

# SLICE SAMPLING

Luiz Fernando Maia

29 de novembro de 2019

# Roteiro

Introdução

A ideia do slice sampling

Slice sampling univariado

Slice sampling multivariado

Overrelaxed slice sampling

Slice sampling reflexivo

Demonstração

Conclusões

Referências

# SLICE SAMPLING<sup>1</sup>

BY RADFORD M. NEAL

*University of Toronto*

Markov chain sampling methods that adapt to characteristics of the distribution being sampled can be constructed using the principle that one can sample from a distribution by sampling uniformly from the region under the plot of its density function. A Markov chain that converges to this uniform distribution can be constructed by alternating uniform sampling in the vertical direction with uniform sampling from the horizontal “slice” defined by the current vertical position, or more generally, with some update that leaves the uniform distribution over this slice invariant. Such “slice sampling” methods are easily implemented for univariate distributions, and can be used to sample from a multivariate distribution by updating each variable in turn. This approach is often easier to implement than Gibbs sampling and more efficient than simple Metropolis updates, due to the ability of slice sampling to adaptively choose the magnitude of changes made. It is therefore attractive for routine and automated use. Slice sampling methods that update all variables simultaneously are also possible. These methods can adaptively choose the magnitudes of changes made to each variable, based on the local properties of the density function. More ambitiously, such methods could potentially adapt to the dependencies between variables by constructing local quadratic approximations. Another approach is to improve sampling efficiency by suppressing random walks. This can be done for univariate slice sampling by “overrelaxation,” and for multivariate slice sampling by “reflection” from the edges of the slice.

# Introdução

- ▶ Método baseados em cadeias de Markov como o amostrador de Gibbs e o algoritmo de Metropolis-Hastings são utilizados para amostrar de muitas das complexas distribuições encontradas na estatística.
- ▶ No entanto,
  - ▶ para implementar o Gibbs, é necessário saber amostrar valores das distribuições condicionais completas
  - ▶ para o Metropolis, é preciso uma escolha adequada da proposta

# Introdução

- ▶ Essas necessidades limitam o uso destes métodos e o desenvolvimento de softwares que automatizam a construção dessas cadeias a partir da especificação de modelos.
- ▶ Vários amostradores de cadeias de Markov são ineficientes pois, em geral:
  - ▶ eles propõe mudanças que não se adaptam bem às propriedades locais da função, o que faz com que as mudanças devem ser feitas em passos pequenos.
  - ▶ esses pequenos passos são passeios aleatórios que precisam de muito mais passos para realizar um deslocamento que poderia ser feito mais facilmente se estes passos fossem feitos consistentemente em uma direção.

# Introdução

- ▶ Os métodos da classe slice sampling podem ser aplicados para uma grande variedade de distribuições.
- ▶ Formulações simples do slice sampling univariado são uma alternativa para o Gibbs pois evitam a necessidade de amostrar das distribuições condicionais.
- ▶ Os métodos de slice sampling podem adaptativamente mudar a escala das mudanças, o que os faz mais fáceis de ajustar do que os algoritmos de Metropolis e evita problemas que podem surgir quando a escala adequada das mudanças varia na distribuição.

# Introdução

- ▶ Para muitas distribuições, estes métodos não vão amostrar de maneira mais eficiente do que o Gibbs ou um Metropolis com proposta adequada, mas estes métodos, geralmente, vão demandar menos esforço para implementação e ajuste.
- ▶ Métodos mais complexos de slice sampling podem se adaptar às dependências entre as variáveis e permitem mudanças maiores do que seria possível com Gibbs ou Metropolis. O que faz com que eles sejam mais eficientes em algumas distribuições.

## A ideia do slice sampling

- ▶ Suponha que queremos gerar valores da distribuição de uma variável  $x$ , que tem suporte em algum subconjunto do  $\mathbb{R}^n$  e densidade proporcional a  $f(x)$ .
- ▶ Para isso, podemos amostrar uniformemente da região  $(n + 1)$ -dimensional abaixo do gráfico de  $f(x)$ .



## A ideia do slice sampling

- Seja  $y$  uma variável auxiliar, define-se a distribuição conjunta de  $x$  e  $y$  como sendo uniforme na região

$$U = \{(x, y) : 0 < y < f(x)\}.$$

Logo,

$$p(x, y) = \begin{cases} 1/Z, & \text{se } 0 < y < f(x) \\ 0, & \text{caso contrário} \end{cases}$$

em que  $Z = \int f(x)dx$ , portanto,

$$p(x) = \int_0^{f(x)} (1/Z) dy = f(x)/Z$$

## A ideia do slice sampling

- ▶ Para amostrar de  $x$ , podemos amostrar de  $(x,y)$  e ignorar  $y$ .
- ▶ No entanto, gerar pontos independentes uniformemente de  $U$  pode não ser fácil, podemos então definir uma cadeia de Markov que convergirá para esta distribuição uniforme.
- ▶ Uma opção é o amostrador de Gibbs: amostramos alternadamente de  $Y \mid X$  que é uniforme em  $(0, f(x))$  e de  $X \mid Y$  que é uniforme em  $S = \{x : y < f(x)\}$  (slice).
- ▶ Porém, gerar um ponto de  $S$  pode ainda ser difícil, neste caso, podemos utilizar algum valor de  $x$  que deixe a distribuição uniforme em  $S$  invariante.

