Laboration 2
Matematisk statistik för K, TMA074

1 Introduktion

Syftet med den här laborationen är att få en djupare förståelse för punktskattningar och konfidensintervall samt lära oss hur vi kan använda Matlab för att beräkna dessa storheter.

2 Stora talens lag

Ett exempel på en punktskattning är att använda stickprovsmedelvärdet $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ som skattning av väntevärdet i en fördelning. För ett stickprov av oberoende likafördelade slumpvariabler $X_1, \ldots, X_n$ med väntevärde μ och en ändlig varians σ^2 har vi tidigare visat att $\mathsf{E}(\bar{X}_n) = \mu$ och $\mathsf{V}(\bar{X}_n) = \sigma^2/n$. Enkelt sagt betyder det att stickprovsmedelvärdet kommer avvika från väntevärdet allt mindre då n växer. Vi kan precisera detta lite mer via den så kallade *stora talens lag* som säger

$$\mathsf{P}(|\bar{X}_n - \mu| > \varepsilon) \to 0, \quad \text{då } n \to \infty$$

för varje $\varepsilon > 0$. Ett sätt att illustrera detta är att kasta en tärning många gånger och se att de successiva medelvärdena konvergerar mot väntevärdet. Simulera först 100 tärningskast:

```
>> X = floor(6*rand(1,100)+1)
```

Funktionen `floor` avrundar nedåt och `rand` drar ett likformigt tal mellan noll och ett. Förvissa er om att elementen i X verkligen har en fördelning som tärningskast. Vi använde i förra laborationen funktionen `randi` för att simulera tärningskast och ett alternativt sätt är att använda den funktionen igen.

Vi vill nu beräkna alla successiva medelvärden av X, ett sätt att göra detta är

```
>> Xbar = cumsum(X)./(1:100)
```

Funktionen `cumsum` returnerar en vektor där element k är summan av de k första elementen X. Notationen ./ betyder elementvis division och $1:100$ skapar en vektor med elementen $1, 2, \ldots, 100$. Plotta `Xbar` och gör sedan om experimentet med 1000 tärningskast.

- vad verkar medelvärdet konvergera mot?
- Vad är det teoretiska väntevärdet?

3 Maximum likelihood

Vi ska nu titta närmare på maximum likelihood-metoden för parameterskattning.

Börja med att simulera data:

```
>> livslangd = exprnd(300,200,1);
```

Antag nu denna data är observationer av exponentialfördelade livslängder av någon viss komponent. Vi känner inte den förväntade livslängden hos komponenterna och vill därför uppskatta den baserat på datan, det vill säga vi vill skatta parametern β i exponentialfördelningen. Börja med att beräkna log-likelihoodfunktionen och plotta den som funktion av β för alla värden mellan $\beta = 50$ och $\beta = 500$:

```
>> beta = linspace(50,500,500);
>> L = -sum(livslangd)./beta  -length(livslangd)*log(beta);
>> plot(beta,L)
>> xlabel('beta'); ylabel('L')
```

Försäkra er om att ni förstår varför uttrycket för log-likelihooden ser ut som det gör! Kom ihåg att ML-skattaren β^* ges av det värde av beta som maximerar log-likelihoodfunktionen. Det kan vara svårt att se var maximat är så vi kan zooma i figuren för att se tydligare.

- Var ligger maximat?
- Vad är det teoretiska uttrycket för ML-skattaren β^*?
- Vilket värde får vi om vi sätter in datan vi har i uttrycket för ML-skattaren? Det vill säga, vad är skattningen av β baserat på den observerade datan.

4 Konfidensintervall

ML-skattaren β^* är en stokastisk variabel som beror av observationerna, och om vi skulle mäta 200 nya komponenter skulle skattningen bli annorlunda. För att illustrera detta tänker vi oss att vi gör 1000 upprepade försök där vi i varje försök mäter $n = 200$ komponenter. Sätt det sanna värdet på β till 300 och simulerar data från 1000 försök:

```
>> beta = 300;
>> X = exprnd(beta,200,1000);
```

Skatta nu β baserat på varje stickprov och plottar resultatet:

```
>> beta_est = mean(X);
>> plot(beta_est,'*')
>> hist(beta_est)
```

- Vilken typ av fördelning verkar skattningen av β följa? Varför är det så?

Beräkna nu ett konfidensintervall för skattningen. Ett konfidensintervall med approximativ konfidensgrad α ges av

$$[L, U] = \left[\beta^* - z_{\alpha/2}\frac{\beta^*}{\sqrt{n}},\ \beta^* + z_{\alpha/2}\frac{\beta^*}{\sqrt{n}}\right].$$

Kom ihåg att $z_{\alpha/2}$ är $\alpha/2$-kvantilen i den standardiserade normalfördelningen och att $\beta^*/\sqrt{n}$ är standardfelet för skattaren. Försäkra er om att ni förstår varför standardfelet ser ut så! Beräkna gränserna för varje stickprov som

```
>> alpha = 0.05;
>> L = beta_est - norminv(1-alpha/2)*beta_est/sqrt(200);
>> U = beta_est + norminv(1-alpha/2)*beta_est/sqrt(200);
```

Plotta de 100 första intervallen

```
>> figure; hold on
>> for(i = 1:100); plot([L(i) U(i)],[i i]); end
```

Eftersom vi har satt $\alpha = 0.05$ så bör ungefär 95% av intervallen täcka det sanna värdet på parametern. Vi kan beräkna täckningsgraden till exempel som

```
>>mean((L<beta).*(U>beta))
```

- Vad är täckningsgraden?

Upprepa nu simuleringen av konfidensintervallen för olika stickprovsstorlekar (dvs byt $n = 200$ mot andra värden på n) och undersök hur intervallen ändras för olika värden på n, testa både stora och små värden, till exempel $n = 5, 20, 200$ och 2000.

Detta är ett bra tillfälle att träna på att skriva egna funktioner för att underlätta analysen. Så om ni vill kan ni istället för att köra koden direkt för olika värden på n kopiera in koden ovan i en funktion som tar n som parameter och simulerar X och beräknar sakerna vi är intresserade av. Ni kan även låta α och antalet upprepningar av försöket vara parametrar till funktionen.

- Hur påverkas bredden av intervallen när vi ändrar n?
- Påverkas även täckningsgraden? Testa speciellt för små värden på n. Kan ni förklara vad som händer? Ett tips för att förstå vad som händer är att titta på fördelningen för skattningen av β genom att plotta histogram.

Antag att det sökta intervallet för livslängderna blev $I_\beta = (299.53, 382.95)$. Med 95% säkerhet kan vi alltså säga att intervallet innehåller den förväntade livslängden β. Observera att tolkningen av intervallet INTE är att sannolikheten är 0.95 att en komponents livslängd ligger i intervallet. Om vi vill uppskatta denna sannolikhet använder vi att $X =$ ”livslängd” är $\text{Exp}(\beta)$, beräknar $\mathsf{P}(299.53 < X < 382.95)$ och använder skattningen av β i beräkningen.

- Uppskatta sannolikheten att livslängden för en komponent ligger mellan 299.53 och 382.95 dagar.

5 Projektuppgift

Utför den andra delen på projektet.