 0$ dvs x' är en minst lika bra punkt som x så går vi dit. Om däremot $\Delta E(x, x') > 0$ (dvs x' är en "sämre" punkt än x) så går vi dit med sannolikhet

$$\alpha(x, x') = e^{-\beta \Delta E(x,x')} = \frac{e^{-\beta E(x')}}{e^{-\beta E(x)}} = \frac{p_\beta(x')}{p_\beta(x)}$$

och stannar kvar med sannolikhet $1 - \alpha(x, x')$. Detta utgör ett enkelt exempel på Metropolis-algoritmen där vi som "förslagsfördelning" $q(x, y)$ har likformig fördelning på grannarna till x. Eftersom alla punkter har samma antal grannar gäller att $q(x, y) = q(y, x)$.

I handelsresandeproblemet väljer vi alltså att byta till den föreslagna rutten om den är bättre än den nuvarande, men om den föreslagna är sämre än den nuvarande väljer vi att acceptera denna bara med en viss sannolikhet beroende på hur mycket sämre den är och aktuellt värde på "temperaturen" β.

Denna Markovkedja får alltså p_β till stationär fördelning. Man kan vidare notera att vi slipper räkna ut den otroligt joxiga nämnaren (kanske 2^{30} termer) i uttrycket för p_β eftersom allt vi behöver göra är att beräkna $\Delta E(x, x') = E(x') - E(x)$.

Övergångsmatrisen

$$\boldsymbol{P}_\beta = (P_\beta(x, y), x, y \in \chi) = (P(X_\beta(n+1) = y | X_\beta(n) = x);\ x, y \in \chi)$$

för denna Markovkedja $(X_\beta(n);\ n = 0, 1, 2, \cdots)$ har $P_\beta(x, x') = 1/N(x)$ om x' är en granne till x som är "bättre" än x eftersom $N(x)$ var antalet grannar till x och vi skulle välja en sådan på måfå. Då är å andra sidan x en granne till x' som är "sämre" än x' och alltså gäller att

$$P_\beta(x', x) = \frac{1}{N(x')} \frac{p_\beta(x)}{p_\beta(x')}.$$

På samma sätt ser man att

$$P_\beta(x, x') = \frac{1}{N(x)} \frac{p_\beta(x')}{p_\beta(x)}$$

om x' är en "sämre" granne till x än x självt. Omvänt är då x en granne till x' som är bättre än x' så vi har $P_\beta(x', x) = \dfrac{1}{N(x')}$.

Eftersom $N(x) = N(x')$, (dvs det finns lika många grannar till både x och x') så får vi att man har

$$p_\beta(x)P_\beta(x, x') = p_\beta(x')P_\beta(x', x)$$

för två grannar både om x är bättre och sämre än x'. Denna relation är trivialt uppfylld om x' ej skulle vara granne till x (då är ju $P_\beta(x, x') = P_\beta(x', x) = 0$ eftersom man i ett steg bara kan gå till närmaste grannar). Denna Markovkedja är alltså tidsreversibel med fördelningen $p_\beta(x)$ och och har alltså denna till stationär fördelning. Detta kan också inses genom att förfarandet (inklusive acceptanssannolikheten) innebär att man använt sig av Metropolis-Hastings algoritm där förslagsfördelningen är den likformiga på de närmsta grannarna.

Vidare är kedjan (åtminstone i exemplet) uppenbarligen irreducibel (man kan komma från vilket tillstånd x till vilket annat tillstånd som helst. Vidare gäller att åtminstone $P_\beta(\widehat{x}, \widehat{x}) > 0$ (alla grannar till $\widehat{x}$ är ju sämre än $\widehat{x}$) så kedjan är alltså även aperiodisk. Detta medför att den också är ergodisk. Oavsett starttillstånd kommer den alltså efter lång tid att hamna i den stationära fördelningen (dvs p_β).

Vi gör nu följande algoritm:

1) Välj β ganska liten, dvs hög temperatur

2) Välj $X_\beta(0) = x_0$ godtycklig

3) Simulera X_β tills stationaritet uppnåtts

4) Öka β, dvs minska temperaturen

5) Upprepa från punkt 3 med gamla slutvärdet som startvärde

Det är svårt att ge tumregler (även om många har försökt göra det) om hur man skall öka på β utan man får pröva sig fram. T ex kan man öka β med en faktor $K > 1$ i varje steg 4. Vidare får man fundera över hur långa simuleringar som behövs för varje enskilt β-värde i steg 3.

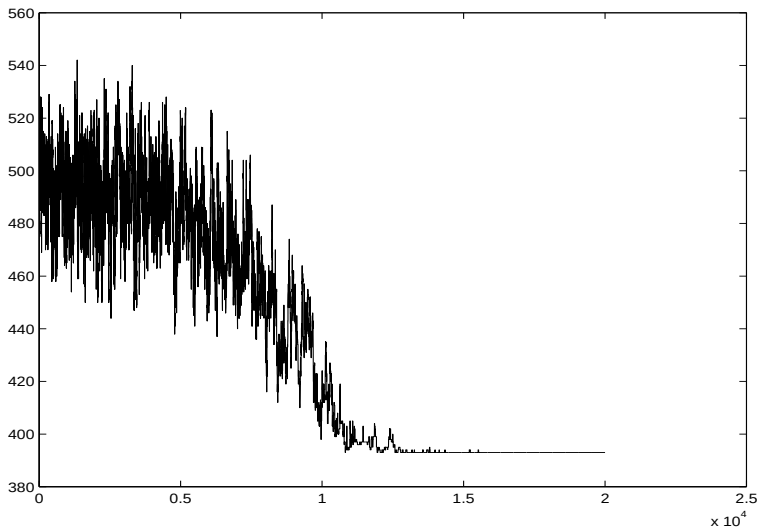
Man kan notera att ovanstående algoritm är ett sorts "icke-girigt" famlande efter successivt bättre punkter. Föreslås en granne som är sämre har man ändå positiv sannolikhet att gå dit, framför allt vid låga β, dvs höga temperaturer. Efterhand som β ökas (temperaturen sänks) så blir man mer "girig", dvs mindre benägen att flytta sig från en bra punkt.

Vid höga temperaturer, dvs i början då β är litet, undviker detta förfarande att man fastnar i ett lokalt minimum genom att man då har en icke-försumbar sannolikhet att förflytta sig från en (eventuellt bara lokal) minimipunkt. Efterhand som β ökas, dvs temperaturen sänks, så blir man mer obenägen att lämna en "bra" punkt.

Exempel 17.6 *Ett överbestämt binärt ekvationssystem*
Vi har två binära vektorer, som alltså består av 0:or och 1:or, $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$ och $\boldsymbol{b} = (b_1, b_2, \cdots, b_m)^T$ (med $m > n$) samt en binär $m \times n$-matris A. Vi vill hitta vektorn $\boldsymbol{x}$ som är en god lösning till det överbestämda ekvationssystet $A\boldsymbol{x} = \boldsymbol{b}$. Matrismultiplikationen skall ske "modulo 2", dvs $A\boldsymbol{x}$ blir för alla val av $\boldsymbol{x}$ en binär vektor. Som "bästa" lösning till ekvationssystemet tar vi den vektor $\boldsymbol{x}$ som minimerar $E(\boldsymbol{x}) = |A\boldsymbol{x} - \boldsymbol{b}|$ där $|\cdot|$ är komponentvis addition av absolutbeloppen. Detta betyder att "lösningen" $\boldsymbol{x}$ gör så få av ekvationerna som möjligt "felaktiga". Detta något okonventionella problem har (så vitt vi vet) inte någon explicit lösning. Man kan ju jämföra med minsta kvadratlösningen $\widehat{\boldsymbol{x}} = (A^T A)^{-1} A^T \boldsymbol{b}$ för "vanliga" vektorer då man vill minimera Euklidiska avståndet $|A\boldsymbol{x} - \boldsymbol{b}|$, t ex i samband med Allmänna linjära modellen.