# Slice sampling univariado

- ▶ Pode ser utilizado para distribuições multivariadas, amostrando de cada variável em sequência.
- ▶ Seja  $x$  a variável a ser atualizada, com densidade  $\propto f(x)$ , para atualizar o valor atual  $x_0$  por um novo valor  $x_1$ , realizamos o seguinte procedimento:
  - a. Amostrar  $y$  uniformemente de  $(0, f(x_0))$ , definindo o slice  $S = \{x : y < f(x)\}$ .
  - b. Encontrar um intervalo,  $I = (L, R)$ , ao redor de  $x_0$  que contenha todo, ou muito, do slice.
  - c. Amostrar  $x_1$  da parte do slice deste intervalo.

## Slice sampling univariado

- ▶ No primeiro passo, ao invés de computar  $f(x)$ , podemos utilizar  $g(x) = \log(f(x))$  para evitar problemas de underflow.
- ▶ Neste caso, utilizamos a variável auxiliar  $z = \log(y) = g(x_0) - e$  em que  $e \sim \text{Exp}(1)$  e definimos  $S = \{x : z < g(x)\}$ .
- ▶ Os passos b. e c. podem ser implementados de várias maneiras desde que a cadeia de Markov resultante deixe a distribuição definida por  $f(x)$  invariante.

# Slice sampling univariado

## Encontrando um intervalo apropriado

- ▶ Nessa etapa, o objetivo é encontrar um intervalo  $I = (L, R)$  que contém o  $x_0$  e em que será amostrado  $x_1$ .
- ▶ Queremos esse intervalo contenha tanto quanto for factível do slice para que o  $x_1$  seja o mais diferente possível de  $x_0$ .
- ▶ Mas também queremos esse intervalo não seja muito maior do que o slice pois isso irá fazer o próximo passo menos eficiente.

# Slice sampling univariado

## Encontrando um intervalo apropriado

- ▶ Existem diversas estratégias para encontrar um intervalo:
  1. Idealmente, podemos fazer  $L = \inf(S)$  e  $R = \sup(S)$ , ou seja,  $I$  é o menor intervalo que contém  $S$ .
  2. Se o suporte de  $x$  é limitado, podemos definir  $I$  como esse suporte.
  3. Dada uma estimativa,  $w$ , do tamanho de  $S$ , podemos escolher aleatoriamente um intervalo inicial de tamanho  $w$  contendo  $x_0$  e, se necessário, expandí-lo em passos de tamanho  $w$  até que os extremos do intervalo estejam fora do slice (procedimento “stepping out”).
  4. Similarmente, podemos escolher aleatoriamente um intervalo de tamanho  $w$  e dobrar este intervalo até que os extremos estejam fora do slice (procedimento “doubling”).

# Slice sampling univariado

## Encontrando um intervalo apropriado

- ▶ Para cada estratégia, também precisamos encontrar o conjunto  $A$  de sucessores aceitáveis dado por

$$A = \{x : x \in S \cap I \text{ e } P(\text{Selecionar } I | \text{No estado } x) = P(\text{Selecionar } I | \text{No estado } x_0)\}$$

- ▶ Ou seja,  $A$  é o conjunto de estados dos quais seria tão provável escolher  $I$  quanto é escolher  $I$  do estado atual.
- ▶ Para 1. e 2.  $A = S$ , para 3.,  $A = S \cap I$  e para 4. pode ser necessário testar se um estado está em  $A$ .

# Slice sampling univariado

## Encontrando um intervalo apropriado

- ▶ A estratégia 1. é interessante quando todas as soluções de  $f(x) = y$  podem ser encontradas analiticamente ou por métodos numéricos.
- ▶ A estratégia 2. pode ser empregada utilizando transformações no suporte de  $x$  mas se o slice é usualmente muito menor do que o suporte, o algoritmo será ineficiente.
- ▶ O procedimento “stepping out” é apropriado para qualquer distribuição desde que tenhamos uma estimativa razoável de  $w$ .



