Giovanni Alvarenga Silva 726525
Lucas Granja Toniello 726560
Vanderlei Jesus de Andrade 726590

Introdução

O trabalho consistiu na implementação de três algoritmos de agrupamento: k-médias, single-link e average-link. A partir disso, os algoritmos foram utilizados para agrupar três conjuntos de objetos. A saída de cada algoritmo foi então transformada em gráficos e, em seguida, estes gráficos foram comparados com os gráficos reais dado pela professora.

Uma análise então pôde ser feita com base na comparação dos resultados entre os algoritmos, assim como entre cada algoritmo e o resultado real. Então, foi utilizado o Índice Rand Ajustado, implementado por nós, para calcular a similaridade entre clusters.

Uma análise sobre estes resultados foi feita, buscando encontrar a melhor partição para cada algoritmo.

Algoritmos

O primeiro passo para iniciar o trabalho foi implementar os três algoritmos necessários. O algoritmo k-médias foi implementado em Python. Seu funcionamento é baseado na classificação de um determinado número pré-definido K de clusters e de iterações, e tem como função de classificação a distância entre o objeto e o centróide. A cada iteração, o centróide dos clusters é atualizado e os cálculos são refeitos.

Em seguida, os outros dois algoritmos, single-link e average-link, também foram implementados em python. O single-link forma a matriz de distâncias calculando todas as distâncias entre todos o objetos em busca da menor delas; os dois objetos mais próximos formam um cluster; e a matriz de distâncias é atualizada com base na distância entre esse cluster e os outros objetos. Em cada iteração será formado um cluster entre os dois objetos ou clusters com menor distâncias entre eles.

Já o average-link funciona de forma semelhante ao single-link, exceto na forma em que calcula as distâncias, onde a distância entre dois clusters se dá pela média entre os objetos dos clusters.

Os três algoritmos recebem como entrada os arquivos c2ds1-2sp.txt, c2ds3-2g.txt e monkey.txt, cada um representando um conjunto de objetos. Como saída, eles produzem um arquivo com as partições com os números de clusters fornecidos separando eles por quebras de linha.

Para formatar a saída no formato correto, foi desenvolvido um algoritmo secundário chamado formatador.py que usa como base a saída dos algoritmos para gerar saídas .clu.

Rand

O Índice Rand Ajustado foi implementado por nós para este trabalho. Sua implementação foi feita em Python. Ele funciona gerando uma matriz de confusão a partir dos valores reais e dos valores previstos e, em seguida, calcula o índice Rand, que gera a similaridade entre as partições reais e previstas que foram produzidas.

Seja R o valor gerado pelo índice Rand Ajustado, $R \in \text{Reais t.q. } R \in [0,1]$. Quanto mais próximo de 1, melhor o agrupamento realizado.

Comparando Resultados

A partir da criação dos gráficos, da geração da matriz de confusão e o cálculo de similaridade do algoritmo Rand, é possível fazer uma comparação de cada algoritmo para identificar quais são mais semelhantes com os gráficos reais.

Abaixo, encontram-se as comparações entre a partição real e a partição mais próxima encontrada por cada algum dos algoritmos:

Para o gráfico em espiral, o melhor algoritmo foi o Single-Link.

