Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Algoritmos para Bioinformática I - Primeiro Semestre 2020

Trabalhos Práticos 1 e 2 - Mineração de Dados utilizando álgebra linear

Aluno: Guilherme de Abreu Lima Buitrago Miranda - Matrícula: 2018054788

1. Introdução

Como funcionam os buscadores internet, como o Google e o Bing? E de que forma isso pode ser correlacionado com compostos orgânicos que regem a vida na Terra? Embora, a princípio, não pareçam estar conectados, ambos podem se aproveitar de um conceito computacional moderno chamado Mineração de dados, que será objeto de estudo destes trabalhos.

Em particular, inicialmente, busca-se entender os fundamentos básicos da mineração de dados usando a álgebra linear. Eles são importantes para que, posteriormente, a técnica seja bem aplicada nos problemas de bioinformática. Sobretudo em contextos que os algoritmos clássicos não geram resultados tão precisos, a mineração de dados pode ser uma valiosa aliada. Afinal, ela traz uma abordagem diferente ao problema, utilizando técnicas de estatística e de reconhecimento de padrões, por exemplo.

Primeiramente, busca-se entender melhor a entrada do programa e quais as saídas esperadas. Em seguida, explica-se o significado de cada uma das informações geradas pelo programa e suas respectivas importâncias, bem como quais foram os métodos utilizados para tal, e suas fundamentações teóricas. Por fim, discute-se algumas conclusões obtidas com o trabalho, vislumbrando outras possibilidades da aplicação da técnica estudada na bioinformática.

2. Entradas e Saídas

a. Trabalho 1

Como entrada para o trabalho, espera-se uma matriz de Termos por Documentos, ou seja, cada linha representa um termo e cada coluna representa um documento. Em nosso exemplo, portanto, cada linha representa uma palavra e cada

coluna representa uma frase (no caso, um título de uma página na web). Assim, cada célula responde à seguinte pergunta: “O documento dessa coluna contém a palavra relativa a essa linha?” Se sim, seu valor é 1. Caso contrário, seu valor é 0.

Temos, para esse problema, os cinco documentos abaixo, com seus respectivos termos destacados em negrito.

1. The **Google matrix** *P* is a model of the **Internet.**
2. P_{ij} is nonzero if there is a **link** from **web page** *j* to *i*.
3. The **Google matrix** is used to **rank** all **web page**s.
4. The **ranking** is done by solving a **matrix eigenvalue** problem.
5. **England** dropped out of the top 10 in the **FIFA ranking.**

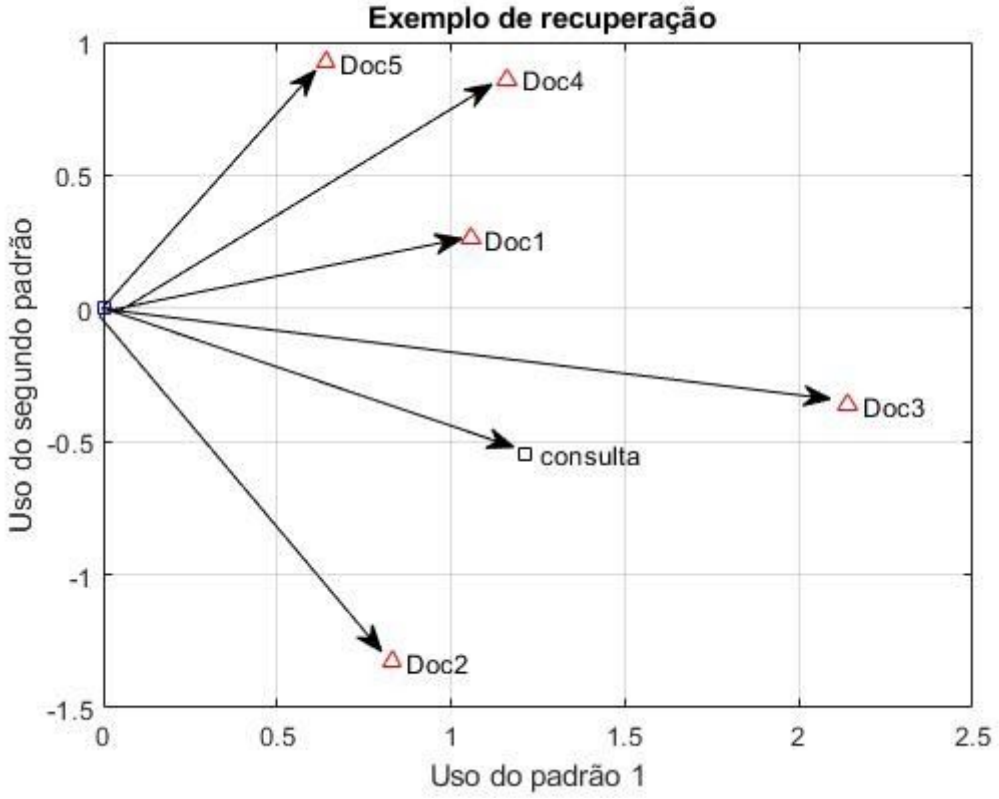
Assim, a entrada será a seguinte matriz:

