Avd. Matematisk statistik

KTH Matematik

Tobias Rydén

2011-09-30

SF1905 Sannolikhetsteori och statistik: Lab 2 ht 2011

Förberedelser. Innan du går till laborationen, läs igenom den här handledningen. Repetera också i läroboken hur man kan göra konfidensintervall för väntevärdet i en population, och skillnaden i väntevärde mellan två olika populationer, när värdena i populationerna inte kan anses vara normalfördelade.

Temat för den här datorlaborationen är *bootstrap*, vilket innebär att uppskatta variationen i en skattning med hjälp av simulering.

Till att börja med formulerar vi själva skattningsproblemet. Antag att vi har data $x_1, x_2, \ldots, x_n$ som vi ser som observationer av oberoende stokastiska variabler $X_1, X_2, \ldots, X_n$ med gemensam fördelning (eller fördelningsfunktion) F. Denna fördelningsfunktion beror på någon parameter θ som vi vill skatta, och vi markerar det genom att skriva F_θ. Denna parameter kan vara t ex väntevärdet i en normalfördelning eller proportionen i en binomialfördelning.

För att skatta θ från data använder vi någon funktion h, dvs skattningen är $\theta^*_{\text{obs}} = h(x_1, x_2, \ldots, x_n)$. För att skatta ett väntevärde t ex använder vi i regel stickprovsmedelvärdet $h(x_1, x_2, \ldots, x_n) = n^{-1} \sum_1^n x_k$.

Skattningen θ^*_{obs} är en observation av *skattaren* θ^*, som är en stokastisk variabel. Vi kan representera θ^* som $\theta^* = h(X_1, X_2, \ldots, X_n)$. Den osäkerhet som finns förknippad med θ^*_{obs} ("hur bra är skattningen?") representeras på motsvarande sätt av variationen i θ^*. Det är därför det är så viktigt att försöka ta reda på vilken variation denna stokastiska variabel har, t ex dess varians och/eller dess fördelning.

I tanken skulle vi kunna studera variationen hos θ^* genom att simulera ett stickprov $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n$ från $X_1, X_2, \ldots, X_n$ och sedan bilda $\tilde{\theta} = h(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ vilket då är ett simulerat värde från θ^*. I praktiken går

det förstås inte, eftersom fördelningsfunktionen F för X_k, som vi alltså skall simulera från, inte är känd; den beror ju på en parameter θ som vi inte känner.

Bootstrap bygger på ovanstående tanke, men istället för att simulera från F, som vi inte känner, så simulerar vi från en approximation av denna fördelning. Denna approximation ges av det observerade stickprovet $x_1, x_2, \ldots, x_n$ självt. Med andra ord så skapar vi ett simulerat stickprov $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n$ genom att för $k = 1, 2, \ldots, n$ sätta $\tilde{x}_k$ lika med något av de observerade värdena $x_1, x_2, \ldots, x_n$ (med lika sannolikheter). På så sätt kan vi få ett värde $\tilde{\theta} = h(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ som *approximativt* är en simulering från θ^*.

Naturligtvis räcker det inte med en enda simulering av en stokastisk variabel, utan vi måste göra många. Proceduren ovan upprepas alltså säg M gånger, dvs vi simulerar M stycken bootstrapstickprov $\tilde{x}_1^{(m)}, \tilde{x}_2^{(m)}, \ldots, \tilde{x}_n^{(m)}$, $m = 1, 2, \ldots, M$ som ovan (oberoender av varandra) och bildar motsvarande bootstrappade skattningar $\tilde{\theta}^{(m)} = h(\tilde{x}_1^{(m)}, \tilde{x}_2^{(m)}, \ldots, \tilde{x}_n^{(m)})$.

1 Bootstrap av väntevärdesskattning för simulerade data

Till att börja med skall vi illustrera bootstrap-idén på en liten mängd data, och en enkel skattning – skattning av väntevärde.

Vi simulerar 20 värden från en s k dubbelexponentialfördelning (också kallad Laplacefördelning), som har täthetsfunktion $f(x) = (\theta/2)e^{-\theta|x|}$ för en parameter $\theta > 0$. Denna fördelning har väntevärdet 0, men vi antar att vi varken vet detta, eller att data faktiskt kommer från en sådan fördelning.

```
>> n=20
>> x=exprnd(1,n,1).*sign(rand(n,1)-1/2)
```

Här resulterar `sign(rand(n,1)-1/2)` i en $n \times 1$-vektor med slumpvis valda tecken ± 1 (varför?).

För att konstruera ett bootstrapstickprov kan vi först slumpa vilka index i vi skall välja för de observationer x_i som skall ingå i bootstrapstickprovet, och sedan plocka fram motsvarande x-värden:

```
>> idx=randsample(n,n,true)
>> xboot=x(idx)
```

Tänk efter varför det fungerar! Notera att det finns observationer x_k som återfinns mer än en gång i bootstrapstickprovet.

Om vi nu tar θ som väntevärdet av den fördelning vi har observationer från, så är vår skattning förstås $\theta^*_{\text{obs}} = \overline{x}$. Vi kan tillämpa medelvärdesbildningen på $M = 1000$ bootstrapstickprov för att skapa lika många bootstrapreplikat av skattningen.

```
>> M=1000, thetaboot=zeros(M,1);
>> for i=1:M, thetaboot(i)=mean(x(randsample(n,n,true)));end
```

Här hoppar vi över variablerna `idx` och `xboot` ovan, som användes för mellanlagringar.

Nu har vi alltså en vektor `thetaboot` med bootstrapreplikat av θ^*, och genom att titta på variationen hos dessa simulerade värden så kan vi dra slutsatser om variationen hos θ^*. Skriv

```
>> hist(thetaboot,50)
```

för att rita ett histogram över bootstrapreplikaten (parametern 50 är antalet klasser i histogrammet). Är variationen stor eller liten?

