Aula 14

Roteiro

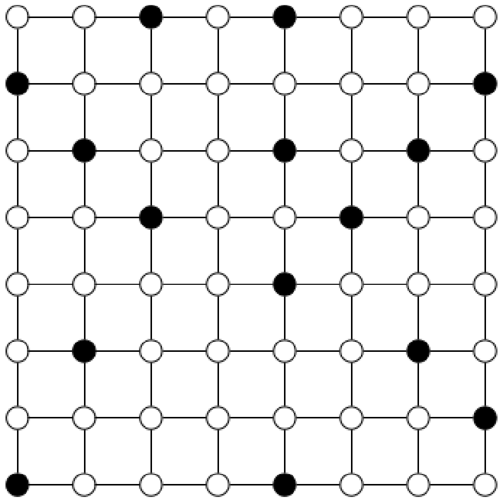
- Modelo Hardcore via Metropolis-Hastings
- Gibbs Sampling
- Exemplo simples
- Modelo Hardcore via Gibbs Sampling
- q-coloração via Gibbs Sampling
- Tempo de mistura (*fast mixing*)

Markov Chain Monte Carlo

- **Problema 1:** gerar amostras de espaço *S* grande e complicado com distribuição π
- **Solução:** Markov chain Monte Carlo
  - construir uma CM com espaço de estado *S* tal que π seja sua distribuição estacionária
  - gerar amostras através de caminhos amostrais de comprimento τ_ε
- **Problema 2:** computar $E_\pi[f(X)]$ para alguma função f no espaço amostral *S* com distribuição π
- **Solução:** Markov chain Monte Carlo!
  - estimador de $E_\pi[f(X)]$ utilizando caminho amostral

Hardcore Model

- Modelo *hardcore*: colorir cada vértice de um grafo com preto ou branco
  - restrição: vértice preto tem todos seus vizinhos brancos
- Conjunto independente (em teoria dos grafos)



15 vértices pretos

- *S:* conjunto de todas as configurações (sem restrição)
  - $|S| = 2^n$, para grafo com n vértices
- *S'* : subconjunto de configurações válidas
- **Problema:** número médio de vértices pretos de uma configuração válida?
  - valor esperado sobre a distribuição uniforme sobre o espaço *S'*

Hardcore Model

- $f(C)$: número de vértices pretos de uma configuração válida C escolhida ao acaso, uniformemente
- Então

$$E[f(C)] = \frac{1}{Z} \sum_{c \in S'} f(c) \qquad Z = |S'| = \sum_{c \in S} 1(c \text{ é válida})$$

- **Problemas:**
  (1) como enumerar *S'* de forma eficiente?

  (2) como lidar com o tamanho de *S'* (exponencial)?
- **Solução:** Markov Chain Monte Carlo

MCMC para *Hardcore Model*

- Construir CM onde estados são elementos de *S'*
  - configurações válidas
- Fazer com que distribuição estacionária desta CM seja uniforme, $\pi_c = 1/Z$ para toda configuração *c*
- Gerar caminho amostral pela CM
  - X_k estado da CM no passo *k* (configuração)
- Estimar valor esperado do número de vértices em preto usando caminho amostral