Termos	**Documentos**				
	Doc_01	Doc_02	Doc_03	Doc_04	Doc_05
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

Observação: na verdade, iremos inserir em nossa ferramenta computacional apenas as sequências de 0s e 1s em forma de matriz, conforme anexo “Exemplo2.m”. Os

nomes dos termos e dos documentos aparecem na documentação apenas a título de explicação.

A principal saída gerada é a projeção no espaço reduzido da semelhança de cada um dos cinco documentos com a consulta "**ranking** of **web pages**". Nela, conforme pode ser observado abaixo, é mostrada a influência do primeiro e do segundo padrão gerados pela decomposição SVD (decomposição em valores singulares) no eixo x e no eixo y. Além disso, essa mesma influência pode ser visualizada nos outros documentos, o que facilita a comparação com a consulta em questão, em especial quando observa-se os ângulos gerados, representados pelas setas da cor preta.



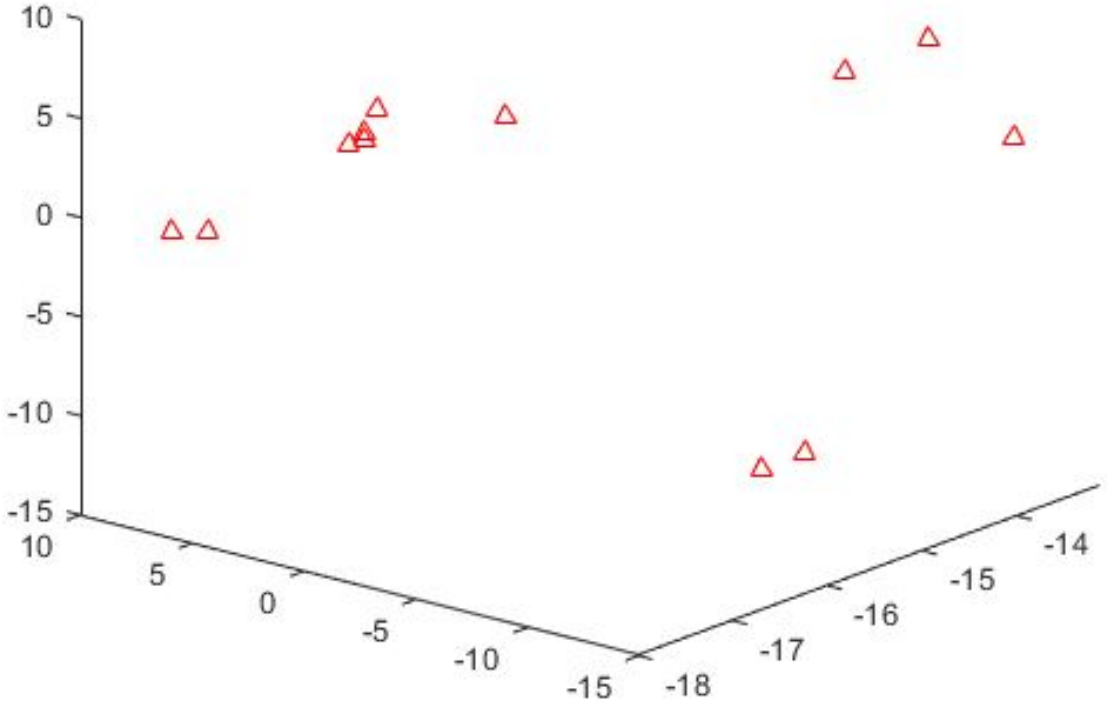
É interessante observar a semelhança da consulta com os documentos 2 e 3, com os quais têm os ângulos mais parecidos. Além disso, a semelhança com o documento 1 também é considerável, e só pôde ser vista em razão do uso das técnicas de visualização de dados, que serão mais detalhadas na seção 3a.