Om man t ex har $n = 50$ finns $2^{50} \approx 1.13 \cdot 10^{15}$ tänkbara $\boldsymbol{x}$-vektorer! Detta problem lämpar sig för "Simulated annealing" där man startar med ett slumpmässigt valt $\boldsymbol{x}$ och sen konstruerar en Markovkedja i enlighet med ovanstående beskrivning.

Vi tog ett kraftigt överbestämt ekvationssytem med $m = 1000$ och startade med $\beta = 0.01$. Kedjan fick stega 500 steg och sen ökades β med en faktor 1.2, kedjan kördes ytterligare 500 steg osv. Detta upprepades 40 gånger, så att β till slut blev $1.2^{40} \cdot 0.01 \approx 15$.



Figur 17.10: Exempel på simulated annealing

Man kan i figur 17.10 notera hur kedjan hoppar rätt vilt i början av $E(\boldsymbol{x})$:s värde att döma. Dock verkar kurvan driva svagt neråt. Vid en någorlunda genomgående "temperatur" skedde ett dramatiskt omslag, då kedjan blir mer "girig" och fastnar i ett (förhoppningsvis globalt) minimum.

Detta förfarande innebär alltså att vi prövat $\leq 40 \cdot 500 = 20000$ $\boldsymbol{x}$-vektorer,

och med tanke på att det fanns över 10^{15} st tänkbara sådana är ju detta en mycket liten andel! □

17.8 *BUGS* Bayesian Inference Using the Gibbs Sampler

Gibbs-sampling är av stort intresse i samband med Bayesiansk statistik där vi vill kunna generera a-posteriori-fördelningen, dvs fördelningen för de ingående parametrarna givet observationerna. Eftersom

$$P(\Theta = \theta | X = x) = \frac{P(\Theta = \theta; X = x)}{P(X = x)} = \frac{P(X = x|\Theta = \theta)P(\Theta = \theta)}{P(X = x)}$$

ser vi att a-posteriori-fördelningen är proportionell mot

$$P(X = x|\Theta = \theta)P(\Theta = \theta) = P(\Theta = \theta; X = x)$$

och nämnaren

$$P(X = x) = \sum_u P(X = X|\Theta = u)P(\Theta = u)$$

utgör bara en normeringskonstant som dock är komplicerad att räkna ut. Eftersom alltså a-posteriori-fördelningen är känd så när som på en normeringskonstant innebär detta att MCMC skulle kunna vara mycket användbar. Speciellt Gibbs-sampling har kommit till stor användning där man alltså uppdaterar varje enskild parameter med betingade fördelningen för denna parameter givet alla övriga (samt observationerna). Det finns speciellt utvecklad programvara som gör detta nästan automatiskt utifrån en enkel grafisk representation av modellstrukturen. Speciellt *BUGS* som är utvecklad i Cambridge har kommit till stor användning. Programvaran som finns både för Unix, Dos och Windows finns att ladda ned via internet på adressen

`http://www.mrc-bsu.cam.ac.uk/bugs/`

tillsammans med ett antal manualer och genomräknade exempel.

Som introduktion till denna typ av modeller ser vi på ett konkret exempel.

17.8.1 Exempel – felintensiteter för pumpar

Vi har observerat 10 st pumpar av liknande typ och räknat antalet fel y_i, $i = 1, 2, \cdots, 10$ under varierande drifttider t_i, $i = 1, 2, \cdots, 10$. Vi ser y_i som ett utfall av Y_i som är $\text{Po}(\lambda_i t_i)$ där alltså λ_i är felintensiteten för pump nr i. Vi skulle traditionellt skatta λ_i med $\widehat{\lambda}_i = y_i/t_i$ enligt följande

Pump	1	2	3	4	5	6	7	8	9	10
t_i	94.3	15.7	62.9	126	5.24	31.4	1.05	1.05	2.1	10.5
y_i	5	1	5	14	3	19	1	1	4	22
$\widehat{\lambda}_i$	0.0530	0.0637	0.0795	0.1111	0.5725	0.6051	0.9524	0.9524	1.9048	2.0952

Traditionell analys skulle möjligen gå vidare genom att anta att pumparna har samma felintensitet λ och skatta λ med

$$\widehat{\lambda} = \frac{\sum_{i=1}^{10} y_i}{\sum_{i=1}^{10} t_i} = \frac{75}{350.34} = 0.2141$$

men detta skulle vara till klen hjälp om man nu t ex vill ha fördelningen för felantalet för en ny pump för t ex drifttiden 100. Skulle man verkligen tro att den hade den fixa felintensiteten 0.2141 och att därmed felantalet var Poissonfördelat med väntevärde $0.2141 \cdot 100 = 21.41$? Vi ser ju av de observerade λ_i-skattningarna att vissa pumpar har ganska höga felintensiteter (ca 2) medan andra har låga (0.05-0.10) så antagligen är pumppopulationen heterogen, dvs består av "bra" respektive "dåliga" pumpar. I denna typ av situation är ett Bayesianskt synsätt att föredra! □

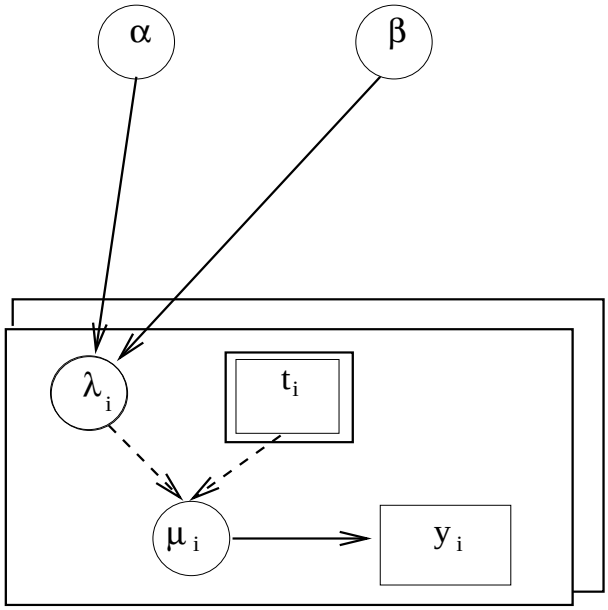
17.8.2 Bayesianska modeller – DAG-modeller

Man gör nu en Bayesiansk modell där λ_i är oberoende utfall av Λ som är (t ex) $\Gamma(\alpha, \beta)$-fördelad. Denna fördelning skall alltså beskriva felintensitetsfördelningen för pump-populationen. En ny vald pump kommer alltså att få en felintensitet enligt denna fördelning. Variabiliteten i denna fördelning för Λ beskriver då förhoppningsvis variabiliteten i pump-populationen.

α och β ger vi i sin tur a-priori-fördelningarna Exp(1) respektive $\Gamma(0.1, 1)$. Den sistnämnda fördelningen har att väntevärde 0.1/1=0.1 så $\Gamma(\alpha, \beta)$-fördelningen får ett väntevärde på ca 10 och en varians på ca 100 och är alltså mycket flack. Man kan uttrycka det så att a-priori-fördelningen för Λ är mycket utspridd (icke-informativ). Detta för att inte vår (ganska ogrundade) a-priori-uppfattning skall påverka a-posteriori-fördelningen för Λ för mycket.

En grafisk bild av ovanstående modell syns i figur 17.11 där vi låter $\mu_i = \lambda_i t_i$, för $i = 1, 2, \cdots, 10$.