Skriv

```
>> std(thetaboot)
```

för att beräkna stickprovsstandardavvikelsen för bootstrapreplikaten. Detta är alltså en uppskattning av standardavvikelsen hos θ^*. Utan att använda bootstrap, hur skulle du då uppskatta denna standardavvikelse, dvs vilket medelfel för skattningen θ^*_{obs} skulle du använda? Räkna ut detta medelfel för de data du har och jämför med uppskattningen du får via bootstrap. Observera att bootstrapberäkningen inte baseras på *några analytiska beräkningar*, utan *bara på simuleringar*!

När vi gör konfidensintervall utgår vi ofta från en skattare som, ofta approximativt, är normalfördelad. Gäller det i det här fallet? Det kan du undersöka genom att använda funktionen `normplot` i Matlab.

```
>> normplot(thetaboot)
```

Hur skulle du göra ett konfidensintervall för väntevärdet baserat på antagandet att θ^* är approximativt normalfördelad? Beräkna ett sådant konfidensintervall!

Ett alternativ är att ta fram gränser för konfidensintervallet som kvantiler ur den simulerade bootstrapfördelningen, dvs som empiriska kvantiler från bootstrapreplikaten $\tilde{\theta}^{(m)}$:

```
>> quantile(thetaboot,[.025 .975])
```

ger ett intervall med (approximativt) 95% konfidensgrad, och där vi lagt 2.5% av felrisken i vänstra respektive högra svansen av fördelningen. Hur överensstämmer det med intervallet du beräknade ovan? Observera att, återigen, bootstrapmetoden *bara använder simuleringar*, och inget antagande om approximativ normalfördelning.

2 Bootstrap av väntevärdesskattning för födelsevikter

I filen `birth.dat`, som finns på kurshemsidan, finns data för 747 nyfödda barn i Malmö kommun. Ladda ned filen och läs in den i Matlab. Du får då en matris `birth` i Matlabs arbetsminne. Filen `birth.txt` innehåller information om vad som finns i matrisen. T ex innehåller kolonn 3 alla födelsevikter, som vi skall arbeta med. Skriv

```
>> x=birth(:,3);
```

för att lägga värdena i den kolonnen i vektorn `x`.

Skatta sedan den förväntade födelsevikten, dvs väntevärdet av den fördelning som vi tänker oss att vikterna kommer ifrån, som i föregående avsnitt. Jämför också med vad du får med de metoder som finns i läroboken!

För att underlätta arbetet kan du använda en funktion `bootstrp` som finns i Matlab, och som gör hela arbetet med att simulera bootstrapreplikat $\tilde{\theta}^{(m)}$.

```
>> thetaboot = bootstrp(M,@mean,x);
```

3 Bootstrap av skattning av skillnad väntevärden för födelsevikter

Vi skall nu går vidare till att studera skillnaden mellan väntevärdena i två populationer, t ex skillnaden i födelsevikt för barn vars mammor röker respektive inte röker under graviditeten. (Om ni vill kan ni ta två andra populationer, och/eller andra variabler att studera!).

I filen `birth.txt` ser man att kolonn 20 i verb+birth+ innehåller rökvanor, och att värdena 1 och 2 betyder att mamman inte röker under graviditeten, medan värdet 3 betyder att hon gör det. Ni kan skapa två variabler `x`

och `y` för födelsevikter hörande till icke-rökande respektive rökande mammor enligt

```
>> x=birth(birth(:,20)<3,3);
>> y=birth(birth(:,20)==3,3);
```

Vad som händer här är att `birth(:,20)<3` returnerar en vektor av "sant" och "falskt", och att bara de rader av kolonn 3 (födelsevikterna) i `birth` för vilka jämförelsen är sann, väljs ut. Använd funktionen `length` eller kommandot `whos` för att se storleken på vektorerna `x` och `y`.

För att skatta skillnaden mellan populationernas väntevärden, använder vi som vanligt skillnaden mellan stickprovsmedelvärdena, `mean(x)-mean(y)`. För att undersöka osäkerheten i denna skattningen kan vi använda bootstrap, och simulera bootstrapreplikat enligt

```
>> thetaboot = bootstrp(M,@mean,x)-bootstrp(M,@mean,y);
```

Ser bootstrapreplikaten ut att komma från en normalfördelning? Vad får du för konfidensintervall för skilladen θ mellan väntevärdena? Vad får du med den metod i boken som du skulle använda?

4 Bootstrap av kvantilskattning

Det kan vara intresse att studera en kvantil för fördelningen av födelsevikterna, t ex för att få ett mått på vad som är en ovanligt låg födelsevikt (vilket i sin tur kan ha medicinska implikationer). Arbeta åter med alla födelsevikter, dvs

```
>> x=birth(:,3);
```

Med bokens beteckningar är 95%-kvantilen den punkt som har 5% av fördelningens sannolikhetsmassa till *vänster* om sig. Denna punkt benämns som 5%-kvantilen i Matlab (och de flesta andra läroböcker!). Du kan beräkna den empiriska kvantilen som `quantile(x,.05)`. Detta är alltså den punkt som har 5% av värdena i `x` till vänster om sig.

Hur osäker är skattningen? Använd bootstrap för att undersöka det! Du kan skapa bootstrapreplikat enligt

```
>> thetaboot = bootstrp(M,@(z) quantile(z,.05),x);
```

Undersök bootstrapreplikaten och gör ett konfidensintervall för 5%-kvantilen hos födelseviktsfördelningen. Observera att du nu analyserat (med bootstrap) ett statistiskt problem (konfidensintervall för kvantilskattning) som inte går att lösa med metoder som finns i läroboken!