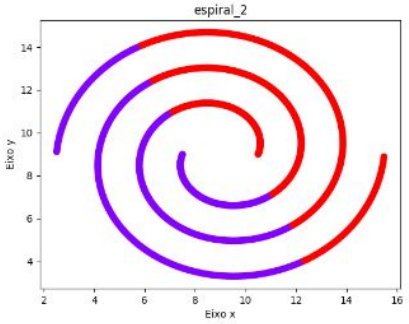

Figura 1 - Average link

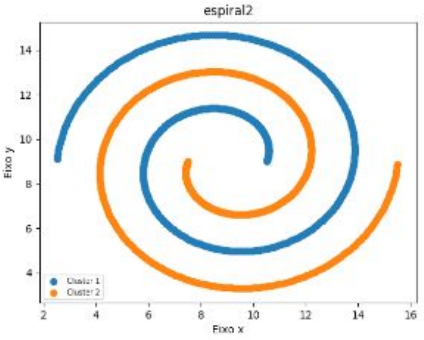

Figura 2 - Single-Link

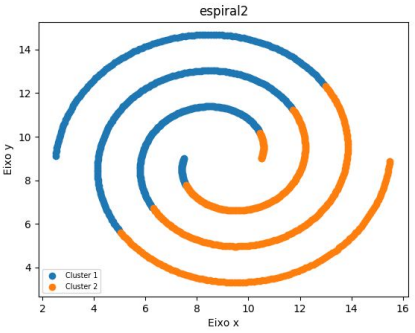

Figura 3 - K-médias

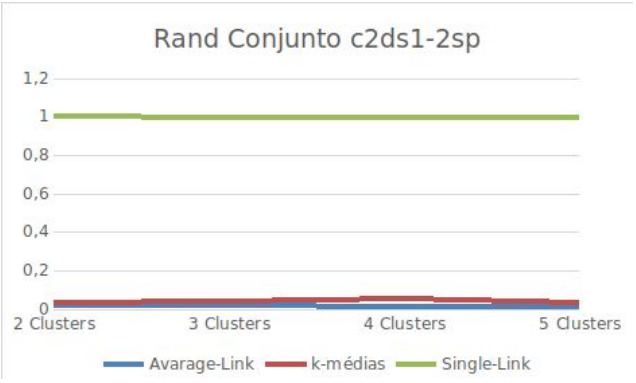

Figura 4 – Índice Rand para conjunto de dados c2ds1-2sp

Através do cálculo do índice Rand, executado a partir do algoritmo do single-link, foi obtido índice Rand de 1,0 de similaridade para 2 partições e muito próximo de 1 para mais de 2 partições. Sendo assim é possível observar que o gráfico que foi gerado pelo algoritmo single-link é mais semelhante à partição real, quando comparado com os outros dois algoritmos, pelo fato de possuir um alto rendimento ao separar conjuntos de dados englobados em clusters.

O fato do single-link ser mais similar a essa partição se dá pois os clusters são encadeados, ou seja, qualquer ponto em comum em um cluster está mais próximo ao mesmo cluster do que qualquer outro ponto que não pertence a esse cluster. Como temos uma partição com dois clusters no gráfico Single-Link, é possível observar que um ponto que pertence ao cluster1 não está próximo de um ponto do cluster2.

Já para o gráfico com dois círculos, o algoritmo que obteve melhor desempenho foi o K-médias.

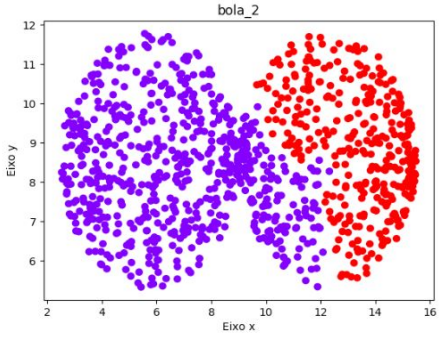

Figura 5 - Average-Link

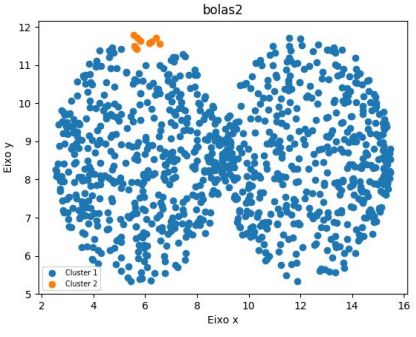

Figura 6 - Single-Link

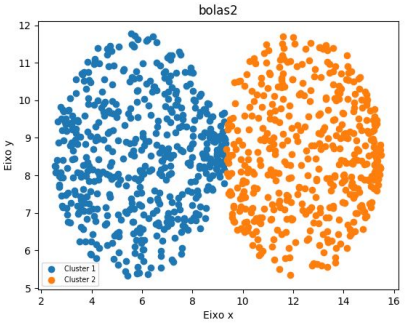

Figura 7 - K-médias

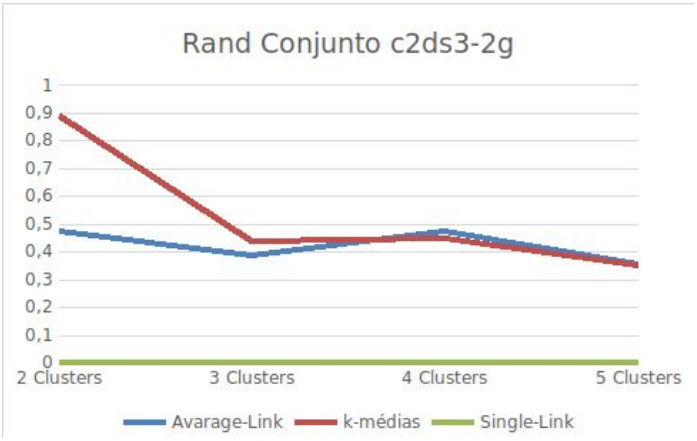

Figura 8 – Índice Rand para conjunto de dados c2ds3-2g

Neste caso o algoritmo K-médias obteve um índice Rand de aproximadamente 0,8872 para 2 Clusters, o que indica alta similaridade. Isso se deve ao fato de que o k-médias possui alto rendimento ao separar 2 conjuntos de dados englobados em clusters, como no caso acima, onde não há nenhuma separação entre os dados.

