Busca em Grafos de Grande Dimensão em Aplicações Responsivas

Leonardo C. Monteiro
PPGI/DCC-IM
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brasil
leonardo.monteiro@ufrj.br

***Resumo* – O objetivo desse trabalho é apresentar as principais estratégias para viabilizar busca em grafos de grande dimensão para o cálculo de distâncias mínimas em aplicações responsivas. Caso sejam toleradas pequenas margens de erro no cálculo das distâncias mínimas, aplicações responsivas podem pré-calcular esses valores apenas para um subconjunto de nós especiais, denominados nós *landmarks*, resultando em grande economia de processamento e armazenamento.**

Palavras-chave – Algoritmos de busca; aplicações responsivas; grafos

I. INTRODUÇÃO

A. Aplicações Responsivas

De acordo com o dicionário Merriam-Webster da língua inglesa, a palavra "*responsive*" significa: "*Quick to respond or react appropriately or sympathetically*". Neste sentido, podemos dizer que uma aplicação responsiva é aquela que responde adequadamente ao usuário em tempo hábil, na faixa dos segundos.

O principal exemplo de aplicação responsiva moderna é o Google Search, que em poucos segundos consegue responder adequadamente, com boas respostas, às consultas dos usuários, apesar de ter que manipular uma grande base de dados (vários índices para praticamente todas as páginas da Web [2]).

B. Grafos de Grande Dimensão

De acordo com [1], em termos práticos, considera-se como um grafo de grande dimensão aquele em que a matriz de distância entre todos os seus pares de nós não pode ser armazenada em memória RAM. Considera-se ainda que os algoritmos de cálculo de distância possuem complexidade quadrática de espaço e tempo de computação.

Exemplos típicos de grafos de grande dimensão incluem redes sociais, redes biológicas, redes de comunicação e grafos de páginas web. Esses grafos podem possuir facilmente milhões de nós e possivelmente centenas de milhões de arestas.

Alguns exemplos de conjuntos de dados que podem formar grafos de grande dimensão são apresentados na Tabela I.

C. Objetivo do Trabalho

Algumas aplicações responsivas precisam trabalhar com grandes conjuntos de dados organizados sob a forma de grafo. Muitas vezes, o principal problema ao lidar com esse tipo de estrutura é realizar buscas não-informadas entre seus nós para o cálculo de menor distância. Que podem quantificar, por exemplo, relações de confiança entre consumidores em aplicações de marketing.

TABELA I. GRAFOS DE GRANDE DIMENSÃO [1]

Conjunto de Dados	Nós	Arestas	Caminho Médio
CA-HEPPH	11,2K	235K	4,66
GOOGLENW	15,7K	297K	2,46
CA-CONDMAT	21,3K	182K	5,47
CIT-HEPTH	27,4K	704K	4,29
ENRON	33,7K	362K	4,05
SLASHDOT0902	82,2K	1,09M	3,94
DBLP	99,3K	1,09M	3,94
M14B	100K	1,28M	52,5
WAVE	156K	2,1M	22,9
WEB-STANFORD	255K	3,88M	7,31
WEB-GOOGLE	856K	5,58M	6,18
WIKI-TALK	2,39M	9,31M	3,91
LIVEJOURNAL	4,00M	69,3M	5,39
HYVES	8,08M	912M	4,75

Surgem os seguintes problemas ao lidar com grafos de grande dimensão em aplicações responsivas:

1. No pior caso, o tempo para realizar uma busca em tempo real pode ser excessivo;
2. Mesmo que o tempo de busca no pior caso seja aceitável, para uma pequena carga de usuários, ela pode ser excessiva em aplicações com muitos usuários simultâneos;
3. Uma alternativa seria computar e armazenar previamente todas as distâncias entre todos os nós do grafo. Entretanto, a quantidade de dados que precisariam ser armazenados (informações e índices) pode ser excessiva.

Caso seja tolerável uma pequena margem de erro no cálculo das menores distâncias, a aplicação responsiva poderia escolher alguns nós especiais, denominados nós *landmarks*, pré-calcular a menor distância apenas para esses nós e fazer aproximações através deles para os demais nós.

O principal objetivo desse trabalho é apresentar as principais estratégias para a seleção de nós *landmarks* para viabilizar o cálculo de distâncias mínimas em aplicações responsivas, o que é detalhado nas seções seguintes.

