BÀI 7: SỰ PHÂN TÍCH NHÓM (TT)

I. Mục tiêu:

Sau khi thực hành xong, sinh viên nắm được phương pháp gom cụm phân cấp (hierarchical) thông qua:

- Phương pháp hợp nhất Bottom-up với single-linkage, complete-linkage và average-linkage.

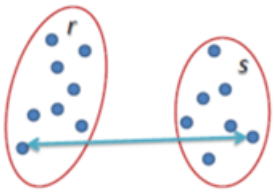
II. Tóm tắt lý thuyết:

Trong phương pháp hợp nhất Bottom-up, các điểm dữ liệu được hợp nhất (agglomerative) liên tục thành các cụm ở mức cao hơn. Sự lựa chọn hàm mục tiêu thường quyết định việc trộn lại của các cụm. Thuật toán được phát biểu như sau:

Algorithm *AgglomerativeMerge*(Data: $\mathcal{D}$)
begin
  Initialize $n \times n$ distance matrix M using $\mathcal{D}$;
  repeat
    Pick closest pair of clusters i and j using M;
    Merge clusters i and j;
    Delete rows/columns i and j from M and create
      a new row and column for newly merged cluster;
    Update the entries of new row and column of M;
  until termination criterion;
  return current merged cluster set;
end

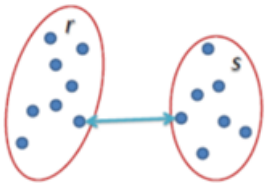
Hàm tiêu chuẩn liên kết (linkage criteria) được sử dụng để xác định khoảng cách giữa 2 cụm hoặc tập hợp các điểm. Các hàm liên kết phổ biến nhất được mô tả như sau:

- Liên kết tối đa hoặc đầy đủ (complete-linkage): Khoảng cách giữa hai cụm được định nghĩa là khoảng cách lớn nhất giữa 2 điểm trong mỗi cụm



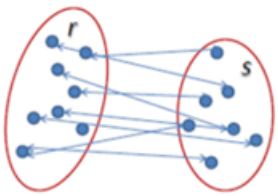
$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

- Liên kết đơn (single-linkage) hoặc tối thiểu: là khoảng cách nhỏ nhất giữa 2 điểm trong mỗi cụm.



$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

- Liên kết trung bình: là khoảng cách trung bình giữa các điểm trong mỗi cụm.



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

- Liên kết tâm: là khoảng cách giữa tâm của mỗi cụm.
- Phương pháp phương sai tối thiểu của Ward: Nó giảm thiểu tổng phương sai trong cụm. Ở mỗi bước, cặp cụm có khoảng cách giữa các cụm tối thiểu được hợp nhất.

Ví dụ: Cho các điểm $A = (1,1), B = (2,3), C = (3,5), D = (4,5), E = (6,6), F = (7,5)$ và gom nhóm chúng.

Tạo ma trận khoảng cách M có kích thước 6×6 (sử dụng khoảng cách Euclidean):

	A	B	C	D	E	F
A	0	2.236068	4.472136	5	7.071068	7.211103
B	2.236068	0	2.236068	2.828427	5	5.385165
C	4.472136	2.236068	0	1	3.162278	4
D	5	2.828427	1	0	2.236068	3
E	7.071068	5	3.162278	2.236068	0	1.414214
F	7.211103	5.385165	4	3	1.414214	0

Các bước để thực hiện gom cụm phân bậc sử dụng single-linkage:

Bước 1: Mỗi điểm dữ liệu là một cụm riêng lẻ nên ta có các cụm sau

$$\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}.$$

Bước 2: Tiếp theo, ta xét khoảng cách nhỏ nhất trong ma trận khoảng cách và hợp nhất các điểm có khoảng cách nhỏ nhất.

	A	B	C	D	E	F
A	0	2.236068	4.472136	5	7.071068	7.211103
B	2.236068	0	2.236068	2.828427	5	5.385165
C	4.472136	2.236068	0	1	3.162278	4
D	5	2.828427	1	0	2.236068	3
E	7.071068	5	3.162278	2.236068	0	1.414214
F	7.211103	5.385165	4	3	1.414214	0

Ở đây, khoảng cách nhỏ nhất là 1 nên chúng ta sẽ hợp nhất điểm C và D. Cập nhật lại ma trận khoảng cách:

	A	B	CD	E	F
A	0	2.236068	4.472136	7.071068	7.211103
B	2.236068	0	2.236068	5	5.385165
CD	4.472136	2.236068	0	2.236068	3
E	7.071068	5	2.236068	0	1.414214
F	7.211103	5.385165	3	1.414214	0

với

$$d(A, CD) = \min\{d(A, C), d(A, D)\} = \min\{4.472136, 5\} = 4.472136$$

$$d(B, CD) = \min\{d(B, C), d(B, D)\} = \min\{2.236068, 2.828427\} = 2.236068$$

$$d(E, CD) = \min\{d(E, C), d(E, D)\} = \min\{3.162278, 2.236068\} = 2.236068$$

$$d(F, CD) = \min\{d(F, C), d(F, D)\} = \min\{4, 3\} = 3$$

Bước 3: lặp lại bước 2 cho đến khi chỉ còn một cụm duy nhất.

