

Simulação Estocástica via Cadeias de Markov I

Ricardo Ehlers
ehlers@icmc.usp.br

Departamento de Matemática Aplicada e Estatística
Universidade de São Paulo

Introdução

Métodos de Monte Carlo

Métodos de Monte Carlo via cadeias de Markov (MCMC)

Algoritmo de Metropolis-Hastings

Introdução

Seja $\mathbf{X} = (X_1, \dots, X_d)$ um vetor aleatório discreto com possíveis valores $\mathbf{x}_1, \mathbf{x}_2, \dots$ e probabilidades

$$P(\mathbf{X} = \mathbf{x}_1), P(\mathbf{X} = \mathbf{x}_2), \dots$$

Suponha que queremos calcular,

$$E[g(\mathbf{X})] = \sum_{j=1}^{\infty} g(\mathbf{x}_j)P(\mathbf{X} = \mathbf{x}_j).$$

- ▶ Em muitas situações não é possível resolver este somatório.
- ▶ Uma possibilidade é aproximar $E[g(\mathbf{X})]$ por simulação.

Alguns exemplos,

1. Se $g(X) = X$, então tem-se que,

$$E[g(X)] = E(X) = \mu.$$

2. Quando $g(X) = (X - \mu)^2$ tem-se que,

$$E[g(X)] = E(X - \mu)^2 = \text{Var}(X).$$

3. Se $g(\mathbf{X}) = (\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})'$ sendo $\boldsymbol{\mu} = E(\mathbf{X})$ tem-se que,

$$E[g(\mathbf{X})] = E[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})'] = \text{Var}(\mathbf{X}),$$

a matriz de variâncias e covariâncias de \mathbf{X} .

4. Se $g(X) = I_A(X)$, sendo

$$I_A(X) = \begin{cases} 1, & \text{se } X \in A, \\ 0, & \text{se } X \notin A, \end{cases}$$

então,

$$\begin{aligned} E[g(X)] &= \sum_{j=1}^{\infty} I_A(x_j) P(X = x_j) \\ &= \sum_{x_j \in A} P(X = x_j) = P(X \in A). \end{aligned}$$

Como aproximar $E[g(\mathbf{X})]$ por simulação?

Métodos de Monte Carlo

Seja $\mathbf{X} = (X_1, \dots, X_d)$ um vetor aleatório discreto com possíveis valores $\mathbf{x}_1, \mathbf{x}_2, \dots$.

- ▶ Simule vetores aleatórios $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$ independentes e identicamente distribuídos com probabilidades,

$$P(\mathbf{X} = \mathbf{x}_j), \quad j = 1, 2, \dots$$

- ▶ Para cada vetor simulado faça,

$$Y^{(i)} = g(\mathbf{X}^{(i)}), \quad i = 1, \dots, n.$$

Lei forte dos grandes números

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n Y^{(i)} = E[g(\mathbf{X})],$$

com probabilidade 1.

Para n finito podemos usar então a média aritmética dos Y 's simulados como aproximação para $E[g(\mathbf{X})]$,

$$\frac{1}{n} \sum_{i=1}^n Y^{(i)} \approx E[g(\mathbf{X})].$$

Exemplo. Para uma variável aleatória discreta X com probabilidades $P(X = x_j)$, $j = 1, 2, \dots$ suponha que foram simulados $X^{(1)}, \dots, X^{(n)}$ independentes e identicamente distribuídos. Então,

$$E(X) \approx \frac{1}{n} \sum_{i=1}^n X^{(i)} = \mu$$

$$\text{Var}(X) \approx \frac{1}{n} \sum_{i=1}^n (X^{(i)} - \mu)^2$$

$$P(X \in A) \approx \frac{1}{n} \sum_{i=1}^n I(X^{(i)} \in A)$$

Como simular valores $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$ do vetor aleatório \mathbf{X} ?

- ▶ Pode ser complicado se \mathbf{X} é composto de variáveis aleatórias dependentes.
- ▶ Também se a distribuição de \mathbf{X} é conhecida parcialmente, i.e.

$$P(\mathbf{X} = \mathbf{x}_j) = C b_j, \quad j = 1, 2, \dots$$

sendo b_j conhecidos e a constante C desconhecida.

