MC886 - Algoritmos Não Supervisionados

Naomi Takemoto
RA 184849
naomitkm1@gmail.com

Thiago Dong Chen
RA 187560
thiagodchen@gmail.com

I. Introdução

A. Objetivos

Descobrir estruturas organizacionais em um *dataset* utilizando técnicas de aprendizado não supervisionado. O conjunto de dados consiste de *posts* com o tema saúde coletados em 2015. [1]

II. Teoria

A. K-means

É um algoritmo de clusterização, baseado na aplicação de dois passos principais:

1) Atribuição de elementos a um *Cluster*
2) Ajuste das posições dos centróides, cada um destes define um agrupamento que idealmente possuem alta coesão interna e baixa coesão com elementos de outros grupos.

O número de *clusters* (k) deve ser fornecido previamente. Para determinar o melhor valor dessa variável utiliza-se normalmente a técnica da "curva do cotovelo"que consiste na análise do gráfico de função de custo *vs* k em busca de um vértice (cotovelo).

B. Principal Component Analysis (PCA)

Principal Component Analysis é um método de redução de dimensionalidade que utiliza transformações ortogonais de vetores que projeta o dado em uma dimensão menor. Em *Machine Learning*, essa redução de dimensionalidade tem como objetivo diminuir o número de *features*, acelerando o processamento dos métodos de classificação e clusterização, preservando parte das informações dos dados iniciais.

C. DBSCAN

DBSCAN é algoritmo de clusterização baseado na densidade, ele agrupa os vizinhos mais próximos em um mesmo *clusters*. O DBSCAN destaca ao marcar os *outliers* em regiões com pouca densidade. Além disso, não é necessário determinar a quantidade de *clusters* como o *k-means*.

III. Metodologia

A. Processamento dos Dados

A limpeza dos dados envolveu:

- Remover pontuação
- Remover dígitos
- Remover palavras precedidas por @ (no Twitter essas palavras são menções a outros usuários).

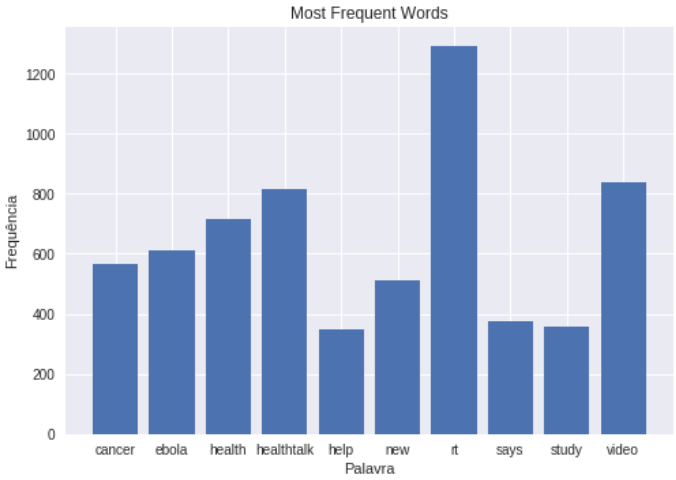

Figura 1. Gráfico das 10 palavras mais frequentes

- Remover palavras recorrentes na lingua inglesa que agregam pouco significado (*stop words*), como por exemplo conectivos.
- Transformar o texto de modo que todas as letras se tornem minúsculas.

B. Montar a Bag of Words

Depois da limpeza dos textos, aplicou-se a função *CountVectorizer* da biblioteca *Scikit-Learn* que gerou uma matriz *Bag of Words*, na qual cada linha representa um *post* e cada coluna um vocábulo. A partir da aplicação desse método obtém-se:

1) Vocabulário de palavras que aparecem no documento como um todo.
2) Uma medida da frequência de cada termo.

