Projeto 3

TI327 - Tópicos em Inteligência Artificial

Prof. Guilherme Macedo

COTUCA - Unicamp

Quando mecanismos de busca como o Google exibem resultados de busca, eles o fazem colocando páginas mais "importantes"e de maior qualidade em posições mais altas nos resultados do que páginas menos importantes. Mas como o mecanismo de busca sabe quais páginas são mais importantes que outras?

Uma heurística pode ser que uma página "importante"é aquela para a qual muitas outras páginas fazem link, já que é razoável imaginar que mais sites vão fazer link para uma página da web de maior qualidade do que para uma de menor qualidade. Portanto, poderíamos imaginar um sistema em que cada página recebe uma classificação de acordo com o número de links recebidos de outras páginas, e classificações mais altas sinalizariam maior importância.

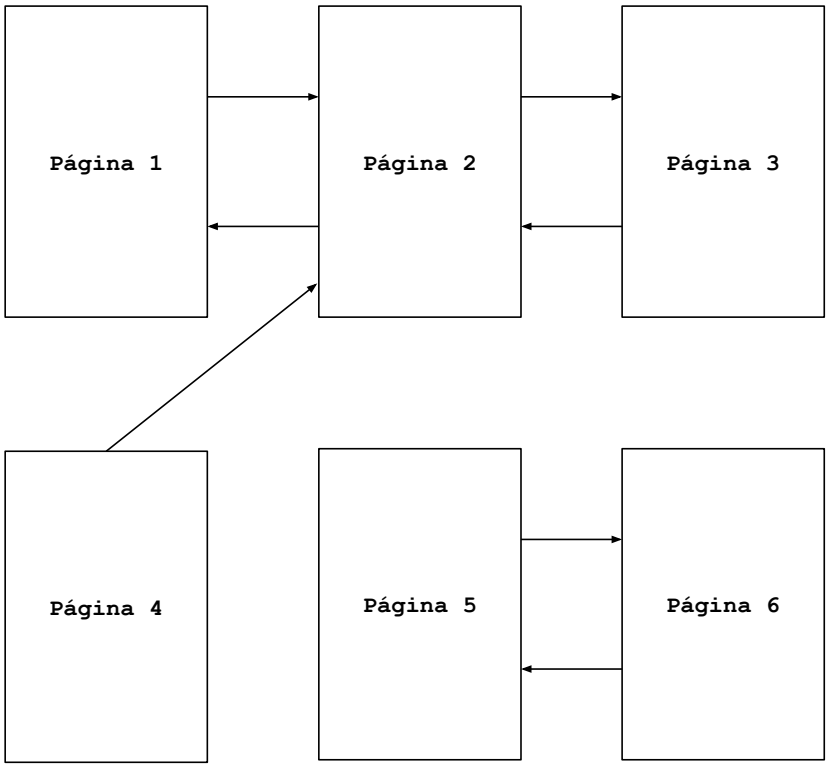
No entanto, essa definição não é perfeita: se alguém quiser fazer sua página parecer mais importante, sob esse sistema, poderia simplesmente criar muitas outras páginas que linkam para a página desejada, para inflacionar artificialmente sua classificação.

Por essa razão, o algoritmo PageRank foi criado pelos cofundadores do Google

(incluindo Larry Page, que deu nome ao algoritmo). No algoritmo PageRank, um site é mais importante se for linkado por outros sites importantes, e links de sites menos importantes têm seu peso reduzido. Essa definição pode parecer um pouco circular, mas, na verdade, existem várias estratégias para calcular essas classificações.

Modelo do Surfista Aleatório

Uma forma de pensar sobre o PageRank é com o modelo do surfista aleatório, que considera o comportamento de um surfista hipotético na internet que clica em links aleatoriamente. Considere o conjunto de páginas da web abaixo, onde uma seta entre duas páginas indica um link de uma página para outra.