Note que,

$$C = 1 / \sum_{j=1}^{\infty} b_j.$$

Exemplo. Sejam X e Y variáveis aleatórias discretas não independentes assumindo valores em $\{0, 1, \dots\}$ de modo que,

$$P(Y = y, X = x) = P(Y = y|X = x) P(X = x),$$

sendo $P(Y = y|X = x)$ e $P(X = x)$ conhecidas.

Suponha que $Y = 0$ e deseja-se calcular,

$$E(g(X)|Y = 0) = \sum_{x=0}^{\infty} g(x) P(X = x|Y = 0).$$

Mas note que,

$$\begin{aligned} P(X = x|Y = 0) &= \frac{P(Y = 0|X = x) P(X = x)}{P(Y = 0)} \\ &= \frac{P(Y = 0|X = x) P(X = x)}{\sum_{x=0}^{\infty} P(Y = 0|X = x) P(X = x)}. \end{aligned}$$

Se o somatório no denominador não pode ser obtido analiticamente então $E(g(X)|Y = 0)$ não pode ser calculada de forma exata.

Note também que,

$$E(g(X)|Y = 0) = \frac{\sum_{x=0}^{\infty} g(x)P(Y = 0|X = x) P(X = x)}{\sum_{x=0}^{\infty} P(Y = 0|X = x) P(X = x)}.$$

Métodos de Monte Carlo via cadeias de Markov (MCMC)

Uma solução consiste em gerar uma sequência de n vetores aleatórios correlacionados $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$ segundo uma cadeia de Markov cuja distribuição estacionária seja $P(\mathbf{X} = \mathbf{x}_j)$, $j = 1, 2, \dots$

- ▶ Se a cadeia de Markov for ergódica e $E[g(\mathbf{X})] < \infty$ o resultado limite permanece válido, i.e.

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=1}^n g(\mathbf{X}^{(t)}) = E[g(\mathbf{X})].$$

- ▶ Para n finito podemos usar então a média aritmética como aproximação para $E[g(\mathbf{X})]$.

Autocorrelações

Os valores simulados segundo uma cadeia de Markov são por definição correlacionados.

Seja $\{X_t, t = 0, 1, \dots\}$ uma cadeia de Markov e $\{x_1, \dots, x_n\}$ uma sequência de valores simulados.

Autocorrelações

As autocorrelações de ordem k podem ser estimadas como,

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k} (x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{n-k} (x_t - \bar{x})^2}.$$

1. Simule vetores aleatórios $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$ segundo uma cadeia de Markov irredutível com probabilidades limite,

$$P(\mathbf{X} = \mathbf{x}_j) = \pi_j, \quad j = 1, 2, \dots$$

2. Para cada vetor simulado no passo t , faça $Y^{(t)} = g(\mathbf{X}^{(t)})$, $t = 1, \dots, n$.
3. Use $\frac{1}{n} \sum_{t=1}^n Y^{(t)}$ como aproximação para $E[g(\mathbf{X})]$.

Como simular a cadeia de Markov?

Algoritmo de Metropolis-Hastings

Seja X uma variável aleatória discreta com possíveis valores x_1, x_2, \dots e probabilidades

$$P(X = x_j), \quad j = 1, 2, \dots$$

Sejam $b_j > 0, j = 1, 2, \dots$ tais que,

$$P(X = x_j) = \frac{b_j}{B}, \text{ e}$$
$$B = \sum_{j=1}^{\infty} b_j < \infty.$$

Se não for possível obter a constante B analiticamente então,

- ▶ a distribuição de X é apenas parcialmente conhecida;
- ▶ não é possível simular valores diretamente desta distribuição.

O algoritmo de Metropolis-Hastings gera uma cadeia de Markov reversível com probabilidades estacionárias,

$$\pi_j = \frac{b_j}{B}, \quad j = 1, 2, \dots$$

Implementação

- ▶ Especifique a matriz de transição auxiliar Q com elementos $q(i, j)$ de uma cadeia de Markov irredutível.
- ▶ Defina uma cadeia de Markov $\{X_t, t = 1, 2, \dots\}$ tal que quando $X_t = i$ um valor de Y é gerado sendo,

$$P(Y = j | X_t = i) = q(i, j).$$

- ▶ Se o valor $Y = j$ foi gerado,
 - ▶ faça $X_{t+1} = j$ com probabilidade $\alpha(i, j)$, ou
 - ▶ $X_{t+1} = i$ com probabilidade $1 - \alpha(i, j)$.

