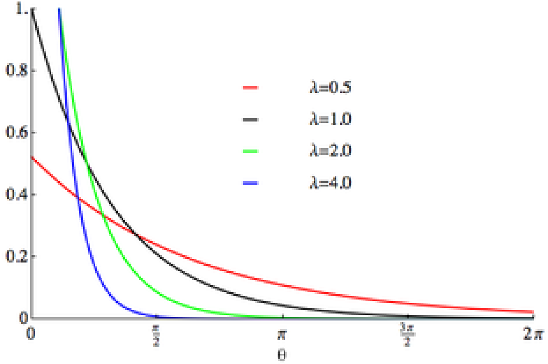
TP 547- Princípios de Simulação de Sistemas de Comunicação

Prof. Samuel Baraldi Mafra

Inatel
Instituto Nacional de Telecomunicações

- Statistics (scipy.stats): https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html
- numpy: https://numpy.org/doc/stable/
- matplotlib: https://matplotlib.org/stable/tutorials/index.html
- Comparação numpy Matlab: http://mathesaurus.sourceforge.net/matlab-numpy.html

Geração de variáveis aleatórias contínuas



Método da transformada inversa

- Seja X uma variável aleatória com função de distribuição acumulada F;
- Definindo $F_X^{-1}(z)$ como a função $X = F_X^{-1}(z)$ com $0 \leq z \leq 1$;
- Seja $U \sim U(0,1)$, a cdf da transformada inversa $F^{-1}(U)$ é

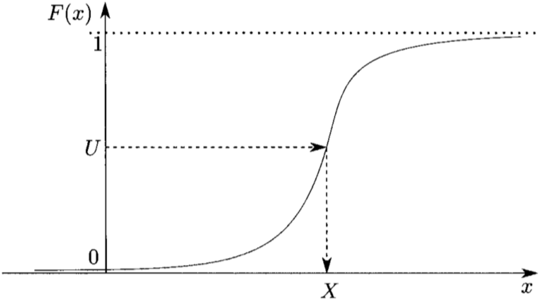
$$P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x)$$

- Este método pode ser utilizado em distribuições que sejam inversíveis.

Algoritmo de geração

Gerar um vetor U de valores com distribuição uniforme entre 0 e 1;
Calcular $X = F^{-1}(U)$;

Método da transformada inversa



Geradores de Variáveis Aleatórias com Distribuição Uniforme em um intervalo [a,b]

Considerando uma variável aleatória com distribuição uniforme no intervalo [a,b], a sua função densidade de probabilidade será:

$$f(x) = \frac{1}{b-a}, \quad a \leq x \leq b$$

$$F(x) = P(X \leq x) = \frac{x-a}{b-a}$$

Algoritmo de geração

Gerar um vetor U de valores com distribuição uniforme entre 0 e 1;
Calcular x=a+(b-a)*U;

```
import numpy as np
import matplotlib.pyplot as plt

A=0
B=40
N=100000
x=np.random.uniform(0,1,N)

Xab=A+(B-A)*x
print(Xab)

X=np.arange(A, B, 0.1)
fx=(1/(B-A))*np.ones(np.size(X))
plt.plot(X,fx)
plt.hist(Xab,bins=100,density=True)
plt.show()

uniabgen.py
```

Geradores de Variáveis Aleatórias com Distribuição Uniforme em um intervalo [a,b]:Exemplo

Uma pessoa chega a um prédio e vai até o elevador. Depois de acionado, o elevador demora entre 0 e 40 segundos para chegar. Considerando que o intervalo entre 0 e 40 segundos é uniformemente distribuído. Qual é a probabilidade de o elevador demorar menos de 15 segundos?
código: uniabgenex.py

Geradores de Variáveis Aleatórias com Distribuição Exponencial

Considerando uma variável aleatória com distribuição exponencial, a sua função densidade de probabilidade será:

$$f(x) = \lambda e^{-\lambda x}, \quad a \leq x \leq b$$
$$F(x) = P(X \leq x) = 1 - e^{-\lambda x}$$

Algoritmo de geração

Gerar um vetor U de valores com distribuição uniforme entre 0 e 1;
Calcular $x = -\ln(U)/\lambda$;