O modelo do surfista aleatório imagina um surfista que começa com uma página da web aleatoriamente e, em seguida, escolhe links aleatoriamente para seguir. Se o surfista estiver na Página 2, por exemplo, ele escolheria aleatoriamente entre a Página 1 e a Página 3 para visitar a seguir. Se ele escolher a Página 3, o surfista escolheria aleatoriamente entre a Página 2 e a Página 4 para visitar a seguir.

Imagine que começamos aleatoriamente pela Página 5. Não teríamos escolha a não ser ir para a Página 6, e depois voltar para a Página 5, depois Página 6 novamente, e assim por diante. Acabaríamos com uma estimativa de $0,5$ para o PageRank das Páginas 5 e 6, e uma estimativa de 0 para o PageRank de todas as outras páginas, já que passamos todo o nosso tempo nas Páginas 5 e 6 e nunca visitamos nenhuma das outras páginas.

Para garantir que sempre possamos chegar a outro lugar no conjunto de páginas da web, introduziremos um fator de amortecimento d em nosso modelo. Com probabilidade d o surfista aleatório escolherá um dos links na página atual aleatoriamente. Mas, de outra forma, com probabilidade $1 - d$, o surfista aleatório escolherá uma página aleatoriamente entre todas as páginas do conjunto.

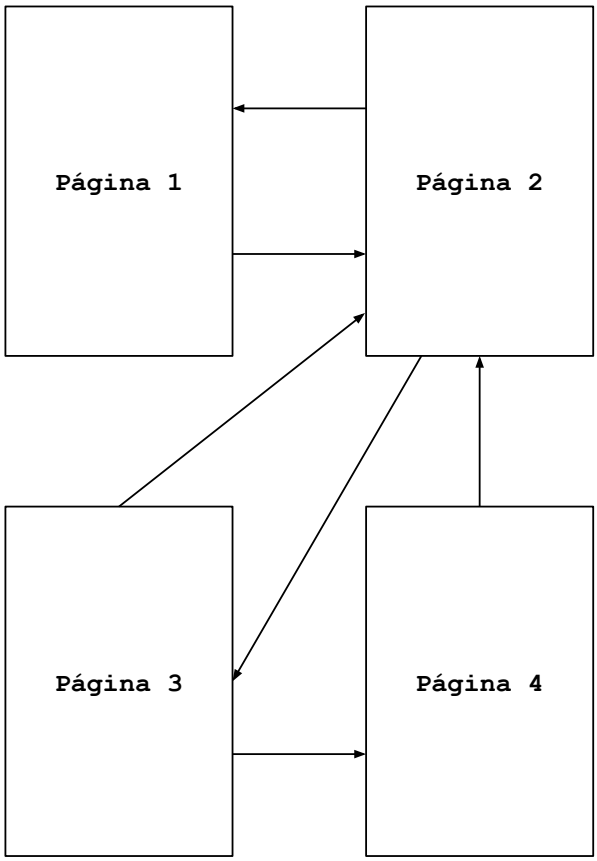
O nosso surfista aleatório agora começa escolhendo uma página aleatoriamente e, para cada amostra adicional que gostaríamos de gerar, escolhe um link da página atual aleatoriamente com probabilidade d e escolhe qualquer página aleatoriamente com probabilidade $1 - d$. Se acompanharmos quantas vezes cada página foi selecionada como amostra, podemos tratar a proporção de visitas a uma página específica como seu PageRank. Assim, o PageRank de uma página pode ser descrito como a probabilidade de que um surfista aleatório esteja nessa página em um dado momento. Afinal, se houver mais

links para uma página específica, é mais provável que um surfista aleatório acabe nessa página. Além disso, um link de um site mais importante é mais provável de ser clicado do que um link de um site menos importante, que recebe menos links, então esse modelo também leva em conta a importância dos links.

Uma forma de interpretar esse modelo é como uma Cadeia de Markov, onde cada página representa um estado, e cada página tem um modelo de transição que escolhe aleatoriamente entre seus links. A cada passo de tempo, o estado muda para uma das páginas vinculadas pelo estado atual.