b. Trabalho 2

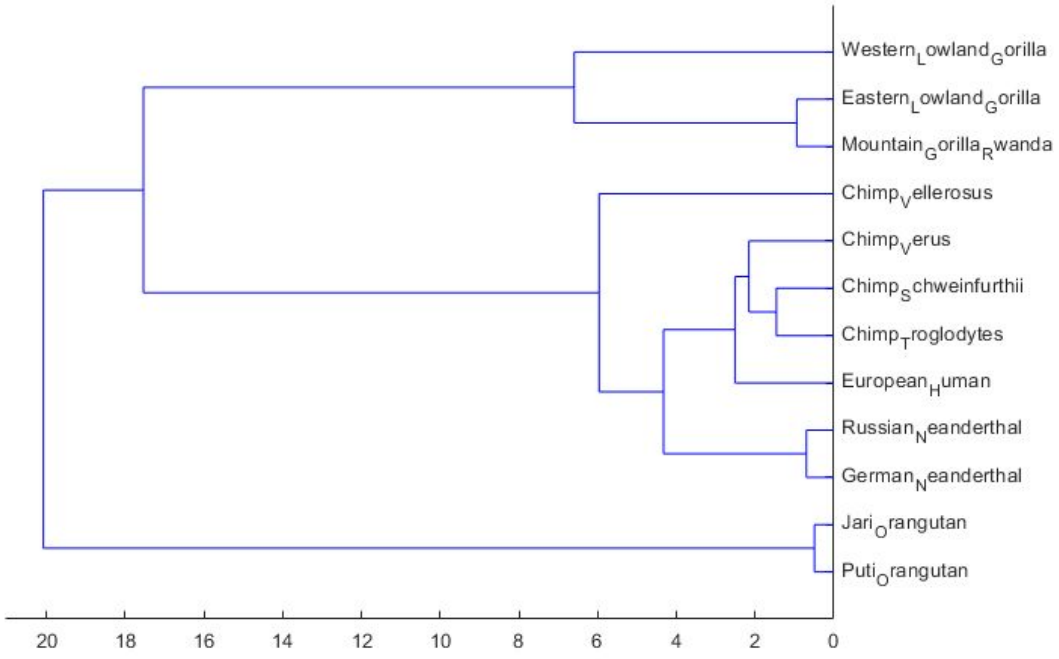
Neste trabalho, utilizou-se como entrada o conjunto de sequências de DNA "primates", presente no toolbox de bioinformática do software MATLAB. Durante e

após o processamento (que será esmiuçado na próxima seção), tem-se três saídas principais. São elas:

- A representação em R^3 das entidades resultantes a partir da decomposição SVD realizada. Nela, pode-se observar 4 clusters bem definidos, que serão melhor trabalhados na sequência



- A árvore hierárquica associada, obtida a partir da manipulação dos valores resultantes da decomposição SVD. É possível constatar, a partir da figura, a maneira com a qual os primatas evoluíram. Assim, nota-se que espécies como os neandertais russos são bastante parecidas com os neandertais germânicos, pois ambos têm um ancestral em comum mais recente. Em contrapartida, o orangotango jari é deveras diferente do gorila das montanhas, por exemplo. Nesse caso, o ancestral em comum das duas espécies é bem distante do restante das espécies.



