Matematisk Statistik

SF1923/SF1924 Sannolikhetsteori och statistik, HT2021
Laboration 1

1 Introduktion

Denna demonstration är inte poänggivande, men utgör en förberedelse för den andra laborationen på kursen, vilken kan ge bonuspoäng på tentamen. Syftet med denna laboration är dels att ge en introduktion till att arbeta med statistik i MATLAB, dels att ge djupare förståelse för några viktiga begrepp i kursen genom att illustrera dem med hjälp av MATLAB.

Denna laboration kan utföras antingen på egen hand eller vid det schemalagda laborationstillfället. Om du väljer det senare alternativet, så kan det vara bra att ha läst igenom specifikationen två gånger och ha börjat försöka lösa uppgifterna före laborationstillfället. Fokusera när du gör labben på att lära dig att använda MATLAB och se till att du förstår de kommandon som du använder.

2 MATLAB

I båda laborationerna på den här kursen används MATLAB som är ett interaktivt program för numeriska beräkningar. Det är också ett programmeringsspråk. MATLAB finns på de flesta datorer på KTH och till fördelarna med programmet hör att det ser i stort likadant ut oberoende av på vilken sorts dator man kör det. Om du vill ha mer att läsa om MATLAB så finns det olika handledningar att ladda ned eller köpa.

Börja med att logga in på ditt vanliga konto. Starta sedan MATLAB genom att klicka på ikonen. Programmet avslutas med kommandot `exit`. Till att börja med kan man tänka på MATLAB som en avancerad räknedosa. Man skriver in vad man vill ha gjort och MATLAB svarar.

```
>> 3*11.5 + 2.3^2/4

ans =

      35.8225
```

Variabler tilldelas värden med tecknet = och finns sedan kvar i minnet. Pröva att tilldela några variabler värden.

```
>> a = 1;
>> b = sqrt(36);
```

```
>> width = 3.89;
>> who

Your variables are:

a         b         width
```

Kommandot `who` visar alltså vilka variabler som finns i minnet. En variabel, t.ex. `b`, kan raderas ur minnet med `clear b`. Läggs ett semikolon, `;`, till efter en kommandorad, skrivs resultatet inte ut på skärmen. MATLAB kan också hantera vektorer och matriser, och de hanteras precis lika enkelt som ovan.

```
>> x = [1 3 7]

x =

     1 3 7

>> y = [2 1 8]'

y =

      2
      1
      8

>> z = [1 2 ; 3 4]'

z =

      1   3
      2   4

>> w = rand(1, 4)

w =

     [0.2190    0.0470    0.678    0.6793]
```

Tecknet `'` betyder som synes transponat och semikolon används för att skilja rader åt i matriser. Funktionen `rand(m, n)` ger en $m \times n$-matris med slumptal som är likformigt fördelade mellan 0 och 1. Notationen för algebraiska uttryck är den vanliga, men kom ihåg att multiplikationstecknet `*` tolkas som matrismultiplikation. Termvis multiplikation av tvåmatriser `A` och `B` med samma dimensioner skrivs istället `A.*B`. I MATLAB finns alla vanliga funktioner inbyggda, t.ex.

```
exp log sin asin cos acos tan atan
```

Observera att `log` är den naturliga logaritmen. Pröva att plotta en funktion, t.ex. genom följande kommandon.

```
>> x = 0.5:0.1:2
>> help log
>> y = log(x)
>> plot(x,y)
```

Den första raden tilldelar `x` en vektor som löper från 0.5 till 2 i steg om 0.1. Hjälpfunktionen `help` ger information om en funktion. Skriver du bara `help` visas en lista med tillgängliga funktioner, sorterade efter funktionspaket (ett sådant paket kallas en *toolbox*).

I MATLAB finns det en stor mängd funktioner som har att göra med sannolikhetsteori och statistik. Se `help stats`. Titta speciellt under rubriken *Random Number Generators*, som kommer i början på den långa listan av funktioner.

Till denna laboration behöver du, utöver de funktioner som redan finns i MATLAB, dessutom följande två filer som du kan ladda ner från kurshemsidan.

- `plot_mvnpdf.m`
- `hist_density.m`

Se till att filerna ligger i den mapp som 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` exekverar innehållet i en cell.

3 Simulering

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 modell. För mer komplicerade system är det ibland inte alls möjligt att göra exakta beräkningar, eller så är det så tidskrävande att man avstår.

I sådana sammanhang kan simulering vara ett alternativ. Simulering innebär att man med hjälp av en dator simulerar 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 enklare problem men grundprinciperna går att använda på långt mer komplicerade problem som vi inte kan lösa med enkla beräkningar.

