PROJETO 3

Rafael Moreira Cavalcante de Souza
cc23333@g.unicamp.br

Vinícius Dos Santos Andrade
cc22333@g.unicamp.br

1 Introdução

O PageRank é um algoritmo utilizado para medir a importância de páginas da web com base na estrutura de links entre elas. Desenvolvido por Larry Page e Sergey Brin, cofundadores do Google, o algoritmo é capaz de classificar páginas de maneira mais eficiente do que contagens simples de links. Neste projeto, implementamos duas abordagens para o cálculo do PageRank: o Modelo do Navegador Aleatório e o Algoritmo Iterativo.

2 Descrição

O PageRank considera que uma página web é importante se muitas outras páginas relevantes apontarem para ela. O algoritmo combate o inflacionamento artificial de classificações, considerando também a qualidade dos links.

O PageRank pode ser representado como um navegador aleatório que percorre a web clicando em links de maneira aleatória. Para evitar que o navegador fique preso em um ciclo de páginas, é introduzido um fator de amortecimento, permitindo que ele escolha uma página aleatoriamente com uma certa probabilidade.

Existem duas formas principais de calcular o PageRank:

- **Modelo do Navegador Aleatório**: Neste modelo, um navegador hipotético escolhe aleatoriamente links a seguir e a probabilidade de ele estar em uma página específica reflete seu PageRank. Esse modelo pode ser representado como uma Cadeia de Markov, onde cada página é um estado e os links entre páginas definem as transições entre esses estados. Para estimar o PageRank, o navegador percorre a web de forma aleatória,

e o número de vezes que ele visita uma página é usado como base para calcular sua importância.

- **Algoritmo Iterativo**: Outra abordagem para calcular o PageRank é iterar sobre as páginas, ajustando continuamente suas classificações. Inicialmente, assume-se que todas as páginas têm o mesmo valor de PageRank. A cada iteração, o valor do PageRank de uma página é atualizado com base nos valores das páginas que apontam para ela, divididos pelo número de links em cada uma dessas páginas. Com o tempo, os valores convergem para uma estimativa estável.

  A fórmula iterativa é dada por:

$$PR(p) = \frac{1-d}{N} + d \sum_{i \in M(p)} \frac{PR(i)}{L(i)}$$

  Onde:

  - $PR(p)$ é o valor do PageRank da página p.
  - d é o fator de amortecimento (geralmente definido como 0.85).
  - N é o número total de páginas no conjunto.
  - $M(p)$ é o conjunto de páginas que apontam para a página p.
  - $PR(i)$ é o valor do PageRank da página i que aponta para p.
  - $L(i)$ é o número de links saindo da página i.

3 Estrutura do Projeto

O projeto é dividido em três componentes principais:

1. **Conjunto de Páginas HTML**: Representa a rede de links a ser analisada, onde cada página contém hiperlinks para outras, formando uma estrutura de grafo direcionado.
2. **Implementações em Python dos algoritmos de PageRank**: Inclui a implementação tanto do método de amostragem quanto do método iterativo para o cálculo do PageRank.
3. **Resultados e Análise**: Realiza a comparação dos resultados obtidos pelos diferentes métodos, discutindo sua convergência e precisão.

3.1 Páginas HTML

O conjunto de páginas HTML simula uma rede de links entre páginas, onde cada página contém links para outras, representando os nós e arestas de uma rede. Esses arquivos formam a base de entrada para o algoritmo de PageRank, sendo processados para calcular a importância de cada página na rede.

Por exemplo, uma página nomeada como "2"pode conter links para as páginas "1"e "3", demonstrando como as páginas do projeto se conectam para formar a rede de links. Cada página do conjunto segue esse padrão, com variações no número e destino dos links, o que proporciona diferentes cenários para testar os algoritmos de PageRank.

3.2 Implementação dos algorítmos em Python

3.2.1 Modelo do Navegador Aleatório

O **Modelo do Navegador Aleatório** simula um navegador que, ao visitar uma página, segue um link de forma aleatória com uma probabilidade d (fator de amortecimento), ou escolhe qualquer outra página da rede com probabilidade $1 - d$. As funções a seguir foram construídas para aplicar o modelo em ambiente da linguagem de programação **Python**. Suas funções são:

- **def transition_model(corpus, page, damping_factor)**: Retorna uma distribuição de probabilidade sobre qual página visitar em seguida, dada uma página atual.
- **def sample_pagerank(corpus, damping_factor, n)**: Retorna os valores do PageRank para cada página, amostrando ‘n‘ páginas conforme o modelo de transição, começando com uma página aleatória.