$$\hat{f}(k) = \frac{1}{k} \sum_{l=1}^{k} f(X_k) \xrightarrow{} E[f(C)]$$

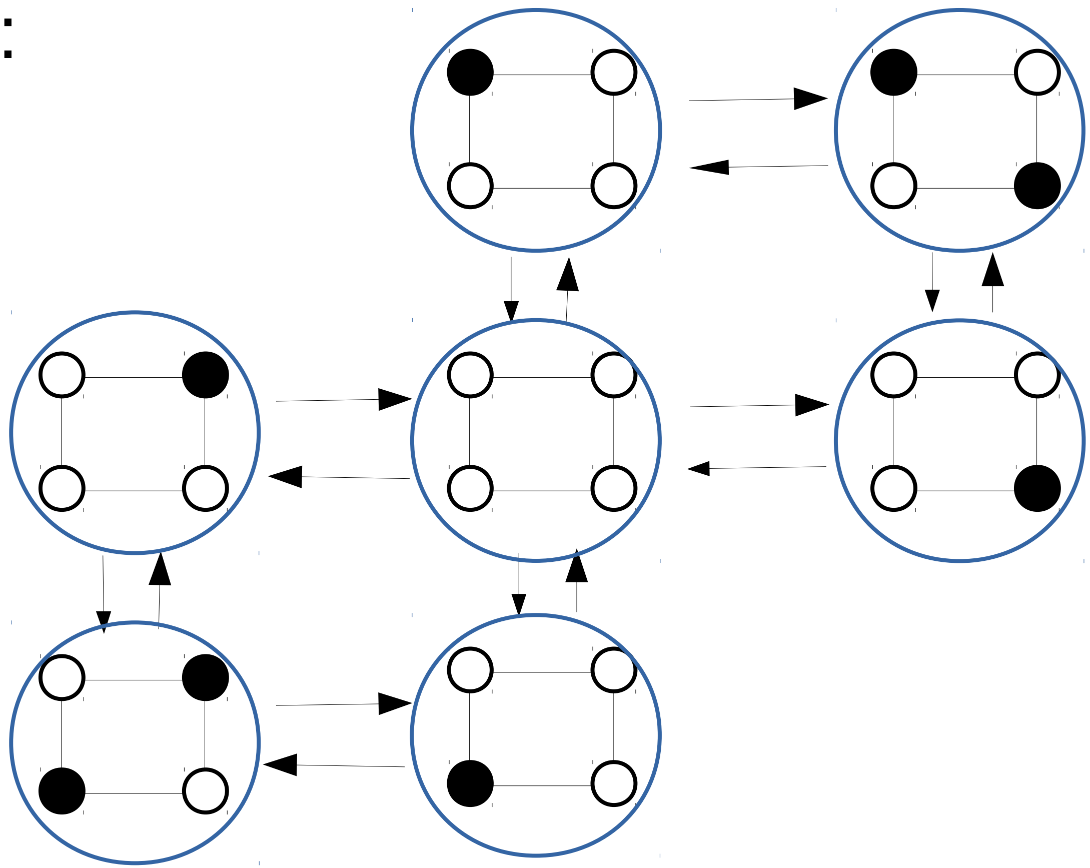
converge quando
k → infinito
(teorema ergódico)

Estimador do número médio de vértices pretos após *k* amostras

Figueiredo 2021

MCMC para *Hardcore Model*

- Construir CM é equivalente a definir as transições para cada estado
- **Ideia:** construir transições entre configurações válidas que diferem na cor de exatamente um vértice
  - usar passeio aleatório para transicionar
- Exemplo:



Figueiredo 2021

MCMC para *Hardcore Model*

- Passeio aleatório não possui distribuição estacionária uniforme
- Construir cadeia de Metropolis-Hastings com distribuição uniforme (aula passada)

$$P'_{i,j} = \frac{1}{d_i} min\{1, \frac{d_i}{d_j}\}$$

para todo *j* vizinho de *i* (estado atual)

- d_i : grau da configuração *i* (na cadeia de Markov)
  - d_i = número de vértices pretos + número de vértices brancos que não possuem vizinhos pretos
- Gerar caminho amostral para calcular o estimador $\hat{f}(k)$