# Slice sampling univariado

## Stepping out

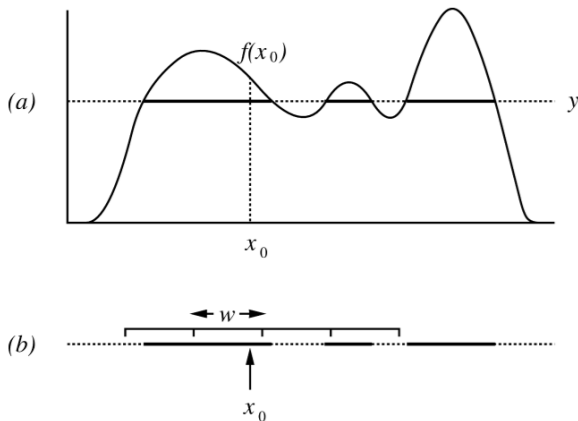


FIG. 1. A single-variable slice sampling update using the stepping-out and shrinkage procedures. A new point,  $x_1$ , is selected to follow the current point,  $x_0$ , in three steps. (a) A vertical level,  $y$ , is drawn uniformly from  $(0, f(x_0))$ , and used to define a horizontal "slice," indicated in bold. (b) An interval of width  $w$  is randomly positioned around  $x_0$ , and then expanded in steps of size  $w$  until both ends are outside the slice.

# Slice sampling univariado

## Doubling

- ▶ O procedimento “doubling” pode expandir o intervalo mais rapidamente do que o “stepping out”, logo, pode ser mais eficiente quando  $w$  é pequeno.

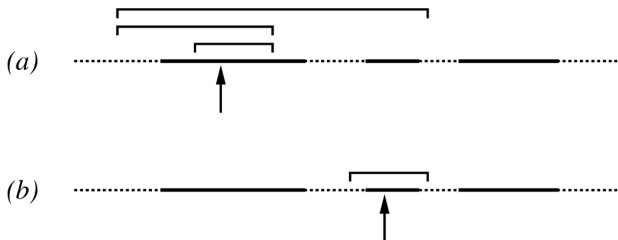


FIG. 2. *The doubling procedure. In (a), the initial interval is doubled twice, until both ends are outside the slice. In (b), where the start state is different, and the initial interval's ends are already outside the slice, no doubling is done.*

# Slice sampling univariado

Amostrando da parte do slice dentro do intervalo

- ▶ Dois métodos podem ser empregados para gerar valores da parte do slice dentro do intervalo  $I = (L, R)$ .
  - i. Amostrando uniformemente de  $I$  até que seja selecionado um ponto de  $A$ .
  - ii. Amostrando uniformemente de um intervalo que inicialmente é igual a  $I$ , e que é encolhido cada vez que um ponto selecionado não é de  $A$ , até que um ponto de  $A$  é encontrado.
- ▶ Em geral, o método ii. é mais adequado pois garante que o número de pontos amostrados não será muito grande.

# Slice sampling univariado

Amostrando da parte do slice dentro do intervalo

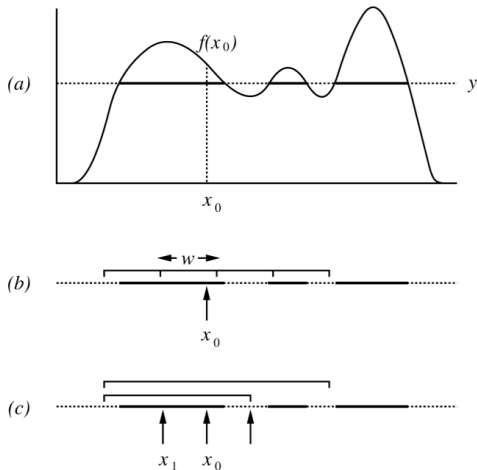


FIG. 1. A single-variable slice sampling update using the stepping-out and shrinkage procedures. A new point,  $x_1$ , is selected to follow the current point,  $x_0$ , in three steps. (a) A vertical level,  $y$ , is drawn uniformly from  $(0, f(x_0))$ , and used to define a horizontal "slice," indicated in bold. (b) An interval of width  $w$  is randomly positioned around  $x_0$ , and then expanded in steps of size  $w$  until both ends are outside the slice. (c) A new point,  $x_1$ , is found by picking uniformly from the interval until a point inside the slice is found. Points picked that are outside the slice are used to shrink the interval.

# Slice sampling univariado

Amostrando da parte do slice dentro do intervalo

- Para testar se o ponto amostrado no procedimento “doubling” é aceito, começamos o “doubling” de trás para frente e verificamos se nenhum dos intervalos tem extremidades fora do slice, o que terminaria o procedimento prematuramente.

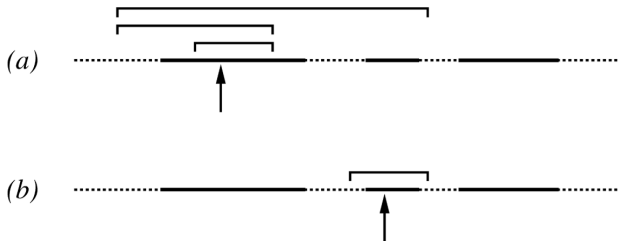


FIG. 2. The doubling procedure. In (a), the initial interval is doubled twice, until both ends are outside the slice. In (b), where the start state is different, and the initial interval's ends are already outside the slice, no doubling is done.

