Relatório Final

Bruno Peres
Carlos Tadeu
Henrique Kodama
Vinícius Salinas

Janeiro 2019

1 Introdução

Este relatório tem como objetivo descrever e, principalmente, discutir os resultados retirados da execução dos algoritmos com a base de dados disponibilizada. Os algoritmos foram desenvolvidos em Python2.7, na qual o cálculo das distâncias foram feitos pelo método euclideano, e para interpretar a qualidade das partições geradas foi utilizada uma versão pronta do índice rand ajustado (ARI), implementada na biblioteca `sklearn` disponível para Python. Todos os algoritmos podem ser encontrados no repositório `https://github.com/hskodama/IA-Trabalho-Final.git`, portanto não serão colocados no relatório. Além disso, todas as figuras, tais como as separações reais dos dados, agrupamentos encontrados pelos algoritmos e os dendogramas gerados se encontram na planilha excel no mesmo url acima, ou na pasta de Resultados. Abaixo estão algumas informações relevantes em relação às bases de dados:

Database	**k mínimo**	**k máximo**	**Quantidade de pontos**	**k real utilizado**
c2ds1-2sp.txt	2	5	1000	2
c2ds3-2g.txt	2	5	1000	2
monkey.txt	5	12	4000	8

Table 1: Informações sobre as bases de dados

2 Distribuições reais dos dados

Todos os algoritmos encontrados nesse trabalho, k-médias, single-link e average-link utilizam a mesma base de dados para comparar os agrupamentos encontrados. Entretanto, para cada conjunto de dados, a quantidade de clusters gerados são diferentes. Para as bases de dados `c2ds1-2sp.txt` e `c2ds3-2g.txt` foram consideradas 2 clusters e para a base `monkey.txt` foi considera 8 clusters, mas para todos os dados, foram executados com todas as quantidades de clusters entre k-mínimo e k-máximo.

3 Algoritmo K-médias

O algoritmo inicialmente recebe 3 entradas: a base de dados desejada e a quantidade máxima de iterações. Em seguida, calcula e gera todos os resultados com as quantidade de clusters possíveis. Vale ressaltar que para este algoritmo, apesar de utilizar a mesma quantidade de clusters, a qualidade das partições pode variar a cada execução, uma vez que as coordenadas de cada centroide são sempre geradas aleatoriamente.

3.1 Base de dados c2ds1-2sp.txt

Os centroides e índice gerados foram:

- C1 = (4.2361, 12.0072)
- C2 = (8.8265, 7.8954)
- ARI = 0.0073488402990068155

3.2 Base de dados c2ds3-2g.txt

Os centroides e índice gerados foram:

- C1 = (6.9023, 7.364)
- C2 = (15.3455, 9.5474)
- ARI = 0.4895350118726768

3.3 Base de dados monkey.txt

Os centroides e índice gerados foram:

- C1 = (12.9134, 12.1981)
- C2 = (-3.8948, 5.0439)
- C3 = (8.6448, -2.9729)
- C4 = (10.5368, 9.5718)
- C5 = (13.6858, 10.6673)
- C6 = (-6.5382, 16.9615)
- C7 = (30.1683, -2.8282)
- C8 = (-12.7568, 10.0774)
- ARI = 0.38374096538634284

4 Algoritmo single-link

O algoritmo recebe como entrada uma das bases de dados, e o intervalo de valores para o número de clusters. Em contrapartida ao algoritmo anterior, todos os valores são sempre os mesmos, contanto que o intervalo estipulado permaneça igual.

4.1 Índices gerados

Os índices rand ajustado gerados para cada base de dados foram:

- ARI c2ds1-2sp.txt = 1.0
- ARI c2ds3-2g.txt = 0.0003603746454815536
- ARI monkey.txt = 0.8344550575452158

5 Algoritmo average-link

Por fim, este algoritmo recebe as mesmas entradas do single-link.

5.1 Índices gerados

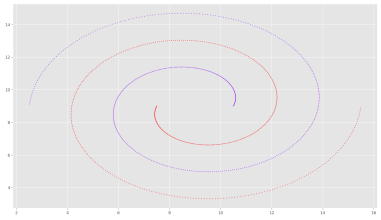
Os índices rand ajustado gerados para cada base de dados foram:

- ARI c2ds1-2sp.txt = 0.022125146292585184
- ARI c2ds3-2g.txt = 0.4728677045753165
- ARI monkey.txt = 0.4967584024740447

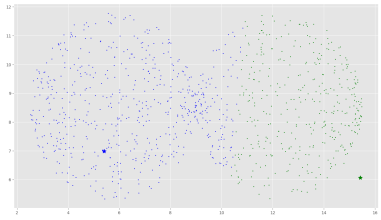
6 Conclusão

O índice rand ajustado compara duas partições geradas e assume a distribuição hiper-geométrica generalizada como modelo de randomização, tendo seu valor entre 0 e 1. O valor 1 indica que as duas partições são perfeitamente iguais, enquanto o valor 0 indica o contrário. Para esse índice, o valor 0 é esperado para clusters gerados aleatoriamente.

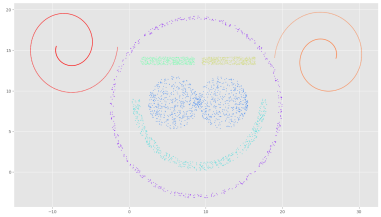
Com relação a execução do algoritmo k-médias, pode-se dizer que a qualidade das partições geradas, quando comparadas com as partições reais, é baixa, pois os pontos são gerados aleatoriamente, podendo resultar em clusters muito pequenos ou muito grandes. Além disso, pontos próximos um ao outro podem não estar no mesmo cluster, pois a divisão utiliza o centróide calculado a cada iteração para dividir os dados. Apesar do algoritmo gerar agrupamentos diferentes a cada iteração, as partições podem acabar sendo semelhantes às distribuições reais, assim como no resultado da segunda base de dados, na qual os centroides gerados contribuíram para uma maior semelhança na comparação,



(a) Single-link, ARI = 1,0



(b) K-médias, ARI = 0,59251



(c) Single-link, ARI = 0,87077

Figure 1: 2 Figures side by side

resultando em um $ARI = 0,59251$. Na maioria dos casos é desejado um valor ARI próximo a 1, e o algoritmo k-médias não é muito eficiente.

Por outro lado, tanto o algoritmo single-link, quando o average-link são métodos de agrupamento hierárquico. A única diferença consiste na forma em é tomada a decisão de quais clusters irão se juntar. Uma delas considera a menor distância e a outra considera a menor distância média entre os grupos. De modo geral, ambos os algoritmos tiveram índices maiores do que o k-médias, devido ao fato de levarem em consideração os pontos próximos, ou agrupamentos já existentes, sem gerar informações aleatoriamente a cada execução. Vale notar que foi obtido um índice perfeito 1, através do qual podemos sugerir que foi executado um algoritmo bastante semelhante ao single-link. Por fim, abaixo estão os agrupamentos com maior índice rand ajustado de cada arquivo base: