Trabalho 3 - MC886 Aprendizado de Máquina

Leo Yuuki Omori Omi
138684
leoyuuki@gmail.com

João Pedro Ramos Lopes
139546
jpedrorl@gmail.com

I. Introdução

Neste trabalho, tivemos que agrupar vários textos utilizando utilizando técnicas de Aprendizado não-supervisionado. Utilizando dados em formato *bag-of-words* utilizamos algumas técnicas para tentar achar agrupamentos que fizessem algum sentido.

A. Soluções Propostas

1) DBSCAN: Tentamos utilizar a técninca Density-Based Spatial Clustering of Applications with Noise (DBSCAN), já implementado na biblioteca Scikit Learn. Parecia uma boa escolha já que o algoritmo agrupa pontos próximos um dos outros, sem precisar receber como parâmetro o número de grupos (que é um dos resultados que gostaríamos). Tentamos variar os parâmetros eps e *min_samples*, no entanto, inicialmente, não conseguimos achar uma configuração que desse um resultado que parecia coerente, portanto desistimos de continuar experimentos com esta técnica. Possivelmente, a distribuição dos dados não favorecia o DBSCAN ou pode ter sido uma escolha infortúnia do parêmtros.

2) Hierarchical Clustering: Foi utilizado a implementação de Hierarchical clustering da biblioteca SciPy para agrupar os textos utlizando a função fclusterdata. Ela recebe como parâmetros os dados, uma limiar de distância máxima entre dois pontos, e um critério de comparação (neste caso foi utilizada a distância). Assim como o DBSCAN, o número de grupos não é uma das entradas para a função, e o algoritmo trata de achar estes grupos.

3) K-means: Foi utilizado o K-means implementado no Scikit learn. Neste caso, precisamos escolher um número de grupos e depois avaliar de alguma forma se a escolha foi boa, diferente das propostas anteriores.

B. Experimentos

Para todos os experimentos foi utilizado o Silhouette Coefficient como uma forma de avaliar a performance entre os métodos.

1) Modelos iniciais: Para analisar o número de grupos presentes no problema, utilizamos como modelo base o Elbow Method, variando esse número no intervalo [10, 1000] com passo 10 encontramos o gráfico disposto na figura 1.

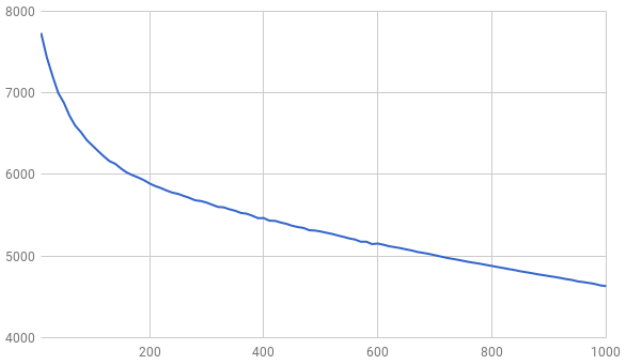

Figure 1. Número de Clusters x Erro médio

Utilizando o modelo de k-médias com 3 reinícios, 10 iterações e inicialização aleatória *k-means++*. Com base no gráfico gerado, escolhemos o range de classes [50, 400] a ser submetido ao método de silhueta. Utilizando passo 50 encontramos o gráfico da figura 2, que representa a pontuação para tais valores de k.

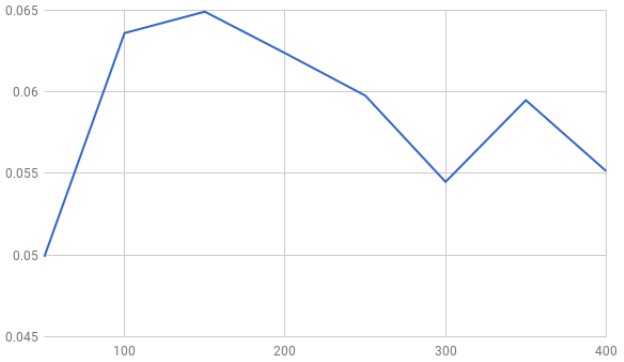

Figure 2. Silhueta no intervalo [50, 400] com passo 50

Finalmente, para determinar o número de classes a ser utilizado pelo modelo, foi percorrido o intervalo [125, 175] e desenhado o gráfico da pontuação encontrada pelo método de silhueta, representado na figura 3. Dessa forma, foram emscolhidas 151 clusters para o problema.

