

Demonstration av laboration 2, SF1901

Thomas Önskog

KTH

29 November 2017

Målet med dagens föreläsning är att repetera några viktiga begrepp från kursen och illustrera dem med hjälp av MATLAB. Laboration 2 har följande delar

- Fördelningsfunktion och täthetsfunktion
- Multivariat normalfördelning
- Simulering av slumpetal
- Stora talens lag
- Monte Carlo-simulering av väntevärden
- Monte Carlo-simulering av talet π
- Centrala gränsvärdessatsen
- Simulering av konfidensintervall

Fördelningsfunktion och täthetsfunktion

Täthetsfunktionen f_X för en kontinuerlig stokastisk variabel X definieras av

$$P(X \in [a, b]) = \int_a^b f_X(x) dx,$$

och fördelningsfunktionen F_X ges av

$$F_X(x) = P(X \leq x) = \int_{-\infty}^x f_X(y) dy.$$

Fördelningsfunktion och täthetsfunktion

Normalfördelning

MATLAB har kommandon för de vanligaste sannolikhetsfördelningarna. För normalfördelningen $N(\mu, \sigma)$ ges exempelvis täthetsfunktionens värde i x ,

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

av kommandot `normpdf(x,mu,sigma)`. Följande kod genererar grafen av täthetsfunktionen för den standardiserade normalfördelningen $N(0, 1)$.

```
1  %% 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)
```

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

Fördelningsfunktion och täthetsfunktion

Gammafördelning

För gammafördelningen med parametrar a och b är täthetsfunktionen

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

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

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

Fördelningsfunktion och täthetsfunktion

Gammafördelning

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

```
1 %% Fördelningsfunktion för gammafördelning
2     dx = 0.01;
3     x = -0:dx:10;           % Skapar en vektor med dx som ...
                             inkrement
4     y = gamcdf(x,1,2);
5     plot(x,y), hold on
6     z = gamcdf(x,5,1);
7     plot(x,z,'r')
```

Fördelningsfunktion och täthetsfunktion

En egendefinierad täthetsfunktion

Betrakta nu en stokastisk variabel X med täthetsfunktion

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

för ett specifikt λ . Genom att lösa ekvationen $\int_1^{10} f_X(x) dx = 1$ numeriskt så kan vi härleda approximationen $\lambda = 0.4267$.

Fördelningsfunktionen för X ges av

$$\begin{aligned} F_X(x) &= \int_1^x f_X(y) dy = \int_1^x \lambda e^{-\frac{y}{\lambda}} + \frac{\lambda}{y} dy = [-\lambda^2 e^{-\frac{y}{\lambda}} + \lambda \ln y]_1^x \\ &= -\lambda^2 e^{-\frac{x}{\lambda}} + \lambda \ln x + \lambda^2 e^{-\frac{1}{\lambda}}, \quad \text{för } x \in (1, 10). \end{aligned}$$

För $x \leq 1$ är $F_X(x) = 0$ och för $x \geq 10$ är $F_X(x) = 1$.

Fördelningsfunktion och täthetsfunktion

En egendefinierad täthetsfunktion

För att få en bättre bild av täthetsfunktionen för X , så kan vi plotta den och exempelvis jämföra med täthetsfunktionen för en exponentialfördelad stokastisk variabel med väntevärde ett.

```
1 %% Jämförelse av täthetsfunktioner
2     dx = 0.1;
3     x = 0:dx:15;           % Skapar en vektor med dx som ...
                             inkrement
4     mu = 1;
5     y = exppdf(x, mu);    % exponential-fordelningen
6     plot(x,y), hold on
7     lambda = 0.4267;
8     f=(lambda*exp(-x/lambda)+lambda./x).*(x >= 1 & x < ...
        = 10);
9     plot(x,f)
```

Skillnader mellan fördelningarna?

Multivariat normalfördelning

Den multivariata normalfördelningen bestäms av fem parametrar

$$\mu_X, \sigma_X, \mu_Y, \sigma_Y, \rho,$$

där $\mu_X, \mu_Y \in \mathbb{R}$, $\sigma_X, \sigma_Y \in (0, \infty)$ och $\rho \in [-1, 1]$. Följande kod ritar upp den simultana täthetsfunktionen

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

Vi testar att variera parametrarna. μ_X och μ_Y är väntevärdena i x - och y -led, σ_X och σ_Y är standardavvikelserna i x - och y -led och ρ är korrelationskoefficienten.

