Matematisk Statistik

SF1910 Tillämpad statistik, HT 2016
Laboration 1 för CSAMHS, CLGYM-TEMI

Introduktion

Detta är handledningen till Laboration 1, ta med en en utskriven kopia av den till laborationen. Läs handledningen två gånger. Försäkra dig om att du förstår hur de MATLAB-kommandon som finns i den bifogade koden fungerar. Laborationen bedöms som godkänd eller ej godkänd. För att få deltaga i laborationen skall svar på förberedelseuppgifter kunna redovisas **individuellt**. Arbete i grupp är tillåtet (och uppmuntras) med **högst två** personer per grupp. Godkänd laboration ger 3 poäng på ordinarie tentamenstillfälle.

Förberedelseuppgifter

1. Definiera begreppen sannolikhet, fördelningsfunktion, och täthetsfunktion. Skriv upp sambandet mellan dem.

2. Den stokastiska variabeln X har täthetsfunktion

$$f_X(x) = \lambda e^{-\frac{x}{\lambda}} + \frac{\lambda}{x}, \quad x \in [1,\, 10] \tag{1}$$

   för ett specifikt λ.

   (a) Bestäm (med ett närmevärde) λ så att f_X blir en täthetsfunktion.

   Svar: .....

   (b) Bestäm fördelningsfunktionen för X.

   Svar: .....

   (c) Bestäm sannolikheten att X är mindre än 7. Använd $\lambda = 0.4267$.

   Svar: .....

   (d) Bestäm $E[X]$.

   Svar: .....

3. Låt U vara likformigt fördelad över intervallet $[0, 2\pi]$, beräkna

   (a) $E[\cos(U)]$

   Svar: .....

   (b) $E[\sin(U)^2]$

   Svar: .....

4. Redogör för Stora talens lag.

5. Redogör för Centrala gränsvärdessatsen.

6. Läs sid. 272-273 om bootstrap i läroboken.

Syfte och vidare introduktion

Börja laborationen med att ladda ner följande filer från kurshemsidan.

- `plot_mvnpdf.m`
- `hist_density.m`
- `birth.dat`
- `birth.txt` - beskrivning av datat `birth.dat`

Se till att de ligger i den mapp du kommer att arbeta i. För att kontrollera att du har lagt filerna rätt, skriv `ls` och se om filerna ovan listas.
Du kan skriva dina kommandon direkt i MATLAB-prompten men det är absolut att föredra att arbeta i editorn. Om den inte är öppen så kan du öppna den och skapa ett nytt dokument genom att skriva `edit lab1.m`. Koden som ges nedan är skriven i celler. En ny cell påbörjas genom att skriva två procenttecken. `Ctrl+Enter` exkeverar innehållet i en cell.
Temat för den här datorlaborationen är simulering. Sannolikhetsteoridelen av kursen handlar om hur man genom beräkningar kan ta fram olika storheter som sannolikheter, väntevärden osv, för en given stokastisk modellen. För mer komplicerade system är det ibland inte alls möjligt att göra exakta beräkningar, eller så är de så tidskrävande att man avstår. I sådana sammanhang kan simulering vara ett alternativ. Simulering innebär att man med hjälp av en dator drar ett antal replikeringar av det stokastiska systemet, och sedan använder t ex medelvärden eller empiriska kvantiler (mer om det nedan) för att uppskatta de storheter man söker. I den här laborationen skall vi göra detta för några enkla problem, men grundprinciperna går att använda på långt mer komplicerade problem som vi inte kan lösa med enkla beräkningar.

Problem 0 - Beräkna sannolikheter

Läs help för funktionerna `binocdf`, `binopdf`, `normcdf`, `normpdf`, `expcdf` och `exppdf`. Observera att MATLABs `exprnd` har väntevärdet μ som parameter i motsats till [2] som har $1/\mu$ som parameter.
Låt X_1 vara Bin$(10, 0.3)$, $X_2 \in N(5, 3)$, $X_3 \in Exp(7)$ och bestäm (med hjälp av funktionerna ovan) för $k = 1, 2, 3$

1. $P(X_k \leq 3)$

   Svar: .....

2. $P(X_k > 7)$

   Svar: .....

3. $P(3 < X_k \leq 4)$

   Svar: .....

Problem 1 - Täthetsfunktioner

Plotta täthetsfunktionen för en exponentialfördelad stokastisk variabel med väntevärde μ.

```
%% Problem 1: exp-pdf
    dx = 0.1;
    x = 0:dx:15;        % Skapar en vektor med dx som inkrement
    mu = 1;
    y = exppdf(x, mu);  % exponential-fordelningen
    plot(x,y)
```

Gör nu samma sak för täthetsfunktionen i förberedelseuppgift 2.

```
%% Problem 1: lambda-plot
    lambda = 0.4267;
    f=(lambda*exp(-x/lambda)+lambda./x).*(x > = 1 & x < = 10);
    plot(x, f)
```

Diskutera skillnaden mellan fördelningarna.

