TP1 - Algoritmos para Bioinformática I

Lucas Resende Pellegrinelli Machado - 2018126673

Setembro 2020

1 Introdução

Mineração de dados é área da computação que tenta inferir informações úteis a partir de dados que inicialmente não aparentam possuir essas informações.

Ao longo do tempo, novas tecnologias e algoritmos voltados para esse propósito foram inventados, cada vez mais aumentando o poder que temos ao analisar qualquer tipo de dado.

No trabalho atual, a ideia é usar um dos clássicos da mineração de dados, o SVD, para resolver um problema muito comum que é o de recomendação a partir de características. Dado um usuário que quer encontrar algo na internet, como ele pode quantificar o quao próximo uma página é do que ele quer encontrar somente a partir de palavras chaves dadas pelo documento?

2 Entradas e saídas

Dado o método que usaremos nesse trabalho, todos os nossos dados de entrada e saída serão descritos por meio de matrizes e vetores.

Para representar o banco de dados da máquina de pesquisa, teremos uma matriz contendo diversas linhas correspondentes a termos e diversas colunas correspondendo a documentos. Cada uma das células é responsável por armazenar se um documento possuí ou não um dado termo, sendo assim representado por um 1. Caso contrário será armazenado um 0 naquela célula.

A matriz de entrada dada é a seguinte

eigenvalue	0	0	0	1	0
England	0	0	0	0	1
FIFA	0	0	0	0	1
Google	1	0	1	0	0
Internet	1	0	0	0	0
link	0	1	0	0	0
matrix	1	0	1	1	0
page	0	1	1	0	0
rank	0	0	1	1	1
web	0	1	1	0	0

Essa matriz representa um modelo da internet em um dia onde um dos assuntos mais comentados é a queda da Inglaterra dos 10 primeiros colocados no ranking de melhores seleções de futebol no mundo da FIFA.

É interessante notar que podemos ter uma idéia do que cada página tem a falar a partir dos valores 1 na matriz. Por exemplo a quinta página claramente fala do tema dito acima visto que ela possui como palavras chaves atives "England", "FIFA" e "rank", representando que algo aconteceu com a Inglaterra no rank da FIFA.

Também temos como entrada uma query, que nada mais é que um vetor correspondendo com o que queremos procurar nessa máquina de pesquisa.

$$q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}^T$$

Ao resolver o sistema linear utilizando essa query, teremos como resposta uma projeção no espaço reduzido que corresponde a similaridade de cada um dos documentos armazenados na matriz que representa a internet como a nossa query, ou seja, as páginas que o sistemas de busca nos recomendaria caso procurassemos essa query.

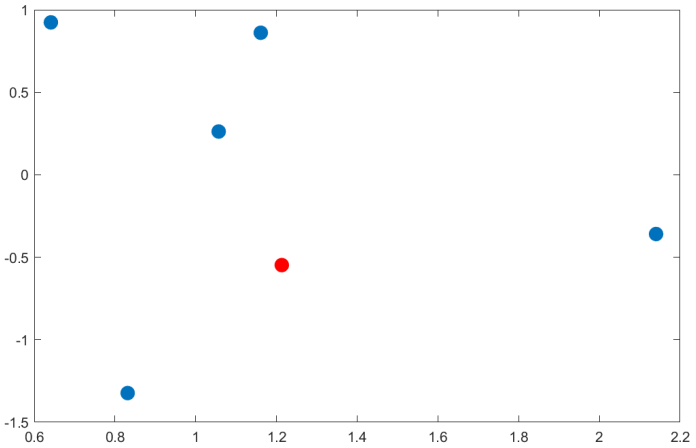

Figure 1: Plot mostrando em azul os pontos correspondentes a cada uma das páginas que estão no modelo da internet e o ponto vermelho mostrando onde a query fica no espaço reduzido

3 Metodologia

Para que possamos resolver esse sistemas de equações e conseguir o resultado que queremos, temos que primeiro decompor a matriz original representando o modelo da internet. Isso é feito usando um algoritmo chamado de Singular Value Decomposition (SVD).

Essa decomposição nos retornará três matrizes U, Σ e V^*, onde a matriz Σ possuí uma propriedade interessante onde ela só possuí valores em sua diagonal e cada um de seus valores (chamados valores singulares) corresponde a importância de uma certa característica para a representação do modelo inteiro.

Esses valores singulares nos dão um ótimo jeito de encontrar uma representação em dimensionalidade reduzida dos nossos dados visto que ao ordenalos, temos agora um ranking de quais características são as mais importantes, então podemos escolher as n mais importantes e reconstruir o modelo original apenas com essas n características. Essa é a ideia principal em diversas aplicações.

No caso desse trabalho, para calcular o resultado da query a partir do vetor de entrada, precisamos resolver o sistema linear definido por

$$U_{1,2}x = q$$

Onde a matriz $U_{1,2}$ corresponde à matriz U reduzida a apenas suas 2 dimensões com maiores valores singulares, ou seja, as duas dimensões mais importantes para representar os dados. O vetor coluna q representa a query.

O vetor x por sua vez é o vetor que queremos descobrir ao resolver esse sistema linear. Ele vai ser um vetor de 2 posições visto que estamos lidando apenas com as 2 dimensões mais importantes e ele dirá onde no espaço reduzido uma página que corresponde a exatamente o que estamos procurando estaria.

Para descobrir quais as páginas mais adequadas de serem mostradas ao executar esse query, basta comparar a distância do ponto gerado pela projeção da query no espaço reduzido (x) para a projeção das páginas no mesmo espaço ($U_{1,2}$), quanto mais próximo estiverem, mais adequada aquela página é.

4 Conclusões

Esse trabalho é importante visto que ele exemplifica de forma visual como métodos conhecidos na computação operam e geram resultados interessantes e práticos.

É evidente a robustez do método e a base teórica interessante sobre ele, que reduz um problema de natureza complexa para um problema de comparar a distância em um espaço 2D. De certa forma é bonito ver esse tipo de problema sendo resolvido com um método elegante como esse.

É inevitável também pensar em outras aplicações para métodos como esse em procura de padrões em textos ou imagens ou compactação de informações, gerando ainda mais interesse pelo decorrer da disciplica e trabalhos propostos usando técnicas parecidas.

5 Referências bibliográficas

1. Série de vídeos "Singular Value Decomposition" por Steve Brunton.
   https://www.youtube.com/playlist?list=PLMrJAkhIeNNSVjnsviglFoY2nXildDCcv

2. Relatório de aula sobre SVD do departamento de ciência da computação da universidade de Carnegie Mellon.

   https://www.cs.cmu.edu/ venkatg/teaching/CStheory-infoage/book-chapter-4.pdf

6 Anexos

O script abaixo foi usado para gerar o gráfico da representação compacta da matriz original e da query dada além de comparar o valor da matriz original com a recomposição da matriz usando suas decomposições.

```
% Definindo a matriz 'A'
A = [0 0 0 1 0;
     0 0 0 0 1;
     0 0 0 0 1;
     1 0 1 0 0;
     1 0 0 0 0;
     0 1 0 0 0;
     1 0 1 1 0;
     0 1 1 0 0;
     0 0 1 1 1;
     0 1 1 0 0];

% Definindo a query 'q'
q = transpose([0, 0, 0, 0, 0, 0, 0, 1, 1, 1]);

% Calculando o singular value decomposition da matriz A
[T, S, D] = svd(A);

% Caso essa norma seja 0, então A = TSD'
norm(A - T * S * transpose(D), 2)

% O resuldado obtido pelo comando acima é 2.8812e-15

% Calcula os pontos no sistema coordenada definido pelo SVD
S2 = S(1:2, 1:2);
D2 = D(:, 1:2);
Aux = S2 * transpose(D2);
x = Aux (1, :);
y = Aux (2, :);

% Plote os pontos
plot(x, y, '*');

```

```
hold on

% Projetando a query no espaço reduzido
query_pt = linsolve(T(:, 1:2), q)

% Plotando o ponto no gráfico
plot(query_pt(1), query_pt(2), 'r*');
```