I det allra enklaste fallet kan det vara fråga om att uppskatta väntevärdet för en fördelning. Antag att vi har en fördelningsfunktion F och låt X vara en stokastisk variabel med denna fördelningsfunktion. Antag också att vi vill uppskatta tillhörande väntevärde, μ säg. Om vi nu drar observationer $x_1, x_2, \ldots, x_n$ från X kan vi uppskatta μ med hjälp av

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k. \tag{1}$$

Att detta är en rimlig uppskattning följer av stora talens lag. Om vi vill uppskatta fördelningsfunktionen $F(a) = \mathbb{P}(X \leq a)$ för något tal a, så kan vi göra detta genom att räkna ut hur stor andel av de n simulerade observationerna som är $\leq a$. Vi kan skriva detta som

$$\hat{F}(a) = \frac{\text{antal } x_k \text{ som är} \leq a}{n} = \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}(x_k \leq a). \tag{2}$$

Här är $\mathbb{I}$ en s.k. *indikatorfunktion*, som antar värdet 1 om villkoret inom parentes är uppfyllt och annars antar värdet 0. Alltså är $\mathbb{I}(x_k \leq a)$ lika med 1 precis för de k sådana att $x_k \leq a$, så att summan ovan räknar antalet index k som uppfyller villkoret.

4 Laborationsuppgifter

Problem 1 - Simulering av exponentialfördelade stokastiska variabler

MATLABs funktion för att simulera exponentialfördelade stokastiska variabler, dvs för att skapa exponentialfördelade slumptal, heter `exprnd`. Använd gärna MATLABs hjälpkommando `help` för att ta reda på precis hur funktionen `exprnd` fungerar, dvs. skriv `help exprnd`. Observera att MATLABs `exprnd` har väntevärdet μ som parameter i motsats till [2] som har $1/\mu$ som parameter. Funktionen kan också användas för att simulera vektorer (eller matriser) av oberoende exponentialfördelade variabler. Följande kod genererar N stycken Exp(1/10)-fördelade slumptal, ritar upp ett histogram av slumptalen samt plottar den sanna täthetsfunktionen för Exp(1/10) ovanpå histogrammet som jämförelse.

```
%% Problem 1: Simulering av exponentialfordelade 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
```

Hur förhåller sig histogrammet till den röda linjen och hur förklaras variationen kring denna linje?

Problem 2 - Stora talens lag

Stora talens lag säger att för oberoende, likafördelade stokastiska variabler $X_1, X_2, ...$, så konvergerar det aritmetiska medelvärdet mot väntevärdet när antalet termer i medelvärdet går mot oändligheten. Vi ska nu undersöka denna konvergens genom att simulera de stokastiska variablerna X_i (som vi här låter vara exponentialfördelade) och studera beteendet hos medelvärdet $S_n = (X_1 + \cdots + X_n)/n$.

```
%% Problem 2: 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
```

Punkten med x-värde `k` är medelvärdet av `k` exponentialfördelade stokastiska variabler. Ser det ut som förväntat?

Problem 3 - Monte Carlo-skattning av väntevärde

Antag att vi nu inte vet att exponentialfördelningen som vi simulerade i Problem 1 har väntevärdet just 10 och att vi vill uppskatta detta från våra simulerade data. Det kan vi göra genom att beräkna medelvärdet av slumptalen.

```
%% Problem 3: Vantevarde av exp.fordelad stokastisk variabel
    N = 1e4;
    y = exprnd(mu, N, 1);
    mean(y);
```

Hur bra blir din uppskattning? Prova att göra om simuleringen och medelvärdesberäkningen några gånger. Prova också olika värden på `N`.

Att använda medelvärdet av simulerade slumptal för att beräkna väntevärden kallas för Monte Carlo-metoder. Idén bakom Monte Carlo-metoder har funnits inom matematiken åtminstone sedan 1700-talet, men kom till praktisk användning först under andra halvan av 1900-talet då det blev möjligt att utföra stora beräkningar med dator. De första Monte Carlo-simuleringarna utfördes under 1940-talet av Stanislaw Ulam och John von Neumann [1] i samband med Manhattanprojektet vars syfte var att ta fram den första atombomben. Metoden namngavs efter casinot Monte Carlo i Monaco.

