Bài tập 04

Clustering

Thông tin cá nhân

Họ và tên	Vũ Lê Thế Anh	Nguyễn Lê Hồng Hạnh
MSSV	1612838	1612849
Email	{1612838, 1612849}@student.hcmus.edu.vn	
SĐT	0961565087	0902719551

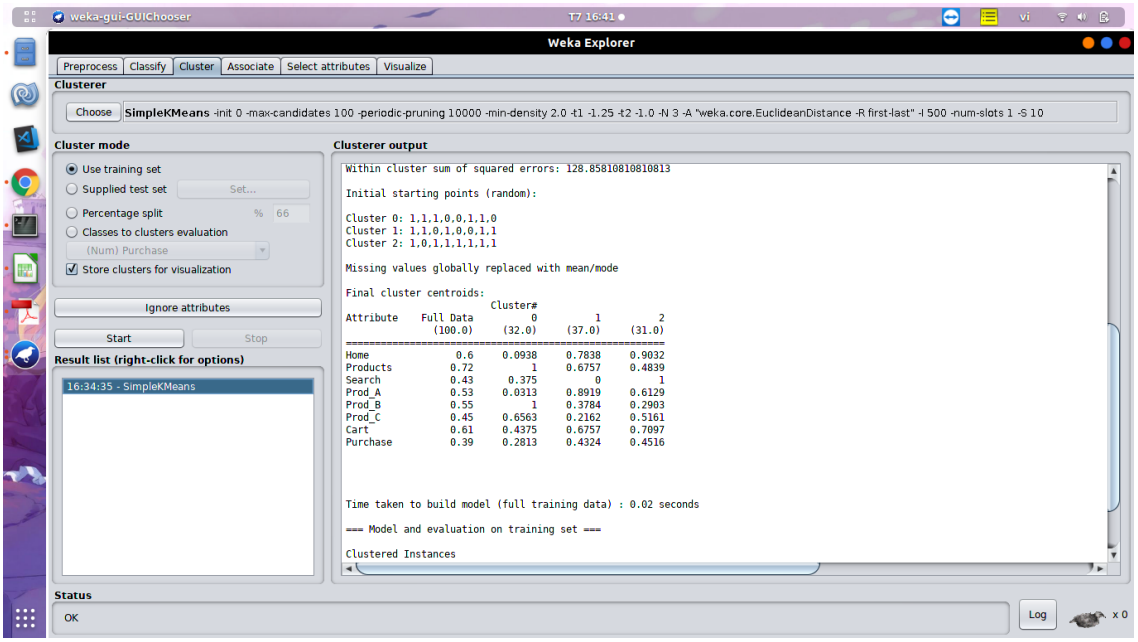
Yêu cầu bài tập

STT	Yêu cầu	Hoàn thành
1	Weka: thực hiện chạy trên bộ session và điền kết quả	100%
2	Weka: trả lời các câu hỏi	100%
3	Tự cài đặt thuật toán kmeans với đầu vào và đầu ra theo yêu cầu	100%
4	So sánh cài đặt và Weka	100%
5	Tìm hiểu cải tiến kmeans	100%