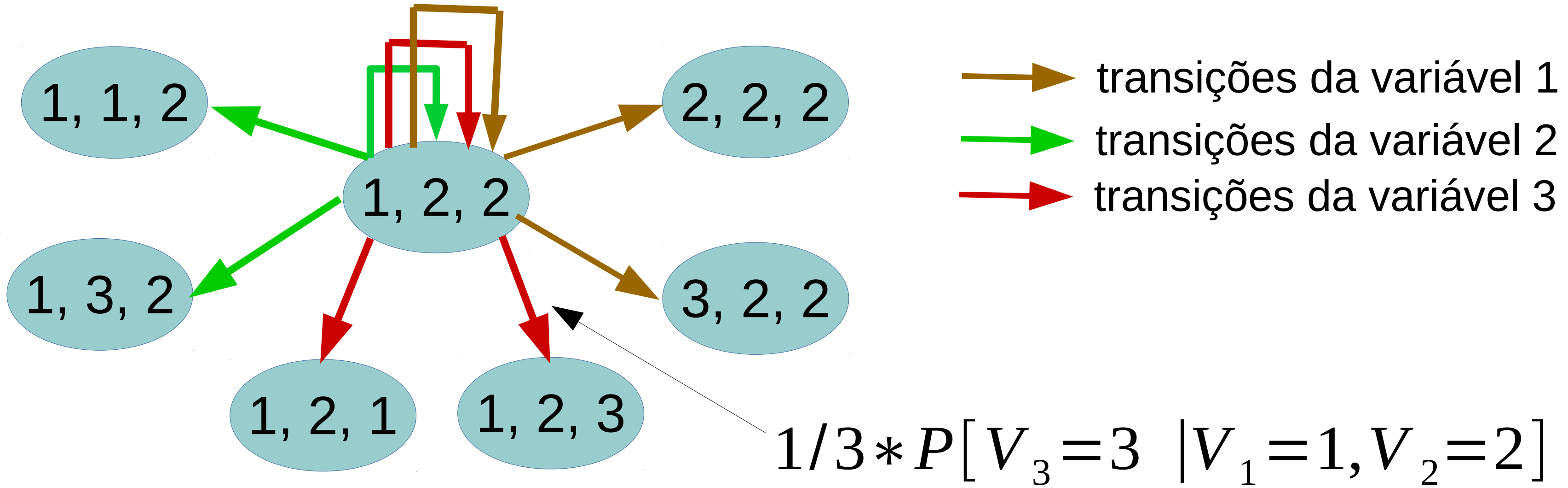
Gibbs Sampling

- Outro algoritmo de Markov Chain Monte Carlo
  - também chamado de *Glauber Dynamics*
- Algoritmo para construir CM com distribuição estacionária π sobre o espaço S (ambos são entrada para o problema)
- Elemento do espaço amostral é um vetor $V = (V_1, \ldots, V_n)$
- Cada variável V_i assume valores do conjunto K
  - $|S| <= |K|^{|V|}$: nem todas as combinações precisam estar no espaço amostral
- Baseado na distribuição condicional de uma variável dado o valor de todas as outras
  - precisa ser conhecida a priori ou construída para induzir π

$$P[V_k = a_k \mid V_1 = a_1, \ldots, V_n = a_n]$$

Gibbs Sampling

- CM é construída considerando todos possíveis valores para uma variável, dado o valor atual de todas as outras
- Probabilidade de transição é proporcional a probabilidade condicional
  - dividida pelo número de variáveis, *n*
- Exemplo: $n = 3$, $|K| = 3$



$$1/3 * P[V_3 = 3 \ | V_1 = 1, V_2 = 2]$$

Gibbs Sampling

- CM construída é aperiódica, reversível e possui distribuição estacionária π
- Considerando duas variáveis (ideia idêntica outros casos)

$$\begin{tikzcd} a_i, a_j \arrow[r, shift left, "1/2*P[V_1=a_k \mid V_2=a_j]"] & a_k, a_j \arrow[l, shift left, "1/2*P[V_1=a_i \mid V_2=a_j]"] \end{tikzcd}$$

- Para reversibilidade temos que para toda transição

$$\pi_{a_i,a_j} P_{(a_i,a_j),(a_k,a_j)} = \pi_{a_k,a_j} P_{(a_k,a_j),(a_i,a_j)}$$

$$\pi_{a_i,a_j} \frac{1}{2} P[V_1=a_k | V_2=a_j] = \frac{1}{2} \pi_{a_k,a_j} P[V_1=a_i | V_2=a_j]$$

$$\pi_{a_i,a_j} P[V_1=a_k | V_2=a_j] P[V_2=a_j] = \pi_{a_k,a_j} P[V_1=a_i | V_2=a_j] P[V_2=a_j]$$

$$\pi_{a_i,a_j} P[V_1=a_k, V_2=a_j] = \pi_{a_k,a_j} P[V_1=a_i, V_2=a_j]$$

$$\pi_{a_i,a_j} \pi_{a_k,a_j} = \pi_{a_k,a_j} \pi_{a_i,a_j}$$

Algoritmo

- Dado estado atual $X_t = (V_1(t), V_2(t), \ldots, V_n(t))$
- Escolher uma das variáveis uniformemente
  - $K \sim$ Uniforme$(1, n)$
- Escolher valor para V_K dado o vetor $V_1(t), \ldots, V_n(t)$
  - usar probabilidade condicional para V_K
  - seja a_k o valor escolhido
- Transicionar para o estado X_{t+1} copiando valores de $V_i(t)$ e atualizando apenas V_K com a_k
- Repetir por τ_ε passos para que distribuição de X_t esteja próxima de π

Exemplo

- Duas variáveis (X, Y) com seguinte distribuição conjunta

X

Y

	1	2	3	4	5	6	7	8	9
1	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100
2	0	3/100	3/100	3/100	3/100	3/100	3/100	3/100	0
3	0	0	5/100	5/100	5/100	5/100	5/100	0	0
4	0	0	0	7/100	7/100	7/100	0	0	0
5	0	0	0	0	24/100	0	0	0	0

- Distribuição condicional é obtida a partir da conjunta
  - dado Y, X tem distribuição uniforme sobre valores que dependem de Y
- Como gerar amostras de (X, Y) ?

Gibbs Sampling!

Exemplo

- Escolher $Z_0 = (X_0, Y_0)$ de forma arbitrária, ex. $Z_0 = (1,1)$
- Escolher uniformemente entre as variáveis X e Y
- Escolher novo valor para X ou Y utilizando a distribuição condicional (dado o valor atual da outra variável)
  - seja a_t o valor escolhido
- Atualizar $Z_{t+1} = (X_t, a_t)$ ou $Z_{t+1} = (a_t, Y_t)$
- Repetir τ_ε vezes e retornar Z_t como sendo a amostra

