Aula 13

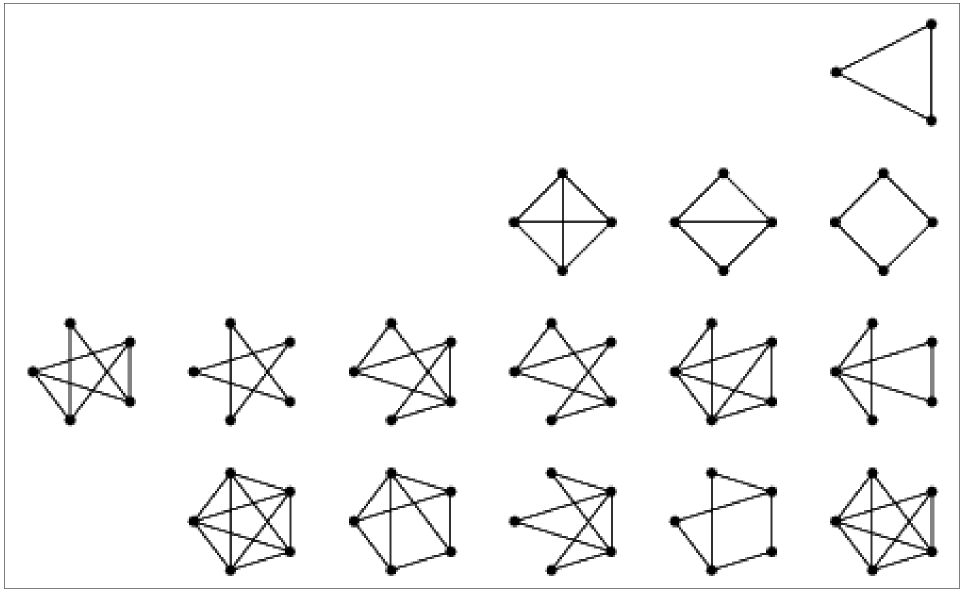
Roteiro

- Amostrando espaços complicados
- Markov Chain Monte Carlo
- Caso simétrico
- Exemplo
- Metropolis-Hastings
- Amostrando vértices (sem conhecer a rede)

Espaço Amostral de Objetos

- Considere um espaço amostral de objetos

Frutas



Grafos

Como gerar amostras destes espaços?

- amostra é um objeto

Gerando Amostras Uniformes

- **Ideia 1:**

  1) Mapear cada elemento para um número natural

  2) Escolher um número entre [1, |S|]

  3) Usar o mapeamento para retornar objeto

- Exemplo

- banana = 1, maça = 2, pera = 3, manga = 4, morango = 5, …
- escolher uniforme em [1, 19]
- retornar a fruta

- Construir mapeamento: O(|S|)
- Gerar amostra: O(1)

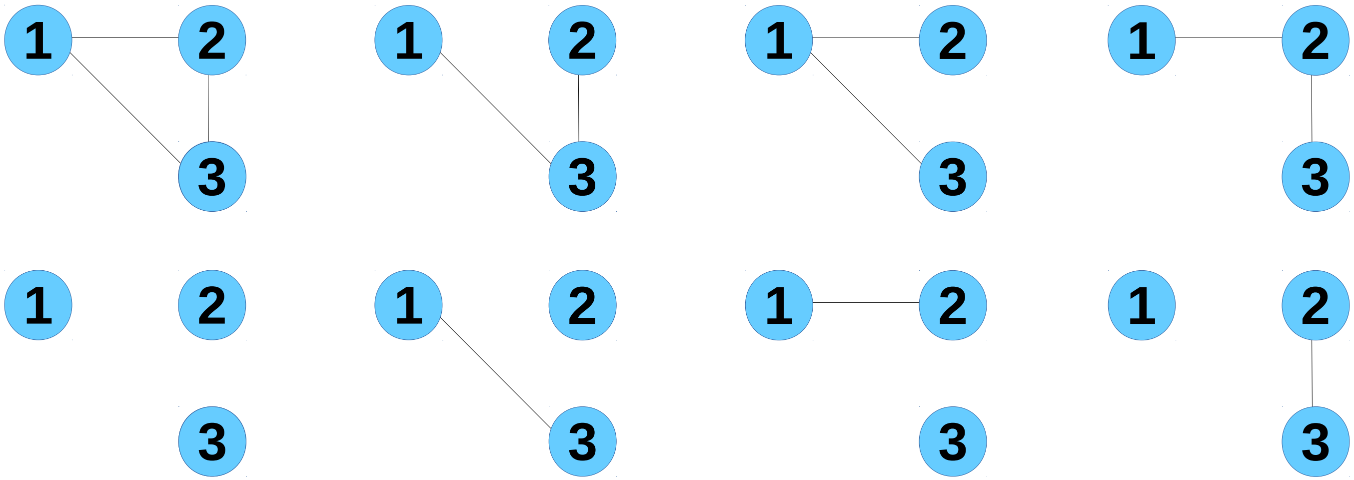
Proibitivo em alguns casos!

