Inteligência Artificial

Thiago Henrique Leite da Silva, RA: 139920

AULA11: Exercício teórico Aprendizado não supervisionado

1) Execute o passo a passo do algoritmo complete linkage nos dados a seguir:

$$\mathbf{D}_1 = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & & & & \\ \boxed{2} & 0 & & & \\ 6 & 5 & 0 & & \\ 10 & 9 & 4 & 0 & \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix}$$

No algoritmo complete linkage, nós escolhemos a menor distância para saber quais nós juntar, e na hora de atualizarmos a matriz, pegamos o maior valor dentre os possíveis. Portanto, para estes dados, teríamos o seguinte passo a passo:

Começamos escolhendo a menor distância para fazer o primeiro agrupamento. Neste caso, a menor distância encontrada foi o 2, então juntamos 1 e 2.

	1	**2**	**3**	**4**	**5**
1	0				
2	**2**	0			
3	6	5	0		
4	10	9	4	0	
5	9	8	5	3	0

Agora temos o grupo 12, para atualizar os valores de d(3,12), por exemplo, escolheremos o maior valor entre as distâncias de d(3,1) e d(3,2), que é 6. E assim faremos para d(4,12) e d(5,12).

	12	**3**	**4**	**5**
12	0			
3	6	0		
4	10	4	0	
5	9	5	**3**	0

A menor distância da nova matriz obtida é 3, portanto, agruparemos agora 4 e 5.

	12	3	45
12	0		
3	6	0	
45	10	**5**	0

Tendo o grupo 45, seguiremos a mesma lógica acima para atualizar os valores da matriz. Também encontramos novamente o elemento com a menor distância, que foi o 5. Sendo assim, a última junção que faremos será de 3 com 45.

	12	345
12	0	
345	**10**	0

Por fim, caso realizássemos mais uma operação de agrupamento, teríamos o grupo 12345.

2) Execute o passo a passo do algoritmo k-means com k=3 nos dados abaixo a partir dos protótipos [6 6], [4 6] e [5 10]:

	A1	A2
1	1	2
2	2	1
3	1	1
4	2	2
5	8	9
6	9	8
7	9	9
8	8	8
9	1	15
10	2	15
11	1	14
12	2	14

Para me auxiliar na aplicação do algoritmo KMeans conforme pede o exercício, implementei um script em Ruby que estará anexado no classroom e que realiza os passos necessários no KMeans, posteriormente, coloquei os resultados obtidos em uma planilha.

O código é muito simples e de fácil interpretação. Dado um valor K de protótipos, formaremos K grupos, onde para cada um dos nossos pontos, calcularemos a distância euclidiana para os K protótipos, e iremos inserir o ponto no array do grupo com menor distância.

Estando todos os pontos em um dos K grupos, iremos redefinir os protótipos, onde cada uma das coordenadas X e Y serão as médias das mesmas coordenadas de todos os pontos do grupo. Após a redefinição dos protótipos, fazemos uma nova varredura em todos os pontos rearranjando cada um para o seu respectivo grupo.

Então, verificaremos se nossa divisão ficou balanceada, em caso positivo, o algoritmo se encerrou, caso contrário, repetimos o processo de correção dos protótipos e divisão dos pontos entre os grupos.

Para nosso protótipo inicial, as distâncias e divisão dos grupos ficaram da seguinte maneira:

Protótipos
(6, 6)
(4, 6)
(5, 10)

A1	A2	(6, 6)	(4, 6)	(5, 10)
1	2	6.4	5.0	8.94
2	1	6.4	5.39	9.49
1	1	7.07	5.83	9.85
2	2	5.66	4.47	8.54
8	9	3.61	5.0	3.16
9	8	3.61	5.39	4.47
9	9	4.24	5.83	4.12
8	8	2.83	4.47	3.61
1	15	10.3	9.49	6.4
2	15	9.85	9.22	5.83
1	14	9.43	8.54	5.66
2	14	8.94	8.25	5.0