3.2.2 Algoritmo Iterativo

A função foi construída para aplicar o **modelo iterativo** em um ambiente da linguagem de programação **Python**. Sua função é calcular os valores de PageRank para cada página de forma iterativa até que os valores converjam e retornar um dicionário com os nomes das páginas e seus respectivos PageRanks.

- **def iterate_pagerank(corpus, damping_factor)**: Calcula os valores de PageRank para cada página de forma iterativa até que os valores de PageRank converjam e retorna um dicionário com os nomes das páginas e seus respectivos PageRanks.

4 Resultados

4.1 Resultados dos Testes

Os valores de PageRank estimados pelo modelo do navegador aleatório e pelo algoritmo iterativo são apresentados a seguir:

Resultados do Corpus0:

Corpus0	PageRank (Aleatório)	PageRank (Iterativo)
1.html	21.41% - 22.45%	21.9777%
2.html	42.74% - 43.00%	42.9358%
3.html	21.73% - 22.31%	21.9777%
4.html	12.69% - 13.42%	13.1088%

Resultados do Corpus1:

Corpus1	PageRank (Aleatório)	PageRank (Iterativo)
bfs.html	10.92% - 11.46%	11.5193%
dfs.html	7.59% - 8.03%	8.0933%
games.html	22.77% - 23.56%	22.7705%
minesweeper.html	11.98% - 12.74%	11.7971%
minimax.html	12.98% - 13.28%	13.1199%
search.html	20.25% - 20.95%	20.9029%
tictactoe.html	11.55% - 11.99%	11.7971%

Resultados do Corpus2:

Corpus2	PageRank (Aleatório)	PageRank (Iterativo)
ai.html	17.99% - 19.18%	18.89%
algorithms.html	10.52% - 10.85%	10.6435%
c.html	12.38% - 12.89%	12.3811%
inference.html	11.89% - 13.02%	12.8851%
logic.html	2.48% - 2.69%	2.6355%
programming.html	23.06% - 23.93%	23.0132%
python.html	12.23% - 12.81%	12.3811%
recursion.html	6.77% - 6.95%	7.1703%

4.2 Análise dos Resultados

Ao analisar os resultados obtidos, notamos que, em geral, os valores de PageRank obtidos pelo modelo do navegador aleatório e pelo algoritmo iterativo estão próximos, mas não coincidem exatamente. Essa diferença pode ser explicada por vários fatores, incluindo a aleatoriedade inerente ao modelo do navegador e a maneira como o algoritmo iterativo converge para um valor estável. O modelo do navegador aleatório é influenciado pelo número de iterações realizadas e pela escolha inicial do ponto de partida, o que pode levar a variações em seus resultados. Por outro lado, o algoritmo iterativo se baseia na estrutura de links e na propagação de valores ao longo das iterações, o que pode resultar em um cálculo mais preciso em um cenário estável.

Essa diferença entre os resultados evidencia a importância de escolher a abordagem adequada com base nos objetivos do projeto. O modelo do navegador aleatório pode ser mais fácil de implementar e entender, enquanto o algoritmo iterativo pode fornecer resultados mais confiáveis em cenários complexos.

5 Conclusão

Neste projeto, exploramos a implementação e o funcionamento de duas abordagens distintas para o cálculo do PageRank: o Modelo do Navegador Aleatório e o Algoritmo Iterativo. Ambas as abordagens demonstraram ser eficazes na estimativa de PageRank, embora apresentem características e resultados um pouco diferentes.

O modelo do navegador aleatório oferece uma forma intuitiva e simples de visualizar a navegação na web, mas pode ser afetado pela aleatoriedade e pelo número de iterações. Em contrapartida, o algoritmo iterativo permite um controle mais preciso e uma convergência estável dos valores, sendo mais adequado para cenários onde a precisão é crucial.

Os resultados obtidos nos testes reforçam a ideia de que o PageRank é uma ferramenta poderosa para a análise de relevância na web e pode ser aplicada em diversos contextos.