Aula 16

Aula passada

- Metropolis-Hasting
- Amostrando vértices sem conhecer a rede
- Modelo Hardcore

Aula de hoje

- Otimização
- Caixeiro viajante
- *Hill Climbing*
- Distribuição de Boltzman
- Simulated Annealing
- De volta ao caixeiro

Otimização

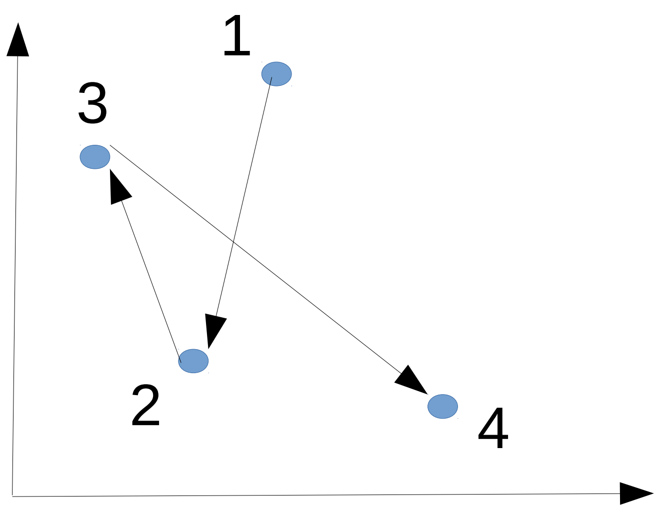
- Considere conjunto *S* e função *f* que avalia cada elemento, $f : S \rightarrow R$
- **Problema:** encontrar elemento(s) de *S* que minimizam (ou maximizam) *f*

$$S^o = \{ e \mid e = arg\, min_{s \in S} f(s) \}$$

- **Ideia 0:** enumerar elementos de *S*, encontrar valores mínimos
  - funciona apenas se |S| é pequeno (ex. 10^9)
- **Ideia 1:** criar uma *estrutura* sobre os elementos de *S*, procurar utilizando a estrutura
  - estrutura é um grafo, vértices são elementos de S

Caixeiro Viajante

- Considere *n* pontos no plano
  - $v_i = (x_i, y_i)$ → coordenadas de cada ponto
- Considere a distância euclideana entre pares de pontos
- **Problema:** encontrar percurso com menor comprimento
  - percurso é permutação dos pontos
- S = todas as permutações; f = soma das distância entre pares consecutivos na permutação
- Exemplo: $n = 4$

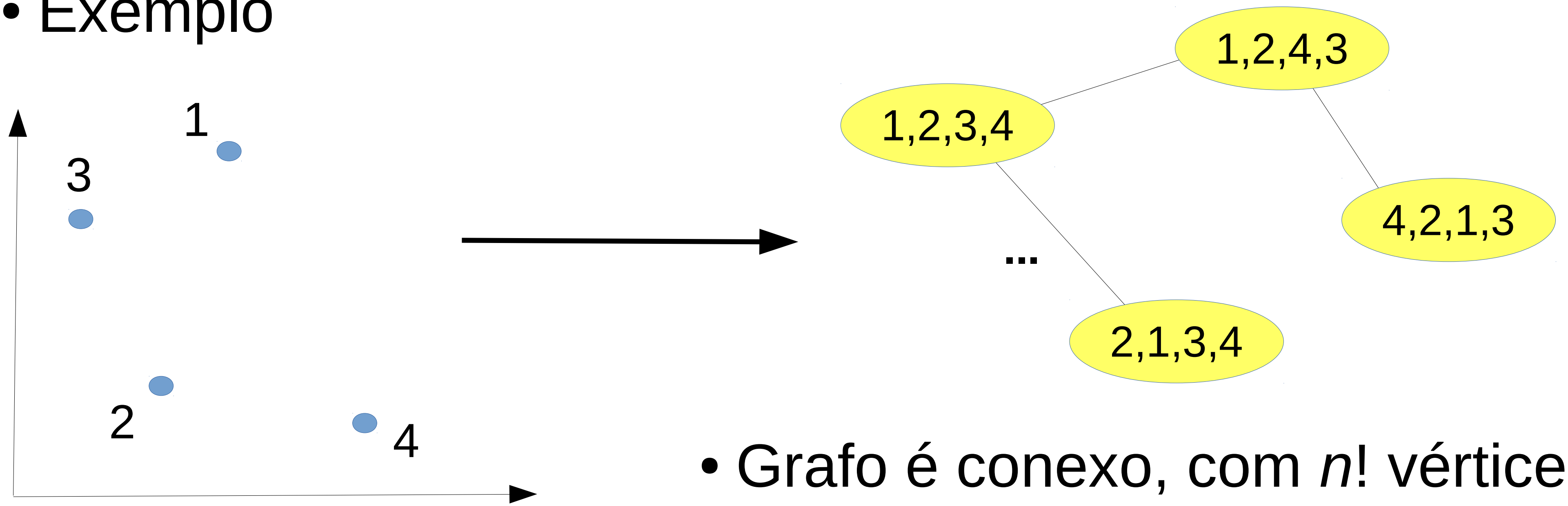


- $P_1 = 1,2,3,4$ $f(P_1) = d(1,2)+d(2,3)+d(3,4)$
- $P_2 = 4,2,1,3$ $f(P_2) = d(4,2)+d(2,1)+d(1,3)$
- $P_3 = 1,4,3,2$
- Qual é o menor percurso?
  - número de percursos é $n!$