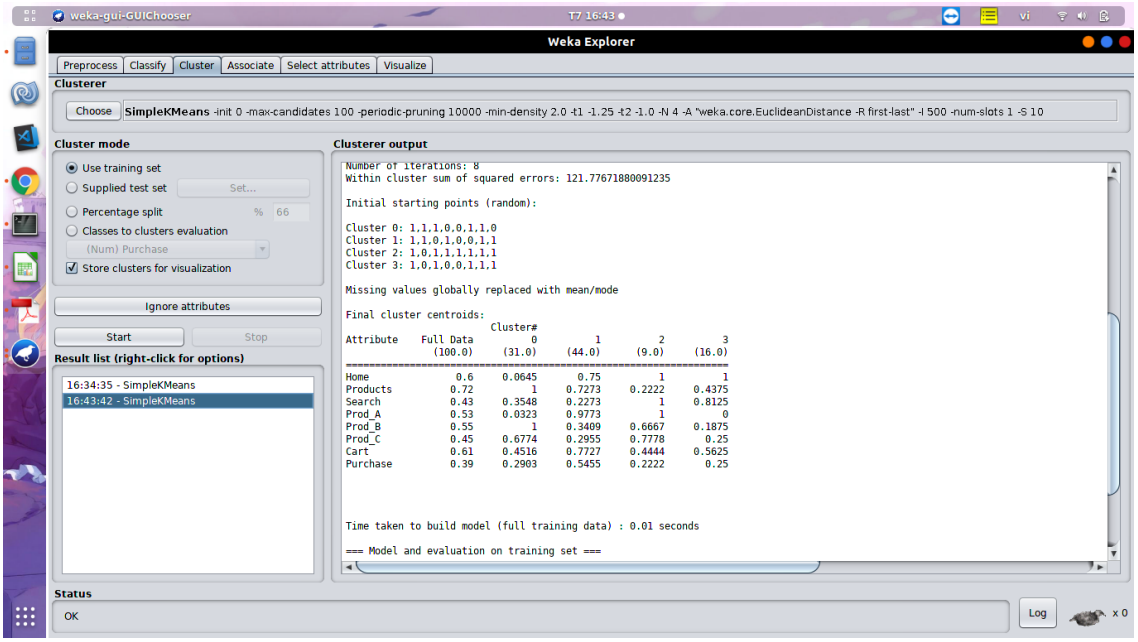
Mục lục

1 Nội dung thực hiện báo cáo với ứng dụng WEKA

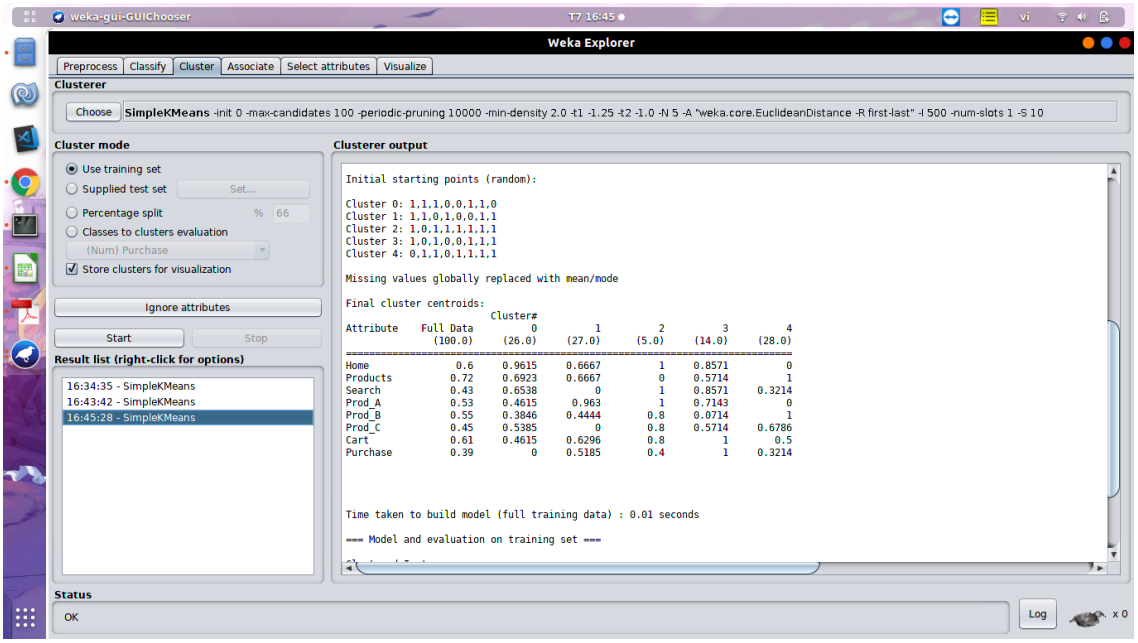
1. Bảng 1 báo cáo kết quả chạy thực nghiệm `SimpleKMeans` với các giá trị `k` thay đổi từ 2 đến 8 trên WEKA. Hình ảnh màn hình Clusterer Output ứng với mỗi giá trị k được cho trong Hình 1, 2, 3, 4, 5, 6.