	A	B	CD	E	F
A	0	2.236068	4.472136	7.071068	7.211103
B	2.236068	0	2.236068	5	5.385165
CD	4.472136	2.236068	0	2.236068	3
E	7.071068	5	2.236068	0	1.414214
F	7.211103	5.385165	3	1.414214	0

Khoảng cách nhỏ nhất là 1.414214 nên nên chúng ta sẽ hợp nhất điểm E và F. Cập nhật lại ma trận khoảng cách:

	A	B	CD	EF
A	0	2.236068	4.472136	7.071068
B	2.236068	0	2.236068	5
CD	4.472136	2.236068	0	2.236068
EF	7.071068	5	2.236068	0

với

$$d(A, EF) = \min\{d(A, E), d(A, F)\} = \min\{7.211103, 7.071068\} = 7.071068$$

$$d(B, EF) = \min\{d(B, E), d(B, F)\} = \min\{5, 5.385165\} = 5$$

$$d(CD, EF) = \min\{d(C, E), d(C, F), d(D, E), d(D, F)\} = \min\{3.162278, 4, 2.236068, 3\}$$
$$= 2.236068$$

	A	B	CD	EF
A	0	2.236068	4.472136	7.071068
B	2.236068	0	2.236068	5
CD	4.472136	2.236068	0	2.236068
EF	7.071068	5	2.236068	0

Khoảng cách nhỏ nhất là 2.236068 nên nên chúng ta có thể hợp nhất điểm A và B, CD và B, hoặc CD và EF. Trong bài này, chúng ta sẽ hợp nhất điểm A và B. Cập nhật lại ma trận khoảng cách:

	AB	CD	EF
AB	0	2.236068	5
CD	2.236068	0	2.236068
EF	5	2.236068	0

với

$$d(CD, AB) = \min\{d(C, A), d(C, B), d(D, A), d(D, B)\}$$
$$= \min\{4.472136, 2.236068, 5, 2.828427\} = 2.236068$$

$$d(AB, EF) = \min\{d(A, E), d(A, F), d(B, E), d(B, F)\}$$
$$= \min\{7.071068, 5, 7.211103, 5.385165\} = 5$$

Tương tự, khoảng cách nhỏ nhất là 2.236068 nên nên chúng ta có thể hợp nhất AB và CD hoặc CD và EF. Chúng ta sẽ chọn hợp nhất AB và CD. Cập nhật lại ma trận

	AB	CD	EF
AB	0	2.236068	5
CD	2.236068	0	2.236068
EF	5	2.236068	0

khoảng cách:

	ABCD	EF
ABCD	0	2.236068
EF	2.236068	0

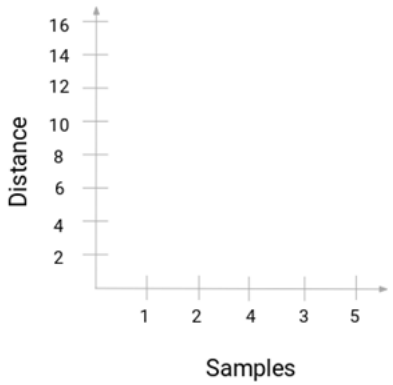
với

$$d(ABCD, EF) = \min\{d(A,E), d(A,F), d(B,E), d(B,F), d(C,E), d(C,F), d(D,E), d(D,F)\}$$
$$= \min\{7.071068, 7.211103, 5, 5.385165, 3.162278, 4, 2.236068, 3\} = 2.236068.$$

Cuối cùng, chúng ta sẽ hợp nhất ABCD và EF lại thành ABCDEF.

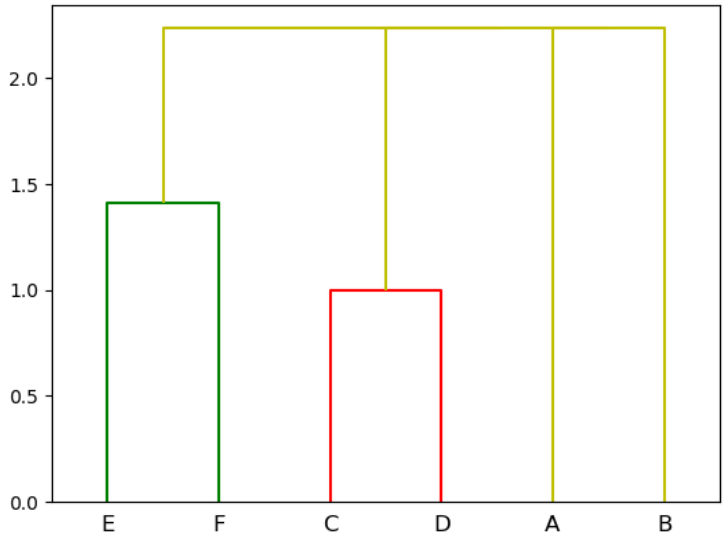
Để có được số lượng cụm cho gom cụm phân bậc, chúng ta sử dụng một khái niệm có tên là Dendrogram.

Dendrogram là một sơ đồ dạng cây ghi lại các chuỗi hợp nhất hoặc phân tách. Bất cứ khi nào chúng ta hợp nhất hai cụm, một dendrogram sẽ ghi lại khoảng cách giữa các cụm này và biểu thị nó dưới dạng biểu đồ.