# Slice sampling univariado

## Exemplo

- ▶ Exemplo beta.

## Slice sampling multivariado

- ▶ Ao invés de utilizar técnicas de slice sampling univariado para cada componente de  $x = (x_1, \dots, x_n)$ , podemos generalizar estes métodos.
- ▶ Uma generalização é substituir o intervalo  $I = (L, R)$  por um hiper-retângulo  $H = \{x : L_i < x_i < R_i \text{ para todos } i = 1, \dots, n\}$ .

## Slice sampling multivariado

- ▶ O procedimento para encontrar o  $x_1 = (x_{1,1}, \dots, x_{1,n})$  a partir de  $x_0 = (x_{0,1}, \dots, x_{0,n})$  é:
  - a. Amostrar  $y$  uniformemente de  $(0, f(x_0))$ , definindo o slice  $S = \{x : y < f(x)\}$ .
  - b. Encontrar um hiper-retângulo  $H = (L_1, R_1) \times \dots \times (L_n, R_n)$ , ao redor de  $x_0$ , que contenha, preferencialmente, boa parte do slice.
  - c. Amostrar  $x_1$  da parte do slice deste hiper-retângulo.



# Slice sampling multivariado

- ▶ O procedimento “stepping out” não é bem generalizado pois um hiper-retângulo  $n$ -dimensional possui  $2^n$  vértices, e testar se todos estes vértices estão fora do slice é muito custoso.
- ▶ O “doubling” possui uma generalização melhor e podemos usar como critério de parada se um ponto uniformemente amostrado do hiper-retângulo atual está fora do slice.
- ▶ Outro esquema possível é utilizar um hiper-retângulo posicionado aleatoriamente sem nenhuma expansão, neste caso, é crucial que as estimativas  $w_i$  não sejam muito menores do que a realidade.

## Slice sampling multivariado

- ▶ Essa técnica de hiper-retângulos, em geral, é menos eficiente do que aplicar o slice sampling univariado em uma variável por vez.
- ▶ Encolher todas as dimensões do hiper-retângulo até que um ponto dentro do slice seja encontrado pode não ser eficiente pois a função densidade pode não variar muito em algumas dessas dimensões.
- ▶ Uma solução é realizar o encolhimento em apenas uma dimensão, baseado no gradiente de  $\log f(x)$  no último ponto.

# Slice sampling multivariado

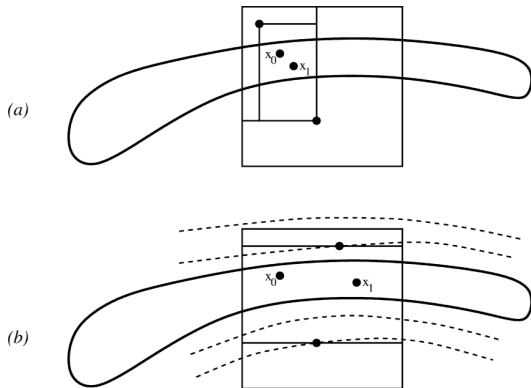


FIG. 7. Multivariate slice sampling with hyperrectangles. The heavy line outlines the slice, containing the current point,  $x_0$ . The large square is the initial hyperrectangle. In (a), the hyperrectangle is shrunk in all directions when the point drawn is outside the slice, until a new point,  $x_1$ , inside the slice is found. In (b), the hyperrectangle is shrunk along only one axis, determined from the gradient and the current dimensions of the hyperrectangle. The dashed lines are contours of the density function, indicating the direction of the gradient.

# Overrelaxed slice sampling

- ▶ Quando variáveis são atualizadas sem levar em conta suas dependências, as mudanças devem ser pequenas e muitas iterações são necessárias para mover de uma parte da distribuição para outra.
- ▶ É possível melhorar a eficiência do amostrador evitando o comportamento de passeio aleatório.
- ▶ Os métodos de “overrelaxation” atualizam uma variável de cada vez como o Gibbs.
- ▶ Ao invés de amostrar um valor da distribuição condicional independente do valor atual, o novo valor é escolhido do outro lado da moda do valor atual.
- ▶ Em geral, espera-se que a distribuição alvo seja unimodal.

# Overrelaxed slice sampling

- Uma das maneiras de fazer “overrelaxation” com slice sampling é:
  1. Aplicamos o procedimento “stepping out” para encontrar um intervalo ao redor de  $x_0$ .
  2. Se o intervalo inicial do “stepping out” já possui os extremos fora do slice, encolhemos seus extremos pela metade até que a mediana do intervalo esteja no slice.
  3. Localizamos os extremos do intervalo de forma mais precisa utilizando várias bissecções.
  4. Aproximamos o slice pelo intervalo  $(\hat{L}, \hat{R})$  formado pelos extremos do intervalo após as bissecções.
  5. Fazemos  $x_1 = \frac{\hat{L} + \hat{R}}{2} - (x_0 - \frac{\hat{L} + \hat{R}}{2}) = \hat{L} + \hat{R} - x_0$ , ou seja,  $x_1$  tem a mesma distância da mediana de  $(\hat{L}, \hat{R})$  do que  $x_0$ , só que está do lado oposto.