No python: random.exponential(β, N). β representa a média da distribuição ($\beta = 1/\lambda$)

```
import numpy as np
import matplotlib.pyplot as plt



N=10000
lambda1=1
x=np.random.uniform(0,1,N)

Xexp=-np.log(x)/lambda1

print(Xexp)
X=np.arange(0, 10, 0.1)
fx=lambda1*np.exp(-lambda1*X)
plt.plot(X,fx)
plt.hist(Xexp,bins=100,density=True)
plt.show()

expgen.py
```

Geradores de Variáveis Aleatórias com Distribuição Exponencial:Exemplo

Um componente eletrônico, de marca "A", tem duração de vida que segue uma distribuição exponencial com vida média de 100 horas. Qual é a probabilidade do componente parar de funcionar em menos de 40 horas de uso?
Código: expgenex.py

- O método de aceitação rejeição (ou método de rejeição) é usado para simular valores de uma variável aleatória, geralmente quando o método da inversa não pode ser aplicado;
- Considere uma v.a. com densidade $f(x)$ da qual é necessário gerar valores e $g(x)$ a densidade de outra v.a., da qual é possível gerar valores;
- No método, é gerado um valor y da v.a. com densidade $g(x)$ dado que $f(y)/g(y) \leq c$ para todo y.
- O valor c é dado por:

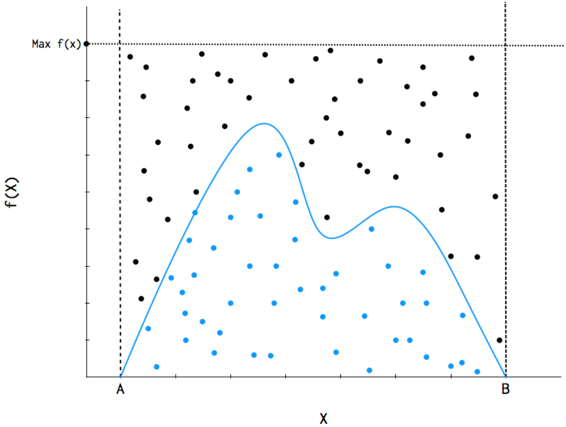
$$c = \max \frac{f(x)}{g(x)}.$$

- Baseado na geração de uma variável com distribuição uniforme, a amostra é aceita ou rejeitada.

Método da Aceitação-Rejeição

Algoritmo de Geração

1. Gerar uma variável aleatória Y de uma distribuição conhecida $g(x)$;
2. Gerar U independente de Y;
3. Se $U \leq \frac{f(Y)}{cg(Y)}$. Então aceita-se a amostra $(X = Y)$, senão rejeita e volta ao passo 1;


Max f(x)
f(X)
A
B
X

Método da Aceitação-Rejeição:Exemplo

Aplicar o método da aceitação-rejeição para gerar valores de uma v.a. com densidade:

$$f(x) = 20x(1-x)^3, \quad 0 < x < 1$$

Método da Aceitação-Rejeição:Exemplo

- Usa-se como função conhecida $g(x) = 1 \quad 0 < x < 1$, portanto:

$$\frac{f(x)}{g(x)} = 20x(1-x)^3.$$

- O valor de c é calculado pelo máximo valor de $\frac{f(x)}{g(x)}$, que acontece em $x = \frac{1}{4}$:

$$c = \max \frac{f(x)}{g(x)} = \max 20x(1-x)^3 = 20*0.25(1-0.25)^3 = \frac{135}{64}$$

- Então:

$$\frac{f(x)}{cg(x)} = \frac{256}{27}x(1-x)^3$$

Algoritmo de Geração

1. Gerar uma variável aleatória Y de uma distribuição uniforme U_1;
2. Gerar U_2 independente de U_1;
3. Se $U_2 \leq \frac{256}{27} Y(1-Y)^3$. Então aceita-se a amostra $(X = Y)$, senão rejeita e volta ao passo 1;

```
import numpy as np
import matplotlib.pyplot as plt
n=100000
xfunction=np.array([])
for i in range(n):
        x1=np.random.uniform(0,1)
        x2=np.random.uniform(0,1)
        while (x2 >=(256/27)*x1*pow((1-x1),3)):
                x1=np.random.uniform(0,1)
                x2=np.random.uniform(0,1)
        xfunction = np.append(xfunction, x1)
print(xfunction)
X=np.arange(0, 1, 0.1)
fx=20*X*pow((1-X),3)
plt.plot(X,fx)
plt.hist(xfunction,bins=100,density=True)
plt.show()
código: accept1.py
```