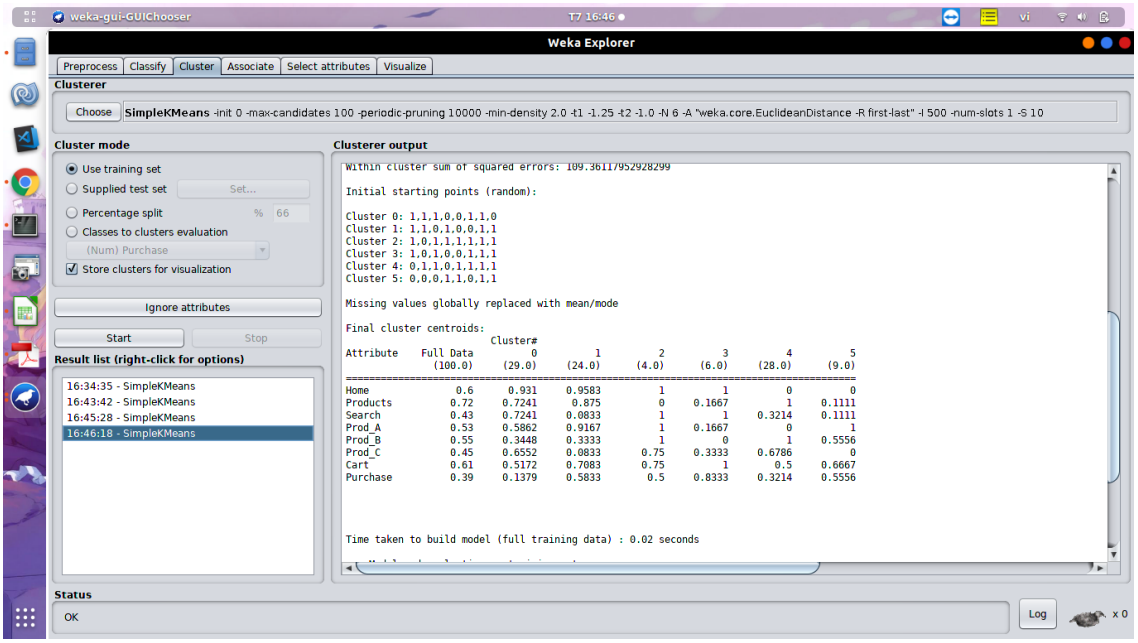
Hình 1: Màn hình Weka Output Classifier với $k = 3$.



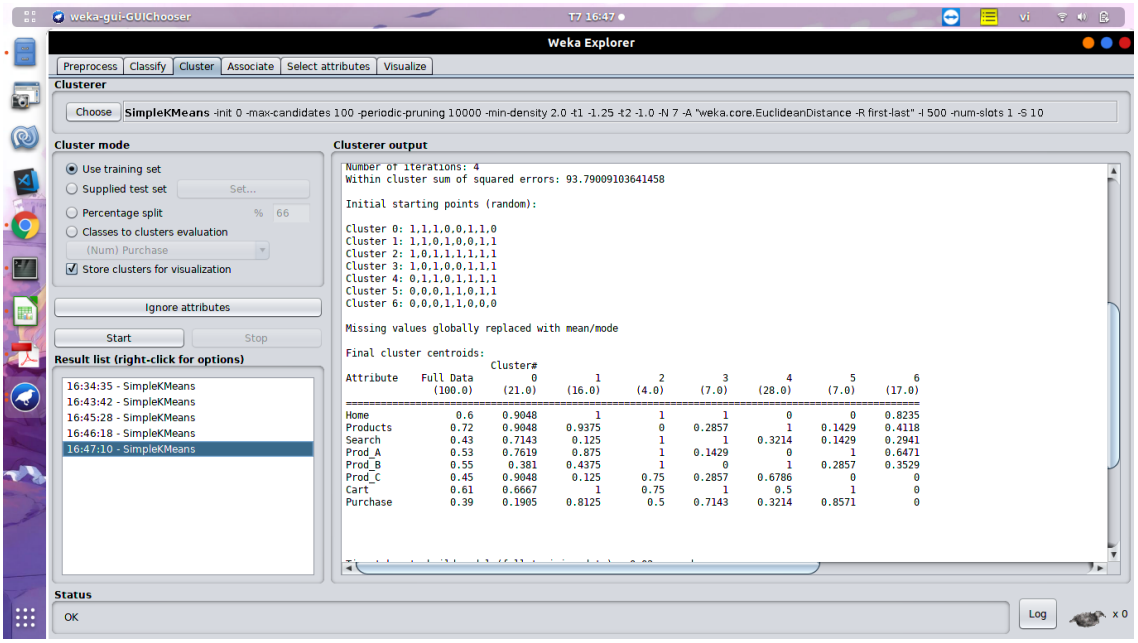
Hình 2: Màn hình Weka Output Classifier với $k = 4$.