# Overrelaxed slice sampling

- ▶ As vezes é necessário rejeitar o candidato a  $x_1$  e fazer  $x_1 = x_0$ .
- ▶ Rejeitamos  $x_1$  se:
  1.  $x_1$  está fora do intervalo que foi encontrado antes das bisecções, pois o intervalo encontrado a partir de  $x_1$  seria diferente. Isso não pode acontecer em distribuições unimodais.
  2.  $x_1$  está fora do slice. Isso acontece até para uma distribuição unimodal quando os extremos do intervalo não foram localizados de forma exata.

# Overrelaxed slice sampling

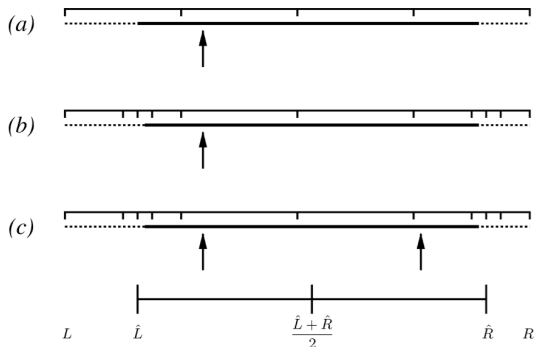


FIG. 9. *Overrelaxation using the stepping out procedure and bisection. In (a), an interval,  $(L, R)$ , with both ends outside the slice is found by stepping out from the current point. In (b), the endpoints of the slice are located more accurately using bisection. In (c), a candidate point is found by flipping through the point half-way between the approximations to the endpoints. In this case, the candidate point is accepted, since it is within the slice, and within the interval prior to bisection.*

## Slice sampling reflexivo

- ▶ Também podemos desenvolver métodos multivariados de slice sampling para evitar passeios aleatórios.
- ▶ Uma forma são os métodos que “refletem” nas fronteiras do slice.
- ▶ Esses métodos podem ser vistos como uma especialização para distribuições uniforme das dinâmicas do Hamiltonian Monte Carlo (DUANE et al; 1987).
- ▶ Novamente, suponhamos que temos uma distribuição em  $\mathbb{R}^n \propto f(x)$  e que conseguimos calcular  $f(x)$  e seu gradiente. Em cada iteração, vamos amostrar  $y$  de uma  $Unif(0, f(x))$  e definir o slice  $n$ -dimensional  $S = \{x : y < f(x)\}$ .



## Slice sampling reflexivo

- ▶ Além disso, introduziremos  $n$  variáveis auxiliares de “momento” em um vetor  $p$  que indicarão a velocidade e direção do movimento.
- ▶ No início de cada iteração, geramos um valor de  $p$ , independente de  $x$ , de alguma distribuição rotacionalmente simétrica, geralmente Gaussiana com média 0 e covariância identidade.
- ▶ Dados  $y$  e  $p$ , atualizamos  $x$  caminhando um número pré-determinado de passos na direção de  $p$ . Em cada passo, tentamos fazer  $x' = x + wp$  para algum valor de  $w$  que determina o tamanho do passo.
- ▶ Se o  $x'$  resultante está fora do slice, devemos trazê-lo de volta utilizando algum tipo de reflexão.

## Slice sampling reflexivo

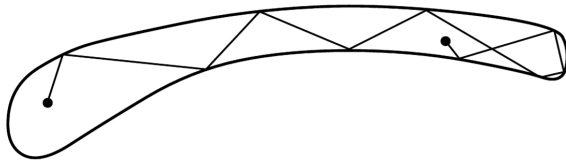


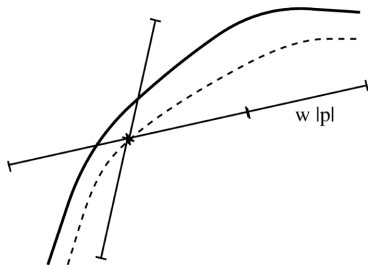
FIG. 11. *Moving around a two-dimensional slice by reflection from the exact boundaries.*

- ▶ A reflexão ideal é difícil de implementar pois é necessário um cálculo exato de onde a trajetória atual intersecta a fronteira do slice.
- ▶ Podemos utilizar duas estratégias de aproximação: reflexão “de dentro” ou reflexão de “de fora”.
- ▶ É necessário rejeitar alguns valores para que estas estratégias sejam válidas.