Os outros algoritmos obtém resultados muito ruins neste exemplo, com o single-link em particular apresentando índice rand próximo de 0. Devido a aglomeração dos dados, o single-link não consegue distinguir quais dados pertencem a cada cluster, pois faz seus pareamentos essencialmente de maneira aleatória.

Por fim, para o gráfico do macaco, o algoritmo mais efetivo foi, novamente, o single-link.

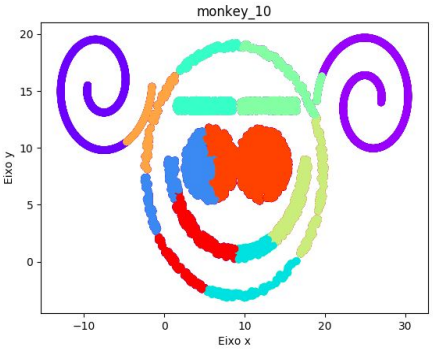

Figura 9 - Average-Link

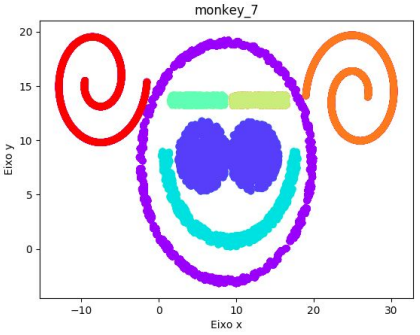

Figura 10 - Single-Link

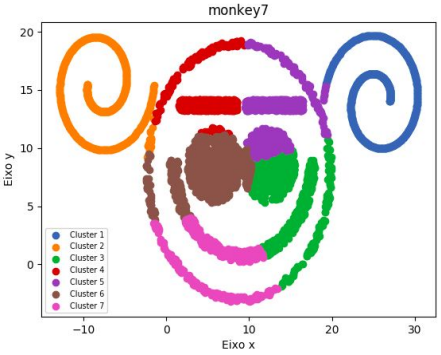

Figura 11 - K-médias

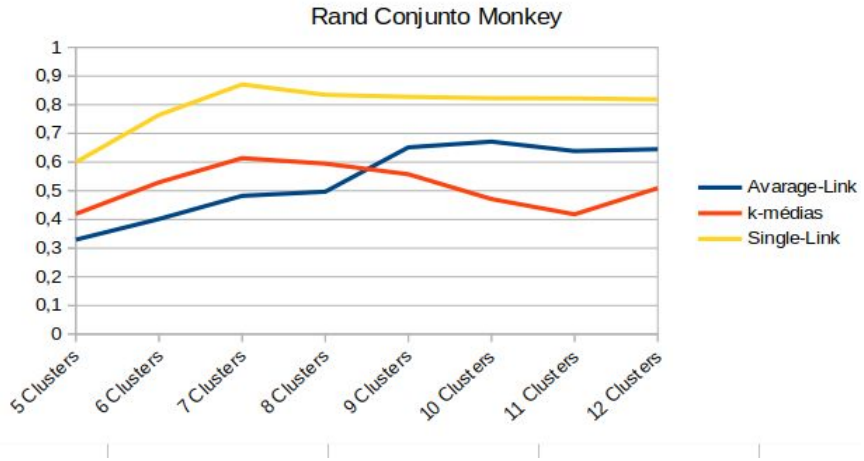

Figura 12 - Índice Rand para conjunto de dados monkey

Neste caso, o algoritmo Single-Link obteve um índice Rand de aproximadamente 0,87077 com 7 clusters. Como foi discutido no gráfico de espiral, isso se deve ao fato do algoritmo conseguir agrupar conjuntos de dados mais próximos do mesmo cluster. Os outros algoritmos tiveram índice Rand semelhantes entre si: o k-médias obteve 0,6137, e o average-link 0,6713, que são índices ruins de aproximação para ver qual é o melhor algoritmo.

Conclusão

A partir dos gráficos obtidos pelos algoritmos, é importante observar que cada algoritmo tem sua similaridade de acordo com o agrupamento dos cluster.

Para os casos aqui apresentados, o algoritmo single-link apresentou um comportamento mais próximo ao esperado em duas ocasiões. Por outro lado, o algoritmo average-link se mostrou vantajoso somente no gráfico de círculos com 4 clusters. Entretanto, podemos ver no gráfico de círculos que o single-link, até então supostamente o melhor algoritmo para a tarefa, obteve o pior índice Rand de todos: 0,00036.

Com isso, fica claro que os três algoritmos são úteis para situações mais específicas: o single-link é útil para partições com clusters separados; já o k-médias, para partições com clusters aglomerados; por fim, o average-link, apesar de não se mostrar o melhor algoritmo para nenhuma solução, obteve resultados medianos sem muitos valores fora da média.

Concluímos então que a disposição dos objetos influenciam consideravelmente no resultado esperado dos algoritmos, tornando-os complementares e de igual importância para a classificação dos dados.