

Redes Complexas

Aula 12

Roteiro

- Busca em redes
- Busca com informação
- Navigabilidade em redes
- Modelo de Kleinberg
- Exemplo real

Aula passada

- Configuration Model
- Propriedades
- Stochastic Block Model (SBM)
- Propriedades

Busca em Redes

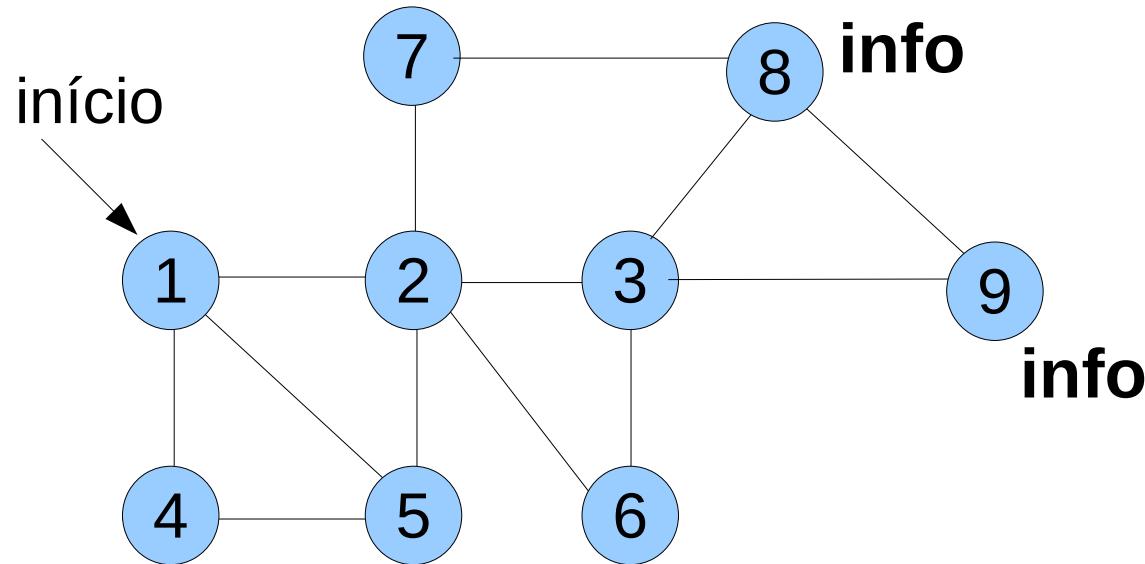
- Vértices possuem informação
 - conteúdo das páginas web
 - arquivos em uma rede P2P
 - conhecimento em uma rede social
- Informação distribuída
 - não existe repositório centralizado de quem conhece o que
- **Problema:** como localizar a informação?



Algoritmos de busca em redes!

Busca em Redes

- Vértices possuem informações
 - possivelmente replicadas na rede
- Um vértice deseja localizar uma informação
 - ponto de entrada na rede
- Utiliza as arestas para navegar pela rede
 - primitivas de perguntar e repassar aos vizinhos



BFS (*Breadth First Search*)

- BFS = busca em largura
- Iniciado pelo vértice fazendo a busca
 - busca tem identificador único na rede
- Ao receber a busca pela primeira vez, se vértice não possui informação, repassa a **todos** vizinhos
 - se possui informação, responde ao vértice inicial
- **Vantagem**
 - Rápido! Percorre caminho mais curto até informação
- **Desvantagem**
 - Pesado! Todos os vértices serão atingidos
 - Não há como interromper (eficientemente) o processo de busca

Flooding Limitado

- Busca em largura também chamado de *flooding* (inundação)

Como controlar a carga?