Kommentar: .....

.....

Problem 2 - Multivariat Normalfördelning

Täthetsfunktionen för den multivariata normalfördelningen ritas upp av funktionen `plot_mvnpdf`. Undersök hur funktionen fungerar och testa med lite olika parametervärden. Parametrarna `mux` och `muy` kan anta alla reella värden, parametrarna `sigmax` och `sigmay` kan anta alla positiva värden och parametern `rho` kan anta alla värden på intervallet $[-1, 1]$. Observera att plotfönstret i funktionen `plot_mvnpdf` är fixt, så för parametervärden som är av storleksordningen tio eller större, så kommer merparten av täthetsfunktionen att hamna utanför plotfönstret.

```
%% Problem 2: Multivariat normal
    mux = 0; muy = -2; sigmax = 1; sigmay = 4; rho = 0.7;
    plot_mvnpdf(mux, muy, sigmax, sigmay, rho)
```

Hur påverkar olika parametervärden utseendet på plotten?

Kommentar: .....

.....

Problem 3 - Simulering av slumptal

I denna övning så genererar vi ett stort antal slumptal, ritar upp deras histogram och plottar slutligen den sanna täthetsfunktionen ovanpå .

```
%% Problem 3: Simulering av slumptal
    mu = 10;
    N = 1e4;
    y = exprnd(mu, N, 1);   % Genererar N exp-slumptal
    hist_density(y);    % Skapar ett normaliserat histogram
    t = linspace(0, 100, N/10); % Vektor med N/10 punkter
    hold on
    plot(t, exppdf(t, mu), 'r') % 'r' betyder rod linje
    hold off
```

Upprepa simuleringarna och studera hur histogrammet förändras. Hur förhåller sig histogrammet till den röda linjen och hur förklaras variationen kring denna linje?

Kommentar: .....

.....

Problem 4 - Stora talens lag, Monte Carlo och CGS

I detta avsnitt kommer vi igen att generera stora mängder slumptal, nu för att beräkna väntevärden och sannolikheter. Antag att man vill veta väntevärdet på en tärning, det är inte svårt att beräkna för hand men det går också att kasta tärningen många gånger och sedan räkna ut medelvärdet på dessa kast. Om $X_1, X_2, \ldots, X_n$ är likafördelade med väntevärde μ så gäller enligt stora talens lag i [2] att

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| < \varepsilon\right) \to 1 \tag{2}$$

för varje $\varepsilon > 0$ när $n \to \infty$. Sannolikheten att skillnaden mellan medelvärdet och det sanna väntevärdet är mindre än ε går alltså mot ett när antalet observationer går mot oändligheten. Att använda detta för att beräkna väntevärden kallas för Monte Carlo-metoder.
Idén bakom Monte Carlo är gammal och har funnits inom matematiken åtminstone sedan 1700-talet, men synen kom att förändras under andra halvan av 1900-talet då det på allvar blev möjligt att utföra stora beräkningar. Under 1940-talet utvecklade Stanislaw Ulam och John von Neumann metoder för att göra dessa "tärningskast" med hjälp av dator enligt [1]. Arbetet var kopplat till Manhattanprojektet vars syfte var att ta fram den första atombomben. Metoden namngavs efter casinot Monte Carlo i Monaco.

Illustration av Stora talens lag

Koden nedan simulerar exponentialfördelade stokastiska variabler och plottar medelvärdet efter hand. Plotten som visas är alltså din skattning av medelvärdet upp till och med variabel nummer `k`. Om du tröttnar på att vänta, så kan du kommentera bort `pause`-raden.

```
%% Problem 4: Stora talens lag
    mu = 0.5;
    M = 500;
    X = exprnd(mu, M, 1);
    plot(ones(M, 1)*mu, 'r-.')
    hold on
    for k = 1:M
        plot(k, mean(X(1:k)), '.')
        if k == 1
            legend('Sant \mu', 'Skattning av \mu')
        end
        xlabel(num2str(k)), pause(0.001)
    end
    hold off
```

Ser det ut som förväntat?

Svar: .....

Illustration av Centrala Gränsvärdessatsen

Koden nedan simulerar exponentialfördelade slumptal och summerar sedan dessa. Studera koden och förklara vad `N` representerar.

Svar: .....

```
%% Problem 4: CGS
    M = 1e3;
    N = 4;
    mu = 5;
    X = exprnd(mu, M, N);
    S = cumsum(X, 2);
    for k = 1:N
        hist(S(:, k), 30)
        xlabel(num2str(k))
        pause(0.1)
    end
```

Justera `N`, vad händer när du ökar respektive minskar värdet? Varför?

Kommentar: .....

.....

Vid vilket `N` ser det ut som att det inte gör någon skillnad att öka N?

Svar: .....

Vilken fördelning verkar summorna ha? Varför har de denna fördelning?