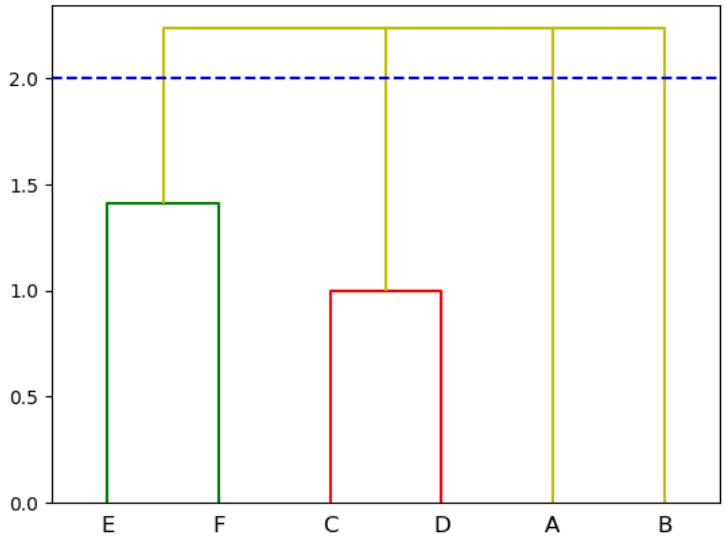


Các mẫu của tập dữ liệu trên trục x và khoảng cách trên trục y. Khi hai cụm được hợp nhất, ta sẽ nối chúng lại và chiều cao của phép nối sẽ là khoảng cách giữa các điểm

này. Khi đó, ta có



Bây giờ, ta có thể đặt khoảng cách ngưỡng và vẽ một đường ngang (Nói chung, ta cố gắng đặt ngưỡng sao cho nó cắt đường thẳng đứng cao nhất). Hãy đặt ngưỡng này là 2 và vẽ một đường ngang:



Số cụm sẽ là số đường thẳng đứng được cắt bởi đường được vẽ bằng ngưỡng. Trong ví dụ trên, do đường màu xanh cắt 4 đường thẳng đứng nên ta sẽ có 4 cụm. Cụm 1 sẽ có một mẫu (E, F), cụm 2 có một mẫu (C, D), cụm 3 có một mẫu (A) và cụm 4 có một mẫu là (B).

Các bước để thực hiện gom cụm phân bậc sử dụng complete-linkage:

Bước 1: Mỗi điểm dữ liệu là một cụm riêng lẻ nên ta có các cụm sau

$$\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}.$$

Bước 2: Tiếp theo, ta xét khoảng cách nhỏ nhất trong ma trận khoảng cách và hợp nhất các điểm có khoảng cách nhỏ nhất.

Ở đây, khoảng cách nhỏ nhất là 1 nên chúng ta sẽ hợp nhất điểm C và D. Cập nhật lại ma trận khoảng cách:

	A	B	C	D	E	F
A	0	2.236068	4.472136	5	7.071068	7.211103
B	2.236068	0	2.236068	2.828427	5	5.385165
C	4.472136	2.236068	0	1	3.162278	4
D	5	2.828427	1	0	2.236068	3
E	7.071068	5	3.162278	2.236068	0	1.414214
F	7.211103	5.385165	4	3	1.414214	0

	A	B	CD	E	F
A	0	2.236068	5	7.071068	7.211103
B	2.236068	0	2.828427	5	5.385165
CD	5	2.828427	0	3.162278	4
E	7.071068	5	3.162278	0	1.414214
F	7.211103	5.385165	4	1.414214	0

với

$$d(A, CD) = \max\{d(A, C), d(A, D)\} = \max\{4.472136, 5\} = 5$$

$$d(B, CD) = \max\{d(B, C), d(B, D)\} = \max\{2.236068, 2.828427\} = 2.828427$$

$$d(E, CD) = \max\{d(E, C), d(E, D)\} = \max\{3.162278, 2.236068\} = 3.162278$$

$$d(F, CD) = \max\{d(F, C), d(F, D)\} = \max\{4, 3\} = 4$$

Bước 3: lặp lại bước 2 cho đến khi chỉ còn một cụm duy nhất.

	A	B	CD	E	F
A	0	2.236068	5	7.071068	7.211103
B	2.236068	0	2.828427	5	5.385165
CD	5	2.828427	0	3.162278	4
E	7.071068	5	3.162278	0	1.414214
F	7.211103	5.385165	4	1.414214	0

Khoảng cách nhỏ nhất là 1.414214 nên nên chúng ta sẽ hợp nhất điểm E và F. Cập nhật lại ma trận khoảng cách:

	A	B	CD	EF
A	0	2.236068	4.472136	7.211103
B	2.236068	0	2.236068	5.385165
CD	4.472136	2.236068	0	4
EF	7.211103	5.385165	4	0

với

$$d(A, EF) = \max\{d(A, E), d(A, F)\} = \max\{7.211103, 7.071068\} = 7.211103$$

$$d(B, EF) = \max\{d(B, E), d(B, F)\} = \max\{5, 5.385165\} = 5.385165$$

$$d(CD, EF) = \max\{d(C, E), d(C, F), d(D, E), d(D, F)\} = \max\{3.162278, 4, 2.236068, 3\}$$
$$= 4$$