(6, 6) 2 pontos
(4, 6) 4 pontos
(5, 10) 6 pontos

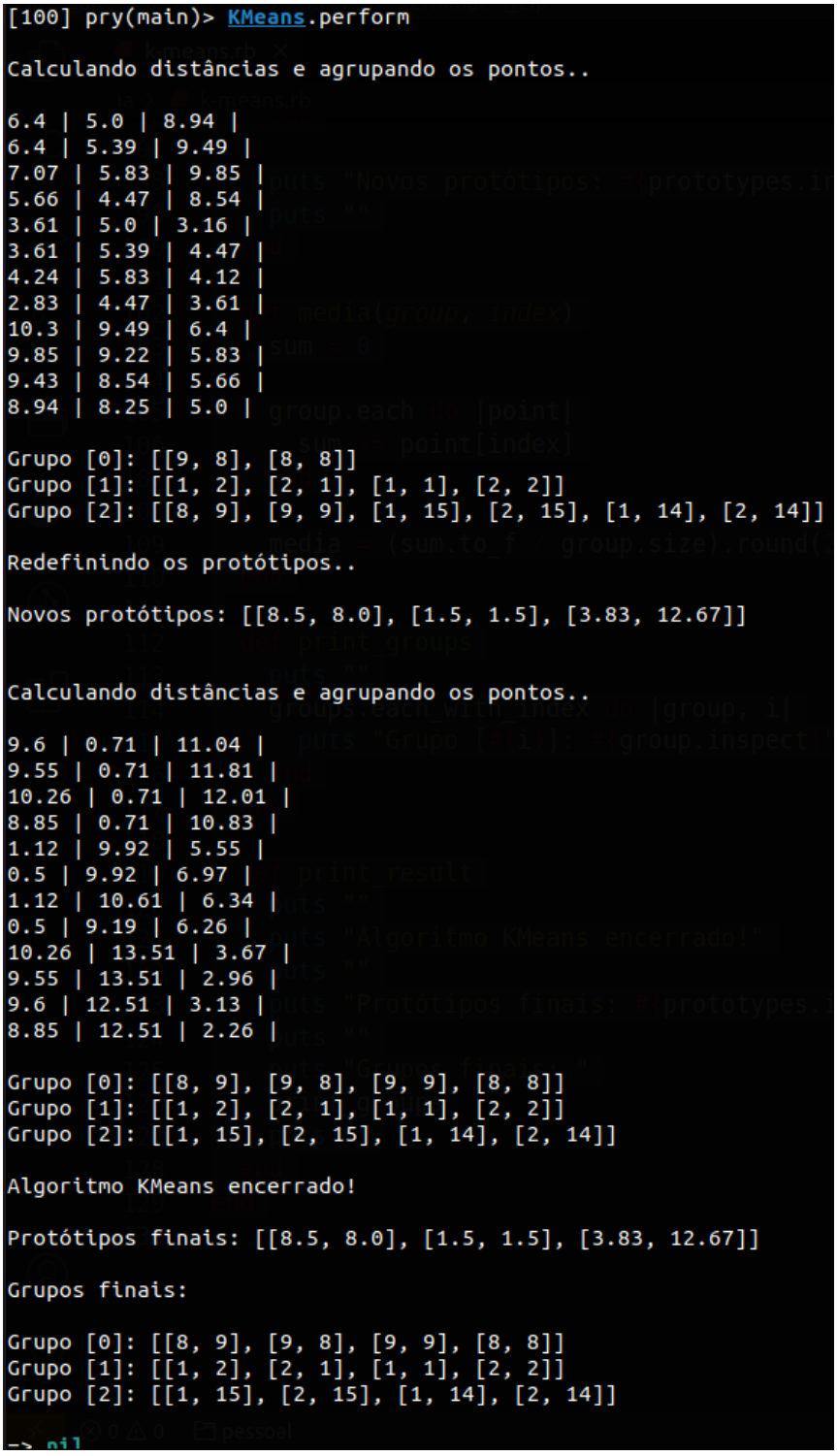
Após a redefinição dos protótipos, as distâncias e divisão dos grupos ficaram balanceadas:

Protótipos
(8.50, 8.00)
(1.50, 1.50)
(3.83, 12.67)

A1	A2	(6, 6)	(4, 6)	(5, 10)
1	2	9.6	0.71	11.04
2	1	9.55	0.71	11.81
1	1	10.26	0.71	12.01
2	2	8.85	0.71	10.83
8	9	1.12	9.92	5.55
9	8	0.5	9.92	6.97
9	9	1.12	10.61	6.34
8	8	0.5	9.19	6.26
1	15	10.26	13.51	3.67
2	15	9.55	13.51	2.96
1	14	9.6	12.51	3.13
2	14	8.85	12.51	2.26