- **Ideia:** usar um TTL (*Time-to-live*) na busca
 - maior número de saltos que busca pode fazer
 - informação contida na mensagem de busca, decrementada a cada salto, vértice não repassa quando contador é zero
- Problemas: busca pode não chegar ao vértice com a informação (TTL pequeno)
 - tradeoff entre carga e taxa de sucesso

Random Walk

- Passeio aleatório iniciado pelo vértice que inicia a busca
 - cada passeio tem um identificador único na rede
- Ao receber a busca, se vértice não possui a informação, repassa a **um** vizinho escolhido aleatoriamente
 - se possui informação, responde ao vértice inicial
- **Vantagem**
 - Leve! Busca se propaga de vértice em vértice
- **Desvantagem**
 - Lento! Pode demorar para encontrar informação

Random Walk 2

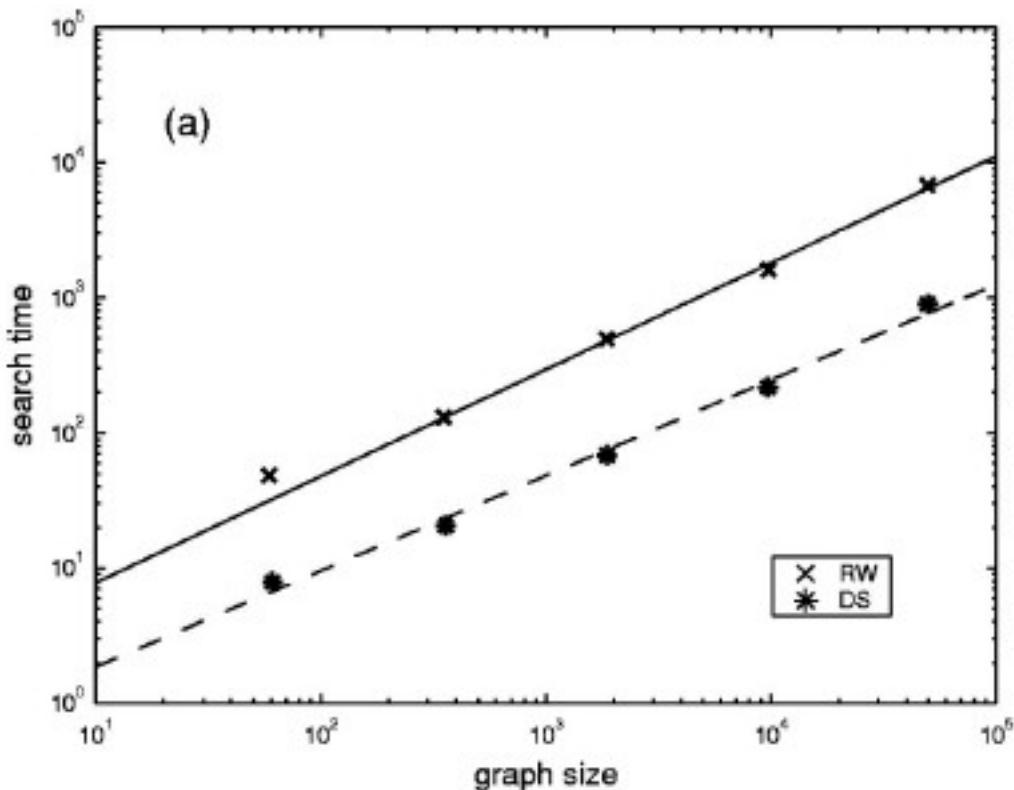
- **Problema:** passeio pode estar ao *lado* da informação e não encontrá-la
 - informação está em um vizinho, mas outro é escolhido ao acaso
- **Solução:** perguntar a todos os vizinhos antes de repassar a busca
 - se algum vizinho tem informação, então para
- Diminui o tempo de busca na prática, mas não na teoria
 - vantagens dependem da estrutura da rede

Explorando a Estrutura

- Como explorar a estrutura da rede para fazer buscas mais eficientes?
- **Idéia:** Propagar a busca para vizinho de maior grau (e não aleatoriamente)
 - vértice de maior grau tem acesso a mais informação
 - direcionar busca para quem sabe mais
- Enviar busca para próximo vizinho de maior grau (em caso de revisita)
- *High-Degree Seeking Walks*
 - vantagens dependem da rede

Avaliação (1/2)