Detta utgör ett exempel på en DAG-modell (Directed Acyclical Graph), dvs en riktad graf som saknar cykler, dvs man kan inte genom att följa pilarna komma tillbaka till utgångspunkten. Varje kvantitet i modellen dyker upp som en nod i grafen i figur 17.11. Direkta beroenden anges genom pilarna. Heldragna pilar svarar mot ett statistiskt beroende, medan streckade pilar svarar mot ett direkt funktionellt (deterministiskt) beroende. Dessa deterministiska beroenden (t ex att väntevärdet i Poissonfördelningen är $\mu_i = \lambda_i t_i$) införs egentligen mest för att förenkla förståelsen av figuren och när man identifierar de statistiska beroendena kan man se det som att dessa deterministiska beroenden "kollapsar". Cirklar betecknar genuint stokastiska kvantiteter medan rektanglar står för fixa storheter. De som är givna av försöksuppläggningen (i vårt fall t_i:na) betecknas med dubbla rektanglar, medan observationer (i vårt fall y_i:na) betecknas med enkla rektanglar. Upprepade storheter (i vårt fall de enskilda pumparna svarande mot index i) betecknas med "staplade ark".



Figur 17.11: DAG-graf för pump-exemplet

För att tolka grafen i figur 17.11 inför man följande terminologi. Låt v vara en nod i grafen. De övriga noder som har pilar direkt till v kallas v:s "föräldrar" medan de noder som kan nås direkt via en pil från v kallas v:s "barn". I denna process "kollapsar" man eventuella mellanliggande deterministiska noder. I vår figur är alltså λ_i:s föräldrar α och β, medan α:s och β:s barn är λ_i:na. På samma sätt är λ_i förälder till y_i och y_i är barn till λ_i. Vissa noder (i vårt fall α och β) saknar föräldrar och dessa noder måste vi ange a-priori-fördelningar för.

Vi gör antagandet att fördelningen för en viss nod v specificeras av v:s föräldrar genom de betingade fördelningarna.

Vi inför nu

$$\boldsymbol{X} = (X_1, X_2, \cdots, X_{22}) = (\alpha, \beta, \lambda_1, \cdots, \lambda_{10}, Y_1, Y_2, \cdots, Y_{10})$$

dvs vi har en 22-dimensionell fördelning. Dock kommer vi genomgående att betinga m a p de 10 sista komponenterna som ju svarar mot våra observationer $y_1, \cdots, y_{10}$.

Om vi kallar de d (i vårt fall $d = 22$) stokastiska noderna för $\boldsymbol{X} = (X_1, X_2, \cdots, X_d)$ så gäller alltså att

$$P(\boldsymbol{X} = \boldsymbol{x}) = \prod_{i=1}^{d} P(X_i = x_i | \text{föräldrarna till } X_i).$$

Man kan uppfatta det så att vi startar med noderna utan föräldrar och sen "arbetar oss igenom" grafen och med successiva betingar erhåller ovanstående

resultat. Att grafen är "icke-cyklisk" innebär att detta förfarande fungerar – vi riskerar inte att komma tillbaka till någon nod.

Vi får t ex (med lite slarvigt skrivsätt) i vårt pump-exempel att den simultana fördelningen blir

$$P(\alpha, \beta, \lambda_1, \cdots, \lambda_{10}, Y_1, \cdots, Y_{10}) =$$
$$= P(\alpha)P(\beta)\prod_{i=1}^{10}(P(\lambda_i|\alpha, \beta)P(Y_i|\lambda_i))$$

som alltså specificerar den simultana fördelningen för samtliga stokastiska noder i vårt diagram.

Eftersom vi vill göra en Gibbs-sampling är vi intresserade av fördelningen för t ex $X_i|\boldsymbol{X_{-i}} = \boldsymbol{x_{-i}}$ där som tidigare $\boldsymbol{X_{-i}}$ står för samtliga komponenter (noder) utom nr i. I praktiken kommer vi bara att vara intresserad av $i = 1, 2, \cdots, 12$ i vårt konkreta exempel.

Man inser att vi får

$$P\left(X_i = x|\boldsymbol{X_{-i}} = \boldsymbol{x_{-i}}\right) = \frac{P(X_i = x; \boldsymbol{X_{-i}} = \boldsymbol{x_{-i}})}{P(\boldsymbol{X_{-i}} = \boldsymbol{x_{-i}})} \propto P(X_i = x; \boldsymbol{X_{-i}} = \boldsymbol{x_{-i}}).$$

Men detta innebär att $P\left(X_i = x|\boldsymbol{X_{-i}} = \boldsymbol{x_{-i}}\right)$ är proportionellt mot de faktorer i $P(X_i = x; \boldsymbol{X_{-i}} = \boldsymbol{x_{-i}})$ som beror av x dvs vi får att

$$P\left(X_i = x|\boldsymbol{X_{-i}} = \boldsymbol{x_{-i}}\right) \propto$$
$$\propto P(X_i = x|\text{föräldrar till } X_i) \prod_{j \in \text{barn till } X_i} P(X_j = x_j|\text{alla föräldrar till } X_j)$$

Detta innebär att fördelningen för $X_i|\boldsymbol{X_{-i}}$ beror av föräldrarna till X_i samt av barn till X_i samt dessas eventuella andra föräldrar.

Detta betyder i pumpexemplet ovan att t ex betingade fördelningen för α givet alla de övriga dvs (med samma förkortade och slarviga skrivsätt som tidigare) blir

$$P(\alpha|\beta, \lambda_1 \cdots, \lambda_{10}, Y_1, \cdots, Y_{10}) \propto$$
$$\propto P(\alpha)P(\beta)\prod_{i=1}^{10}(P(\lambda_i|\alpha, \beta)P(Y_i|\lambda_i))$$

och vi är intresserade av hur detta beror av just α. Visste vi detta α-beroende skulle vi ju kunna normera på rätt sätt så att vi får en sannolikhetsfördelning. Man ser att α-beroendet dyker upp i $P(\alpha)$ (dvs a-priori-fördelningen för α) och i $\prod_i P(\lambda_i|\alpha, \beta)$ dvs fördelningen för "barnen" λ_i och dessas övriga "föräldrar" β. Alltså har vi att

$$P(\alpha|\beta, \lambda_1 \cdots, \lambda_{10}, Y_1, \cdots, Y_{10}) \propto$$
$$\propto P(\alpha)\prod_i P(\lambda_i|\alpha, \beta)$$

Man kan se dessa två delar av α:s betingade fördelning som en a-priori-del och en likelihood-del.

α var i detta fall en nod utan föräldrar men med barn. Om vi tar en nod som λ_2 som har både föräldrar och barn så blir det på ungefär samma sätt eftersom fördelningen för λ_2 är känd om värdet för dess föräldrar α och β är kända. På samma sätt är betingade fördelningen för t ex λ_2 givet alla övriga proportionell mot de faktorer som innehåller λ_2 i simultana fördelningen

$$P(\alpha)P(\beta)\prod_{i=1}^{10}\left(P(\lambda_i|\alpha,\beta)P(Y_i|\lambda_i)\right).$$

Detta ger att

$$P(\lambda_2|\alpha,\beta,\lambda_1,\lambda_3,\cdots\lambda_{10},Y_1,\cdots,Y_{10}) \propto$$
$$\propto P(\lambda_2|\alpha,\beta)P(Y_2|\lambda_2).$$

17.8.3 Pumpexemplet i *BUGS*

Med programvaran BUGS ser koden i pumpexemplet ovan ut på följande sätt (# ger kommentarer)

model pump;

const

N = 10; # antalet pumpar

var

theta[N], # felintensitet hos pumparna
y[N], # antalet fel hos pumparna
t[N], # drifttiderna
alpha,beta, # parametrar för gamma-fördelningen
lambda[N]; # theta[]*t[]

data

t, y in "pump.dat";
inits in "pump.in";

{
for (i in 1:N){
theta[i] ~ dgamma(alpha,beta);
lambda[i] <- theta[i]*t[i];
y[i] ~ dpois(lambda[i])
}
alpha ~ dexp(1.0);
beta ~ dgamma(0.1,1.0);
}

Program-språkets syntax framgår rätt väl och följande delar kan urskiljas:

model ger namn på vår modell,

const ger värden åt dimensionsstorleken,

var "deklarerar" ingående variabler,

data talar om var data och startvärden finns,

nästa del specificerar modellen samt a-priori-fördelning på α respektive β.