Ao amostrar estados aleatoriamente da Cadeia de Markov, podemos obter uma estimativa para o PageRank de cada página. Podemos começar escolhendo uma página aleatoriamente e, em seguida, continuar seguindo links aleatoriamente, mantendo o controle de quantas vezes visitamos cada página. Após coletar todas as nossas amostras, com base em um número que escolhemos antecipadamente, a proporção de vezes que estivemos em cada página pode ser uma estimativa do PageRank daquela página.

No entanto, essa definição de PageRank se torna um pouco problemática se considerarmos uma rede de páginas como a mostrada abaixo.



Algoritmo Iterativo

Nós podemos também definir o PageRank de uma página usando uma expressão matemática recursiva. Seja $P(k)$ o PageRank de uma determinada página k: a probabilidade de que um surfista aleatório acabe nessa página. Como definimos $P(k)$? Bem, sabemos que há duas maneiras pelas quais um surfista aleatório poderia acabar na página:

- o surfista, com probabilidade $1 - d$, escolheu uma página aleatoriamente e acabou na página k

- o surfista, com prababilidade d, seguiu um link de uma página i para uma página k.

A primeira condição é bastante simples de expressar matematicamente: é $1-d$ dividido por N, onde N é o número total de páginas em todo o conjunto. Isso ocorre porque a probabilidade $1-d$ de escolher uma página aleatoriamente é dividida igualmente entre todas as N páginas possíveis.

Para a segunda condição, precisamos considerar cada página i possível que linka para a página k. Para cada uma dessas páginas que estão fazendo links, deixe $\Omega(i)$ ser o número de links na página i. Cada página i que linka para k tem seu próprio PageRank, $P(k)$, representando a probabilidade de que estejamos na página i em um dado momento. E, como de i viajamos para qualquer um dos links daquela página com probabilidade igual, dividimos a $P(i)$ pelo número de links $\Omega(i)$ para obter a probabilidade de que estivemos na página i escolhemos o link para a página k.

Isso nos dá a seguinte definição para o PageRank de uma página k.

$$P(k) = \frac{1-d}{N} + d \cdot \sum_i \frac{P(i)}{\Omega(i)}.$$

Nesta fórmula, d é o fator de amortecimento, N é o número total de páginas no conjunto, i abrange todas as páginas que linkam para a página k, e $\Omega(i)$ é o número de links presentes na página i.

Como procederíamos para calcular os valores de PageRank para cada página, então? Podemos fazer isso por iteração: comece assumindo que o PageRank de cada página é $1/N$. Em seguida, use a fórmula acima para calcular novos valores de PageRank para cada página, com base nos valores anteriores. Se continuarmos repetindo esse processo, calculando um novo conjunto de valores

de PageRank para cada página com base no conjunto anterior, eventualmente os valores de PageRank convergirão, ou seja, não mudarão por mais do que um pequeno limiar a cada iteração.

Neste projeto, você implementará ambas as abordagens para calcular o PageRank – calculando tanto por amostragem de páginas a partir de um surfista aleatório na Cadeia de Markov quanto aplicando iterativamente a fórmula de PageRank.

Atividades

- Completar a implementação, no arquivo `pagerank.py`, das funções `transition_model`, `sample_pagerank` e `iterate_pagerank`.
- Testar as diferentes abordagens do PageRank com os conjuntos de páginas web disponbilizadas pelo professor no ambiente de ensino aberto.
- Escrever um relatório de 3 à 5 páginas com as suas descobertas. O modelo do relatório está disponível em https://bit.ly/43QpBkv. O relatório deverá ser escrito em português ou inglês.

Prazo de entrega

Quarta-feira, 17 de outubro de 2024, até às 23h59. Não é aconselhável que este projeto seja entregue fora do prazo estabelecido. Contudo, no caso de a fazer, a sua nota será penalizada da seguinte maneira:

18/06/2024 23h59	$0,75 \times$ nota
19/06/2024 23h59	$0,50 \times$ nota
20/06/2024 23h59	$0,25 \times$ nota

Submissão

O código-fonte e o relatório deverão ser submetidos pelo GitHub Classroom até os prazos de entrega estabelecidos.