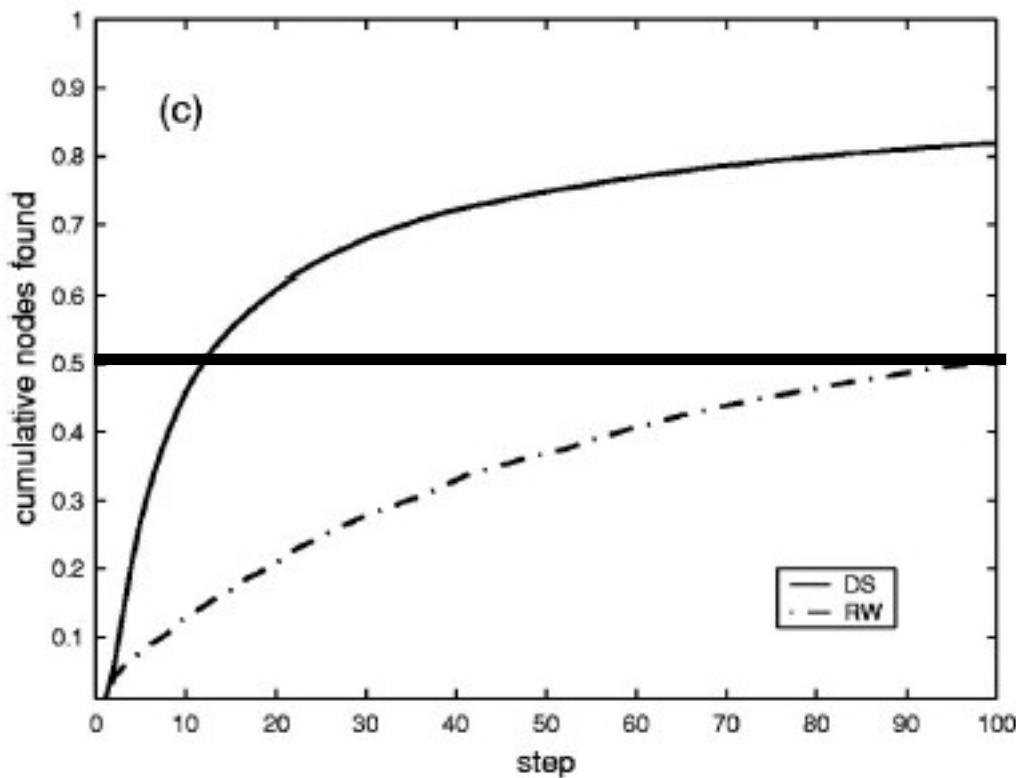
- Rede com lei de potência no grau ($\alpha=2.1$)
- Vértice de origem/destino da busca escolhidos aleatoriamente
- Busca termina quando chega a um vizinho do destino



- Tempo médio para encontrar destino
- Medido em número de passos
- 10 vezes mais rápido que RW clássico (para qualquer n)

Avaliação (2/2)

- “Alcance” da busca: fração de vértices pesquisados em função do número de passos



- $n = 10000$
- DS: metade da rede pesquisada em apenas 10 passos
- RW: 100 passos!
- Problema: carga de busca nos vértices não é uniforme!

Busca com Informação

- Algoritmo de busca anteriores eram cegos
 - não havia informação local sobre onde encontrar o destino
 - ex. qual melhor vizinho para enviar a busca?
- Em muitas redes, temos informação local sobre o destino
 - ex. coordenadas do destino e coordenadas de cada vértice da rede
- Busca pode explorar a informação local

Como explorar informação local?