Problem 4 - Monte Carlo-simulering av talet π

Fördelen med numerisk simulering av väntevärden är att de kan användas även för väntevärden som är svåra att beräkna exakt. Vi ska nu använda Monte Carlo-metoder för att bestämma ett approximativt värde på talet π. Låt U och V vara två oberoende stokastiska variabler som är likformigt fördelade på $[-1, 1]$. Paret (U, V) antar värden i $[-1, 1] \times [-1, 1]$ och kan ses som punkter i en kvadrat i planet. Sannolikheten att punkten (U, V) hamnar i enhetscirkeln är

$$P(\sqrt{U^2 + V^2} \leq 1) = \frac{\text{arean av enhetscirkeln}}{\text{arean av kvadraten } [-1, 1] \times [-1, 1]} = \frac{\pi}{4}.$$

Vi kan skatta π på följande sätt. Vi simulerar först ett stort antal punkter $(U_1, V_1), (U_2, V_2), ..., (U_N, V_N)$. För varje punkt (U_i, V_i) kontrollerar vi om $\sqrt{U_i^2 + V_i^2} \leq 1$ och beräknar andelen punkter som hamnat i enhetscirkeln. Eftersom

$$\frac{\text{Antal punkter som hamnat i enhetscirkeln}}{N} \to P(\sqrt{U^2 + V^2} \leq 1) = \frac{\pi}{4},$$

då $N \to \infty$, så gäller det för stora värden på N att

$$\pi \approx \frac{4 \cdot \text{Antal punkter som hamnar i enhetscirkeln}}{N}.$$

Följande kod genererar N punkter (U_i, V_i), plottar dem i planet samt beräknar motsvarande skattning av värdet på π. Kör koden flera gånger och variera `N`.

```
1  %% Problem 4: Monte Carlo-skattning av talet pi
2      N = 1e2;
3      U = 2*rand(1,N)-1;  % Genererar U(-1,1)-ford. slumptal
4      V = 2*rand(1,N)-1;
5      plot(U,V,'o'), hold on  % Plottar de genererade punkterna
6      X = -1:0.01:1;
7      plot(X,sqrt(1-X.^2),'r')     % Plottar enhetscirkeln
8      plot(X,-sqrt(1-X.^2),'r')
9      Z = (sqrt(U.^2+V.^2)≤1);     % Beraknar narmevarde pa pi
10     pi = 4*mean(Z);
```

Notera att syntaxen `(x>5)` i MATLAB ger en vektor av samma storlek som `x`, men där ett element i vektorn är 1 eller 0 beroende på om motsvarande element i `x` uppfyller villkoret `> 5` eller ej. Denna syntax används på rad 9 i koden ovan.

Problem 5 - Beräkning av sannolikheter

MATLAB har kommandon för de vanligaste sannolikhetsfördelningarna. Läs `help` för funktionerna `binocdf`, `binopdf`, `normcdf`, `normpdf`, `expcdf` och `exppdf`. Notera att `expcdf` och `exppdf`, precis som `exprnd`, har väntevärdet μ 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. $P(X_k \leq 3)$
2. $P(X_k > 7)$
3. $P(3 < X_k \leq 4)$

Problem 6 - Visualisering av sannolikhetsfördelningar

MATLAB:s kommandon kan även användas för att visualisera de vanligaste sannolikhetsfördelningarna. Värdet i x för täthetsfunktionen för normalfördelningen $N(\mu, \sigma)$ ges exempelvis av kommandot `normpdf(x,mu,sigma)`. Följande kod genererar grafen av täthetsfunktionen för den standardiserade normalfördelningen $N(0, 1)$.

```
1  %% Problem 6: Tathetsfunktion for normalfordelning
2      dx = 0.01;
3      x = -10:dx:10;      % Skapar en vektor med dx som inkrement
4      y = normpdf(x,0,1);
5      plot(x,y)
```

Prova även att plotta täthetsfunktionen till normalfördelningen för några andra värden på parametrarna μ och σ, exempelvis $\mu = -1$, $\sigma = 0.1$ respektive $\mu = 2$, $\sigma = 2$.

För gammafördelningen med parametrar a och b ges täthetsfunktionen av

$$f_X(x) = \frac{1}{b^a \Gamma(a)} x^{a-1} e^{-x/b},$$

(observera att [2] använder en annan definition av parametrarna i gammafördelningen). Följande kod kan användas för att plotta denna täthetsfunktion.

```
%% Problem 6: Tathetsfunktion for gammafordelning
    dx = 0.01;
    x = 0:dx:10;        % Skapar en vektor med dx som inkrement
    y = gampdf(x,1,2);
    plot(x,y), hold on
    z = gampdf(x,5,1);
    plot(x,z,'r')
```

Även för fördelningfunktionerna finns kommandon för de vanligaste sannolikhetsfördelningarna. För gammafördelningen gäller exempelvis

```
%% Problem 6: Fordelningsfunktion for gammafordelning
    dx = 0.01;
    x = 0:dx:10;        % Skapar en vektor med dx som inkrement
    y = gamcdf(x,1,2);
    plot(x,y), hold on
    z = gamcdf(x,5,1);
    plot(x,z,'r')
```

Problem 7 - Multivariat normalfördelning

Täthetsfunktionen för den multivariata normalfördelningen ritas upp av funktionen `plot_mvnpdf`. Vi undersöker hur funktionen fungerar och testar med några 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 7: Multivariat normalfordelning
    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? Vad motsvarar de olika parametrarna?

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.