Método da Aceitação-Rejeição

Aplicar o método da aceitação-rejeição para gerar valores de uma v.a. com densidade:

$$f(x) = (1 + sin(x))/2\pi, \quad 0 < x < 2\pi$$

- Usa-se como função conhecida $g(x) = 1/2\pi \quad 0 < x < 2\pi$, portanto:
$$\frac{f(x)}{g(x)} = (1 + sin(x)).$$

Método da Aceitação-Rejeição:Exemplo

- O valor de c é calculado pelo máximo valor de $\frac{f(x)}{g(x)}$, que acontece em $x = \frac{\pi}{2}$:

$$c = \max \frac{f(x)}{g(x)} = (1 + sin(\pi/2)) = 2$$

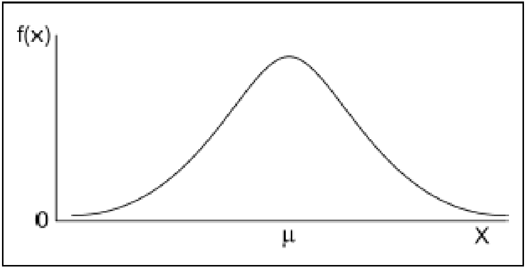
- Então:

$$\frac{f(x)}{cg(x)} = (1 + sin(x))/2$$

```
import numpy as np
import matplotlib.pyplot as plt
n=100000
xfunction=np.array([])
for i in range(n):
        x1=np.random.uniform(0,2*np.pi)
        x2=np.random.uniform(0,1)
        while (x2 >=(1+np.sin(x1))/2):
                x1=np.random.uniform(0,2*np.pi)
                x2=np.random.uniform(0,1)
        xfunction = np.append(xfunction, x1)
print(xfunction)
X=np.arange(0, 2*np.pi, 0.1)
fx=(1+np.sin(X))/(2*np.pi)
plt.plot(X,fx)
plt.hist(xfunction,bins=100,density=True)
plt.show()
código: accept2.py
```

Distribuição Normal: $N(\mu,\sigma^2)$

- A distribuição Normal é uma das mais importantes distribuições contínuas de probabilidade pois descreve muitos fenômenos aleatórios nas áreas da economia, biologia, telecomunicações, etc.

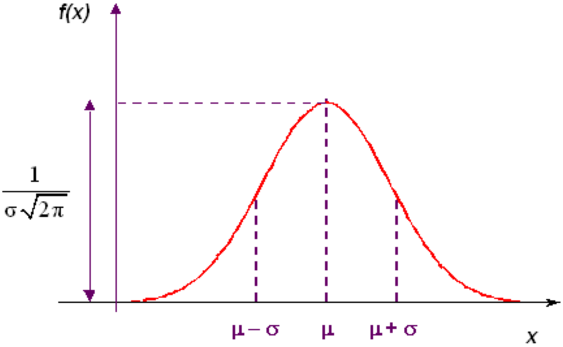


Distribuições Normal: $N(\mu,\sigma^2)$

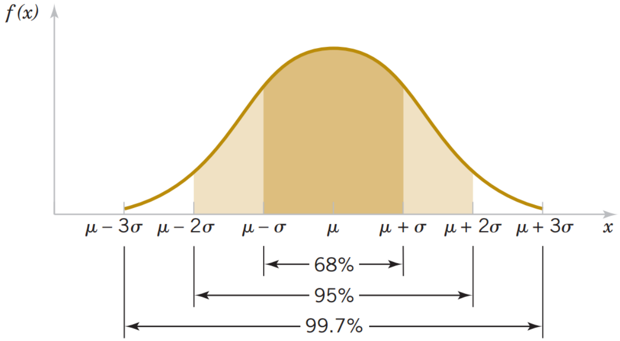
- Parâmetros
  - μ,σ^2.
- $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \qquad -\infty \leq x \leq \infty$;
- Média: μ;
- Variância: σ^2.

Distribuições Normal $N(\mu,\sigma^2)$: Propriedades

- $x = \mu$ é o ponto máximo de $f(x)$;
- $\mu - \sigma$ e $\mu + \sigma$ são os pontos de inflexão da curva;
- A curva normal é simétrica em torno da média μ.