Experimento de Milgram

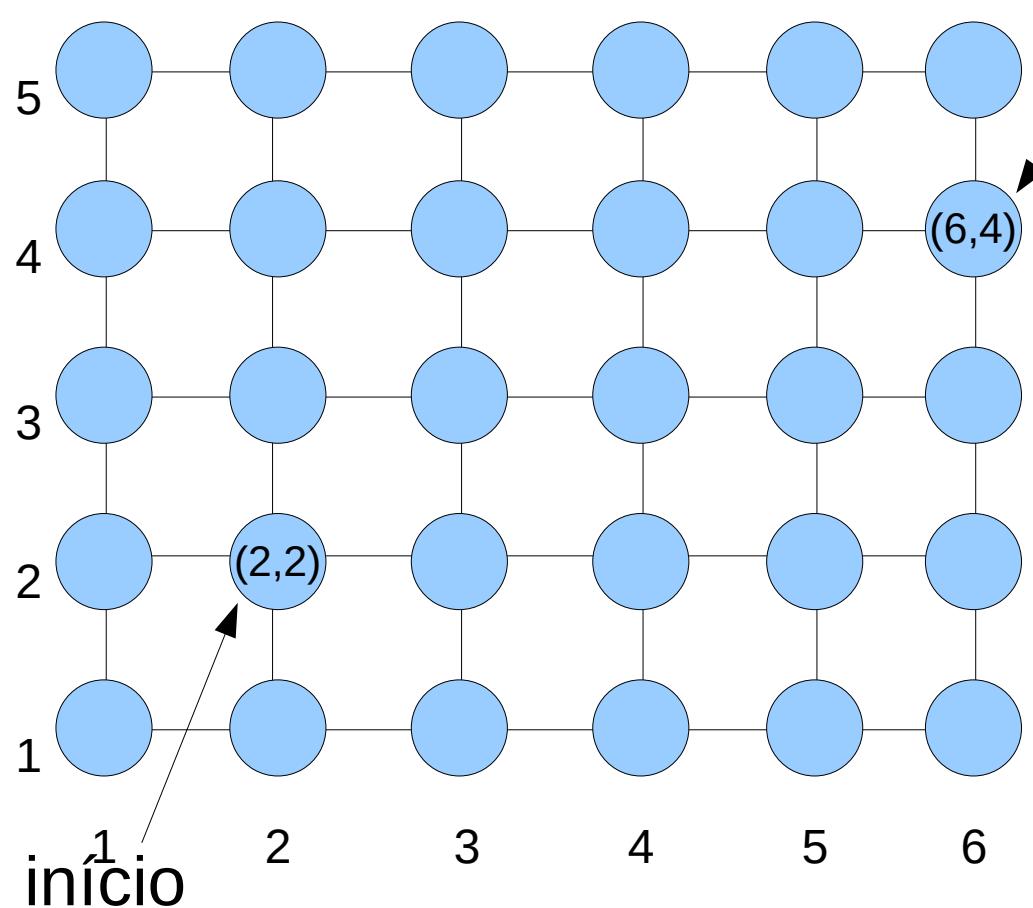
- Enviar carta a um desconhecido através da rede social
- **Surpreendente:** Caminhos curtos existem
 - ~ 6 passos na rede social
- **Igualmente surpreendente:** Pessoas comuns encontraram caminhos curtos
 - busca aleatória na rede social levariam a caminhos muito mais longos
- Pessoas navegam bem a rede social
 - mesmo sem terem uma visão global da rede social

Algoritmo de Busca Guloso

- Considere função (heurística) que avalia (estima) a distância (na rede) entre dois vértices
 - usando atributos dos vértices
- Algoritmo de busca
 - a cada passo, busca é enviada para vizinho **mais próximo** do destino
- Algoritmo é míope e guloso
 - considera apenas próximo passo
 - tenta se aproximar o máximo do destino a cada passo
- Função de distância depende do contexto
 - distância física, distância social, etc
- Similar ao usado por pessoas para navegar na rede social real

Exemplo

- Rede é um grid em 2D, atributo é a coordenada (x, y) do vértice
- Distância de Manhattan no grid (em saltos)



- Vértice inicial conhece coordenadas do destino
- Vértices conhecem coordenadas dos vizinhos
- Escolhe vizinho que está mais próximo do destino
- Passa busca para vértice vizinho (que repete)

- **Eficiente:** número de passos é a distância!