Esta sequência de estados constitui uma cadeia de Markov com as seguintes probabilidades de transição,

$$P_{ij} = P(X_{t+1} = j | X_t = i) = q(i, j) \alpha(i, j)$$

$$P_{ii} = P(X_{t+1} = i | X_t = i) = q(i, i) + \sum_{k \neq i} q(i, k) [1 - \alpha(i, k)].$$

Esta cadeia será reversível e terá probabilidades estacionárias π_j se,

$$\pi_i P_{ij} = \pi_j P_{ji}, \text{ para } j \neq i,$$

ou equivalentemente,

$$\pi_i q(i, j) \alpha(i, j) = \pi_j q(j, i) \alpha(j, i), \text{ para } j \neq i.$$

que são chamadas *equações de equilíbrio*.

Probabilidade de Aceitação

Definindo-se as probabilidades de aceitação como,

$$\alpha(i, j) = \min \left\{ 1, \frac{\pi_j}{\pi_i} \frac{q(j, i)}{q(i, j)} \right\}$$

as equações de equilíbrio são satisfeitas pois, se

$$\alpha(i, j) = \frac{\pi_j}{\pi_i} \frac{q(j, i)}{q(i, j)}$$

então $\alpha(j, i) = 1$ e se $\alpha(i, j) = 1$ segue que,

$$\alpha(j, i) = \frac{\pi_i}{\pi_j} \frac{q(i, j)}{q(j, i)}.$$

Segue então que a cadeia de Markov simulada desta forma é reversível com probabilidades estacionárias $\pi_j, j = 1, 2, \dots$

Finalmente, como $\pi_j = b_j/B$ segue que,

$$\alpha(i,j) = \min \left\{ 1, \frac{b_j}{b_i} \frac{q(j,i)}{q(i,j)} \right\}$$

e o valor (desconhecido) de B não é necessário para construir a cadeia de Markov.

Algoritmo de Metropolis-Hastings (forma geral)

1. Defina uma matriz de transição Q ,
2. Especifique um valor inicial para a cadeia de Markov $X_0 = x_0$,
3. Para $t = 1, 2, \dots$
 - ▶ Gere Y com probabilidade,

$$P(Y = j | X_{t-1} = i) = q(i, j).$$

- ▶ Faça $X_t = j$ com probabilidade $\alpha(i, j)$, ou
- ▶ mantenha $X_t = i$ caso contrário,

- ▶ $Y = j$ é um valor *proposto* ou *candidato* e $\alpha(i, j)$ é a probabilidade de aceitação.
- ▶ Se o valor proposto não for aceito a cadeia permanece no mesmo estado.
- ▶ Na prática o passo de aceitação/rejeição é implementado gerando-se $U_t \sim U(0, 1)$ e fazendo,

$$X_t = j, \text{ se } U_t < \min \left\{ 1, \frac{\pi_j}{\pi_i} \frac{q(j, i)}{q(i, j)} \right\}$$

e mantendo $X_t = i$ caso contrário.

- ▶ A distribuição π é chamada *distribuição alvo*.

Exemplo. Deseja-se gerar valores de uma variável aleatória com distribuição de Poisson com média λ .

Vamos simular uma cadeia de Markov $\{X_t, t = 0, 1, \dots\}$ com espaço de estados $\{0, 1, \dots\}$ cuja distribuição limite seja,

$$\pi_j = P(X_t = j) = \frac{\lambda^j e^{-\lambda}}{j!}, \quad j = 0, 1, \dots$$

Note que,

$$\pi_j = B \frac{\lambda^j}{j!}, \quad j = 0, 1, \dots$$

e neste caso a constante B é conhecida,

$$B^{-1} = \sum_{j=0}^{\infty} \frac{\lambda^j}{j!} = e^{\lambda}.$$

Aplicando o algoritmo de Metropolis-Hastings usaremos um passeio aleatório para propor valores com matriz de transição Q simétrica tal que,

$$q_{0,0} = 1/2$$

$$q_{i,i+1} = 1/2$$

$$q_{i,i-1} = 1/2$$

$$q_{i,j} = 0, \text{ caso contrário.}$$

Note que neste caso,

$$r = \frac{\pi_j}{\pi_i} \frac{q(j, i)}{q(i, j)} = \frac{\pi_j}{\pi_i} = \frac{\lambda^j}{j!} \frac{i!}{\lambda^i} = \begin{cases} 1, \text{ se } i = j = 0 \\ \lambda/j, \text{ se } j = i + 1 \\ i/\lambda, \text{ se } j = i - 1 \end{cases}$$