Svar: .....

Väntevärden

Uppgiften är nu att beräkna samma väntevärden som i förberedelseuppgift 3. Det första väntevärdet kan du beräkna enligt nedan.

```
%% Problem 4: Monte Carlo
    N = 1e5;
    U = rand(N, 1)*2*pi;
    mean(sin(U).^2);
```

Kontrollera att resultatet stämmer med vad du förväntade dig i båda fallen. Låt nu X och Y vara oberoende stokastiska variabler där $X \in Exp(4)$ och $Y \in N(0,1)$. Använd Monte Carlo-metoden för att beräkna $E[(e^X)^{\cos(Y)}]$.

Svar: .....

Problem 5 - Deskriptiv statistik

Vi skall nu gå vidare till att studera skillnaden mellan väntevärden 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 `birth.txt` 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 koden nedan för att visuellt inspektera datat.

```
%% Problem 5: Deskriptiv statistik
    load lab2data/birth.dat
    x = birth(birth(:, 20) < 3, 3);
    y = birth(birth(:, 20) == 3, 3);
    subplot(2,2,1)
    boxplot(x)
    axis([0 2 500 5000])
    subplot(2,2,2)
    boxplot(y)
    axis([0 2 500 5000])
    subplot(2,2,3:4)
    ksdensity(x)
    hold on
    [fy, ty] = ksdensity(y);
    plot(ty, fy, 'r')
    hold off
```

Vad betyder plotarna? Vilka slutsatser kan ni dra?

Svar: .....

Problem 6 - Bootstrap (gör i mån av tid)

Namnet bootstrap syftar till metaforen att dra sig upp ur ett knivig situation genom att ta tag i sina stövelskaft. Ett klassiskt exempel är historien om Baron von Münchausen i vilken han ska ha räddat sig och sin häst ur ett träsk genom att dra ur de båda genom att lyfta sig själv i håret. Detta förfarande beskriver idén bakom den statistiska varianten av metoden mycket väl: Man har observerat en begränsad mängd data och man vill bilda sig en uppfattning om vad som hade hänt om man hade haft fler observationer. I vårt fall innebeär det att vi gör upprepade slumpvisa dragningar från vårt observerade data med funktionen `randsample`.
Vi ska börja med att titta på ett simulerat exempel där vi vet vad som *borde* hända och fortsätter sedan i nästa laboration med att titta på riktiga data, vilket är det som vi huvudsakligen är intresserade av.

Simulering av summerade exponentialfördelade variabler

Om $X_1, X_2, \ldots, X_n$ alla är oberoende och exponentialfördelade med intensitet λ så gäller

$$Y_n = \sum_{k=1}^{n} X_k \in \Gamma(n, \frac{1}{\lambda}). \tag{3}$$

Med andra ord är summan av de exponentialfördelade variablerna gammafördelad. (Specialfallet när Γ-fördelningen har ett positivt heltal n som första parameter kallas mer specifikt för *Erlang*-fördelning.)
Vi ska nu simulera M stycken summor av typen (3), med n noterat som `n_sum` i koden nedan, och studera ett histogram för dem. Börja med att studera koden nedan. Läs `help randsample` och testa hur funktionen fungerar. Kommandot `reshape` används nedan för att omforma matrisen `x` till en vektor så att den ska kunna användas i `randsample`. Notera att vi använder väntevärdet $\mu = \frac{1}{\lambda}$ i koden nedan.

```
%% Problem 6: Bootstrap - Simulering
    % lambda = 1/5;
    mu = 5;
    M = 1e3;
    n_sum = 5;
    X = exprnd(mu, M, n_sum);
    g = sum(X, 2);
    subplot(211)
    hist_density(g)
    hold on
    t = 0:0.01:mu*10;
    plot(t, gampdf(t, n_sum, mu), 'r')
    hold off
    B = 1e3;    % Antal bootstrapreplikat
    totalNoSamples = M*n_sum;
    X = reshape(X, totalNoSamples, 1); % Gor om X till vektor
    yBoot = zeros(B, 1);
    for j = 1:B
        sampleDraws = X(randsample(totalNoSamples, n_sum, 1));
        yBoot(j) = sum(sampleDraws);
    end
    subplot(212)
    hist_density(yBoot)
    hold on
    plot(t, gampdf(t, n_sum, mu), 'r')
    hold off
```

Förklara vad de två raderna i for-loopen gör!

Kommentar: .....

.....

Justera `B`, `M`, respektive `mu`. Vad händer? Varför?

Kommentar: .....

.....

Referenser

[1] Eckhardt, Roger (1987) Stan Ulam, John von Neumann and the Monte Carlo, Method *Los Alamos Sci.*, Vol **15**, p. 131-43.

[2] Blom, G., Enger, J., Englund, G., Grandell, J., och Holst, L., (2005). Sannolikhetsteori och statistikteori med tillämpningar.