II. Notações

Notações utilizadas neste trabalho:

- G = (V, E), grafo não-direcionado e sem peso, onde V é o conjunto de nós e E o conjunto de arestas.
- n = |V|, número de nós no grafo G.
- m = |E|, número de arestas no grafo G.
- B ⊆ V, conjunto de nós *landmarks*.
- k = |B|, número de nós *landmarks*, com k ≪ n.
- d(u, v), distância entre os nós u e v ∈ V.
- N(v) ⊆ V, conjunto de nós conectados diretamente a v ∈ V.
- deg(v) = |N(v)|, número de nós conectados diretamente a v ∈ V.
- σ(u, w), número de caminhos mínimos que passam por u e w ∈ V.
- σ_v(u, w), número de caminhos mínimos que passam por u e w ∈ V interceptando v ∈ V.

III. Seleção de *Landmarks*

Conforme mencionado acima, o uso de nós *landmarks* pode viabilizar o uso de grafos de grande dimensão em aplicações responsivas. Resta saber, entretanto, como selecionar esses nós. Conforme mencionado em [1], existem basicamente 5 estratégias:

- Seleção Randômica (*Random*)
- Grau de Centralidade (*Degree*)
- Grau de Proximidade (*Closeness*)
- Grau de Intermediação (*Betweenness*)
- PageRank©

Cada uma dessas estratégias é detalhada a seguir. O objetivo é selecionar os nós de melhor classificação (*ranking*) em função da estratégia escolhida (os 1, 5 ou 10% melhores, por exemplo). Na Figura 1 é mostrado um gráfico comparativo utilizando o conjunto de dados CA-CONDMAT, sendo o desempenho medido através da taxa de sucesso nas buscas.

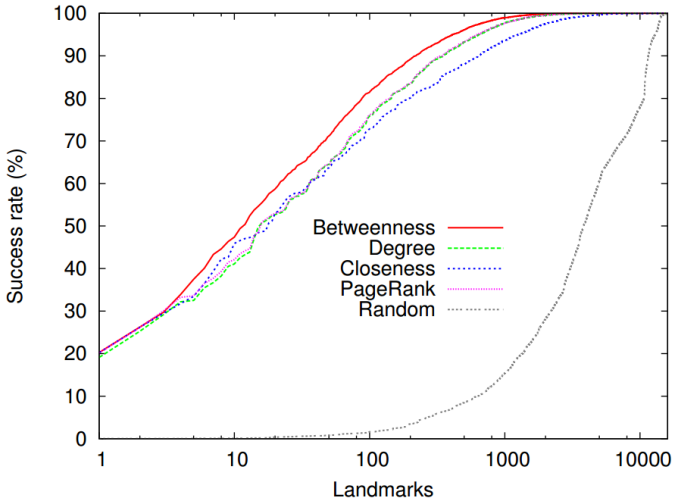

Figura 1. Taxa de sucesso das estratégias de busca através de *landmarks*, aplicadas ao conjunto de dados CA-CONDMAT [1].

A. *Seleção Randômica (Random)*

Essa estratégia consiste em selecionar aleatoriamente os nós *landmasks*.

Vantagem: É a estratégia de seleção mais simples e rápida de computar. Particularmente interessante quando é necessário atualizar o grafo com frequência.

Desvantagem: Conforme pode ser observado na Figura **1**, só apresenta bom resultado quando o percentual de nós escolhidos é alto.

B. *Grau de Centralidade (Degree)*

O grau de centralidade (C_{deg}) é calculado da seguinte forma:

$$C_{deg}(v) = \frac{1}{n-1} \deg(v), \text{ onde } v \in V$$

Quanto maior o valor, melhor é a classificação do nó.

Vantagem: É uma estratégia de seleção simples e rápida de computar, pois depende apenas do número de vizinhos de um nó. Particularmente interessante quando é necessário atualizar o grafo com frequência.

Desvantagem: Só utiliza informações disponíveis no nó, não considerando a estrutura do grafo.

C. *Grau de Proximidade (Closeness)*

O grau de proximidade (C_c) é calculado da seguinte forma:

$$C_c(v) = \frac{1}{n-1} \sum_w d(v, w), \text{ onde } v \text{ e } w \in V$$

Quanto menor o valor, melhor é a classificação do nó.