Figueiredo 2018

Grafo do Caixeiro Viajante

- Cada permutação é um vértice do grafo
- Aresta entre P_i e P_j sse P_i e P_j diferem em apenas um par de elementos
  - uma única troca entre as duas permutações
- Exemplo



- Grafo é conexo, com n! vértices
- Cada vértice do grafo define um custo: $f(v)$

Explorando o Grafo

- Como explorar o grafo
  em busca do ótimo?

Algoritmo guloso

1) Começar em um vértice qualquer
2) Avaliar qualidade de todos os vizinhos
3) Transicionar para vizinho de menor valor
4) Repetir enquanto puder

- Algoritmo conhecido como *Hill Climbing*
  - sobe (desce) gulosamente pela montanha
- **Problema:** mínimos (ou máximos) locais
  - vértice cujos vizinhos são todos superiores

MCMC to the Rescue

- **Ideias:** Não ser tão guloso assim; usar aleatoriedade para controlar a gula!
  - permite transicionar para vizinho superior a outros, e também superior a vértice atual

- Construir uma CM finita, irredutível e aperiódica
  - garante que todos os estados serão visitados
- Mas com qual distribuição estacionária?

π_v inversamente proporcional a $f(v)$

- vértices com menor valor serão visitados com maior probabilidade
- Usar *Metropolis-Hasting* para construir CM com π_s

Distribuição de Boltzman

- Conjunto *S* e função *f* que avalia cada elemento
- Considere parâmetro *T* > 0, chamado de temperatura
- Probabilidade associado ao elemento *s* dada por

$$\pi_s = \frac{e^{-\frac{f(s)}{T}}}{Z} \qquad Z = \sum_{s \in S} e^{-\frac{f(s)}{T}}$$

Constante de normalização

- Para um *T* fixo, menor *f(s)* maior π_s, exponencialmente
  - valores menores de *f*(*s*) tem probabilidade bem maiores
- Para problema de maximização, trocar sinal da exponencial

Distribuição de Boltzman

- O que acontece quando T é muito pequeno?

$$\pi_s = \frac{e^{-\frac{f(s)}{T}}}{Z}$$

Aumenta as diferenças entre π_v

- Mínimo se destaca mais

- Seja $\alpha(T)$ a probabilidade de um elemento X escolhido aleatoriamente com probabilidade π_s ser mínimo
- Então temos $\lim_{T \to 0} \alpha(T) = 1$

- Boa notícia: probabilidade de escolher algum elemento mínimo vai a um
  - Amostrar CM usando π_s como distribuição estacionária

Figueiredo 2018

Simulated Annealing

- Técnica para construir uma sequência de CM para resolver problema de máximo/mínimo global
  - usando mecânica de *Metropolis-Hasting* e distribuição de *Boltzman* com diferentes valores para T
- Escolher $T_1 > T_2 > T_3$ … com $T_i \to 0$ quando i → infinito
- Escolher N_1, N_2, N_3, …. e vértice inicial qualquer
- Amostrar CM com T_1 por N_1 passos, depois mudar para T_2 e amostrar por N_2 passos, e assim por diante
  - (T_1, N_1), (T_2, N_2), ... chamado de *annealing* (*agenda de resfriamento*)
- **Ideia:** “resfriar” enquanto gera amostras, ficando mais difícil voltar para valores mais longes de ótimo

Simulated Annealing

- $(T_1, N_1), (T_2, N_2), \ldots$ chamado de *annealing*
- Escolha do *annealing* é fundamental para garantir convergência correta do método
  - se resfriar muito rápido, CM pode ficar presa em mínimo local
- **Teorema:** se T_i decresce devagar o suficiente, então P[X_t ser ótimo] $\rightarrow$ 1 quando $t \rightarrow$ infinito
- **Problemas**