Gerando Amostras Uniformes

- **Ideia 2:** Construir elemento de forma iterativa

  1) sequência de escolhas aleatórias levam à escolha do elemento

- Exemplo: S = { todos os grafos com *n* vértices }
- $|S| = 2^{n(n-1)/2}$
- n =3, |S| = 8



- Se cada aresta é uma Bernoulli(½), resultado é um grafo uniforme!
- Realizar cada aresta gera grafo em tempo n(n-1)/2

Espaços Grandes e Complicados

- Considere um espaço amostral grande e complicado
  - grande = enorme quantidade de elementos
  - complicado = não é fácil construir elementos de forma iterativa
- Espaço amostral combinatorial (grande) com restrições (complicado)

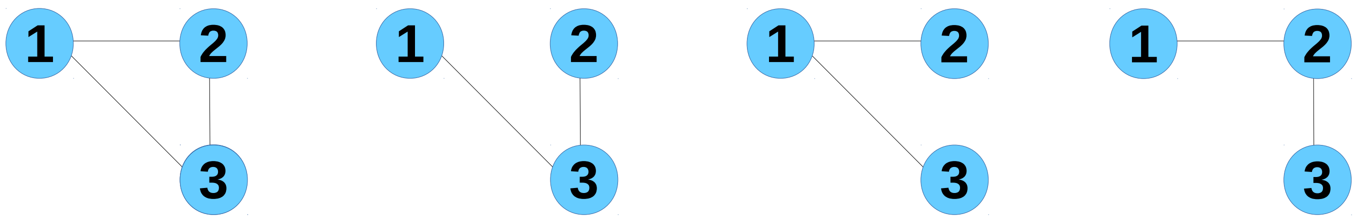
Como gerar amostras destes espaços?

- Ideia 1 e Ideia 2 não funcionam!

Exemplos

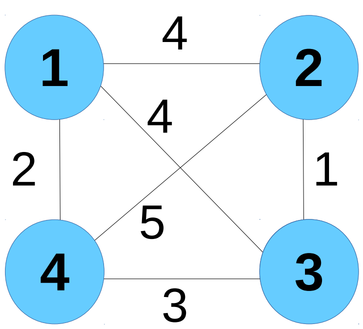
- Todos os grafos conexos com *n* vértices

n = 3,
|S| = 4



- Todos os percursos por *n* cidades de comprimento *L* ou menor

n = 4,
L = 10,
|S| = ?



Percursos e comprimento:
1,2,3,4 → 8
1,2,4,3 → 12
2,3,4,1 → 10
...

- |S| cresce exponecialmente com *n*
- Não é fácil construir amostra de forma iterativa

Gerando Amostras Uniformes

- **Ideia 3:**

  1) Aumentar espaço de estado amostral (ex. removendo restrição) para usar Ideia 2

  2) Usar *rejection sampling* para rejeitar amostras que não atendem a restrição

- Exemplo: Amostrar grafos conexos com *n* vértices

  1) gerar grafo *G* uniforme (usando Ideia 2)

  2) se G não é conexo vai para 1), se não retorna G

Problema?

- Tempo médio para gerar amostra: $1/p$, onde p é a fração de amostras que atendem a restrição
- Se p é muito baixo (ex. vai a zero com n), algoritmo é muito ineficiente

Markov Chain to the Rescue

- **Ideia 4:** Construir e simular uma cadeia de Markov
- Construir CM cujos estados correspondem aos elementos de *S* (espaço amostral)
- Construir matriz de transição *P* tal que distribuição estacionária π seja uniforme
- Simular CM para gerar amostras
  - dar τ_ε passos para gerar uma amostra

***Markov Chain Monte Carlo* (MCMC)**

- Método baseado em CM para gerar amostras de espaços arbitrários com qualquer distribuição
  - não precisa ser uniforme

MCMC

Como construir CM, se *S* é muito grande?

- Não precisamos construir a priori
- Podemos simular CM iterativamente (aula passada)

Como determinar as transições?

- Fácil descrição dos estados vizinhos a partir do estado atual
- Poucas transições de saída de cada estado (ex. log |S|)
- Baixo tempo de mistura, *fast mixing:* τ_ε <= log |S|
- Tema de muita pesquisa!