	A	B	CD	EF
A	0	2.236068	4.472136	7.211103
B	2.236068	0	2.236068	5.385165
CD	4.472136	2.236068	0	4
EF	7.211103	5.385165	4	0

Khoảng cách nhỏ nhất là 2.236068 nên nên chúng ta có thể hợp nhất điểm A và B hoặc CD và B. Trong bài này, chúng ta sẽ hợp nhất điểm A và B. Cập nhật lại ma trận khoảng cách:

	AB	CD	EF
AB	0	5	7.211103
CD	5	0	4
EF	7.211103	4	0

với

$$\begin{aligned} d(CD, AB) &= \max\{d(C, A), d(C, B), d(D, A), d(D, B)\} \\ &= \max\{4.472136, 2.236068, 5, 2.828427\} = 5 \\ d(AB, EF) &= \max\{d(A, E), d(A, F), d(B, E), d(B, F)\} \\ &= \max\{7.071068, 5, 7.211103, 5.385165\} = 7.211103 \end{aligned}$$

	AB	CD	EF
AB	0	5	7.211103
CD	5	0	4
EF	7.211103	4	0

Tương tự, khoảng cách nhỏ nhất là 4 nên chúng ta sẽ hợp nhất CD và EF. Cập nhật lại ma trận khoảng cách:

	AB	CDEF
AB	0	2.236068
CDEF	2.236068	0

với

$$\begin{aligned} d(AB, CDEF) &= \max\{d(A, C), d(A, D), d(A, E), d(A, F), d(B, C), d(B, D), d(B, E), d(B, F)\} \\ &= \max\{4.472136, 5, 7.071068, 7.211103, 2.236068, 2.828427, 5, 5.385165\} \\ &= 7.211103. \end{aligned}$$

Cuối cùng, chúng ta sẽ hợp nhất AB và CDEF lại thành ABCDEF. Khi đó, ta có

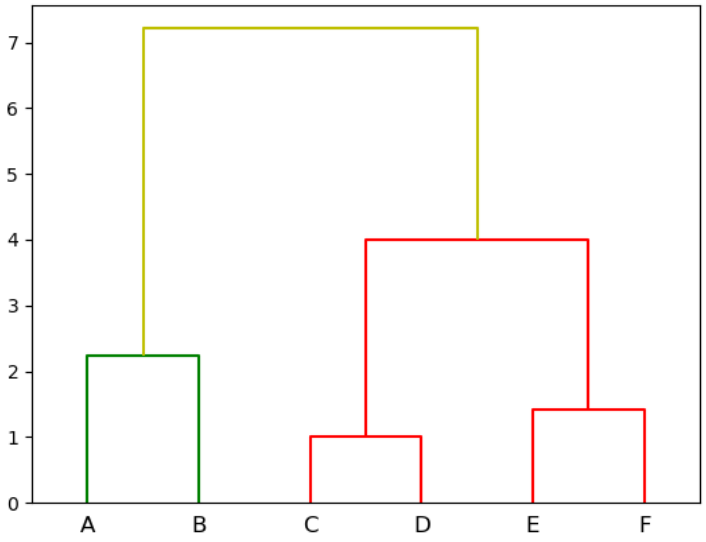
Ta đặt ngưỡng này là 5 và vẽ một đường ngang:

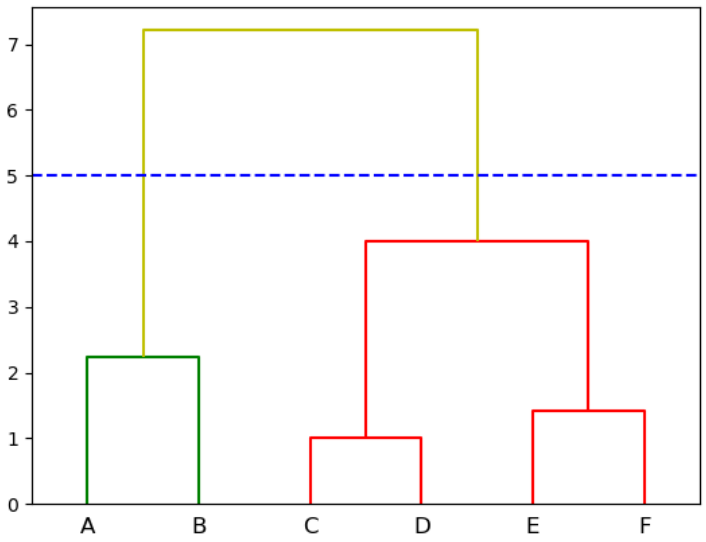
Trong ví dụ trên, do đường màu xanh cắt 2 đường thẳng đứng nên ta sẽ có 2 cụm. Cụm 1 sẽ có một mẫu (C,D,E,F), cụm 2 có một mẫu (A, B).

Các bước để thực hiện gom cụm phân bậc sử dụng average-linkage:

Bước 1: Mỗi điểm dữ liệu là một cụm riêng lẻ nên ta có các cụm sau

$$\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}.$$


7
6
5
4
3
2
1
0
A
B
C
D
E
F


7
6
5
4
3
2
1
0
A
B
C
D
E
F

Bước 2: Tiếp theo, ta xét khoảng cách nhỏ nhất trong ma trận khoảng cách và hợp nhất các điểm có khoảng cách nhỏ nhất.