A representação em *Bag of Words* tem a vantagem de ser bastante simples, no entanto se perdem registros relacionados à ordem das palavras em cada *post*. No caso de estudo desta tarefa, isso não necessariamente é um problema, pois é possível extrair informações pela medida de frequência dos termos. A figura 1 por exemplo mostra um gráfico das palavras mais recorrentes nos *tweets* analisados. Com ela é possível perceber que o termo "ebola"está entre os 10 mais citados, tomando como referência a data de coleta dos dados, 2015, percebe-se que o tema era relevante na época, pois coincide com o período de surto da doença (final de 2013 a 2015). [2]

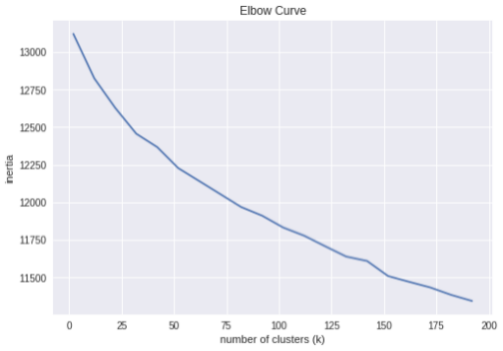

Figura 2. Gráfico do erro pelo número de *clusters*

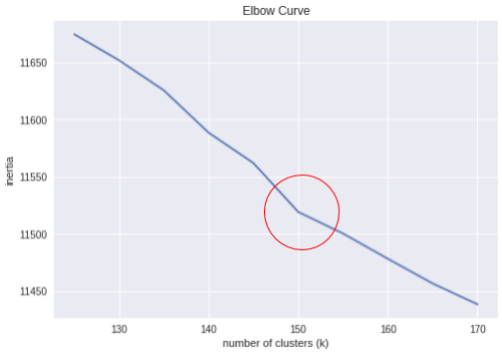

Figura 3. Gráfico identificação da curva do cotovelo

Um processamento adicional utilizado consistiu na aplicação da função TfidfTransformer [4]. O objetivo de utilizar essa função, em detrimento da frequência absoluta de cada termo é diminuir o impacto de *tokens* que ocorrem com alta frequência, pois empiricamente são menos informativos do que palavras que ocorrem poucas vezes.

IV. Resultados

A. K-Means

Na aplicação do *k-means*, foram tomados alguns cuidados, uma vez que o desempenho deste algoritmo é bastante sensível à inicialização dos centroides. Para achar o número ideal de *clusters* foi necessário minimizar esse efeito, para tanto, o número de vezes em que o algoritmo inicializava com os centroides em posições diferentes foi colocado em 10, sendo que o melhor resultado é tomado como o o que obteve o menor erro dentre essas tentativas. Para fazer os experimentos consistentes, foi fixada uma *seed* para o *random state*. Uma das métricas usadas para medir a qualidade dos agrupamentos encontrados foi o coeficiente de silhueta, que mede o quanto os elementos de um grupo são coesos entre si contraste a outros grupos. Esse coeficiente varia de -1 a 1, quando o valor está próximo de -1, há um indicativo de que o elemento foi atribuído para o *cluster* errado, quando o está próximo de 0 há a indicação de que o elemento está na fronteira entre dois *grupos* e valores próximos de 1 são os desejáveis. Um problema encontrado com o uso dessa métrica para avaliar os *clusters* é que para *datasets* com alta dimensionalidade, como é o caso de textos, o valor tende a ser baixo por conta do fenômeno conhecido como "Curse of Dimensionality". [5]. Para escolher o melhor valor k, aplicou-se o *k-means* para múltiplos valores no intervalo de 2 a 200, plotando o gráfico de erro (inércia - distância euclidiana) pelo número de grupos.2 Depois, observando o gráfico percebeu-se que um possível cotovelo estava entre as posições 130 e 170, como mostra a figura 3.

Como a clusterização realizada pelo *k-means* é um processo não supervisionado, a verificação da qualidade semântica dos grupos ocorreu analisando-se os elementos pertencentes a cada um deles. Examinando o conteúdo dos grupos gerados pra k = 150, foi possível perceber que eram agrupados por possuir ou não uma ou mais palavras chaves. Posteriormente, listou-se as palavras mais frequentes nos *tweets* de cada agrupamento. Os principais vocábulos são: heart, sex, million, chocolate, look, ways, worst,sugar,changed, diet, etc. Muitos dos quais possuem relação direta com conceitos de saúde.