Suponha que $x_{t-1} = i$,

1. Gere $u_1 \sim U(0, 1)$.
2. Se $i \neq 0$, faça
 - ▶ $j = i + 1$ se $u_1 < 1/2$, ou
 - ▶ $j = i - 1$ se $u_1 \geq 1/2$.
3. Se $i = 0$, faça
 - ▶ $j = 0$ se $u_1 < 1/2$, ou
 - ▶ $j = 1$ se $u_1 \geq 1/2$.
4. Faça,
 - ▶ $r = 1$ se $i = j = 0$, ou
 - ▶ $r = \lambda/j$ se $j = i + 1$, ou
 - ▶ $r = i/\lambda$ se $j = i - 1$.
5. Se $r \geq 1$ faça $x_t = j$.
6. Se $r < 1$ gere $u_2 \sim U(0, 1)$, faça $x_t = j$ se $u_2 < r$, ou $x_t = x_{t-1}$ caso contrário.

Função em R para executar o algoritmo anterior.

```
> poisson.metro <- function(lambda,i,n) {  
+ y = seq(n)  
+ for (k in 1:n) {  
+   u1 = runif(1)  
+   if (u1 < 0.5) j= ifelse(i==0,i,i-1) else j= i+1  
+   r = switch(i+2-j,lambda/j,1,i/lambda)  
+   u2 = runif(1)  
+   new= if(r>=1) j else {if(u2<r) j else i}  
+   i=new  
+   y[k]=i  
+ }  
+ return(y)  
+ }
```

Exemplo. Comandos do R para gerar valores de $X \sim \text{Poisson}(\lambda)$ com $\lambda = 5$ usando 10000 valores de uma cadeia de Markov.

```
> niter= 10000  
> nburn= 0  
> lam= 5  
> x = poisson.metro(lambda=lam,i=1,n=niter)  
> xx= as.matrix(x[(nburn+1):niter])  
> colnames(xx)="x"
```

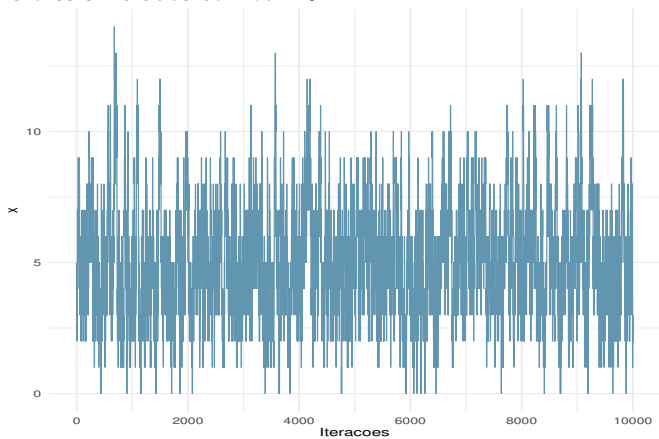
Para fazer gráficos com valores simulados e autocorrelações podemos usar os pacotes [bayesplot](#) e [ggplot2](#).

```
> library(bayesplot)
> library(ggplot2)

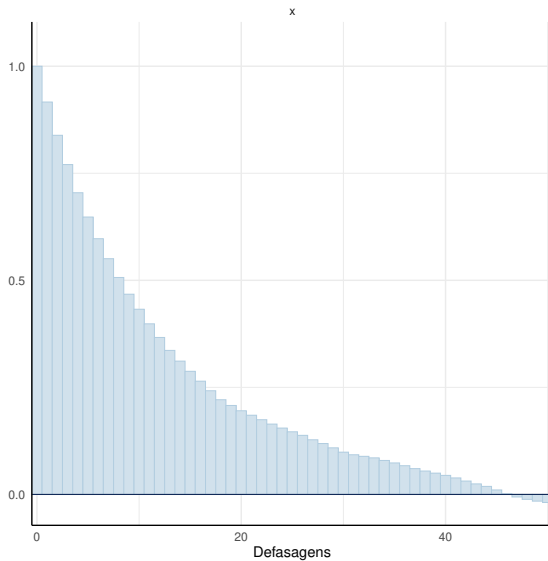
> bayesplot_theme_set(theme_minimal())
> color_scheme_set("blue")
> mcmc_trace(as.matrix(xx)) + labs(x="Iteracoes")

> mcmc_acf_bar(xx,lags=50)
```

10000 valores simulados com $\lambda = 5$.