Desempenho do Guloso

- Desempenho do algoritmo depende da estrutura da rede
- Guloso pode **não encontrar** caminho mais curto na rede
 - mesmo quando caminho curto existir
- Estrutura da rede é fundamental para desempenho do algoritmo
 - caminhos de comprimento logarítmicos
- Observação e demonstração feita por Kleinberg em artigo influente

Modelo de Kleinberg

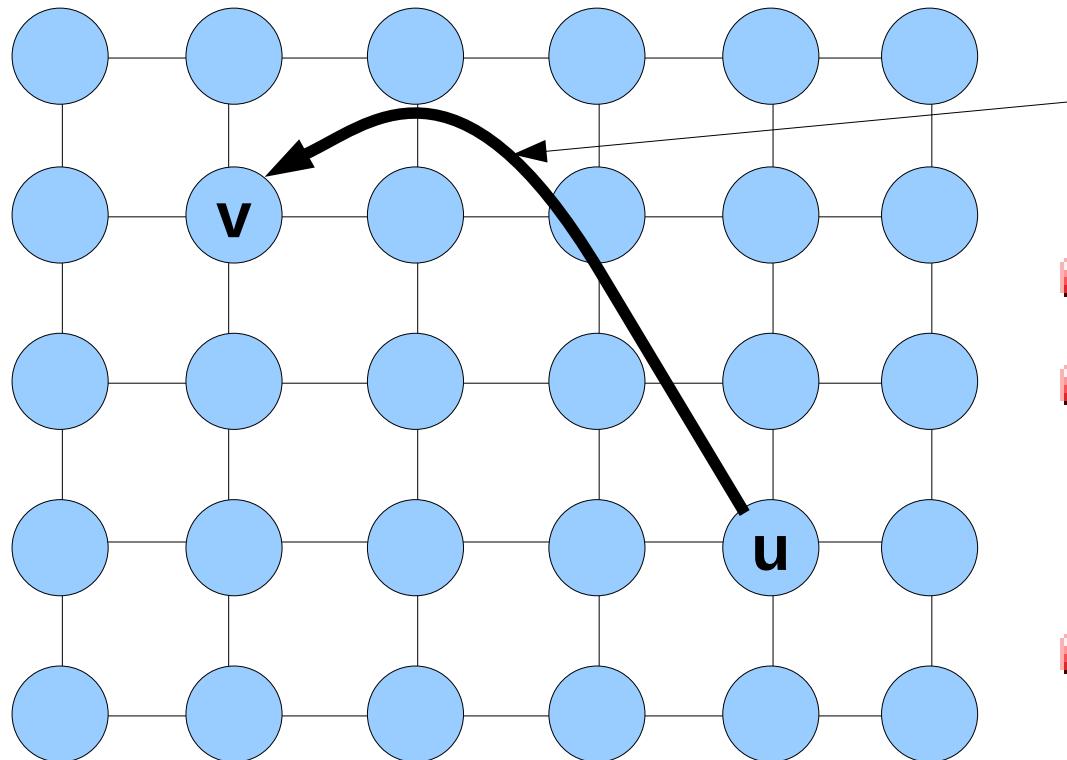
- Rede é um grid em 2D, atributo é a coordenada (x,y) do vértice
- Distância de Manhattan no grid (em saltos)
- Adicionar atalhos no grid
 - criação de caminhos curtos na rede
 - cada vértice adiciona 1 atalho
- Atalhos aleatórios
 - probabilidade inversamente proporcional a distância entre os vértices no grid

$$p_{uv} \propto \frac{1}{d(u, v)^\alpha}$$

, onde $d(u, v)$ é a distância entre os vértices u e v no grid, α é constante (parâmetro)

Modelo de Kleinberg

- Rede: grid em 2D + atalhos probabilísticos



Atalho ocorre com probabilidade $\sim 5^{-\alpha}$

- Atalho é função de α
- α grande: baixa probabilidade de atalhos longos
- α pequeno: atalhos muito longos podem ocorrer

- **Problema:** Calcular desempenho do algoritmo guloso em função de α

Modelo de Kleinberg

- Gerar atalhos no grid
 - atalhos são vizinhos e podem ser usados
- Escolher dois vértices aleatoriamente
- Executar algoritmo guloso
 - sempre termina pois pode andar pelo grid
- Medir número de passos
 - aleatório, pois depende dos atalhos e par origem/destino
- Considerar número médio de passos em função de n (tamanho da rede)
 - limitante inferior e superior para tempo médio