~ betecknar "fördelad som"

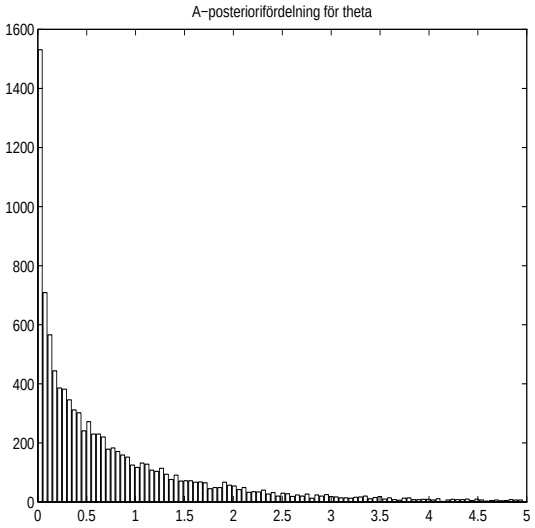
$<-$ betecknar funktionella samband.

En körning av BUGS kan se ut som följande

compile("pump.bug") – som kompilerar programmet pump.bug
monitor(alpha) – som säger att vi vill få ut α
monitor(beta) – som säger att vi vill få ut β
update(10000) – antalet uppdateringar i Markovkedjan
stats(alpha) – ger statistik om α
stats(beta) – ger statistik om β,
q() – utloggning ur BUGS (q står för "quit")

Programmet levererar då bl a en utdata-fil pump.out med 10000 observationer av (α, β) och denna kan plockas in i t ex Matlab. Denna tvådimensionella fördelning utgör alltså a-posteriori-fördelningen för (α, β) givet observationerna. Man kan sedan generera den prediktiva fördelningen för Λ, dvs den fördelning som felintensiteten hos en slumpmässigt vald pump har. Med lite slarvigt skrivsätt kan man säga att Λ är $\Gamma(\alpha, \beta)$ där (α, β) har a-posteriori-fördelningen. Denna prediktiva fördelning beskriver alltså uppfattningen om fördelningen av felintensiteten hos pump-populationen. Denna fördelning kan ju sedan användas för att lotta fram en ny pump-felintensitet för t ex simuleringsändamål. Detta innebär i princip att man genererar den prediktiva fördelningen för felintensiteten givet observationerna, dvs den fördelning för felintensiteten som en ny slumpmässigt vald pump skulle ha. Jämför diskussionen i avsnitt 15.9.2. Denna prediktiva fördelning kan sedan användas för att göra utsagor och beräkningar om egenskaper hos en slumpmässigt vald pump t ex sannolikheten att den kommer att ha 3 fel under 100 drifttimmar.

Vad som gjorts simuleringsmässigt är att med hjälp av BUGS simulera 10000 utfall av (α, β) och sen i Matlab simulerat 10000 utfall av $\Gamma(\alpha, \beta)$-fördelningen utgående från dessa simulerade värden på α och β. Utifrån den simulerade a-posteriori-fördelningen för (α, β) har alltså den prediktiva fördelningen för Λ simulerats.
Fördelningen i figur 17.12 avspeglar alltså vad vi tror om felintensitets-fördelningen för pumpar av den aktuella typen. Man kan notera att även rätt stora felintensiteter Λ inte är helt osannolika (t ex $P(\Lambda > 4) = 0.0347$ enligt figuren). Med tanke på att 2 av de 10 observerade pumpar hade en felintensitet på ca 2 så är det inte konstigt att pumpar med så stor felintensitet som 4 kan förutsägas finnas.



Figur 17.12: Prediktiv fördelning för felintensiteten Λ i pump-exemplet.

Med hjälp av denna fördelning för Λ (felintensitetsfördelningen för pump-populationen) enligt figur 17.12 kan man naturligtvis även beräkna sannolikheten att en sådan framlottad pump på tiden 10 uppvisar k st fel. Vi vet ju att om Y=antalet fel för en ny pump under tiden 10 är $\text{Po}(10 \cdot \lambda)$ givet att $\Lambda = \lambda$ och alltså är

$$P(Y = k) = \int_0^\infty P(Y = k|\Lambda = \lambda) f_\Lambda(\lambda) d\lambda = \int_0^\infty \frac{(10\lambda)^k}{k!} e^{-10\lambda} f_\Lambda(\lambda) d\lambda$$

där $f_\Lambda(\lambda)$ är just den prediktiva fördelningen som vi hade simulerat. För t ex $k = 0$ försöker vi alltså beräkna $E(e^{-10\Lambda})$ i prediktiva fördelningen för Λ.

Notera att detta utgör ett exempel på att vi vill beräkna $E(g(\boldsymbol{X}))$ för en given funktion g. Detta kan alltså erhållas ur MCMC-simuleringen genom att

$$\lim_{B\to\infty} \frac{1}{B} \sum_{i=1}^{B} g(\boldsymbol{x}_i) = E(g(\boldsymbol{X}))$$

enligt Ergod-satsen.

BUGS levererar också skattningar och konfidensintervall för parametrarna, t ex för α

mean	sd	2.5% :	97.5% CI	median	sample
6.972E-1	2.696E-1	2.882E-1	1.320E+0	6.566E-1	10000

eller β

mean	sd	2.5% :	97.5% CI	median	sample
9.271E-1	5.507E-1	1.901E-1	2.274E+0	8.161E-1	10000

Notera dock att i a-posteriori-fördelningen är α och β beroende – de har ju gemensamma barn, dvs λ_i:na och dessa ger i sin tur upphov till barnen

$y_1, y_2, \cdots, y_{10}$. Alltså är α och β beroende i a-posteriori-fördelningen, dvs givet observationerna! Man kan alltså inte dra någon direkt slutsats av de endimensionella a-posteriori-fördelningarna för α respektive β.

17.8.4 Regressionsmodell som DAG-modell

Vi ger ytterligare ett exempel på en DAG-modell nämligen en regressionsmodell.

Vi har observationer y_{ij} =värde för patient i vid tidpunkt t_{ij}, $i = 1, 2, \cdots, n$ och $j = 1, 2, \cdots, n_i$. Vi tror att dessa data kan beskrivas av regressionsmodeller – en för varje enskild patient – dvs att y_{ij} är utfall av

$$Y_{ij} = \alpha_i + \beta_i t_{ij} + \epsilon_{ij},\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n_i.$$

där ϵ_{ij} är oberoende $N(0, \sigma^2)$. Vi har alltså en regressionsmodell för varje enskild patient och vi är t ex intresserade av variabiliteten i β_i:na. Här är en Bayesiansk ansats utmärkt. Vi ser alltså α_i:na och β_i:na som utfall av bakomliggande variabler α respektive β. De enskilda β_i:na är inte speciellt intressanta utan det är fördelningen för dessa som är målet för inferensen. Detta i analogi med pump-exemplet där de enskilda pumparna inte var av primärt intresse utan snarare dessa pumpars variabilitet vad gällde felintensiteten.