Simulering av slumpstal

Vi ska undersöka hur MATLAB kan användas för att generera slumpstal. Följande kod genererar N stycken $\text{Exp}(1/10)$ -fördelade slumpstal, ritar upp ett histogram av slumptalen samt plottar den sanna täthetsfunktionen för $\text{Exp}(1/10)$ ovanpå histogrammet som jämförelse.

```
1 %% Simulering av slumpstal
2     mu = 10;
3     N = 1e4;
4     y = exprnd(mu, N, 1);    % Genererar N exp-slumpstal
5     hist_density(y);        % Skapar ett normaliserat histogram
6     t = linspace(0, 100, N/10); % Vektor med N/10 punkter
7     hold on
8     plot(t, exppdf(t, mu), 'r') % 'r' betyder rod linje
9     hold off
```

Upprepa simuleringarna. Variationen i histogrammet beror på att olika slumpstal dras varje gång, men att fördelningen är densamma.

Stora talens lag säger att för oberoende, likafördelade stokastiska variabler $X_1, X_2, \dots$, så gäller det att

$$S_n := \frac{1}{n} \sum_{i=1}^n X_i \rightarrow E[X_1], \quad \text{då } n \rightarrow \infty,$$

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 .

Stora talens lag

En illustrerande simulering

```
1 %% Stora talens lag
2 mu = 0.5;
3 M = 500;
4 X = exprnd(mu, M, 1);
5 plot(ones(M, 1)*mu, 'r-.')
6 hold on
7 for k = 1:M
8     plot(k, mean(X(1:k)), 'b.')
9     if k == 1
10         legend('Sant \mu', 'Skattning av \mu')
11     end
12     xlabel(num2str(k)), pause(0.001)
13 end
14 hold off
```

Punkten med x-värde k är en skattning av medelvärdet av k exponentialfördelade stokastiska variabler. Som förväntat så konvergerar medelvärdet mot väntevärdet $\mu = 0.5$.

Monte Carlo-simulering av väntevärden

Antag att vi vill bestämma väntevärdet av antalet ögon som kommer upp vid kast med en sexsidig tärning. Detta är inte svårt att beräkna för hand, men det går också att kasta tärningen många gånger och sedan räkna ut medelvärdet av dessa kast. Om $X_1, X_2, \dots, X_n$ är likafördelade med väntevärde μ så gäller enligt Stora talens lag i Blom att

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

för varje $\varepsilon > 0$ när $n \rightarrow \infty$. Att använda detta samband och en slumpfelsgenerator för att beräkna väntevärden kallas för Monte Carlo-metoder.

Idén bakom Monte Carlo-metoder har funnits i matematiken åtminstone sedan 1700-talet, men synen kom att förändras under andra halvan av 1900-talet då det 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. Arbetet var kopplat till Manhattanprojektet vars syfte var att ta fram den första atombomben. Metoden namngavs efter casinot Monte Carlo i Monaco.

Monte Carlo-simulering av väntevärden

Ett första exempel

Antag att U är en stokastisk variabel som är likformigt fördelad över intervallet $[0, 2\pi]$. Om vi vill beräkna $E[\sin^2(U)]$, så kan vi göra detta analytiskt med hjälp av definitionen som

$$\begin{aligned} E[\sin^2(U)] &= \int_0^{2\pi} \sin^2(x) \frac{1}{2\pi} dx = \int_0^{2\pi} \frac{1 - \cos(2x)}{4\pi} dx \\ &= \left[\frac{x}{4\pi} - \frac{\sin(2x)}{8\pi} \right]_0^{2\pi} = \frac{1}{2}, \end{aligned}$$

men väntevärdet kan också beräknas med Monte Carlo-metoder med följande kod.

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

Resultatet blir nära $1/2$, men varierar något från gång till gång.