Receita para MCMC

- **Dado:** descrição do espaço amostral S, e distribuição de probabilidade sobre os estados, π
- **Saída:** uma amostra aleatória de S de acordo com π
- **Receita:**

  1) Construir CM irredutível onde cada estado corresponde a um elemento do espaço (cadeia base)

  2) Transformar cadeia base em outra CM que é reversível e possui distribuição estacionária π

  3) Simular caminho amostral longo o suficiente e retornar o estado final

     - outra amostra independente requer simular um novo caminho amostral

MCMC – Caso Simétrico

- Cadeia base tem matriz de transição *P* simétrica
  - $P_{ij} = P_{ji}$ para todo estado *i*, *j*
- Modificar *P* para construir uma nova CM que seja reversível com distribuição estacionária π
  - π é entrada para o problema
- **Ideia:** nova cadeia não aceita todas as transições da cadeia base
  - continuar no mesmo estado para induzir π
  - parecido com método da rejeição
- Aceite é probabilístico e independente para cada transição
  - $a(i, j)$: probabilidade de aceitar transição $i \rightarrow j$

MCMC – Caso Simétrico

- Considere cadeia base no estado *i* e uma “proposta” de transição para estado *j*
  - nova cadeia aceita transição com probabilidade a(*i*, *j*)
- Matriz de transição *P’* da nova cadeia

$$P'_{ij} = \begin{cases} P_{ij}\, a(i,j), & se\, i \neq j \\ 1 - \sum_{k:k\neq i} P_{ik}\, a(i,k), & se\, i = j \end{cases}$$

- Temos que escolher *a*(*i*, *j*) tq *P’* tenha distribuição estacionária dada por π, ou seja:

$$\pi_i P_{ij}\, a(i,j) = \pi_j P_{ji}\, a(j,i)$$

Garantindo também que *P’* é reversível

MCMC – Caso Simétrico

- Como $P_{ij} = P_{ji}$ (por simetria em P), temos

$$\pi_i a(i,j) = \pi_j a(j,i)$$

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

- Temos infinitas soluções para o par $a(i, j)$ e $a(j, i)$
- Queremos maximizar a probabilidade de aceite, logo

$$a(i,j) = 1, \quad se\ \pi_i \leq \pi_j$$

$$a(i,j) = \frac{\pi_j}{\pi_i}, \quad se\ \pi_i > \pi_j$$

- Ou seja,

$$a(i,j) = min\left\{1, \frac{\pi_j}{\pi_i}\right\} \text{ e } a(j,i) = min\left\{1, \frac{\pi_i}{\pi_j}\right\}$$

Exemplo

- Gerar amostras de pares ordenados (X, Y) em [1,n] x [1,n]
  - grid 2D discreto e quadrado com n^2 pontos

$$P[(x,y)] = \frac{(x+y)^2}{Z} \qquad Z = \sum_{(x,y)\in[1,n]^2} (x+y)^2$$

constante de normalização

- CM base é torus 2D com n x n vértices
  - estado dado por (x, y), com x, y = 1, …, n
  - todo estado tem 4 transições: norte, sul, leste e oeste, P_{ij} = ¼ (*i* e *j* são estados vizinhos)
  - CM é simétrica, $P_{ij} = P_{ji}$
- Transformar *P* em *P'* para induzir π conforme definição

Exemplo

- Precisamos CM tal que $\pi_{(x,y)} = \frac{(x+y)^2}{Z}$
- Seja estado i = (x,y) e j = (x',y'). Desta forma, podemos definir a(i, j) como

$$a(i,j) = min\{1, \frac{\pi_j}{\pi_i}\} = min\{1, \frac{(x'+y')^2}{(x+y)^2}\}$$

- Por fim, definimos *P'* como anteriormente
  - $P'_{ij} = P_{ij}$ a(i,j) se i != j, etc.
- **Boa notícia:** não precisamos calcular *Z* para definir a(i, j)
  - poderia ser proibitivo calcular *Z*
- Para gerar uma amostra, simular CM definida por *P'* por τ_ε e retornar o estado atual

MCMC – Caso Geral

- Considere cadeia base no estado *i* e uma "proposta" de transição para estado *j*
  - nova cadeia aceita transição com probabilidade a(*i*, *j*)
- Matriz de transição *P'* da nova cadeia

$$P'_{ij} = \begin{cases} P_{ij}\, a(i, j), & se\ i \neq j \\ 1 - \sum_{k:k\neq i} P_{ik}\, a(i, k), & se\ i = j \end{cases}$$

- Temos que escolher *a(i, j)* tal que

$$\pi_i P_{ij}\, a(i, j) = \pi_j P_{ji}\, a(j, i)$$

  - garante que π é distribuição estacionária de *P'*
  - garante que *P'* é reversível

Metropolis-Hastings

- Temos infinitas soluções para o par *a*(*i*, *j*) e *a*(*j*, *i*)
- Queremos maximizar a probabilidade de aceite, logo

$$a(i,j)=1, \quad se\, \pi_i P_{ij} \leq \pi_j P_{ji}$$

$$a(i,j)=\frac{\pi_j P_{ji}}{\pi_i P_{ij}}, \quad se\, \pi_i P_{ij} > \pi_j P_{ji}$$