Vi ansätter följande modell:

Y_{ij} är $N(\mu_{ij}, \sigma^2)$

$\mu_{ij} = \alpha_i + \beta_i t_{ij}$

α_i är $N(\alpha_0, \sigma_\alpha^2)$

β_i är $N(\beta_0, \sigma_\beta^2)$.

Grafiskt kan denna modell representeras av grafen i figur 17.13
För att få en fullständig modell måste vi ge a-priori-fördelningar åt noderna utan "föräldrar", dvs σ^2, α_0, σ_α^2, β_0 och σ_β^2. Vi vill ju välja dessa a-priori-fördelningar så att de inte påverkar resultatet av analysen särskilt mycket. Vi väljer

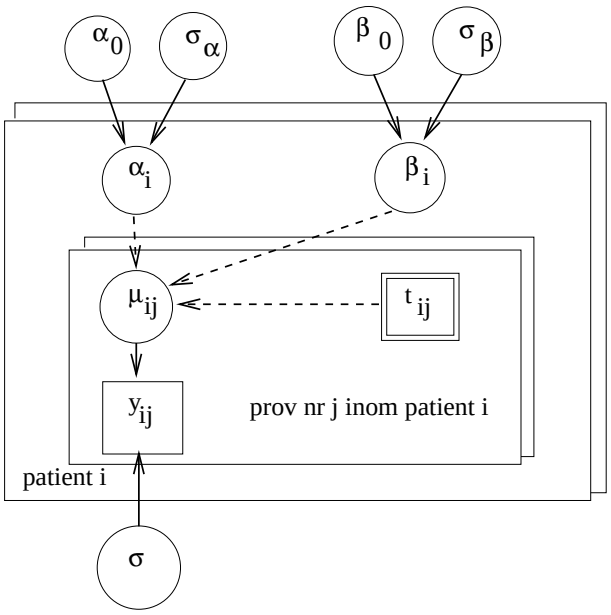
α_0 och β_0 som $N(0, 100^2)$

σ^{-2}, σ_α^{-2}, σ_β^{-2} som $\Gamma(0.01, 0.01)$.

$\Gamma(0.01, 0.01)$ har väntevärde 1 och varians $0.01/0.01^2 = 100$ så dessa fördelningar är synnerligen utspridda och nästan konstanta i det aktuella området. Dessutom är ju Γ-fördelningar positiva som passar för varianser.

Vi får följande *BUGS*-kod för att specificera modellen:

```
{
 for(i in 1:I)
  {
   for j in 1:n[i])
```



Figur 17.13: DAG för regressions-exemplet.

```
   {
     y[i,j]  ~ dnorm(mu[i,j],tau);
     mu[i,j] <- alpha[i]+beta[i]*t[i,j];
    }
   beta[i] ~ dnorm(beta0, tau.beta);
   alpha[i] ~ dnorm(alpha0, tau.alpha);
  }
tau ~ dgamma(0.01,0.01);
tau.alpha ~ dgamma(0.01,0.01);
tau.beta ~ dgamma(0.01,0.01);
alpha0 ~ dnorm(0,0.0001);
beta0 ~ dnorm(0,0.0001);

sigma <- 1/sqrt(tau);
sigma.alpha <- 1/sqrt(tau.alpha);
sigma.beta <- 1/sqrt(tau.beta);
}
```

Man kan notera att funktionen `dnorm(a,b)` i *BUGS* svarar mot en normalfördelning med väntevärde a och varians $1/b$, dvs att andra-parametern står för 1/variansen.

Vi har därför inför de "inversa" varianserna tau, tau.alpha och tau.beta för parametrarna σ^{-2}, σ_α^{-2} respektive σ_β^{-2}. Det avslutande avsnittet transformerar tillbaka till den mer traditionella skalan. Denna lite okonventionella parametrisering beror på att man vill ha parametrar som gör fördelningarna log-konkava av skäl som framgår senare.

I ovanstående modell skulle man få snabbare insvängning mot den stationära fördelningen om man avcentrerar t_{ij}-na med $\bar{t} = \sum_{ij} t_{ij} / \sum_i n_i$ eftersom α_i och β_i då blir mer oberoende. Detta helt i analogi med situationen då vi simulerade från en två-dimensionell normalfördelning med kraftig korrelation.

17.8.5 Variansanalys med ordningsrestriktioner

Som ett exempel på en typ av modell som det skulle vara i stort sett omöjligt att analysera på traditionellt sätt kan man ge följande som är hämtat ur exempelsamlingen volym 2 för *BUGS* ("Marsbars")

Vi har följande data:

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$i=1$	0.982	1.902	3.797	-1.531	0.570
$i=2$	-1.414	1.356	1.287	-3.629	-3.413
$i=3$	-1.601	4.713	0.814	0.834	-2.082
$i=4$	-4.912	-4.541	-4.768	-9.051	-2.744

Vi låter Y_{ij} vara $\mathrm{N}(\mu_{ij}, \sigma^2)$ med $\mu_{ij} = \alpha_i + \beta_j$ där vi har restriktionerna $\alpha_1 > \alpha_2 > \alpha_3 > \alpha_4$ samt $\beta_1 < \beta_2 < \beta_3$ samt $\beta_3 > \beta_4 > \beta_5$. Dessa data kan tänkas uppstå vid test av konsumentpreferenser för olika choklad där i anger priset (växande i i) och j mängden tillsatt socker där man vet att mellanläget $j = 3$ är optimalt.

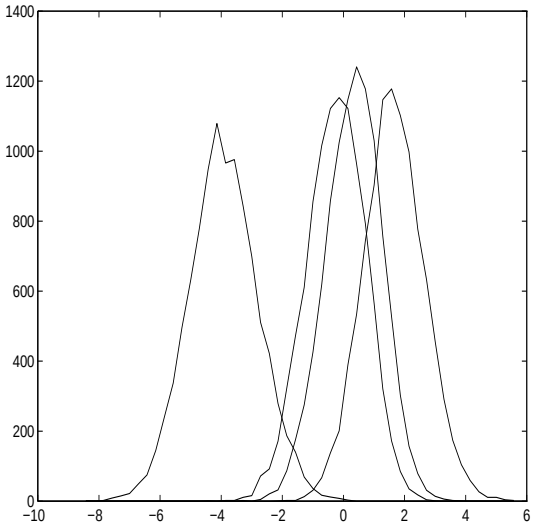
Följande *BUGS*-kod kan användas:

```
model MarsBars;
const
   cols = 5, rows = 4,
   tau.alpha = 0.20, mu.alpha = 0.0,
   tau.beta = 0.20, mu.beta = 0.0;
var
   Y[rows,cols], mu[rows,cols], tau, beta[cols],
   alpha[rows], bound, sigma;
data Y in "marsbars.dat";
inits in "marsbars.in";
{
   for(j in 1:cols){
      for (i in 1:rows){
          mu[i,j] <- alpha[i] + beta[j];
          Y[i,j]   ~ dnorm(mu[i,j],tau)
      }
   }
   tau      ~ dgamma(1.0E-3,1.0E-3);
   sigma   <- 1/sqrt(tau);
   alpha[1] ~ dnorm(mu.alpha,tau.alpha)I(alpha[2],);
   alpha[2] ~ dnorm(mu.alpha,tau.alpha)I(alpha[3],alpha[1]);
   alpha[3] ~ dnorm(mu.alpha,tau.alpha)I(alpha[4],alpha[2]);
   alpha[4] ~ dnorm(mu.alpha,tau.alpha)I(,alpha[3]);
```

```
   bound   <- max(beta[2],beta[4]);
   beta[1]  ~ dnorm(mu.beta,tau.beta)I(,beta[2]);
   beta[2]  ~ dnorm(mu.beta,tau.beta)I(beta[1],beta[3]);
   beta[3]  ~ dnorm(mu.beta,tau.beta)I(bound,);
   beta[4]  ~ dnorm(mu.beta,tau.beta)I(beta[5],beta[3]);
   beta[5]  ~ dnorm(mu.beta,tau.beta)I(,beta[4]);
}
```

Denna kod svarar mot modellen :
Y_{ij} är $\mathrm{N}(\mu_{ij}, 1/\tau)$
$\mu_{ij} = \alpha_i + \beta_j$
α_i är $\mathrm{N}(\mu_\alpha, 1/\tau_\alpha)$ med $\mu_\alpha = 0$ och $\tau_\alpha = 0.2$ (dvs α_i är $\mathrm{N}(0, 5)$)
β_i är $\mathrm{N}(\mu_\beta, 1/\tau_\beta)$ med $\mu_\beta = 0$ och $\tau_\beta = 0.2$ (dvs β_i är $\mathrm{N}(0, 5)$)
τ är $\Gamma(10^{-3}, 10^{-3})$
samt med de restriktioner rörande α_i:na respektive β_j:na som angavs ovan. Notera att man bygger in restriktionerna genom att använda sig av funktionen $I(a, b)$ som anger att värdena måste ligga mellan a och b.