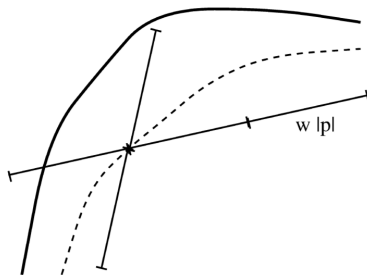
## Slice sampling reflexivo

- ▶ Quando caminhar de  $x$  para  $x' = x + wp$  nos tira do slice, podemos tentar a reflexão do último ponto de dentro,  $x$ , ao invés do ponto exato em que a trajetória intersecta com a fronteira, utilizando o gradiente de  $f(x)$  neste ponto.
- ▶ Para este método ser válido, precisamos checar se a trajetória reversa também refletiria neste ponto, verificando se um passo na direção oposta nos tiraria do slice.

# Slice sampling reflexivo



(a)



(b)

FIG. 12. *Reflection from an inside point. The trajectories here go in steps of size  $w|p|$ , starting from the top right, until a point outside the slice is reached, when a reflection is attempted based on the inner contour shown. In (a), the reflection is successful; in (b), it must be rejected, since the reverse trajectory would not reflect at this point.*

## Slice sampling reflexivo

- ▶ Outra alternativa é refletir do ponto exterior,  $x'$ , utilizando o gradiente neste ponto.
- ▶ Depois de realizar um número pré-determinado de passos, aceitamos a trajetória se o ponto final está no slice.
- ▶ Para este método ser válido, devemos refletir sempre que o ponto atual estiver fora do slice, até mesmo se a reflexão não traga o ponto de volta para o slice.

# Slice sampling reflexivo

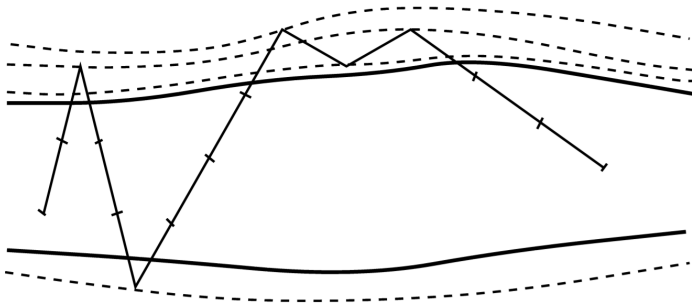


FIG. 13. *Reflection from outside points. Starting from the left, two reflections based on outside contours lead back inside the slice after the next step. The step after the third reflection is still outside the slice, so further reflections must be done. In this case, the trajectory eventually returns to the slice, and its endpoint would therefore be accepted.*

# Demonstração

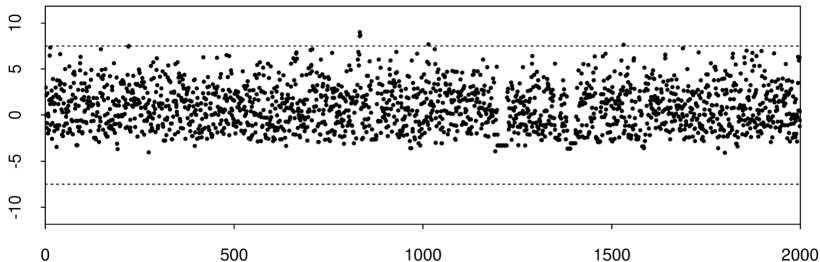
- ▶ Para demonstrar os benefícios da natureza adaptativa do slice sampling, o autor mostra como ele pode evitar um cenário em que a resposta errada é obtida sem indicações de que algo está errado.
- ▶ Suponha que queremos amostrar de uma distribuição composta de 10 variáveis,  $v$  e  $x_1$  até  $x_9$ .
- ▶  $v$  tem distribuição marginal normal com média 0 e desvio-padrão 3.
- ▶ Condicionais a  $v$ ,  $x_1$  até  $x_9$  são independentes e têm distribuição normal com média 0 e variância  $e^v$ .
- ▶ Suponha que não sabemos estas distribuições, caso contrário, poderíamos gerar valores de  $v$  e depois das distribuições condicionais.

# Demonstração

## Simulação 1

- ▶ 2000 iterações em que cada iteração consiste de 10000 atualizações de Metropolis-Hastings multivariado.
- ▶ Proposta Gaussiana centrada no valor atual com variância 1 para cada uma das 10 variáveis.
- ▶ Valores iniciais  $v = 0$  e  $x_i = 1$ .