(8.50, 8.00) 4 pontos = [8, 9] [9, 8] [9, 9] [8, 8]
(1.50, 1.50) 4 pontos = [1, 2] [2, 1] [1, 1] [2, 2]
(3.83, 12.67) 4 pontos = [1, 15] [2, 15] [1, 14] [2, 14]

Por fim, podemos comprovar os resultados com a saída do algoritmo implementado:



3) Considere a seguinte lista de compras de 6 clientes. Aplique o algoritmo Apriori considerando suporte >=2 e confiança > 60%

1 biscoito, cerveja, pão, salaminho

2 cerveja, couve, linguiça, pão, queijo

3 café, brócolis, couve , pão

4 brócolis, café, cerveja, couve, pão, salaminho

5 brócolis, café, couve, pão, refrigerante

6 couve, linguiça

Construindo a tabela de ocorrências:

Cliente	Biscoito	Cerveja	Pão	Salaminho	Couve	Linguiça	Queijo	Café	Brócolis	Refri
1	A	B	C	D						
2		B	C		E	F	G			
3			C		E			H	I	
4		B	C	D	E			H	I	
5			C		E			H	I	J
6					E	F				

Seguindo os passos vistos em aulas na aplicação do algoritmo Apriori, obtemos os seguintes itemsets:

sup >= 2

1 Itemset	sup
A	1
B	3
C	5
D	2
E	5
F	2
G	1
H	3
I	3
J	1

=>

2 Itemset	sup
{B,C}	3
{B,D}	2
{B,E}	2
{B,F}	1
{B,H}	1
{B,I}	1
{C,D}	2
{C,E}	4
{C,F}	1
{C,H}	3
{C,I}	3
{D,E}	1
{D,F}	0
{D,H}	1
{D,I}	1
{E,F}	2
{E,H}	3
{E,I}	3
{F,H}	0
{F,I}	0
{H,I}	3

=>

3 Itemset	sup
{B,C,D}	2
{B,C,E}	2
{B,C.F}	1
{B,C,H}	1
{B,C,I}	1
{B,D,E}	1
{B,D,F}	0
{B,D,H}	1
{B,D,I}	1
{B,E,F}	1
{B,E,H}	1
{B,E,I}	1
{C,D,E}	1
{C,D,F}	0
{C,D,H}	1
{C,D,I}	1
{C,E,F}	1
{C,E,H}	3
{C,E,I}	3
{C,H,I}	3
{E,F,H}	0
{E,F,I}	0
{E,H,I}	3

=>

4 Itemset	sup
{B,C,D,E}	1
{B,C,D,H}	1
{B,C,D,I}	1
{B,C,E,H}	1
{B,C,E,I}	1
{C,E,H,I}	3

Paramos, portanto, no 4º itemset pois caso continuássemos, possivelmente teríamos um conjunto vazio.

Tendo os itemsets descobertos em mãos, conseguimos agora transformá-los em uma regra, ou um conjunto de regras, com uma precisão mínima especificada.

Realizando a análise a partir da frequência mínima especificada e nos itemsets obtidos, temos as seguintes regras:

Itens	Confiança
{C,E,H -> I}	2/3=66,66%
{C,E,I -> H}	2/3=66,66%
{C,H,I -> E}	2/5=40,00%
{E,H,I -> C}	2/5=40,00%
{C -> E,H,I}	2/3=66,66%
{E -> C,H,I}	2/3=66,66%
{H -> C,E,I}	2/3=66,66%
{I -> C,E,H}	2/3=66,66%

Sendo assim, com a lista de compras fornecida, suporte >= 2 e confiança > 60%, podemos concluir que:

Quando um cliente comprar	Ele comprará	Precisão
Pão, couve e café	Brócolis	66,66%
Pão, couve e brócolis	Café	66,66%
Pão	Couve, café e brócolis	66,66%
Couve	Pão, café e brocólis	66,66%
Café	Pão, couve e brócolis	66,66%
Brócolis	Pão, couve e café	66,66%