En körning där man genererade 10000 observationer av vektorn $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ tog ca 30 sekunder på en 486-a och gav resultat enligt figur 17.14 för de 4 endimensionella fördelningarna.



Figur 17.14: A-posteriori-fördelningar för $\alpha_1, \alpha_2, \alpha_3$ och α_4 (från höger till vänster).

Loggen från *BUGS* (körd i DOS-versionen) blir

```
Bugs>compile("marsbars.bug")
Bugs>update(10000)      time for    10000  updates was  00:00:12
Bugs>monitor(alpha)
Bugs>update(10000)      time for    10000  updates was  00:00:12
Bugs>q()
```

där man kan se att kedjan uppdateras 10000 gånger innan vi började följa *alpha*-värdena – detta för att beroendet av initialvärdena 1.0, 0.5, 0.0, -1.0 för α_1 till α_4 och -0.5, 0.0, 1.0, -0.5, -1.0 för β_1 till β_5 samt $\tau = 1.0$ skulle klinga av. Outputen är en 40000 × 2-fil med α-värden i kolumn 2 och organiserad så att de första 10000 värdena är α_1-värden, de följande 10000 är α_2-värden etc. Notera att det naturligtvis finns beroende mellan de olika α_i:na och att man får den 4-dimensionella a-posteriori-fördelningen för $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ genom att para ihop dem. Lämpligen skapar man i Matlab en vektor `alpha` genom operationen

```
for i=1:4;
 alpha(:,i)=bugs((i-1)*10000+1:i*10000,2);
end;
```

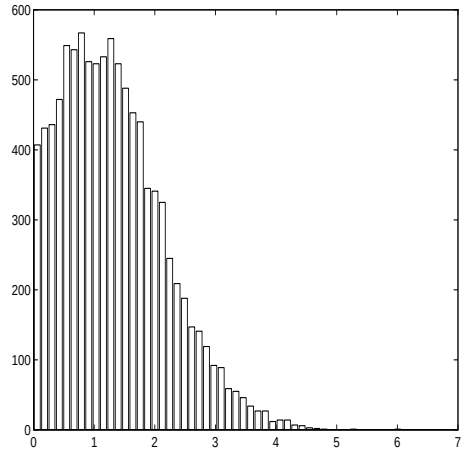
där `bugs` är den vektor som kommer ut ur *BUGS* (defaultnamnet är `bugs.out`).

Skattningarna av parametrarna $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ (dvs väntevärdena i respektive a-posteriori-fördelning) blir $1.6533,\ 0.3349,\ -0.2538,\ -3.9247$.

Om man t ex skulle vara intresserad av a-posteriori-fördelningen för $\alpha_1 - \alpha_2$ kan man få ett histogram (se figur 17.15) av den genom kommandot

```
hist(alpha(1,:)-alpha(2,:),50)
```

som ger 50 "fack":



Figur 17.15: A-posteriori-fördelning för $\alpha_1 - \alpha_2$.

Man ser tydligt att restriktionen $\alpha_1 > \alpha_2$ automatiskt byggs in i a-posteriori-fördelningen för $\alpha_1 - \alpha_2$. Naturligtvis kan man också konstruera konfidensintervall för de enskilda parametrarna eller funktioner av dem (t ex för $\alpha_1 - \alpha_2$).

17.9 Acceptance-Rejection sampling

Vi vill som tidigare generera utfall av $\boldsymbol{X}$ med en täthet $f_{\boldsymbol{X}}(\boldsymbol{x})$ proportionell mot $\pi(\boldsymbol{x})$ dvs enligt tätheten

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{\int_{R^d} \pi(\boldsymbol{u}) d\boldsymbol{u}}.$$

Det kan ibland vara enklast att skaffa sig en (onormerad) kandidattäthet $a(\boldsymbol{x})$ som dominerar $\pi(\boldsymbol{x})$ dvs som uppfyller

$$0 \leq \pi(\boldsymbol{x}) \leq a(\boldsymbol{x})$$

och generera utfall enligt $a(\boldsymbol{x})$ dvs enligt tätheten

$$f_{\boldsymbol{Y}}(\boldsymbol{y}) = \frac{a(\boldsymbol{x})}{\int_{R^d} a(\boldsymbol{u}) d\boldsymbol{u}}$$

och sedan välja att acceptera eller förkasta dessa förslag med korrekt avpassad sannolikhet så att fördelningen för de accepterade förslagen får en täthet proportionell mot målfördelningen $\pi(\boldsymbol{x})$, dvs $f_{\boldsymbol{X}}(\boldsymbol{x})$.

Följande algoritm utför detta:

1) Generera ett förslag $\boldsymbol{Y}$ med tätheten proportionell mot $a(\boldsymbol{y})$ dvs med tätheten

$$f_{\boldsymbol{Y}}(\boldsymbol{y}) = \frac{a(\boldsymbol{y})}{\int_{R^d} a(\boldsymbol{u}) d\boldsymbol{u}}$$

2) Generera ett $R(0,1)$-fördelat slumptal U

3) Om

$$U \leq \frac{\pi(\boldsymbol{Y})}{a(\boldsymbol{Y})}$$

accepteras förslaget. Om det inte accepteras så genererar man ett nytt förslag enligt punkt 1.

Man genererar alltså förslag tills något av dessa accepteras. Ju närmare a-funktionen ligger π-funktionen desto större andel av förslagen kommer att accepteras.

Sats 17.5 *Det accepterade förslaget har täthet proportionell mot* $\pi(\boldsymbol{x})$.