Avaliação Teórica

- $0 \leq \alpha < 2 \longrightarrow T \geq c n^{(2-\alpha)/3}$

- $\alpha > 2 \longrightarrow T \geq c n^{(\alpha-2)/(\alpha-1)}$

Tempo médio é polinomial

- $\alpha = 2 \longrightarrow T \leq c \log^2 n$

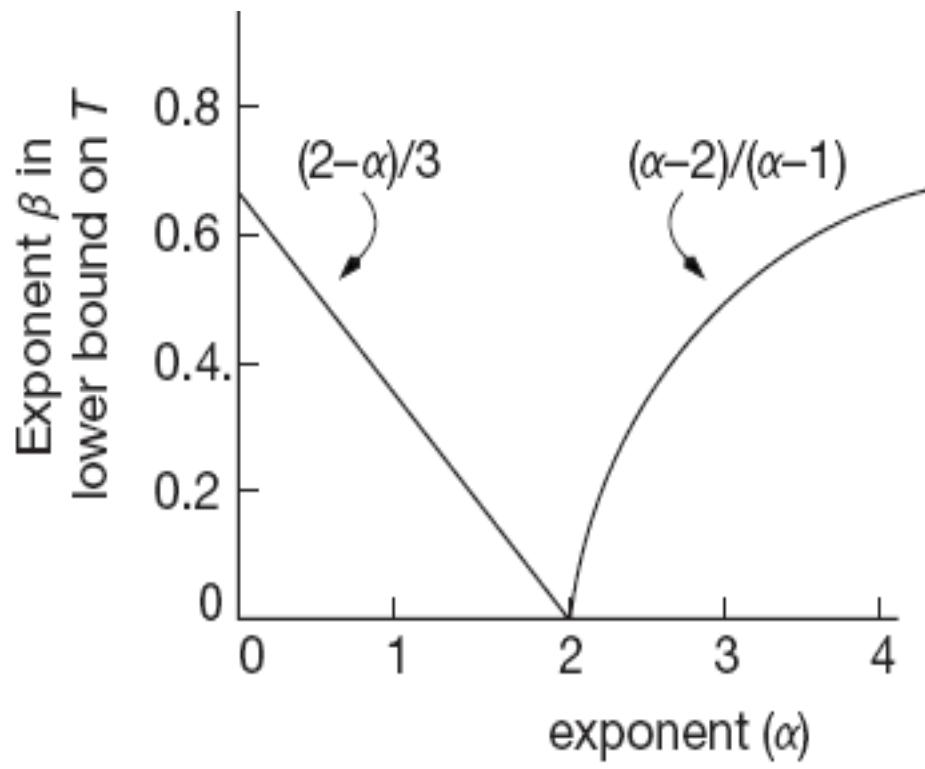
Tempo médio é poly-log

- Exponencialmente menor que tamanho da rede

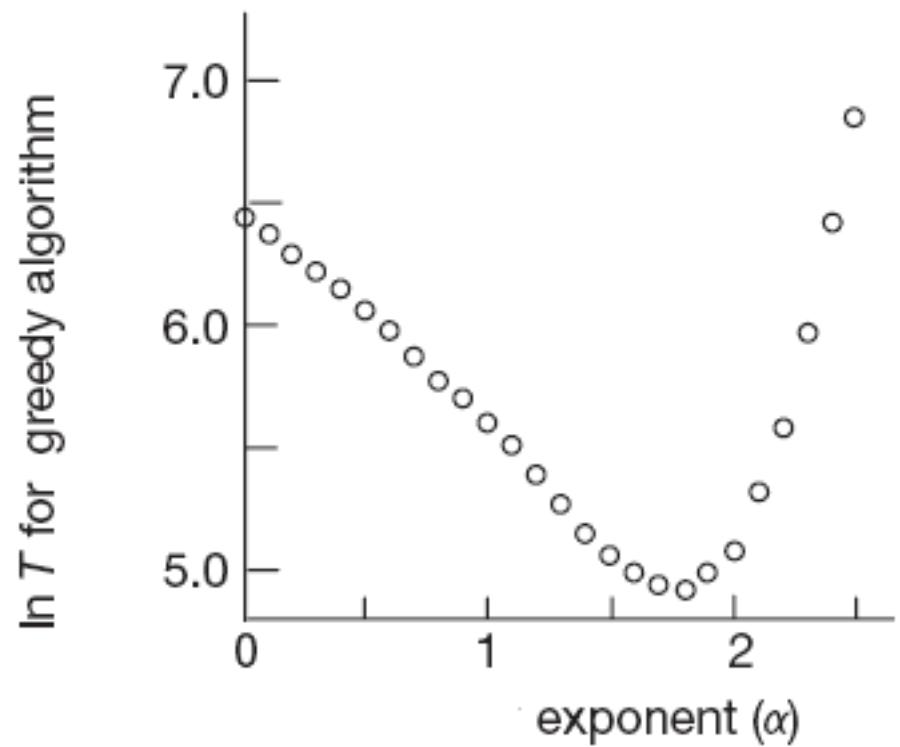
Avaliação

- Gráfico do expoente β

$$T \geq c n^\beta$$



- Avaliação empírica (simulação), $n = 20K$, 1K rodadas, $\log T$



Observações

- $\alpha = 2$: vértices tem em média o mesmo número de vizinhos em todas as distâncias
 - efeitos se cancelam
- $\alpha < 2$: distribuição de atalhos muito uniforme
 - algoritmo não consegue explorar os atalhos
- $\alpha > 2$: não há atalhos longo suficientes
 - não existem caminhos curtos o suficiente

Generalizações

- Generalizado para qualquer número constante de atalhos
 - não muda complexidade
- Generalizado para qualquer algoritmo que utiliza apenas informação local
 - guloso que vimos não usa o passado
- Generalizado para grids com d dimensões
- Ponto crítico ocorre quando $\alpha = d$
 - guloso encontra caminhos poly-log apenas nestes casos
 - mesma intuição que $d=2$