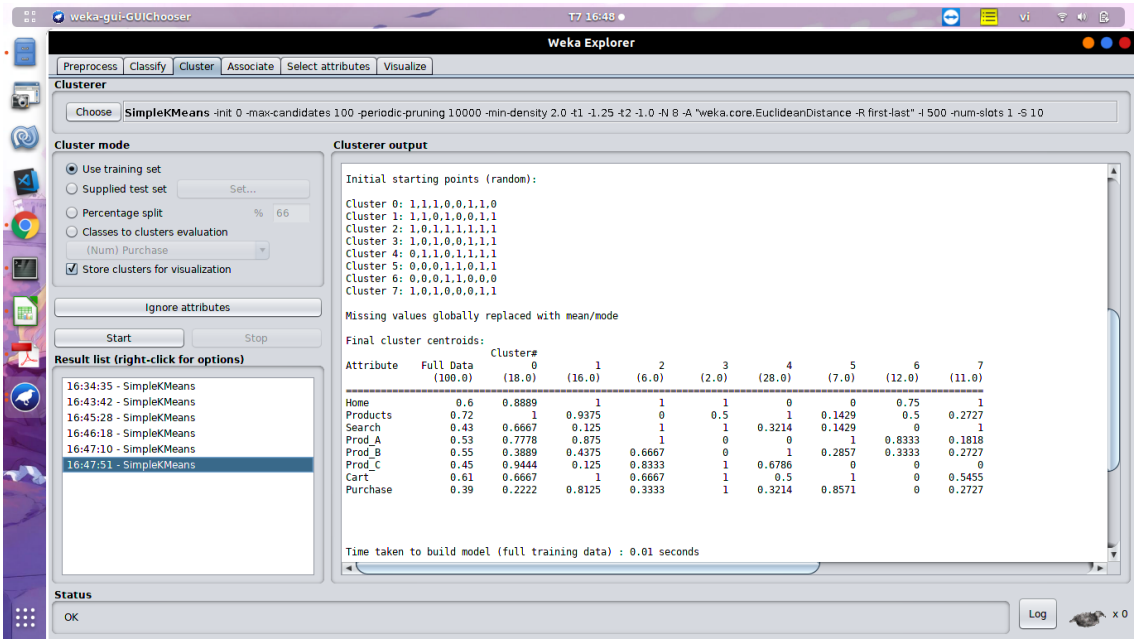
Hình 3: Màn hình Weka Output Classifier với $k = 5$.



Hình 4: Màn hình Weka Output Classifier với $k = 6$.



Hình 5: Màn hình Weka Output Classifier với $k = 7$.



Hình 6: Màn hình Weka Output Classifier với $k = 8$.

k	SSE	Cluster centroids								
			Home	Products	Search	ProdA	ProdB	ProdC	Cart	Purchase
3	128.8581	0	0.0938	1	0.375	0.0313	1	0.6563	0.4375	0.2813
		1	0.7838	0.6757	0	0.8919	0.3784	0.2162	0.6757	0.4324
		2	0.9032	0.4839	1	0.6129	0.2903	0.5161	0.7097	0.4516
4	121.7767	0	0.0645	1	0.3548	0.0323	1	0.6774	0.4516	0.2903
		1	0.75	0.7273	0.2273	0.9773	0.3409	0.2955	0.7727	0.5455
		2	1	0.2222	1	1	0.6667	0.7778	0.4444	0.2222
		3	1	0.4375	0.8125	0	0.1875	0.25	0.5625	0.25
5	113.5826	0	0.9615	0.6923	0.6538	0.4615	0.3846	0.5385	0.4615	0
		1	0.6667	0.6667	0	0.963	0.4444	0	0.6296	0.5185
		2	1	0	1	1	0.8	0.8	0.8	0.4
		3	0.8571	0.5714	0.8571	0.7143	0.0714	0.5714	1	1
		4	0	1	0.3214	0	1	0.6786	0.5	0.3214
6	109.3611	0	0.931	0.7241	0.7241	0.5862	0.3448	0.6552	0.5172	0.1379
		1	0.9583	0.875	0.0833	0.9167	0.3333	0.0833	0.7083	0.5833
		2	1	0	1	1	1	0.75	0.75	0.5
		3	1	0.1667	1	0.1667	0	0.3333	1	0.8333
		4	0	1	0.3214	0	1	0.6786	0.5	0.3214
		5	0	0.1111	0.1111	1	0.5556	0	0.6667	0.5556
7	93.79	0	0.9048	0.9048	0.7143	0.7619	0.381	0.9048	0.6667	0.1905
		1	1	0.9375	0.125	0.875	0.4375	0.125	1	0.8125
		2	1	0	1	1	1	0.75	0.75	0.5
		3	1	0.2857	1	0.1429	0	0.2857	1	0.7143
		4	0	1	0.3214	0	1	0.6786	0.5	0.3214
		5	0	0.1429	0.1429	1	0.2857	0	1	0.8571
		6	0.8235	0.4118	0.2941	0.6471	0.3529	0	0	0
8	88.9319	0	0.8889	1	0.6667	0.7778	0.3889	0.9444	0.6667	0.2222
		1	1	0.9375	0.125	0.875	0.4375	0.125	1	0.8125
		2	1	0	1	1	0.6667	0.8333	0.6667	0.3333
		3	1	0.5	1	0	0	1	1	1
		4	0	1	0.3214	0	1	0.6786	0.5	0.3214
		5	0	0.1429	0.1429	1	0.2857	0	1	0.8571
		6	0.75	0.5	0	0.8333	0.3333	0	0	0
		7	1	0.2727	1	0.1818	0.2727	0	0.5455	0.2727