Bevis:
Det accepterade förslaget $\boldsymbol{X}$ uppfyller för en godtycklig d-dimensionell delmängd A

$$P(\boldsymbol{X} \in A) = P\left(\boldsymbol{Y} \in A | \boldsymbol{Y} \text{ accepteras}\right) = \frac{P\left(\boldsymbol{Y} \in A; U \leq \frac{\pi(\boldsymbol{Y})}{a(\boldsymbol{Y})}\right)}{P\left(U \leq \frac{\pi(\boldsymbol{Y})}{a(\boldsymbol{Y})}\right)}.$$

Vi får genom att betinga på utfallet av $\boldsymbol{Y}$ att

$$P\left(\boldsymbol{Y} \in A; U \leq \frac{\pi(\boldsymbol{Y})}{a(\boldsymbol{Y})}\right) = \int_A P\left(U \leq \frac{\pi(\boldsymbol{Y})}{a(\boldsymbol{Y})} \middle| \boldsymbol{Y} = \boldsymbol{y}\right) f_{\boldsymbol{Y}}(\boldsymbol{y}) d\boldsymbol{y} =$$

$$= \int_A P\left(U \leq \frac{\pi(\boldsymbol{y})}{a(\boldsymbol{y})}\right) f_{\boldsymbol{Y}}(\boldsymbol{y}) d\boldsymbol{y} = \int_A \frac{\pi(\boldsymbol{y})}{a(\boldsymbol{y})} \cdot \frac{a(\boldsymbol{y})}{\int_{R^d} a(\boldsymbol{u}) d\boldsymbol{u}} d\boldsymbol{y} = \frac{\int_A \pi(\boldsymbol{y}) d\boldsymbol{y}}{\int_{R^d} a(\boldsymbol{u}) d\boldsymbol{u}}$$

och med $A = R^d$ får vi på samma sätt (eller genom att bara låta $A = R^d$ i ovanstående uttryck)

$$P\left(U \leq \frac{\pi(\boldsymbol{Y})}{a(\boldsymbol{Y})}\right) = \frac{\int_{R^d} \pi(\boldsymbol{y}) d\boldsymbol{y}}{\int_{R^d} a(\boldsymbol{u}) d\boldsymbol{u}}$$

dvs vi får

$$\int_A f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} = P(\boldsymbol{X} \in A) = \frac{P\left(\boldsymbol{Y} \in A; U \leq \frac{\pi(\boldsymbol{Y})}{a(\boldsymbol{Y})}\right)}{P\left(U \leq \frac{\pi(\boldsymbol{Y})}{a(\boldsymbol{Y})}\right)} = \frac{\int_A \pi(\boldsymbol{y}) d\boldsymbol{y}}{\int_{R^d} \pi(\boldsymbol{y}) d\boldsymbol{y}} =$$

$$= \int_A \frac{\pi(\boldsymbol{x})}{\int_{R^d} \pi(\boldsymbol{u}) d\boldsymbol{u}} d\boldsymbol{x}$$

dvs $\boldsymbol{X}$ har tätheten

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{\int_{R^d} \pi(\boldsymbol{u}) d\boldsymbol{u}}$$

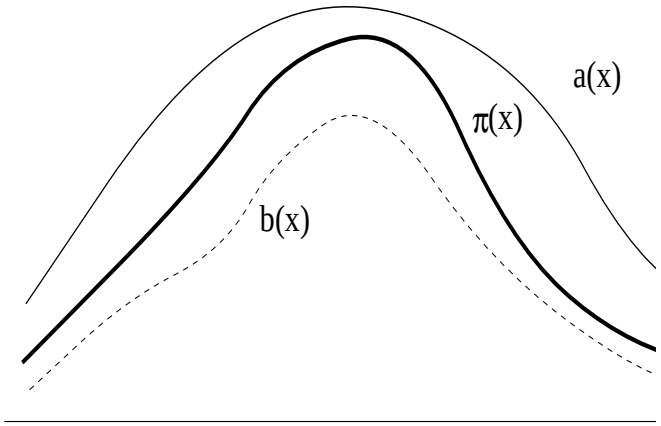
dvs den åstundade tätheten. □

Om vi dessutom stänger inne $\pi(\boldsymbol{x})$ både neråt och uppåt, dvs hittar en funktion $b(\boldsymbol{x})$ så att

$$0 \leq b(\boldsymbol{x}) \leq \pi(\boldsymbol{x}) \leq a(\boldsymbol{x})$$

(se figur 17.16) där tanken är att $b(\boldsymbol{x})$ ska vara lättare att beräkna än $\pi(\boldsymbol{x})$ så kan vi komplettera algoritmen ovan med att innan vi kollar om $U \leq \pi(\boldsymbol{y})/a(\boldsymbol{y})$ så kollar vi om $U \leq b(\boldsymbol{y})/a(\boldsymbol{y})$ och accepterar i så fall förslaget.
Om speciellt $\pi(\boldsymbol{x})$ är log-konkav, dvs att $\log(\pi(\boldsymbol{x}))$ är en konkav funktion kan man successivt förbättra a respektive b eftersom en konkav funktion ligger över sina kordor. Man kan alltså successivt konstruera a och b i enlighet med figur 17.17. De tidigare erhållna slumptalen betecknas med pilar längs x-axeln i figur 17.17.
Detta betyder att om man genererar många utfall kommer dessa uppdaterade a och b att ligga mycket nära π-funktionen vilket gör att nästan alla förslag kommer att accepteras. Simuleringen av fördelningen går alltså allt snabbare.

De resulterande a- respektive b-funktionerna är styckvisa exponentialfunktioner, som det är lätt att generera utfall av repektive beräkna. Ovanstående förfarande brukar benämnas "Adaptive Rejection Sampling" och *BUGS* använder sig av sådan och kräver i princip att de angivna fördelningarna är just



Figur 17.16: Måltätheten $\pi(\boldsymbol{x})$ instängd mellan två funktioner.

log-konkava. Detta är t ex orsaken till att man lägger a-priori-fördelning på 1/variansen i stället för på variansen för normalfördelning. Med $\tau = 1/\sigma^2$ som parameter i stället för σ^2 blir ju t ex normalfördelningstätheten log-konkav i τ.

Detta innebär att man i *BUGS* är hänvisad till vissa speciella parametriseringar och a-priori-fördelningar även om friheten är relativt stor. Speciellt praktiskt är att man mycket lätt kan bygga in restriktioner på parametrarna av typen $\theta_1 < \theta_2$ eftersom dessa restriktioner är lätta att hantera vid uppdatering eftersom man gör detta med den betingade fördelningen.

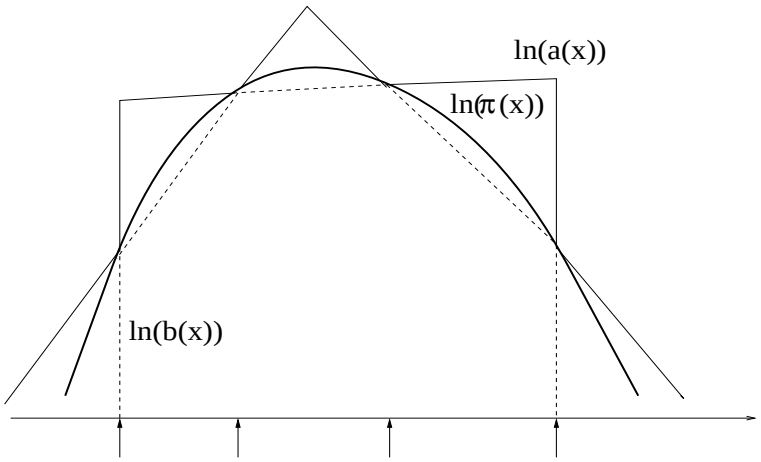
Den intresserade hänvisas till manualen för *BUGS*.

17.10 Lite om statistisk mekanik

Som avslutning på detta avsnitt om MCMC ger vi ett mycket kort diskussion om Statistisk Mekanik vilket kan vara på sin plats eftersom det var just inom det området som MCMC (framför allt Metropolis-algoritmen) hade sitt ursprung. Det kan också ge en motivering till varför det kan vara intressant att kunna simulera t ex spinn-system som vi gjorde i ett tidigare avsnitt.

Vi har ett fysikaliskt system med ett stort (men ändligt antal) tillstånd $\boldsymbol{x}$ där $\boldsymbol{x} \in \Omega$ och varje sådant tillstånd har en energi $E(\boldsymbol{x})$. Systemet befinner sig i termisk jämvikt med ett omgivande (oändligt) system som har en temperatur $T = 1/\beta$. Systemet är alltså isolerat men inte för energiutbyte med det omgivande systemet ("värmebadet"). Sannolikheten att systemet befinner sig i tillståndet $\boldsymbol{x}$ är

$$P_\beta(\boldsymbol{x}) = \frac{e^{-\beta E(\boldsymbol{x})}}{Z(\beta)}$$



Figur 17.17: Måltätheten $\pi(\boldsymbol{x})$ stängs inne ovanifrån och underifrån succesivt allt bättre.

där

$$Z(\beta) = \sum_{\boldsymbol{y}\in\Omega} e^{-\beta E(\boldsymbol{y})}$$

är en normeringskonstant – den s k tillståndssumman. Detta är den kanoniska fördelningen eller Gibbs-fördelningen.