Hardcore com Gibbs

- Estado como vetor de vértices e suas cores
  - $|V| = n$, $V_i = \{0, 1\}$, para cada vértice *i* do grafo
- Distribuição π é uniforme
- Probabilidade condicional do vértice *i*
  - dado a cor de todos os outros vértices, temos apenas um ou dois estados restantes (de probabilidade uniforme)
  - se *i* não possui vizinhos em preto

$$P[V_i=1 \mid V_1=a_{1,}\ldots,V_n=a_n]=1/2$$

$$P[V_i=0 \mid V_1=a_{1,}\ldots,V_n=a_n]=1/2$$

  - se *i* possui vizinhos em preto

$$P[V_i=0 \mid V_1=a_{1,}\ldots,V_n=a_n]=1$$

Hardcore com Gibbs

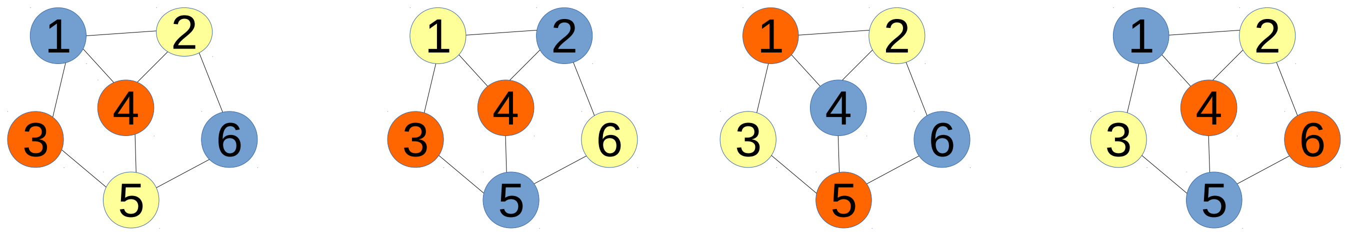
- **Algoritmo**
  - Inicia em uma configuração válida qualquer
  - Escolhe vértice *i* ao acaso, uniformemente
  - Se *i* não tem nenhum vizinho em preto
    - *i* é branco com probabilidade ½ ou preto com probabilidade ½
  - Atualiza estado (considerando *self-loops*)
  - Repete por *k* passos para gerar caminho amostral

Mais simples e eficiente que Metropolis-Hastings

- não precisa do número de vértices em preto para transicionar

q-Coloração

- Coloração de vértices em grafos
  - colorir vértices tal que vértices adjacentes tenham cores distintas
  - *q*-coloração: coloração que usa exatamente *q* cores
    - grafo geralmente tem muitas *q*-colorações distintas
- Exemplos de 3-colorações
  (amarelo = 1, azul = 2, laranja = 3)



- **Problema:** gerar uma *q*-coloração uniformemente para um grafo qualquer
  - conexão com problema de coloração mínima, que é NP-Completo

Figueiredo 2021

Gibbs Sampler para *q*-coloração

- V = vetor de vértices do grafo, K = cores 1, …, q
  - V = (V_1, V_2, …, V_n) com as cores de cada vértice
- *Lema*: dado uma coloração válida, a distribuição condicional das cores que um determinado vértice pode assumir é uniforme nas possíveis cores

$$P[V_i = c \mid V_1 = c_1, \ldots, V_n = c_n] = \frac{1}{|G(c_1, \ldots, c_n)|}$$

Conjunto de cores que *i* pode assumir dado as outras cores

- *Lema*: dado a distribuição condicional acima, cadeia de Gibbs é simétrica e aperiódica
  - Para dois vetores de coloração V^1 e V^2 que diferem em apenas uma cor, temos que $P_{V1,V2} = P_{V2,V1}$ (pelo lema)
  - Todos estados possuem self-loop, logo é aperiódica

Gibbs Sampler para *q*-coloração

- **Algoritmo**
  1) Dado X_t , uma coloração válida no tempo t
  2) Escolher vértice v uniformemente
  3) Escolher $X_{t+1}(v)$ (nova cor para v) uniformemente entre as cores válidas para v em X_t
  4) Não modificar nenhuma outra cor: $X_{t+1}(w) = X_t(w)$ para todo outro vértice w
- Gerar caminho amostral com comprimento τ_ε, e retornar amostra $X_{\tau\varepsilon}$

Tempo de Mistura

- Considere variação onde vértices são escolhidos sequencialmente (ao invés de aleatoriamente)
  - $v_1, v_2, v_3, \ldots, v_n, v_1, \ldots.$
- Considere que $q > 2d$, onde d é o grau máximo do grafo (q é número de cores), e um $\varepsilon > 0$
- Teorema: para CM definida acima, existe uma constante C (que não depende de n ou ε), tal que

$$\tau_\epsilon = C\, n \log\left(n + \log 1/\epsilon\right)$$

- CM é *fast mixing:* tempo de mistura é polylog em n (número de vértices) enquanto número de estados é exponencial em n
- Na prática, funciona bem!