Vantagem: É uma estratégia de seleção global pois considera a estrutura do grafo.

Desvantagem: É necessário pré-calcular todas as distâncias entre todos os pares de nós do grafo, o que pode ser computacionalmente intensivo.

D. *Grau de Intermediação (Betweenness)*

O grau de intermediação (C_{bc}) é calculado da seguinte forma:

$$C_{bc}(v) = \sum_{u \neq v \neq w} \frac{\sigma_v(u,w)}{\sigma(u,w)}, \text{ onde u, v e w} \in V$$

Quanto maior o valor, melhor é a classificação do nó.

Vantagem: É uma estratégia de seleção global pois considera a estrutura do grafo.

Desvantagem: É necessário pré-calcular todas as distâncias entre todos os pares de nós do grafo, o que pode ser computacionalmente intensivo.

E. *PageRank*©

O PageRank [2] é a mesma estratégia utilizada pelo Google Search para qualificar páginas Web em função de um termo. Quanto maior o PageRank, mais significativa é a página para aquele termo.

Para um conjunto de páginas que citam um termo (digamos, "*compiler book*"), o PageRank de cada página é calculado considerando: 1. O número de páginas com hyperlinks apontando para a página calculada; e 2. O PageRank das páginas que apontam para a página calculada. Essa estratégia é ilustrada na Figura **2**, onde o tamanho dos nós representa seu PageRank.

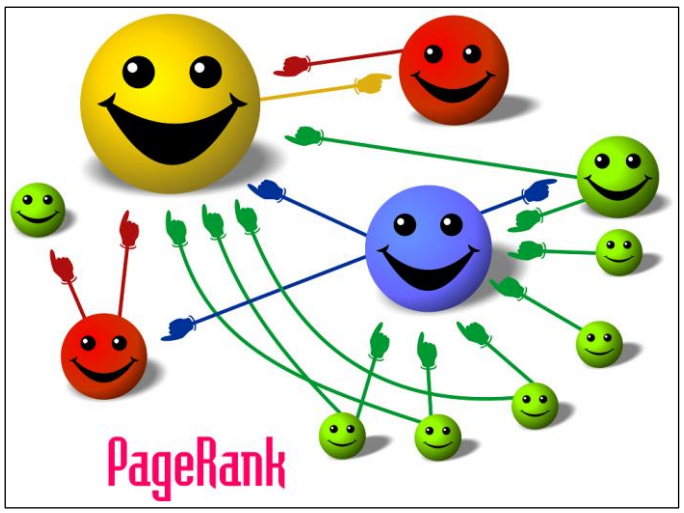

Figura 2. Ilustração do cálculo do PageRank de uma página Web.

Para um número r de interações, o PageRank (PR) é calculado da seguinte forma:

para cada v ∈ V **faça**

PR(v) = 1/n

para i = 1 **até** r **faça**

para cada v ∈ V **faça**

$$PR(v) = \frac{1-0{,}15}{n} + 0{,}15\left(\sum_{w \in N(v)} \frac{PR(w)}{deg(w)}\right)$$

Quanto maior o valor, melhor é a classificação do nó.

Vantagem: É uma estratégia de seleção local e pode ser computada rapidamente para um número limitado de interações (em torno de 100).

Desvantagem: Não leva em consideração a estrutura global do grafo. Para um grafo com diâmetro muito grande (milhares de arestas) um número limitado de interações pode ser muito restritivo.

IV. Conclusão

O uso de nós *landmarks* para o cálculo de distâncias mínimas pode resultar em grande vantagem para aplicações responsivas, viabilizando até mesmo sua execução. A escolha da estratégia para a seleção desses nós, entretanto, deve ser criteriosa, levando em conta as vantagens e desvantagens de cada uma. O ideal é que o desenvolvedor teste todas as estratégias disponíveis e use aquela que mais se adeque à limitações da sua aplicação.

Referências

[1] F. W. Takes and W. A. Kosters, "Adaptive Landmark Selection Strategies for Fast Shortest Path Computation in Large Real-World Graphs," in 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014, vol. 1, pp. 27–34.

[2] S. Brin and L. Page, "The anatomy of a large scale hypertextual Web search engine," Comput. Networks ISDN Syst., vol. 30, no. 1/7, pp. 107–17, 1998.