Monte Carlo-simulering av väntevärden

Ett svårare exempel

Fördelen med Monte Carlo-metoder är att de kan användas även för väntevärden som är svåra att beräkna exakt. Låt exempelvis X och Y vara oberoende stokastiska variabler där $X \in \text{Exp}(4)$ och $Y \in N(0, 1)$. Väntevärdet $E[e^{X \cos(Y)}]$ ges då av

$$\begin{aligned} E[e^{X \cos(Y)}] &= \int_0^\infty \int_{-\infty}^\infty e^{x \cos(y)} f_{X,Y}(x, y) dy dx \\ &= \int_0^\infty \int_{-\infty}^\infty e^{x \cos(y)} \frac{1}{4} e^{-x/4} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy dx, \end{aligned}$$

vilket är en rätt knepig integral.

Monte Carlo-simulering av väntevärden

Ett svårare exempel

Med Monte Carlo-metoder beräknas väntevärdet med följande kod. Vi simulerar slumpvariabler $X_1, \dots, X_N$ och $Y_1, \dots, Y_N$ från $\text{Exp}(4)$ respektive $N(0, 1)$. Vi bildar sedan $Z_i = e^{X_i \cos(Y_i)}$ och beräknar $\bar{Z}$

```
1 %% Monte Carlo, del 2
2     N = 1e5;
3     X = exprnd(1/4,N,1);
4     Y = randn(N,1);
5     mean(exp(X.*cos(Y)) );
```

Om vi upprepar simuleringen av väntevärdet, så kommer resultatet att variera, men vara nära det korrekta värdet. Om vi ökar N , så kommer variationen hos uppskattningarna av väntevärdet att minska.

Monte Carlo-simulering av talet π

Bakomliggande teori

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 $(U_1, V_1), (U_2, V_2), \dots, (U_N, V_N)$. 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

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

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

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

Monte Carlo-simulering av talet π

Kod för simulering

Följande kod genererar N punkter (U_i, V_i) , plottar dem i planet samt beräknar motsvarande skattning av värdet på π . Vi kör koden några gånger för olika val av N .

```
1 %% Monte Carlo, del 3.14
2     N = 1e2;
3     U = 2*rand(1,N,1)-1;      % Genererar U(-1,1)-ford. ...
        slumpstal
4     V = 2*rand(1,N,1)-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);  % Beräknar narmevarde på pi
10    pi = 4*mean(Z);
```

Centrala gränsvärdessatsen

Koden nedan simulerar exponentialfördelade slumpstal och summerar sedan dessa.

```
1 %% Centrala gransvardessatsen
2     M = 1e3;
3     N = 40;
4     mu = 5;
5     X = exprnd(mu, M, N);
6     S = cumsum(X, 2);
7     for k = 1:N
8         hist(S(:, k), 30)
9         xlabel(num2str(k))
10        pause(0.1)
11    end
```

Talet N anger hur många exponentialfördelade slumpvariabler som vi summerar. För $N = 1$ får vi en exponentialfördelning och för stora värden på N så kommer summorna att vara normalfördelade (enligt CGS). Vid vilket N ser det ut som att det inte gör någon skillnad att öka N ?

Simulering av konfidsensintervall

Ett konfidsensintervall med konfidsensgrad $1 - \alpha$ för en (okänd) parameter μ innehåller det sanna μ med sannolikhet $1 - \alpha$. Vi ska försöka förstå innebörden av detta begrepp med hjälp av simuleringar. Koden på nästa sida använder $n = 25$ oberoende observationer från $N(2, 1)$ -fördelningen för att skatta ett konfidsensintervall för väntevärdet med konfidsensgrad 95%. Detta upprepas 100 gånger så vi har 100 konfidsensintervall. Vi förväntar oss att 95% av dessa intervall innehåller den sanna parametern?

Simulering av konfidensintervall

De horisontella strecken visar de 100 olika konfidensintervallen och det vertikala strecket motsvarar det sanna värdet på parametern μ . Vi kör simuleringarna några gånger och ser att antalet intervall som innehåller det sanna värdet på μ varierar från gång till gång. Vi varierar även μ , σ , n och α (en i taget) och ser hur de olika parametrarna påverkar resultatet. En förändring av μ skjuter konfidensintervallen i sidled, en ökning av σ eller α gör intervallen längre, medan en ökning av n gör intervallen kortare.