*Multivariate Metropolis updates, standard deviation 1*

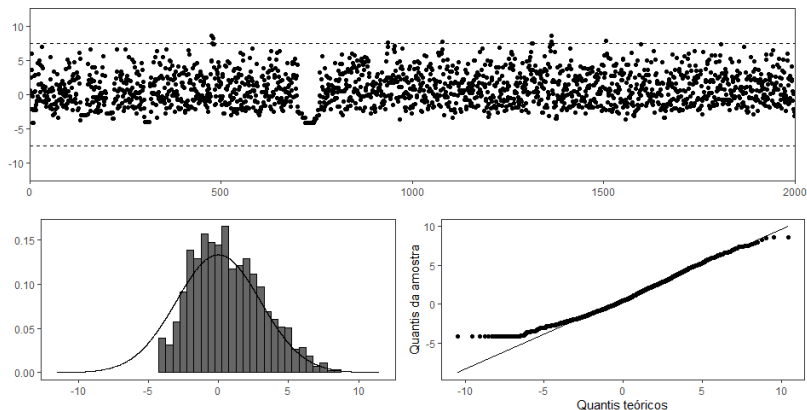




# Demonstração

## Simulação 1

Figura: Replicação da primeira simulação

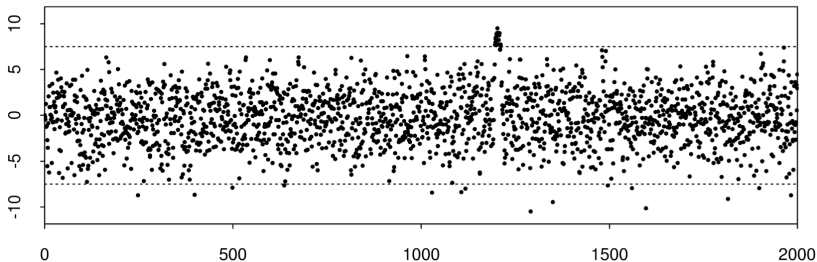


# Demonstração

## Simulação 2

- ▶ 2000 iterações em que cada iteração consiste de 1300 atualizações de Metropolis univariadas para cada variável em sequência.
- ▶ Proposta Normal centrada no valor atual e variância 1.

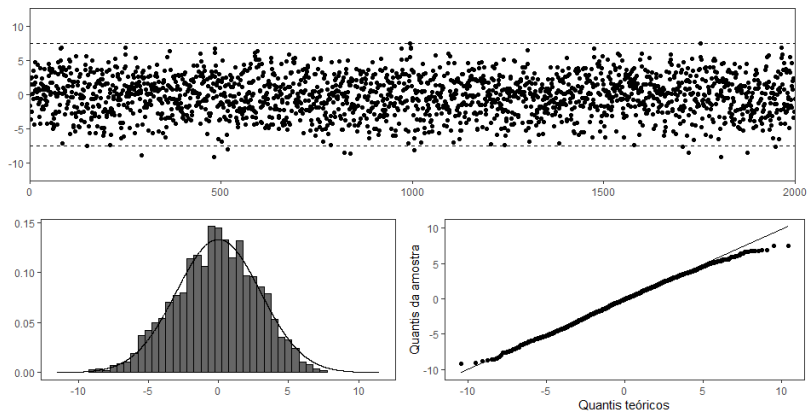
*Single-variable Metropolis updates, standard deviation 1*



# Demonstração

## Simulação 2

Figura: Replicação da segunda simulação

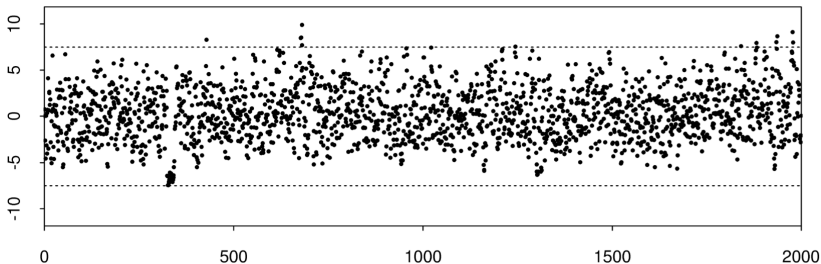


# Demonstração

## Simulação 3

- ▶ 2000 iterações em que cada iteração consiste de 10000 atualizações de Metropolis-Hastings multivariado.
- ▶ Proposta Gaussiana centrada no valor atual e com o  $\log_{10}$  do desvio padrão gerado de uma  $Unif(-3, 3)$ .
- ▶ Valores iniciais  $v = 0$  e  $x_i = 1$ .

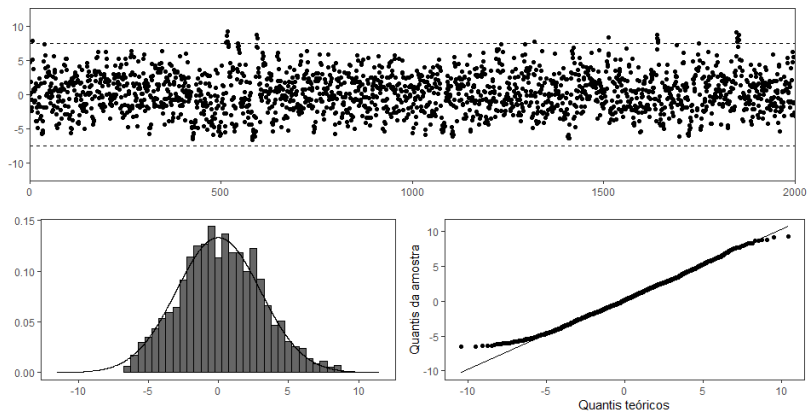
*Multivariate Metropolis updates, random standard deviation*