- A separação de cada espécie em um subgrupo (cluster) utilizando o agrupamento Kmeans. Conforme o comando apresentado na seção de anexos, é possível observar a separação das espécies em quatro grupos principais. Assim, a saída é uma sequência de 12 números, cada qual representando um primata dos nossos dados de entrada. Dessa forma, tem-se:

  German_Neanderthal: grupo 1
  Russian_Neanderthal: grupo 1
  European_Human: grupo 1
  Mountain_Gorilla_Rwanda: grupo 2
  Chimp_Troglodytes: grupo 3
  Puti_Orangutan: grupo 4
  Jari_Orangutan: grupo 4
  Western_Lowland_Gorilla: grupo 2
  Eastern_Lowland_Gorilla: grupo 2
  Chimp_Schweinfurthii: grupo 3
  Chimp_Vellerosus: grupo 3
  Chimp_Verus: grupo 3

Conforme esperado, a separação em quatro grupos é bastante satisfatória. Ademais, é interessante observar que espécies com ancestrais em comum mais recentes são agrupadas juntas. Além disso, o contrário também é verdadeiro, ou seja, espécies com ancestrais em comum mais distantes são agrupadas separadamente.

3. Metodologia

a. Trabalho 1

Para determinar a semelhança de uma dada query a outros documentos, utilizou-se alguns conceitos de álgebra linear, normalmente aplicados na área de mineração de dados. Em particular, aplicou-se uma decomposição SVD (decomposição em valores singulares), a fim de se observar, mais profundamente, tal semelhança.

Dentre os algoritmos clássicos da computação, uma maneira possível de determinar o grau de afinidade entre dois documentos é, por exemplo, analisar os termos que os compõem um a um. Assim, quando aplicado no dataset do presente trabalho 1, um algoritmo de busca de cadeia de caracteres encontraria semelhanças entre a consulta e o documento 2 e 3. Isso ocorre pois os termos “web” e “pages” aparecem em ambos os documentos. Além disso, um algoritmo clássico um pouco mais sofisticado poderia encontrar uma maior semelhança entre a consulta e o documento 3, pois eles compartilham uma palavra com o mesmo prefixo (rank). Contudo, a semelhança entre a consulta e o documento 1 não seria observada. Isso ocorre pois não há nenhum termo em comum, apesar das duas tratarem do mesmo assunto e, portanto, serem semelhantes.

Por conseguinte, é perceptível a importância da inovação trazida pela decomposição SVD nesse contexto. Nela, foi possível observar a semelhança entre o documento 1 e a consulta, apesar de não compartilharem nenhum termo. Isso ocorre pois, na decomposição, é possível determinar que o documento 1 se parece com o documento 3 e, se o documento 3 é bastante parecido com a consulta em questão, então, o documento 1 também se parece com a consulta.

Essa conclusão só é possível pois a decomposição SVD é capaz de separar os documentos em padrões usando a álgebra linear e, posteriormente, evidenciar o quanto um determinado texto se parece com um padrão encontrado. Assim, para comparar a consulta criada com os padrões já existentes, basta resolver um pequeno sistema de equações (feito com a ajuda da ferramenta MATLAB). Conforme observado na seção 2a, usando apenas os dois principais padrões dados pela decomposição SVD, é possível constatar a grande semelhança entre a

consulta e os documentos 2 e 3. Outrossim, a afinidade com o documento 1 também desperta atenção.

b. Trabalho 2

O primeiro passo, antes de efetivamente dar início ao uso de ferramentas da álgebra linear, foi transformar o conjunto primates, uma struct contendo o nome da espécie e sua respectiva sequência genômica, em um vector space model. Para tal, utilizou-se o script “montaA6.m”, disponibilizado junto ao enunciado do trabalho, resultando numa matriz A.

Em seguida, aplica-se, na matriz A, a decomposição SVD (já explicada no subtópico anterior), resultando na projeção antecipadamente observada. Logo após, dá-se início à construção de uma árvore hierárquica associada. Ela será útil para que seja possível observar melhor a separação dos primatas em grupos, já antecipada pela visualização das entidades no espaço R^3.

Para tal, utiliza-se algoritmos já presentes no conjunto de ferramentas do MATLAB, como o “linkage” e o “dendrogram”. Assim, tem-se como resultado a árvore hierárquica já apresentada ao leitor na seção 2b.

Por fim, aplica-se o algoritmo de clusterização kmeans para que se obtenha em qual grupo cada primata foi separado. Tal clusterização funciona da seguinte maneira: durante a decomposição SVD, cada genoma de cada primata é decomposto em valores singulares, conforme o próprio nome do algoritmo sugere. Quanto mais próximos são os valores obtidos, mais semelhantes são os animais em questão.