	A	B	C	D	E	F
A	0	2.236068	4.472136	5	7.071068	7.211103
B	2.236068	0	2.236068	2.828427	5	5.385165
C	4.472136	2.236068	0	1	3.162278	4
D	5	2.828427	1	0	2.236068	3
E	7.071068	5	3.162278	2.236068	0	1.414214
F	7.211103	5.385165	4	3	1.414214	0

Ở đây, khoảng cách nhỏ nhất là 1 nên chúng ta sẽ hợp nhất điểm C và D. Cập nhật lại ma trận khoảng cách:

	A	B	CD	E	F
A	0	2.236068	4.736068	7.071068	7.211103
B	2.236068	0	2.532247	5	5.385165
CD	4.736068	2.532247	0	2.699173	3.5
E	7.071068	5	2.699173	0	1.414214
F	7.211103	5.385165	3.5	1.414214	0

với

$$d(A,CD) = \frac{d(A,C)+d(A,D)}{2} = \frac{4.472136+5}{2} = 4.736068$$
$$d(B,CD) = \frac{d(B,C)+d(B,D)}{2} = \frac{2.236068+2.828427}{2} = 2.532247$$
$$d(E,CD) = \frac{d(E,C)+d(E,D)}{2} = \frac{3.162278+2.236068}{2} = 2.699173$$
$$d(F,CD) = \frac{d(F,C)+d(F,D)}{2} = \frac{4+3}{2} = 3.5$$

Bước 3: lặp lại bước 2 cho đến khi chỉ còn một cụm duy nhất.

	A	B	CD	E	F
A	0	2.236068	4.736068	7.071068	7.211103
B	2.236068	0	2.532247	5	5.385165
CD	4.736068	2.532247	0	2.699173	3.5
E	7.071068	5	2.699173	0	1.414214
F	7.211103	5.385165	3.5	1.414214	0

Khoảng cách nhỏ nhất là 1.414214 nên nên chúng ta sẽ hợp nhất điểm E và F. Cập nhật lại ma trận khoảng cách:

	A	B	CD	EF
A	0	2.236068	4.736068	7.141085
B	2.236068	0	2.532247	5.192582
CD	4.736068	2.532247	0	3.099586
EF	7.141085	5.192582	3.099586	0

với

$$d(A, EF) = \frac{d(A,E) + d(A,F)}{2} = \frac{7.211103 + 7.071068}{2} = 7.141085$$

$$d(B, EF) = \frac{d(B,E) + d(B,F)}{2} = \frac{5 + 5.385165}{2} = 5.192582$$

$$d(CD, EF) = \frac{d(C,E) + d(C,F) + d(D,E) + d(D,F)}{4} = \frac{3.162278 + 4 + 2.236068 + 3}{4}$$

$$= 3.099586$$

	A	B	CD	EF
A	0	2.236068	4.736068	7.141085
B	2.236068	0	2.532247	5.192582
CD	4.736068	2.532247	0	3.099586
EF	7.141085	5.192582	3.099586	0

Khoảng cách nhỏ nhất là 2.236068 nên nên chúng ta có thể hợp nhất điểm A và B. Cập nhật lại ma trận khoảng cách:

	AB	CD	EF
AB	0	3.634158	6.166834
CD	3.634158	0	3.099586
EF	6.166834	3.099586	0

với

$$d(CD, AB) = \frac{d(C, A) + d(C, B) + d(D, A) + d(D, B)}{4}$$
$$= \frac{4.472136 + 2.236068 + 5 + 2.828427}{4} = 3.634158$$
$$d(AB, EF) = \frac{d(A, E) + d(A, F) + d(B, E) + d(B, F)}{4}$$
$$= \frac{7.071068 + 5 + 7.211103 + 5.385165}{4} = 6.166834$$

	AB	CD	EF
AB	0	3.634158	6.166834
CD	3.634158	0	3.099586
EF	6.166834	3.099586	0

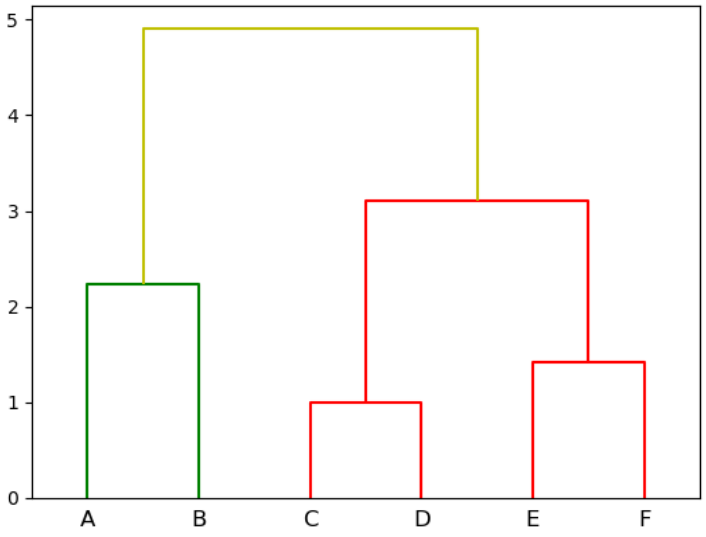
Tương tự, khoảng cách nhỏ nhất là 3.099586 nên chúng ta sẽ hợp nhất CD và EF. Cập nhật lại ma trận khoảng cách:

	AB	CDEF
AB	0	4.900496
CDEF	4.900496	0

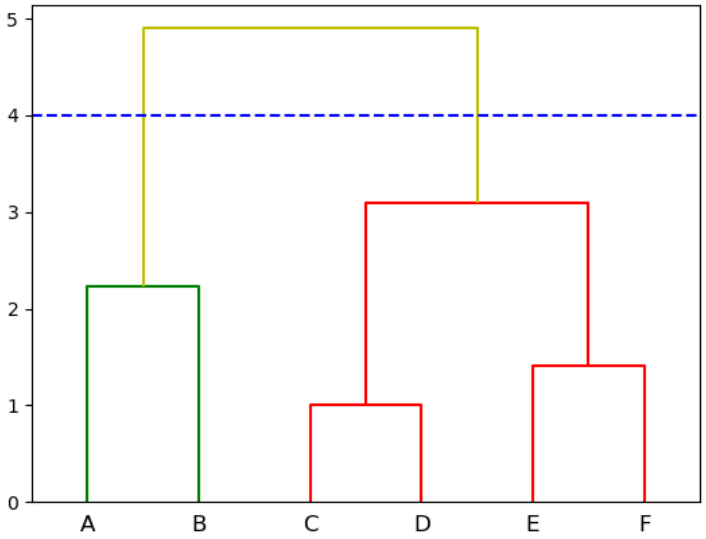
với

$$d(AB, CDEF) =$$
$$\frac{d(A, C) + d(A, D) + d(A, E) + d(A, F) + d(B, C) + d(B, D) + d(B, E) + d(B, F)}{8}$$
$$= \frac{4.472136 + 5 + 7.071068 + 7.211103 + 2.236068 + 2.828427 + 5 + 5.385165}{8}$$
$$= 4.900496.$$

Cuối cùng, chúng ta sẽ hợp nhất AB và CDEF lại thành ABCDEF. Khi đó, ta có



Ta đặt ngưỡng này là 4 và vẽ một đường ngang:



Trong ví dụ trên, do đường màu xanh cắt 2 đường thẳng đứng nên ta sẽ có 2 cụm. Cụm 1 sẽ có một mẫu (C,D,E,F), cụm 2 có một mẫu (A, B).

III. Nội dung thực hành:

1. Cài đặt thuật toán gom cụm phân cấp

- Đọc dữ liệu từ file "data.csv"

  (https://www.dropbox.com/s/ikgzzr30ln6akk2/data.csv?dl=0)

```
>>> import pandas as pd
>>> import seaborn as sns
>>> import scipy.cluster.hierarchy as shc
>>> import matplotlib.pyplot as plt
>>> from sklearn.decomposition import PCA
>>> from sklearn.cluster import AgglomerativeClustering
>>> path_to_file = 'D:\\Huynh\\DataMining_Lab\\data\\tuan7\\data.csv'
>>> customer_data = pd.read_csv(path_to_file)
>>> customer_data.shape
(200, 5)
>>> customer_data.columns
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
>>> customer_data.describe().transpose()
                        count    mean        std  ...    50%     75%    max
CustomerID              200.0  100.50  57.879185  ...  100.5  150.25  200.0
Age                     200.0   38.85  13.969007  ...   36.0   49.00   70.0
Annual Income (k$)      200.0   60.56  26.264721  ...   61.5   78.00  137.0
Spending Score (1-100)  200.0   50.20  25.823522  ...   50.0   73.00   99.0

[4 rows x 8 columns]
>>> customer_data.head()
   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
```

- Chia cột "Age" thành 10 nhóm khác nhau $(15-20, 20-30, 30-40, 40-50, 50-60, 60-70)$

```
>>> intervals = [15, 20, 30, 40, 50, 60, 70]
>>> col = customer_data['Age']
>>> customer_data['Age Groups'] = pd.cut(x=col, bins=intervals)
>>> customer_data['Age Groups']
0      (15, 20]
1      (20, 30]
2      (15, 20]
3      (20, 30]
4      (30, 40]
         ...
195    (30, 40]
196    (40, 50]
197    (30, 40]
198    (30, 40]
199    (20, 30]
Name: Age Groups, Length: 200, dtype: category
Categories (6, interval[int64]): [(15, 20] < (20, 30] < (30, 40] < (40, 50] < (50, 60] < (60, 70]]
>>> customer_data.groupby('Age Groups')['Age Groups'].count()
Age Groups
(15, 20]    17
(20, 30]    45
(30, 40]    60
(40, 50]    38
(50, 60]    23
(60, 70]    17
Name: Age Groups, dtype: int64
```