# Demonstração

## Simulação 3

Figura: Replicação da terceira simulação

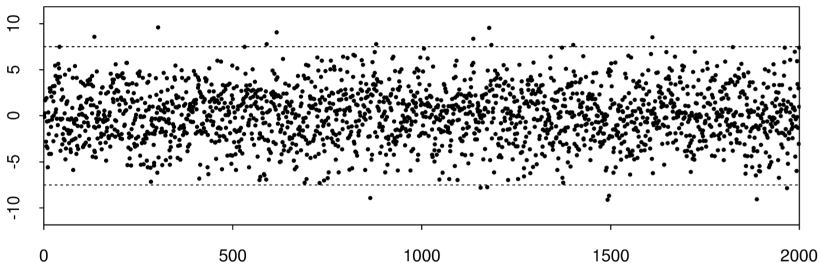


# Demonstração

## Simulação 4

- ▶ 2000 iterações de slicing sample em que cada iteração consiste de 120 atualizações com o procedimento “stepping out” e encolhimento para cada uma das variáveis.
- ▶ Comprimento de intervalo inicial  $w = 1$ .

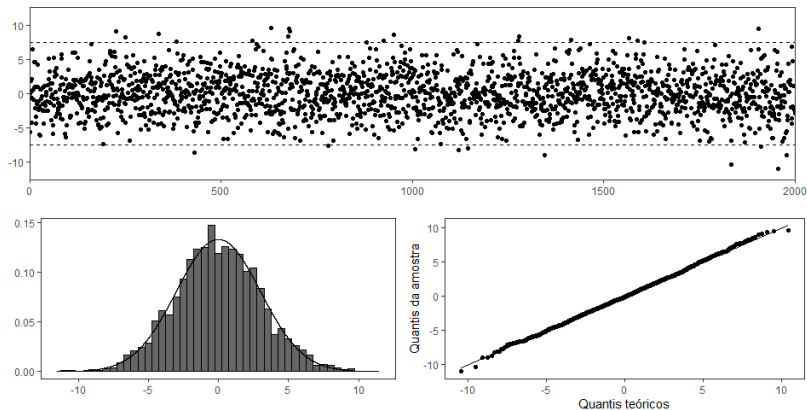
*Single-variable slice sampling, initial width of 1*



# Demonstração

## Simulação 4

Figura: Replicação da quarta simulação



# Demonstração

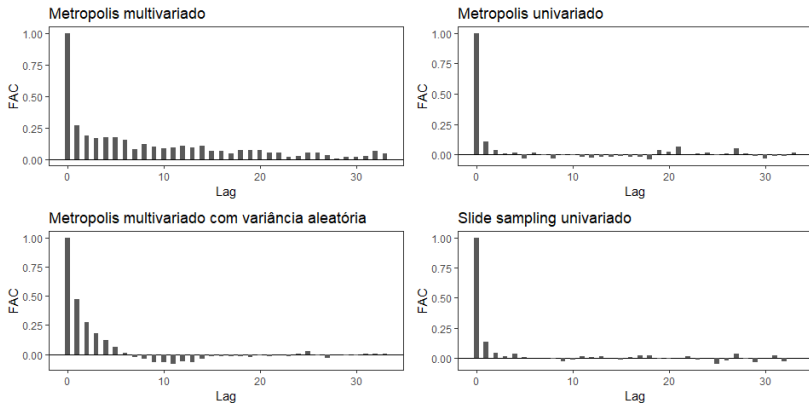
Tabela: Estatísticas das cadeias de  $v$  e taxas de aceitação

Simulação	Média	Desvio-padrão	Taxa de aceitação (%)
1	0,7107	2,5175	20,04
2	-0,1599	2,906	48,56 ( $v$ ) e 62.29 ( $x$ )
3	0,2317	2,9203	40,32
4	-0,0086	3,0357	



# Demonstração

Figura: Funções de autocorrelação das cadeias de  $v$



# Conclusões

- ▶ Métodos de slicing sample são alternativas para Metropolis e Gibbs que precisam de um ajuste menos elaborado e se adaptam melhor a distribuição alvo.
- ▶ No entanto, os métodos de Metropolis e slicing sample devem ter uma performance similar se ambos forem bem ajustados.
- ▶ Métodos como slicing sample reflexivo podem se adaptar as dependências entre as variáveis e possibilitar uma maior eficiência.
- ▶ TLDR: slicing sample é adequado para uso automático de softwares pois é menos dependente do ajuste.

# Referências

NEAL, Radford M. Slice sampling. *The annals of statistics*, v. 31, n. 3, p. 705-767, 2003.

DUANE, Simon et al. Hybrid monte carlo. *Physics letters B*, v. 195, n. 2, p. 216-222, 1987.