  (1) devagar o suficiente depende do problema em questão

  (2) devagar o suficiente pode ser muito lento para ser usado na prática

Estratégias de Resfriamento

- Como T deve decrescer com o número de passos na CM?
- **Ideia:** definir temperatura para cada passo
  - N_i = 1 para todo i
- $T(t)$: temperatura a ser usada no passo t
- Funções geralmente usadas, para um $T_0 > 0$ e $0 < \beta < 1$

$$T(t) = T_0 \beta^t$$

Exponencial

$$T(t) = T_0 - \beta t$$

Linear

$$T(t) = \frac{a}{\log(t+b)}$$

Logarítmico

- Prova se convergência global se a for grande o suficiente e b constante
- na prática, muito lento (t tem que crescer muito)

Estratégias de Resfriamento

- Estratégia de resfriamento no exemplo anteriores era fixa
  - não depende das amostras geradas
- **Ideia:** Estratégia de resfriamento adaptativa (dinâmica)
  - usa valores das amostras para resfriar
  - ex. usar diferença $f(X_t) – f(X_{t-1})$, variação em T ser proporcional a diferença
- Adiciona outra camada de complexidade
- Na prática, pode funcionar bem
  - sem muitas garantias teóricas

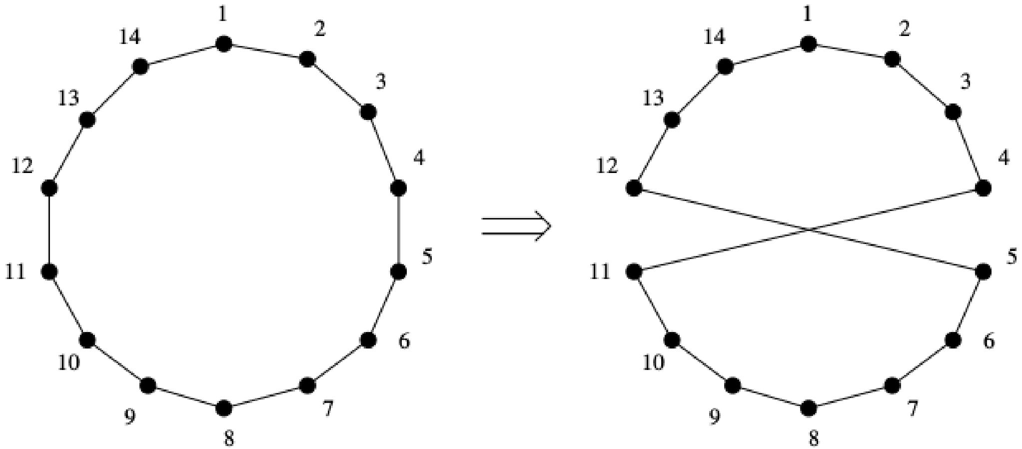
Voltando ao Caixeiro Viajante

- Considerar permutações como vértices da CM
- Transição entre permutações: inverter parte da permutação
  - escolher i < j em [1, n], inverter permutação atual entre os índices i e j
- Exemplo

$1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \xrightarrow{i\,=\,3,\ j\,=\,6} 1\ 2\ 6\ 5\ 4\ 3\ 7\ 8\ 9$

$8\ 1\ 7\ 2\ 5\ 9\ 7\ 3\ 4 \xrightarrow{i\,=\,2,\ j\,=\,4} 8\ 2\ 7\ 1\ 5\ 9\ 7\ 3\ 4$

- Objetivo é redefinir (inverter) pares de arestas



- Exemplo com n=11, i=5, j=11

Figueiredo 2018

Voltando ao Caixeiro Viajante

- Escolher *i* e *j* uniformemente, tal que i < j
  - P[(i, j)] = 1/(n(n-1)/2) → combinação de *n* dois a dois
- Definir π_s de acordo com distribuição de Boltzman
- Montar CM via Metropolis-Hasting, com $T > 0$
- Probabilidade de transição de *s* para *s'*
  - CM original é simétrica (grafo é regular)
  - (i, j) definem a transição (define quem é *s'*)

$$P_{s,s'} = \frac{2}{n(n-1)} \min\{e^{\frac{f(s)-f(s')}{T}}, 1\} \quad , \quad \text{se s != s'}$$

- Se *f*(*s'*) for menor, então aceita com probabilidade 1
  - escolha uniforme entre os *s'* melhores que *s*
- Se *f*(*s'*) for maior, aceita com probabilidade que diminui com a diferença

Figueiredo 2018

Voltando ao Caixeiro Viajante

- Como definir agenda de resfriamento?
  - $(T_1, N_1), (T_2, N_2), (T_3, N_3), \ldots$
- Não temos muita teoria para isto (infelizmente)
  - usar uma das estratégias anteriores
- Na prática, tentativa e erro usando experiência adquirida
  - começar com instâncias pequenas
- Exemplo interativo na web: http://toddwschneider.com/posts/traveling-salesman-with-simulated-annealing-r-and-shiny/

Ainda é tema de muita pequisa em diferentes áreas (física, computação, matemática, engenharia, etc)