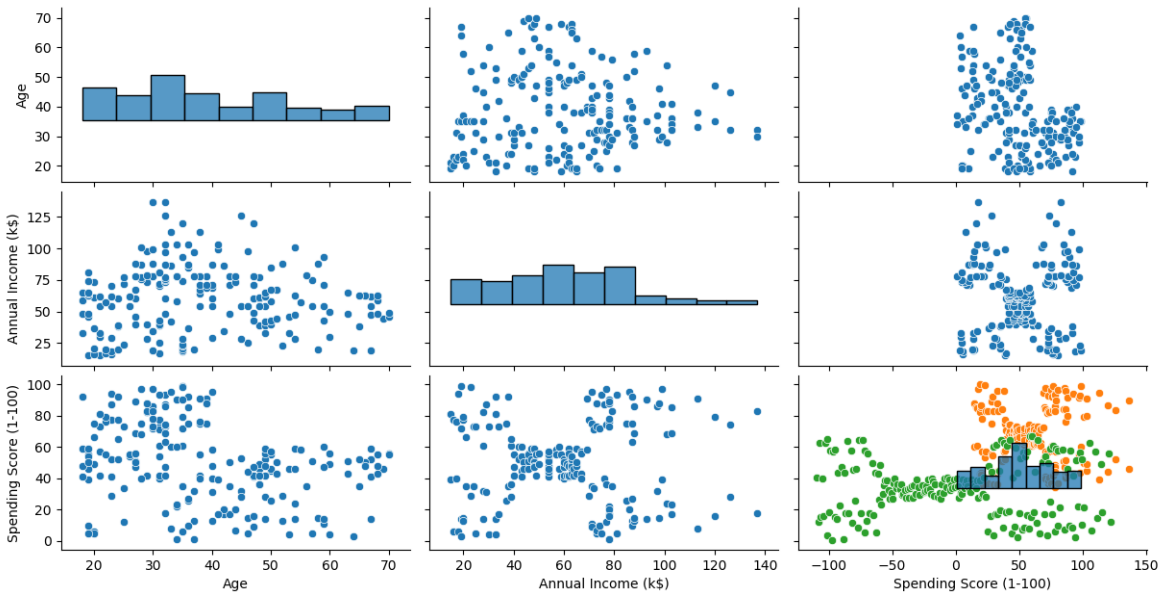
- Chuyển 2 cột Age và Genre thành dạng số

```
>>> customer_data_oh = pd.get_dummies(customer_data)
>>> customer_data_oh
     CustomerID  Age  ...  Age Groups_(50, 60]  Age Groups_(60, 70]
0             1   19  ...                    0                    0
1             2   21  ...                    0                    0
2             3   20  ...                    0                    0
3             4   23  ...                    0                    0
4             5   31  ...                    0                    0
..          ...  ...  ...                  ...                  ...
195         196   35  ...                    0                    0
196         197   45  ...                    0                    0
197         198   32  ...                    0                    0
198         199   32  ...                    0                    0
199         200   30  ...                    0                    0

[200 rows x 12 columns]
```

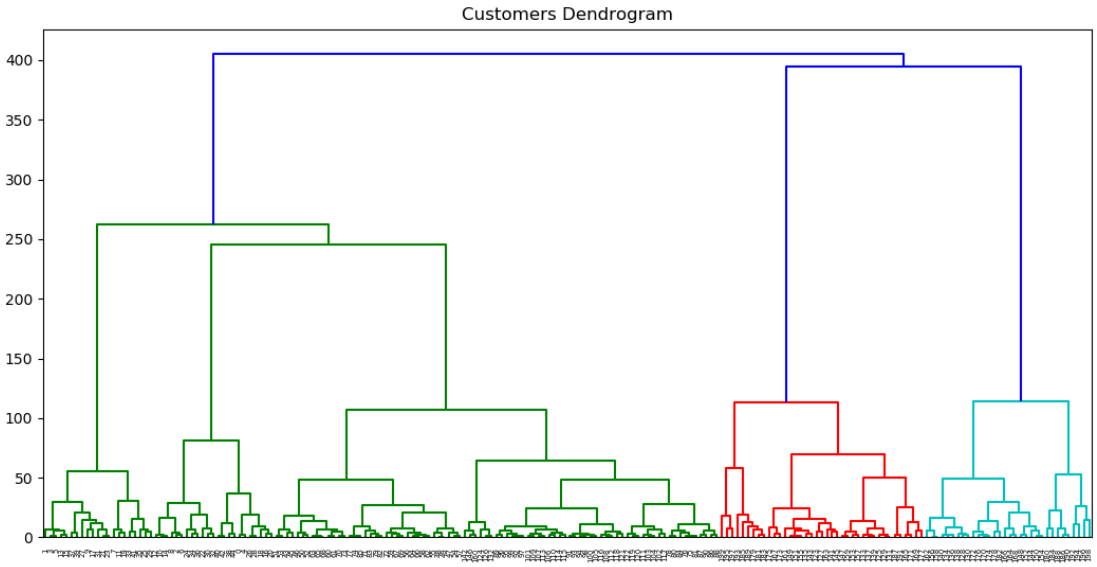
- Bỏ cột "CustomerID"

```
>>> customer_data = customer_data.drop('CustomerID', axis=1)
>>> sns.pairplot(customer_data)
<seaborn.axisgrid.PairGrid object at 0x00000227C6845AC8>
>>> sns.scatterplot(x=customer_data['Annual Income (k$)'],
                    y=customer_data['Spending Score (1-100)'])
<matplotlib.axes._subplots.AxesSubplot object at 0x00000227CA637848>
```