- Ou seja,

$$a(i,j)=min\{1,\frac{\pi_j P_{ji}}{\pi_i P_{ij}}\}$$

- *P'* definida com essa probabilidade de aceite é chamada de cadeia de Metropolis-Hastings
  - dando origem ao algoritmo de Metropolis-Hastings

Metropolis-Hastings

- Mostrando que definição de *a(i, j)* satisfaz a condição
- Assumir que $P_{ij} > 0$ se e somente se $P_{ji} > 0$
- Mostrar que $a(i,j) = min\{1, \frac{\pi_j P_{ji}}{\pi_i P_{ij}}\}$
- Satisfaz $\pi_i P_{ij} a(i,j) = \pi_j P_{ji} a(j,i)$ para todo *i, j*
- Caso *a(i, j)* = 1. Neste caso, temos que $\pi_i P_{ij} \leq \pi_j P_{ji}$
  - Então
$$a(j,i) = min\{1, \frac{\pi_i P_{ij}}{\pi_j P_{ji}}\} = \frac{\pi_i P_{ij}}{\pi_j P_{ji}}$$
  - Consequentemente
$$\pi_i P_{ij} = \pi_j P_{ji} a(j,i) = \pi_j P_{ji} \frac{\pi_i P_{ij}}{\pi_j P_{ji}} = \pi_i P_{ij}$$
- O outro caso é análogo

Amostrando Vértices

- Considere grafo grande e desconhecido (não conhecemos vértices ou arestas, a priori)
- Podemos percorrer o grafo: a partir de um vértice, descobrir seus vizinhos
  - ex. rede de amizades do Facebook
- Como gerar amostras de vértices uniformemente?
  - ex. estimar fração de brasileiros no FB
- **Ideia 1:** BFS de raio k, amostrar uniforme nos vértices descobertos
- **Ideia 2:** Passeio aleatório de comprimento k, retornar vértice X_k

Como garantir que amostra é uniforme?

- Ideias 1 e 2 geram amostras enviesadas (não uniformes)

MCMC to the Rescue

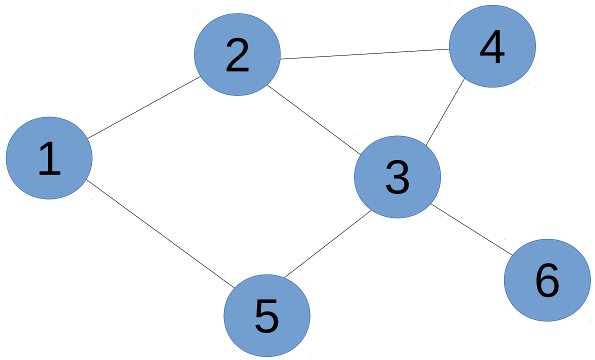
- Construir cadeia de Metropolis-Hasting tq π seja uniforme nos vértices
  - $\pi_v = 1/Z$ onde Z é o número de vértices da rede (desconhecido)
- Cadeia base: passeio aleatório simples
- Definindo probabilidade de aceite
  - CM base, temos: $P_{ij} = 1/d_i$, $P_{ji} = 1/d_j$

$$a(i,j) = min\left\{1, \frac{\pi_j P_{ji}}{\pi_i P_{ij}}\right\} = min\left\{1, \frac{d_i}{d_j}\right\}$$

- Logo $P'_{ij} = 1/d_i * \min\{1, d_i / d_j\}$
- Enviesa o passeio contra vértices de grau alto, adicionando *self-loops*, tornando a distribuição estacionária uniforme

Exemplo

- Grafo



- Matriz de transição do passeio aleatório

$$P=\begin{pmatrix} 0 & 1/2 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 & 1/4 & 1/4 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

- Transformação

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

- Cadeia de Metropolis-Hastings com distribuição estac. uniforme

$$P'=\begin{pmatrix} 1/6 & 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 1/12 & 1/4 & 1/3 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 & 1/4 & 1/4 \\ 0 & 1/3 & 1/4 & 5/12 & 0 & 0 \\ 1/2 & 0 & 1/4 & 0 & 1/4 & 0 \\ 0 & 0 & 1/4 & 0 & 0 & 5/4 \end{pmatrix}$$

Simulação Iterativa

- Mas não conhecemos o grafo!
- Simular caminho amostral iterativamente, a cada passo:
  - descobrir os vizinhos do vértice atual
  - descobrir grau de cada vizinho do vértice atual
  - determinar as probabilidades de transição para cada vizinho (e também *self-loop*)

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

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

  - fazer escolha aleatória, atualizar vértice atual, e repetir

- Método considerado *estado-da-arte* e usado na prática!