Autocorrelações dos valores simulados.



- ▶ As autocorrelações são as correlações entre X_t e X_{t-k} , $k = 1, 2, \dots$ na cadeia de Markov simulada.
- ▶ Os valores da cadeia foram simulados de forma não independente portanto são correlacionados.
- ▶ Autocorrelações decaindo rapidamente a zero conforme aumenta a defasagem é bom sinal.

Dados estes valores simulados da cadeia de Markov, X_1, X_2, \dots, X_n podemos calcular por exemplo,

$$E(X) \approx \frac{1}{n} \sum_{t=1}^n X_t = \mu$$

$$\text{Var}(X) \approx \frac{1}{n} \sum_{t=1}^n (X_t - \mu)^2$$

$$P(X \in A) \approx \frac{1}{n} \sum_{t=1}^n I(X_t \in A)$$

$$P(X = k) \approx \frac{1}{n} \sum_{t=1}^n I(X_t = k), \quad k = 0, 1, 2, \dots$$

No exemplo, podemos comparar probabilidades Poisson $P(X = k)$ com $\lambda = 5$ calculadas de forma exata e aproximada.

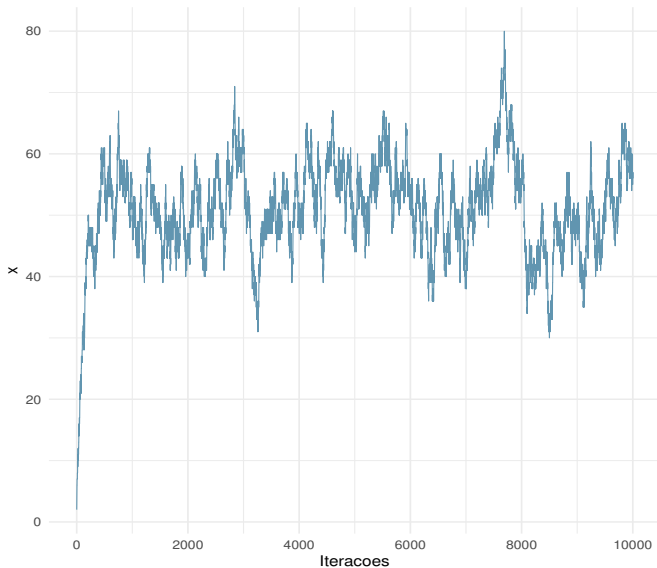
	Aprox	Exata
0	0.0062	0.0067
1	0.0323	0.0337
2	0.0748	0.0842
3	0.1366	0.1404
4	0.1797	0.1755
5	0.1837	0.1755
6	0.1451	0.1462
7	0.1028	0.1044
8	0.0644	0.0653
9	0.0393	0.0363
10	0.0214	0.0181
11	0.0097	0.0082
12	0.0027	0.0034
13	0.0012	0.0013
14	0.0001	0.0005
15	0.0000	0.0002

Comparando probabilidades Poisson $P(X = k)$ com $\lambda = 5$. Use o pacote `xtable` para criar uma tabela no Latex.

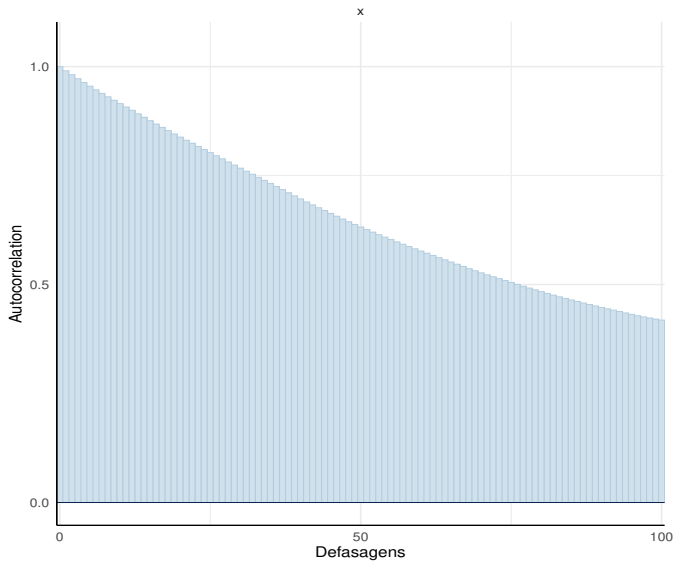
```
> mat= matrix(data=0,nrow=16,ncol=2)
> for (i in 0:15) {
+   mat[i+1,1]= mean(xx==i)
+   mat[i+1,2]= dpois(i,lam)
+ }
> colnames(mat)=c("Aprox","Exata")
> rownames(mat)= 0:15

> library(xtable)
> xtable(mat,digits=4)
```