B. Principal Component Analysis (PCA)

A utilização do PCA nesse trabalho tem como objetivo aumentar a velocidade de processamento do *k-means*, já que terá menos *features* ao reduzir a dimensionalidade da representação dos tweets (*bag-of-words*) com 11400 *features* para 2, 1000, 3000, 5000 *features*.

Para isso, o dado foi reduzido com o *singular-value decomposition* (SVD). SVD é uma fatorização de matriz que decompõe a matriz inicial em autovalores e autovetores. Esses autovalores e autovetores permitem o cálculo das componentes principais do PCA.

Inicialmente, o algoritmo do SVD foi executado com apenas 2 componentes, para permitir a visualização, ver Figura 4, e ver relações entre os dados. Entretanto, os clusters mostraram bem próximos e aparenta não ter realizado uma clusterização adequada. Alem disso, a soma das variâncias foi extremamente baixa de 2.8%, significando que houve uma grande perda das informações na redução de 11400 *features* para 2 *features*, que pode ser uma explicação para a figura.

Executando o SVD para 5000 componentes, a soma das variâncias foi de 96.2%, o que significa que reduzindo aproximadamente pela metade das *features*, houve apenas uma perda de 4% de informação do *bag-of-words*.

Tabela I
Número de componentes no PCA e o impacto no tempo necessário para realizar o k-means

Número de componentes	***Tempo para realizar o SVD (s)***	***Tempo para realizar o k-means (s)***	***Tempo total (s)***	***Métrica silhouette***	***Soma da variância (%)***
2	0.077	65.66	65.73	-0.02	2.8
1000	23.92	1011.72	1035.64	-0.01	69.1
3000	176.38	3872.34	4048.72	-0.00	89.5
5000	501.22	6299.60	6800.82	0.20	96.2

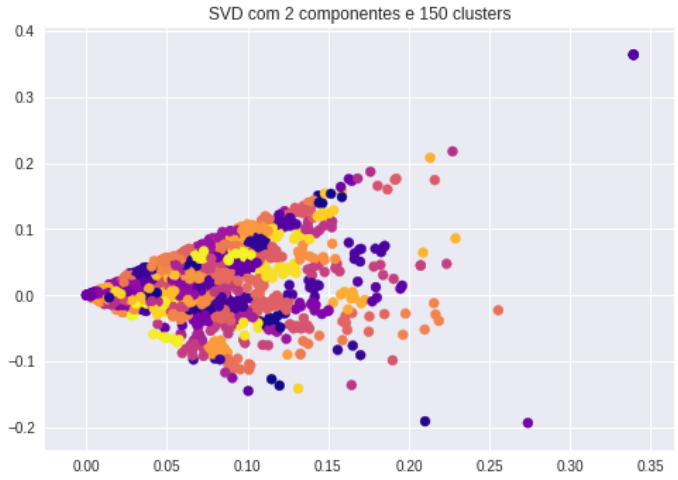

Figura 4. SVD com 2 componentes e 150 clusteres

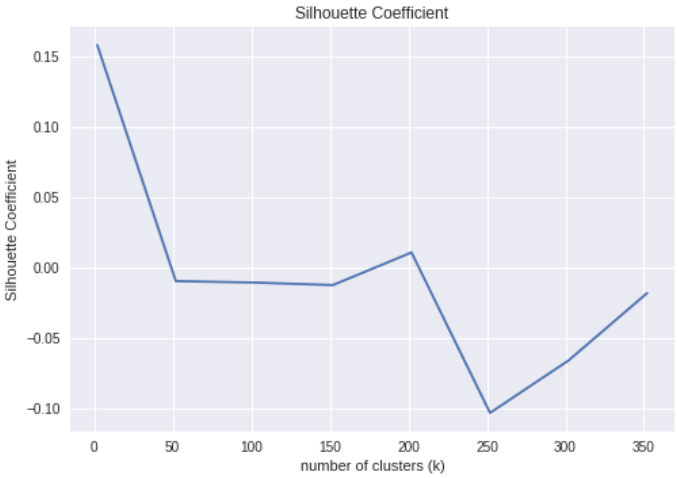

Figura 5. Gráfico de silhouette para 1000 componentes

Analisando a Tabela I, podemos ver que a execução do SVD junto com o *k-means* fica muito custoso ao aumentar o números de componentes do SVD. Apenas no caso em que preservamos duas componentes principais no SVD houve aumento no desempenho no quesito de velocidade. Entretanto com apenas duas componentes, a preservação da informação do dado do *bag-of-words* é baixa.