- A área para certos desvios de σ apresenta alguns valores importantes:



Distribuições Normal $N(\mu,\sigma^2)$: Propriedades

- Se X apresenta uma distribuição normal com média μ e variância σ^2, então para qualquer constantes a e b, $aX + b$ é normalmente distribuído com média $a\mu + b$ e variância $a^2\sigma^2$;
- Portanto uma uma variável $Z = \frac{X-\mu}{\sigma}$ apresenta uma distribuição normal com média zero e variância unitária.
- Esta distribuição é conhecida como normal padrão. A CDF é dada por:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{x^2}{2}} dx.$$

Distribuições Normal $N(\mu,\sigma^2)$: Propriedades

- A distribuição normal padrão é importante pois através dela pode-se calcular as probabilidades em função de X em termos de Φ.

$$
\begin{aligned}
F(x) &= \Pr\{X \leq x\} \\
&= \Pr\{\frac{X-\mu}{\sigma} \leq \frac{x-\mu}{\sigma}\} \\
&= \Pr\{Z \leq \frac{x-\mu}{\sigma}\} \\
&= \Phi\left(\frac{x-\mu}{\sigma}\right)
\end{aligned}
$$

- O valor de $\Phi(x)$ pode ser determinado através de tabelas ou pelo uso de integração numérica. Em python usar norm.cdf da biblioteca scipy.stats.

- Método da aceitação-rejeição;
- O módulo da pdf da distribuição normal padrão é dado por:

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \qquad x \geq 0.$$

- Usa-se como função conhecida $g(x) = e^{-x}$, portanto:

$$\frac{f(x)}{g(x)} = \sqrt{2/(\pi)} e^{x - x^2/2}.$$

- O valor de c é calculado pelo máximo valor de $\frac{f(x)}{g(x)}$:

$$c = \max \frac{f(x)}{g(x)} = \max \sqrt{2/(\pi)} e^{x - x^2/2} = \sqrt{\left(\frac{2e}{\pi}\right)}$$

- Então:

$$\frac{f(x)}{cg(x)} = e^{-\frac{(x-1)^2}{2}}.$$

Geradores de Variáveis Aleatórias com Distribuições Normal $N(0,1)$

Algoritmo de Geração

1. Gerar uma variável aleatória,Y com uma distribuição exponencial de média $\lambda = 1$, isto é gera se U_1 e calcula $Y = -ln(U_1)$;
2. Gerar U_2;
3. Se $U_2 \leq e^{-\frac{(Y-1)^2}{2}}$. Faz $|Z| = Y$, senão volta ao passo 1;
4. Gerar U_3. Faz $Z = |Z|$ se $U_3 \leq 0.5$, senão faz $Z = -|Z|$ se $U_3 > 0.5$.

Para gerar uma distribuição $N(\mu,\sigma^2)$, basta fazer $N(\mu,\sigma^2) = Z * \sigma + \mu$.

```
import numpy as np
import matplotlib.pyplot as plt
n=100000
xfunction=np.array([])
mu=8;
sigma=np.sqrt(4);
for i in range(n):
        x1=np.random.uniform(0,1)
        xexp=-np.log(x1);
        x2=np.random.uniform(0,1)
        while (x2 >=(np.exp(-pow((xexp -1),2)/2))):
                x1=np.random.uniform(0,1)
                xexp=-np.log(x1);
                x2=np.random.uniform(0,1)
        x3=np.random.uniform(0,1)
        if x3<=0.5:
                xfunction = np.append(xfunction, xexp)
        else:
                xfunction = np.append(xfunction, -xexp)
normvalue=sigma*xfunction+mu;
código: acceptnorm.py
```

Geradores de Variáveis Aleatórias com Distribuições Normal $N(\mu,\sigma^2)$:Exemplo

As notas de bioestatística em um determinado curso ocorrem segundo uma distribuição $N(8,2^2)$. Calcule a probabilidade de um aluno:

a) Tirar menos que 6;

b) Tirar acima de 8.

código: acceptnormex1.py

Geradores de Variáveis Aleatórias com Distribuições Normal $N(\mu,\sigma^2)$:Exemplo

Suponha que a espessura média de arruelas produzidas em uma fábrica tenha distribuição normal com média 11,15mm e desvio padrão 2,238mm. Qual a porcentagem de arruelas que tem espessura entre 8,70mm e 14,70mm?
código: acceptnormex2.py