Repita o exemplo anterior agora com $\lambda = 50$.



Autocorrelações dos valores simulados.



- ▶ A partir destes gráficos com valores simulados, o algoritmo parece ser bem mais eficiente quando o parâmetro da Poisson tem valor pequeno.
- ▶ A eficiencia menor para $\lambda = 50$ pode ser porque o número de simulações foi pequeno, ou devido à simetria da matriz Q , ou devido as probabilidades $q(i, j)$.
- ▶ Na prática é preciso checar estas possibilidades.

Exemplo. No exemplo anterior, seja Q uma matriz de transição qualquer (não necessariamente simétrica).

Para $x_{t-1} = i$ o algoritmo fica,

1. Gere um valor j com probabilidade $q(i, j)$, $j = 1, 2, \dots$
2. Faça $r = \frac{\pi_j q(j, i)}{\pi_i q(i, j)}$.
3. Se $r \geq 1$ faça $x_t = j$. Caso contrário, gere $u \sim U(0, 1)$,
 - ▶ se $u < r$ faça $x_t = j$,
 - ▶ se $u \geq r$ faça $x_t = x_{t-1}$.
4. Retorne ao passo 1.

- ▶ Como determinar se a cadeia simulada atingiu estacionariedade? Não há uma resposta teórica e isto deve ser checado na prática.
- ▶ Há diferentes possíveis escolhas para a matriz Q o que pode influenciar no tempo para a cadeia atingir estacionariedade.

Exemplo. Seja uma variável aleatória discreta Y que assume valores em $\{0, 1, \dots\}$ cuja distribuição depende da variável aleatória X . Suponha que a função massa de probabilidade é dada por,

$$P(Y = y|X = x) = \frac{1}{Z(x)} \left(\frac{2^y}{y!} \right)^x, \quad x > 0,$$

com,

$$Z(x) = \sum_{y=0}^{\infty} \left(\frac{2^y}{y!} \right)^x,$$

que não tem solução para $x \neq 1$.

Se X for uma variável aleatória discreta com possíveis valores x_1, x_2, \dots , deseja-se simular valores da distribuição de X condicional a um valor de Y .

Note que,

$$\begin{aligned} P(X = x|Y = y) &= \frac{P(Y = y|X = x) P(X = x)}{P(Y = y)} \\ &= \frac{1}{P(Y = y)} \frac{1}{Z(x)} \left(\frac{2^y}{y!}\right)^x P(X = x). \end{aligned}$$

Metropolis-Hastings para este exemplo

Se $X_t = x$,

1. Gerar um valor candidato x^* com probabilidade $q(x, x^*)$.
2. Calcular a probabilidade,

$$\alpha(x, x^*) = \min \left\{ 1, \frac{(2^y/y!)^{x^*}}{(2^y/y!)^x} \frac{Z(x)}{Z(x^*)} \frac{P(X = x^*)}{P(X = x)} \frac{q(x^*, x)}{q(x, x^*)} \right\}$$

3. Fazer $X_{t+1} = x^*$ com probabilidade $\alpha(x, x^*)$.

Porém a probabilidade de aceitação é impossível de ser calculada!

Resumindo

- ▶ Dada um vetor aleatório discreto \mathbf{X} com probabilidades $P(\mathbf{X} = x_j)$ vimos como aproximar características desta distribuição via simulação.
- ▶ Se \mathbf{X} é um vetor de variáveis aleatórias dependentes e/ou sua função massa de probabilidades é parcialmente conhecida simula-se cadeias de Markov cuja distribuição estacionária é $P(\mathbf{X} = x_j)$.
- ▶ Na próxima aula veremos mais exemplos e será introduzido o *Amostrador de Gibbs*.