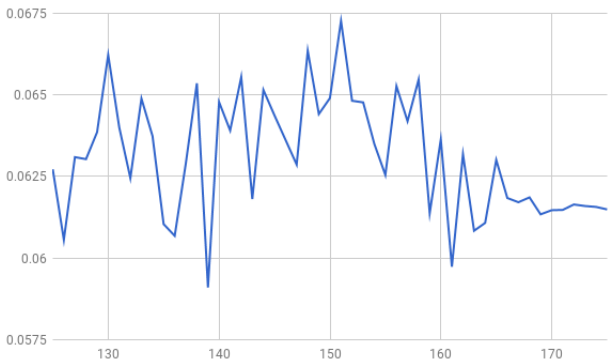

Figure 3. Silhueta no intervalo [125, 175]

C. Resultados

1) K-Means: Para analisar o modelo final do k-means, utilizou-se a mesma configuração que anteriormente, com máximo de 300 iterações e $k = 151$. Com estes parêmetros foi obtido um coeficiente de 0.067. Para checar os resultados encontrados, escolheu-se um agrupamento aleatório e ordenou-se os pontos pertencentes a ele pela distância euclideana. De forma amostral, checamos alguns clusters e percebemos que o assunto geral dos documentos mais próximos foram:

- Cluster 0: pedidos de ajuda e sugestões/dicas em diversos tópicos (carros, hardware
- Cluster 17: discussões relacionadas a *Hockey*
- Cluster 31: 3 grupos principais: carros, esportes e eletrônicos
- Cluster 42: Uma longa discussão sobre o mesmo assunto: *Who's next? Mormons and Jews?*
- Cluster 67: Questões relacionadas a hardware/software da Microsoft/Apple
- Cluster 123: Alguns documentos são relacionados a vendas, outros relacionados a direção de veículos

2) Hierarchical Clustering: Foram 'chutados' alguns valores iniciais para o limiar do algoritmo. Assim, percebemos que valores menores que 1 estavam tendo um Silhoutte Coefficient melhor, portanto foi utilizado um laço para achar valores com melhores resultados. Foi utilizado metade do conjunto de dados por causa do tempo de execução longo. O coeficiente achado foi de 0.055 para o limiar de 0.7, assim, este valor foi utilizado. No entanto, ao analisarmos melhor o resultado, foram criados mais 8500 grupos para as 9962 entradas de dados, ou seja, muitos dos grupos só possuiam um ponto. Analisando alguns dos clusters que tinham multiplos pontos, achamos grupos coerentes:

- Cluster 2: Placares e pontuações da MLB
- Cluster 61: *Accounts of Anti-Armenian Human Right Violations in Azerbaijan*
- Cluster 31: Venda de revistas Playboy do ex colega de quarto

Em clusters que possuem múltiplos pontos, encontramos texto muito parecidos ou iguais. Ou seja, são agrupados textos apenas se extremamente similares, o que podemos até considerar como mérito desta solução. No entanto, como ela forma um número enorme de grupos, esta solução não possui muita generalidade.

II. PCA

Considerando a solução do K-means como a melhor, aplicamos o PCA nos dados. Rodamos alguns testes usando os parâmetros achados e verificamos o Silhoutte Coefficient obtido para alguns valores de dimensionalidade do PCA. Segue uma tabela com alguns dos valores obtidos:

Nº Componentes	Silhoutte
1000	-0.0408837480849
500	0.0116988599587
250	0.0796884585275
100	0.15327313388
10	0.14025108712

Podemos ver que valores maiores das dimensões estão diminuindo o desempenho da solução de acordo com o Silhoutte Coefficient. No entando, conforme o número de dimensões no PCA foi diminuido, chegamos a um valor maior que o original, sendo que nos valores testados, obtemos o melhor resultado com 100 dimensões.

A vanatagem de utilizar o PCA neste caso foi que, conseguimos ganhar em desempenho computacional pois estamos rodando K-Means com um número menor de dados, assim diminuindo o tempo de execução e até melhoramos o desempenho como visto pelo aumento do coeficiente.

III. Conclusões

Não ter uma resposta correta torna muito diferente e difícil a resolução de problemas. Aprendizado não-supervisionado requere uma análise do tipo de dado ou alguma técnica ou heurística para auxiliar na resolução dos problemas. No caso de nossos experimentos do K-Means, conseguimos achar um resultado coerente usando uma técnica conhecida, o Elbow Method. Sendo que DBSCAN, que parece que uma solução até mais robusta, nos trouxe dificuldades para calibrar seus parâmetros.

Não ter uma resposta exata para o problema também traz dificuldade para verificar o desempenho. Utilizamos o Silhoutte Coefficient e uma pequena checagem de alguns casos, mostrando alguns resultados bons.

Ainda assim, e nossos experimentos conseguimos criar modelos relativamente simples que conseguiram servir algum propósito. O K-means conseguiu de fato agrupar de uma manera satisfatória em vário grupos, enquanto o Hierarchical Clustering, apesar de ser muito específico, consegue ter algum tipo de propósito. Além disso, vimos como o PCA pode melhorar um modelo, não só diminuindo o tempo de execução, como melhorando os resultados.

References