Mesmo diminuindo o número de dimensões com o SVD, o desempenho em tempo do *k-means* no é mais lento (534.53s) para o dado não reduzido. Esse fato é contra-intuitivo já que os parâmetros são os mesmos.

Selecionando 5000 componentes principais e depois aplicando o k-means, selecionando o k pelo método do *elbow*, o algoritmo obteve *clusters*, que haviam relações semânticas dos tweets para os mesmos *clusters*.

Observando o gráfico 5 é possível observar que em certas situações selecionar o número de *clusters* pelo coeficiente de silhouette nem sempre é uma boa opção. Nesse se selecionar o k = 2 após reduzir de dimensionalidade, teríamos apenas 2 *clusters*. Com apenas 2 *clusters*, o *dataset* dos tweets não estariam bem separados, apesar da métrica nos dizer o contrário.

C. DBSCAN

O algoritmo DBSCAN comparado com o *k-means*, a resposta é mais veloz e os *clusters* gerados são bem sensíveis aos parâmetros de máxima distância para considerar vizinhos (eps) e número mínimo de amostras na vizinhança para considerar um *core point*, entretanto não é necessário parametrizar o número de *clusters*.

Para eps=1.2 e mínimo de amostras=51, o DBSCAN gerou *clusters* que possuem valores semânticos, ver Tabela II. Por outro lado, se selecionar parâmetros não adequados para o problema, os *clusters* podem ter poucos tweets ou agrupamentos que não fazem sentido.

Tabela II
ALGUNS ASSUNTOS AGRUPADOS PELO DBSCAN

Cluster 0	***Cluster 1***	***Cluster 2***	***Cluster 3***
Alergia	Coisas que amam	Surto de ebola	Perda de peso

V. CONCLUSÃO E TRABALHOS FUTUROS

O algoritmo de *clusterização k-means* é uma boa forma de agrupar automaticamente o dado, entretanto escolher o k pode ser um trabalho difícil, mesmo com o auxílio do método do cotovelo para determinar o k. A métrica de *silhouette* mensura a coesão e a separação entre os *clusters*, mas nem sempre é uma métrica apropriada para escolher o k adequado, pois em alguns casos, o coeficiente de *silhouette* apresenta mais apropriado para valores como k = 2, ver Figura 5, mas ao olhar os tweets nos *clusters*, não há relação semântica.

Comparando-se o *k-means* com o DBSCAN, observou-se que este último é mais rápido e oferece a vantagem de não ter que realizar a busca por um k. Ao gerar 175 *clusters*, a partir deste método observou-se que eles também eram agrupados por conterem ou não determinadas palavras, em particular foram encontrados agrupamentos com as palavras chaves ebola e rt (*retweet*).

REFERÊNCIAS

[1] Assignment 3.Disponível em <http://www.ic.unicamp.br/ sandra/pdf/2018s2-mc886-mo444-assignment-03.pdf> Acessado em: 05 de novembro de 2018.
[2] Surto de ébola na África Ocidental. Disponível em <https://pt.wikipedia.org/wiki/Surto_de_%C3%A9bola_na_%C3%81frica_Ocidental>
[3] SVD. Disponível em <http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html> Acessado em: 05 de novembro de 2018.
[4] TfidfTransformer. Disponível em: <http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html>. Acessado em: 05 de novembro de 2018.
[5] Clustering text documents using k-means. Disponível em http://scikit-learn.org/0.19/auto_examples/text/document_clustering.html>. Acessado em: 6 de novembro de 2018.