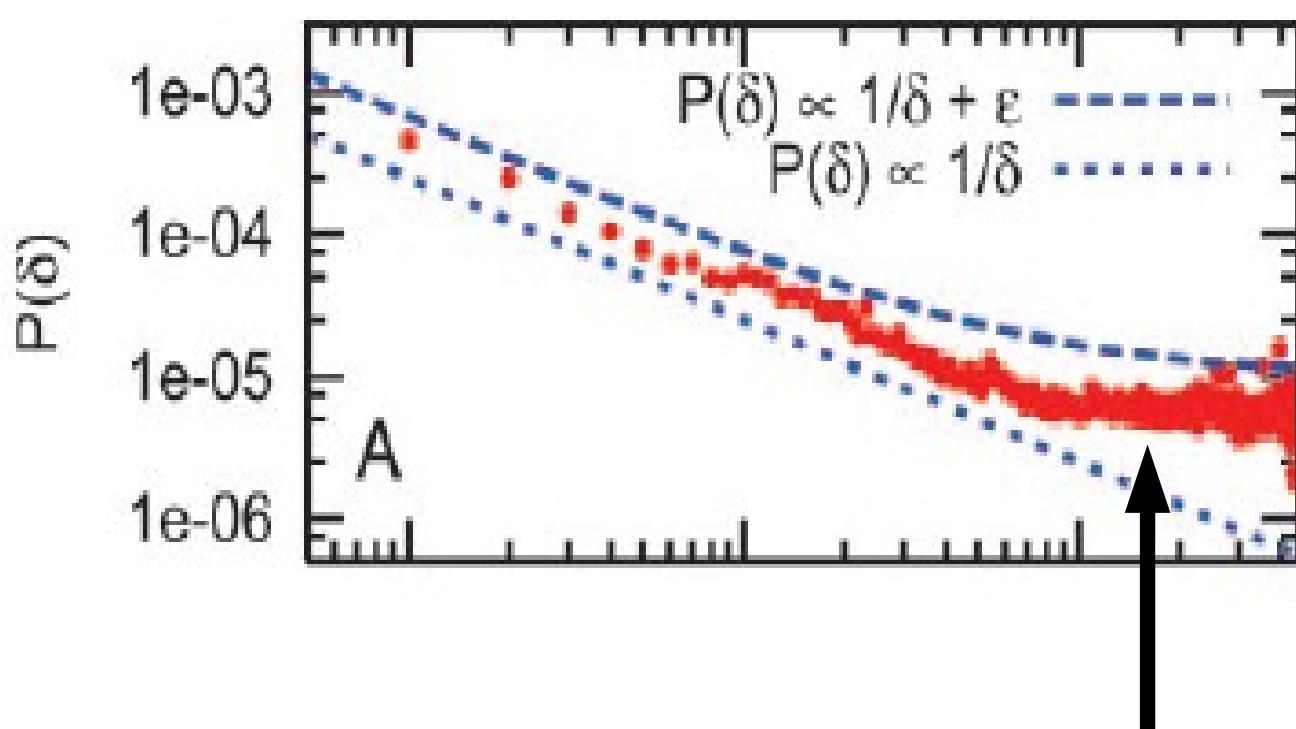
Distâncias e Redes Sociais

- Amizades estão relacionadas com distâncias geográficas?
 - ex. distância entre as cidades onde moram as pessoas
- Estudo empírico de rede social online (2004)
- LiveJournal
 - blogueiros tem identidade, endereço
 - declaram amizade por outros blogueiros
- ~500K usuários que declararam endereço válido nos EUA



Distâncias entre Amigos

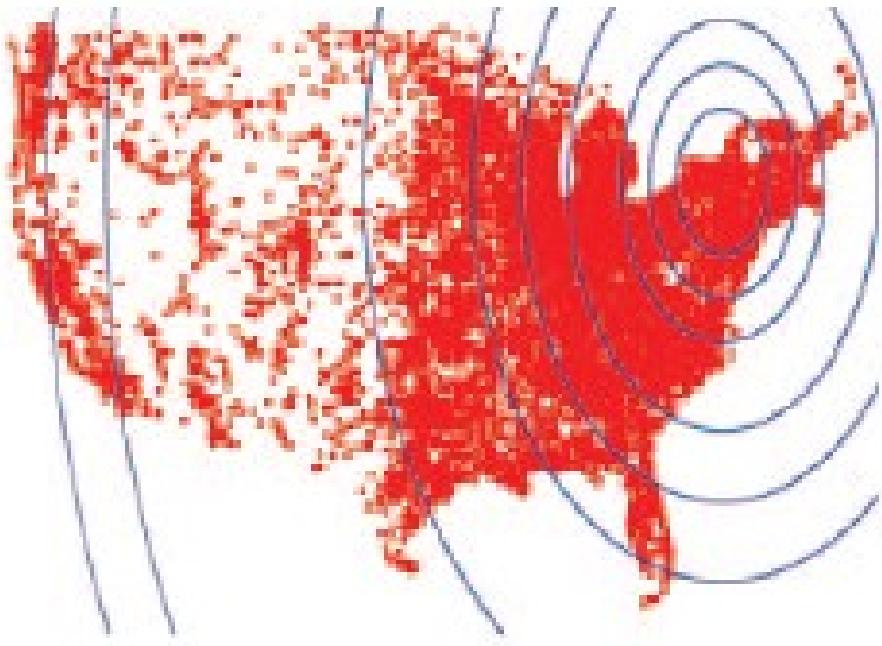
- Distância entre vizinhos (cada aresta)
 - usando CEP do endereço declarado
- $d(u,v) = \delta$, arredondado para múltiplo de 10Km
- $P(\delta)$: fração de arestas com distância δ



- Segue lei de potência até certo ponto
- Não influencia para distâncias muito grandes
- $P(\delta)$ converge para 5×10^{-6}

Densidade Populacional

- Qual relação entre tamanho da vizinhança (em pessoas) e distância geográfica?
- Cada ponto representa uma pessoa (*blog*)



- Círculos concêntricos, centro Ithaca, NY
- Cada círculo representa 50K pessoas
- Densidade populacional influencia amizades

Amizades e Navegação

- Ex. duas pessoas u e v moram a 300m
 - Em Xapuri, certamente são conhecidos
 - Em Copacabana, muito pouco provável
- Probabilidade de amizade depende de distância física e densidade populacional
- Modelo para capturar este efeito
 - extensão de modelo de Kleinberg
- Desempenho do algoritmo guloso é bom (poly-log)

“In a lamentably imperfect world, it is remarkable that people form friendships so close to the perfect distribution for navigating their social structures.”