17.10.1 Härledning av kanoniska fördelningen

Ekvivalent med att vårt system befinner sig ett värmebad är att tänka sig N kopior av systemet som det kan utbyta energi med. Vi kommer att vilja låta $N \to \infty$ och i samband med denna gränsövergång får vi "värmebadet".

Vart och ett av dessa N system har en viss energi och om vi låter a_i st av dem ha energin E_i gäller alltså att

$$\sum_i a_i = N, \text{ och } \sum_i a_i E_i = \text{Totalenergin} = U_{tot}$$

två kvantiteter som alltså är fixa. Dessa system kan permuteras och det mest sannolika tillståndet $a_1^*, a_2^*, \cdots$ är det som kan permuteras på flest sätt.

Antalet sätt Ω som man kan permutera systemen är

$$\Omega = \frac{N!}{a_1! a_2! \ldots a_i! \cdots}$$

och det mest sannolika tillståndet är alltså det som maximerar Ω (eller ekvivalent $\ln(\Omega)$) under bivillkoren $\sum_i a_i = N$ och $\sum_i a_i E_i = U_{tot}$. Detta svarar i princip mot en multinomial fördelning.

Vi kan hitta denna maximalt sannolika fördelning $a_1^*, a_2^*, \cdots$ över de olika energinivåerna $E_1, E_2, \cdots$ genom att med hjälp av Lagrange-multiplikatormetoden maximera $\ln(\Omega)$ under de två bivillkoren.

Vi kommer att låta $N \to \infty$ vilket också innebär att a_i:na $\to \infty$. Vi får med Stirlings formel

$$\ln(n!) \approx \ln(\sqrt{2\pi}) + n\,(\ln(n) - 1) + \ln(\sqrt{n}).$$

Vi kommer att kunna försumma första (konstant) och sista termen (av obetydlig storleksordning i n).

Vi får alltså

$$\ln(\Omega) = \text{konstant } + \ln(N!) - \sum_i \ln(a_i!) \approx$$

$$\approx N(\ln(N) - 1)) - \sum_i a_i(\ln(a_i) - 1) - \frac{1}{2}\sum_i \ln(a_i)$$

Vi får alltså med Lagrange-metoden och Lagrange-variablerna λ och β att (sista summan försummas i $\ln(\Omega)$)

$$\frac{\partial}{\partial a_i}(\ln\Omega - \lambda(\sum_i a_i - N) - \beta(\sum_i a_i E_i - U_{tot})) =$$

$$= -(\ln(a_i) - 1) - a_i\frac{1}{a_i} - \lambda - \beta E_i = -\ln(a_i) - \lambda - \beta E_i$$

och detta uttryck måste alltså vara 0 för att vi skall få ett maximum. Vi får alltså

$$-\ln(a_i^*) - \lambda - \beta E_i = 0$$

dvs

$$a_i^* = e^{-\lambda - \beta E_i}$$

och λ kan bestämmas ur bivillkoret $\sum_i a_i = N$ och vi får

$$e^{-\lambda}\sum_i e^{-\beta E_i} = N.$$

Sannolikheten P_i att systemet har energin E_i är alltså

$$P_i = \frac{a_i^*}{N} = \frac{e^{-\beta E_i}}{\sum_j e^{-\beta E_j}}$$

vilket just är den kanoniska fördelningen.

17.10.2 Entropi

Vi låter

$$g(\beta) = \ln Z(\beta) = \ln\left(\sum_{\boldsymbol{y}\in\Omega} e^{-\beta E(\boldsymbol{y})}\right)$$

dvs vi har

$$P_\beta(\boldsymbol{x}) = e^{-\beta E(\boldsymbol{x}) - g(\beta)}.$$

Om vi deriverar $g(\beta)$ m a p β får vi

$$g'(\beta) = \frac{d\ln(Z(\beta))}{d\beta} = -\frac{\sum_{\boldsymbol{x}} E(\boldsymbol{x})e^{-\beta E(\boldsymbol{x})}}{Z(\beta)} =$$

$$= -\sum_{\boldsymbol{x}} E(\boldsymbol{x})P_\beta(\boldsymbol{x}) = -\sum_{\boldsymbol{x}} E(\boldsymbol{x})e^{-\beta E(\boldsymbol{x}) - g(\beta)}.$$

Vi ser också att $g'(\beta) = -\langle E\rangle_\beta = -$ medelenergin (som funktion av β). Fysiker betecknar ofta väntevärde för en kvantitet q med $\langle q\rangle$. Vi får då

$$g''(\beta) = \sum_{\boldsymbol{x}} \left(E(\boldsymbol{x})^2 + g'(\beta)E(\boldsymbol{x})\right) P_\beta(\boldsymbol{x}) =$$

$$= \langle E^2\rangle_\beta - (\langle E\rangle_\beta)^2 = \mathrm{Var}_\beta(E) = v(\beta) > 0$$

dvs $g(\beta)$ är konvex och vi har (eftersom $g'(\beta) = -\langle E\rangle_\beta$) att

$$\frac{d\langle E\rangle_\beta}{d\beta} = -\mathrm{Var}_\beta(E) < 0$$

som alltså är en avtagande funktion i β, dvs växande i $T = 1/\beta$.

Vi definierar nu Gibbs entropi som

$$h(\beta) = -\sum_{\boldsymbol{x}} P_\beta(\boldsymbol{x})\ln(P_\beta(\boldsymbol{x})) = \sum_{\boldsymbol{x}} P_\beta(\boldsymbol{x})\left(\beta E(\boldsymbol{x}) + g(\beta)\right) =$$

$$= \beta\langle E\rangle_\beta + g(\beta) = -\beta g'(\beta) + g(\beta)$$

som ger

$$h'(\beta) = -g'(\beta) - \beta g"(\beta) + g'(\beta) = -\beta g"(\beta) =$$

$$= -\beta v(\beta) = \beta\frac{d\langle E\rangle_\beta}{d\beta}.$$

Då man varierar β så får man

$$d\langle E\rangle_\beta = \frac{1}{\beta}dh$$

som man kan jämföra med den termodynamiska entropin $dE = TdS$ som ger kopplingen $1/\beta = T$ och $S = h$.

Det specifika värmet $C(T)$ dvs hur energin ändras då temperaturen ändras ges av

$$C(T) = \frac{d\langle E\rangle_T}{dT} = \frac{d\langle E\rangle_\beta}{d\beta} \cdot \frac{d\beta}{dT} = -\frac{1}{T^2}\frac{d\langle E\rangle}{d\beta} =$$

$$= \frac{1}{T^2} v(\beta) = \beta^2 v(\beta).$$

Alltså svarar intressanta storheter som t ex specifikt värme mot kvantiteter som kan beräknas ut P_β-fördelningen. Vid en viss temperatur $T = 1/\beta$ kan man få specifika värmet $C(T)$ som

$$C(T) = \beta^2 \mathrm{Var}(E) = \beta^2 v(\beta).$$

Kan vi alltså simulera den kanoniska fördelningen, dvs fördelningen för energin E kan vi ur dess varians bestämma specifika värmet för systemt. Fasövergångar i systemet karaktäriseras av att specifika värmet ändras dramatiskt – en stor del av en tillförd energi går åt i fasövergången, dvs utan att temperaturen ändras.