Assim, o algoritmo de clusterização kmeans observa, dentre todos os indivíduos que foram inseridos na decomposição SVD, quais devem ser agrupados juntos e quais devem ser agrupados separados. Não obstante, o kmeans deve receber, também, a quantidade de grupos em que a separação deve ser procedida. Contudo, tal número não foi difícil de se obter, pois pode-se facilmente perceber a existência de quatro grupos bem definidos na já apresentada visualização das entidades no espaço R^3.

4. Conclusões

O trabalho mostrou-se bastante interessante para visualizar o poder das técnicas de mineração em comparação aos algoritmos clássicos de computação. É evidente que os algoritmos clássicos ainda são muito importantes, sobretudo quando aplicados aos contextos para os quais foram desenvolvidos. Contudo, para o caso de uso de recuperação da informação, a decomposição SVD mostrou-se muito poderosa, sendo capaz de recuperar informações que ficariam desconhecidas a depender das técnicas empregadas para análise.

Além disso, já com relação à segunda parte do trabalho, é encantador observar as aplicações dessas técnicas na área de bioinformática. Em particular, é bastante poderoso ser capaz de encontrar semelhanças entre as sequências genômicas utilizando técnicas de mineração de dados e fundamentá-las em conhecimentos biológicos.

Ademais, é excepcionalmente estimulante imaginar outras possibilidades da aplicação dessas técnicas na bioinformática, já que boa parte dos estudos se debruçam sobre o entendimento dos principais compostos orgânicos presentes em nosso cotidiano. Assim, certamente, a mineração de dados pode ser bastante útil para que se estude proteínas ou sequências genômicas ainda inexploradas ou, até mesmo, trazer uma visão diferenciada a certos problemas biológicos não resolvidos ou explicados, por exemplo.

5. Referências Bibliográficas

- Video: Singular Value Decomposition (SVD): Overview. URL para acesso: https://youtu.be/gXbThCXjZFM
- Video: Singular Value Decomposition (the SVD). URL para acesso: https://youtu.be/mBcLRGuAFUk
- Agglomerative hierarchical cluster tree - MATLAB linkage. URL para acesso: https://www.mathworks.com/help/stats/linkage.html

6. Anexos

a. Trabalho 1

Script criado durante a aula: Exemplo2.m:

```
disp('Exemplo do trabalho Google')
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];

[T, S, D] = svd(A);
Combinacoes = S*D';
x = Combinacoes(1,:);
y = Combinacoes(2,:);
plot(x, y, '^r')
grid on
hold on
plot(0, 0, 'sb')
q = [0 0 0 0 0 0 0 1 1 1]';
T2 = T(:, 1:2);
qtil = T2'*q;
plot(qtil(1),qtil(2), 'sk')
for i = 1:5
      istr = num2str(i);
      text(x(i), y(i), ['  Doc' istr])
end
text(qtil(1), qtil(2), ['  consulta'])
title(' Exemplo de recuperação')
xlabel('Uso do padrão 1')
ylabel('Uso do segundo padrão')
```

Comando para verificação que A=T*S*D':

```
>> norm(A - T*S*D', 2)
ans =  2.8812e-15
```

Conforme esperado, o resultado da operação é bastante próximo de 0, o que embasa todo o referencial teórico usado no trabalho.

b. Trabalho 2

Além dos scripts “criapos6.m”, “inversa6.m”, “montaA6.m” e “slidwindow6.m” fornecdos pelo professor, foram executados os seguintes comandos no MATLAB:

```
load primates
[nomes, A] = montaA6(primates);
[T, S, D] = svds(A, 12);
```

```
s = diag(S);
plot(s, '*');
s = s*(1/sum(s));
plot(s, '*');
Aux = S*D';
Combinacoes = S*D';
x = Combinacoes(1,:);
y = Combinacoes(2,:);
z = Combinacoes(3,:);
plot3(x, y, z, '^r');
arvore = linkage(transpose(Aux (1:3, :)), 'average');
dendrogram(arvore, 'Orientation', 'left', 'Labels',
{primates.Header});
grupos = kmeans(transpose(Aux(1:3, :)), 4);
```