- Bỏ cột “Age” và vẽ dendrogram

```
>>> customer_data_oh = customer_data_oh.drop(['Age'], axis=1)
>>> customer_data_oh.shape
(200, 11)
>>> plt.figure(figsize=(10, 7))
<Figure size 1000x700 with 0 Axes>
>>> plt.title("Customers Dendrogram")
Text(0.5, 1.0, 'Customers Dendrogram')
>>> selected_data = customer_data_oh.iloc[:, 1:3]
>>> clusters = shc.linkage(selected_data,
            method='ward',
            metric="euclidean")
>>> shc.dendrogram(Z=clusters)
Squeezed text (263 lines).
>>> plt.show()
```

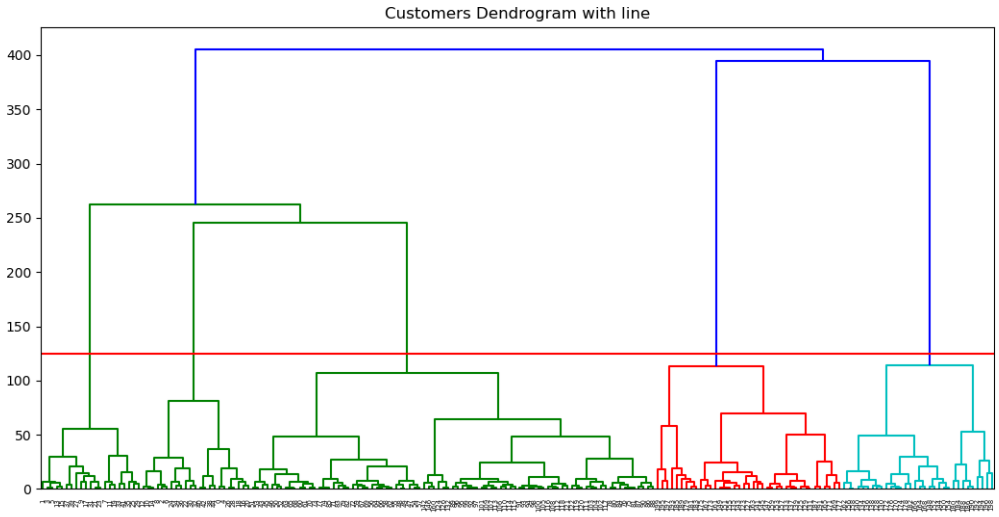


- Vẽ đường nằm ngang đi qua khoảng cách dài nhất

```
>>> plt.figure(figsize=(10, 7))
<Figure size 1000x700 with 0 Axes>
>>> plt.title("Customers Dendrogram with line")
Text(0.5, 1.0, 'Customers Dendrogram with line')
>>> clusters = shc.linkage(selected_data,
            method='ward',
            metric="euclidean")
>>> shc.dendrogram(clusters)
```

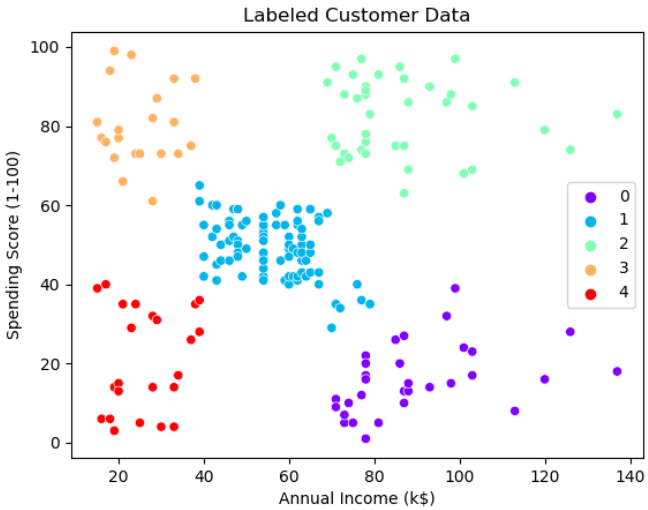
Squeezed text (263 lines).

```
>>> plt.axhline(y = 125, color = 'r', linestyle = '-')
<matplotlib.lines.Line2D object at 0x00000250F020E488>
>>> plt.show()
```



- Thực thi phân cụm với dữ liệu ban đầu

```
>>> clustering_model = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='ward')
>>> clustering_model.fit(selected_data)
AgglomerativeClustering(n_clusters=5)
>>> clustering_model.labels_
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
       4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 1,
       4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 0, 2, 0, 2,
       1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2], dtype=int64)
>>> data_labels = clustering_model.labels_
>>> sns.scatterplot(x='Annual Income (k$)',
                    y='Spending Score (1-100)',
                    data=selected_data,
                    hue=data_labels,
                    palette="rainbow").set_title('Labeled Customer Data')
Text(0.5, 1.0, 'Labeled Customer Data')
>>> plt.show()
```



2. Yêu cầu

- Viết chương trình để thực thi thuật toán hierarchical sử dụng phương pháp hợp nhất Bottom-up với single-linkage, complete-linkage và average-linkage.

- Trình bày tóm tắt phần code do em viết và so sánh với hàm có sẵn trong thư viện (thay 'ward' tương ứng bằng 'single', 'complete' và 'average').