Bảng 1: Kết quả chạy thuật toán `SimpleKMeans` sử dụng Weka

Đối với k = 6, ý nghĩa từng cụm như sau:

- Cụm 0: dựa vào bàng số liệu, ta cũng thấy rằng người dùng truy cập hầu hết các trang (vì xác suất lớn hơn 0). Tuy nhiên, trong số đó, trang thanh toán (`Purchase`) lại có xác suất truy cập thấp (khoảng 0.1379), nghĩa là người dùng này có xu hướng "dạo chơi" qua các trang để xem chứ không có ý định mua hàng cụ thể.

- Cụm 1: nhóm người dùng này hiếm dùng chức năng tìm kiếm (0.0833) mà vào xem trang sản phẩm (0.875). Trong đó, sản phẩm A là họ hứng thú nhất (0.9167) so với B (0.3333) và C là hầu như không hứng thú (0.0833). Khả năng mua hàng của họ ở mức trung bình. Đây là nhóm yêu thích sản phẩm A trong số những người dùng dạo xem các sản phẩm bày bán.

- Cụm 2: trong cụm này, người dùng hoàn toàn không truy cập trang `Products`, mà sau khi truy cập trang `Home`, xác suất người dùng truy cập trang `Search` cao. Tiếp đến, xác suất truy cập trang cả 3 sản phẩm cũng cao (`Prod_A`: 1, `Prod_B`: 1, `Prod_C`: 0.75). Cuối cùng, hai trang `Cart` và `Purchase` cùng có xác suất truy cập khá cao (0.5). Như vậy, loại người dùng này có xu hướng tìm kiếm sản phẩm cần mua, có thể do người dùng chưa rõ về trang web, nên không biết sản phẩm đó có được bán trên web hay không hoặc do họ đã biết trước sản phẩm cần mua và tìm kiếm ngay.

- Cụm 3: tương tự như cụm 2, người dùng này cũng tìm kiếm sản phẩm cần mua; đồng thời người dùng cũng truy cập trang sản phẩm và trang sản phẩm A nhưng với xác suất rất thấp (0.1667). Bên cạnh đó, trạng sản phẩm B thì không được truy cập lần nào, còn trang sản phẩm C mặc dù có xác suất truy cập cao hơn nhưng cũng chỉ 0.3333. Như vậy, người dùng này cũng cần tìm kiếm nhưng không tìm thấy sản phẩm mình cần. Một điều đặc biệt là nhóm người dùng này có xác suất vào `Cart` và `Purchase` rất cao so với xác suất vào trang sản phẩm, lý giải cho điều này có thể là nhóm người dùng này vào để xem thử quy trình thanh toán, hay hình thức thanh toán; cũng có thể đây là nhóm người đã bỏ đồ vào giỏ hàng từ trước và thanh toán khi quay lại.

- Cụm 4: người dùng không truy cập vào trang `Home` mà truy cập thẳng vào trang sản phẩm, và có xác suất truy cập trang sản phẩm B và C cao. Tuy nhiên xác suất mua sản phẩm thấp. Như vậy, có thể người dùng xem danh sách các sản phẩm có ở trang web rồi sau đó mới quyết mua sản phẩm.

- Cụm 5: người dùng không vào trang sản phẩm mà vào thẳng trang chứa sản phẩm A hoặc B. Những người dùng này đã có đường dẫn từ bên ngoài để vào thẳng mà không qua các trang chính. Có thể người dùng đã tìm kiếm sản phẩm trên google, và mặt hàng ở trang web đã được google trả về trong kết quả tìm kiếm. Hoặc người dùng này thấy quảng cáo đặt ở các trang khác. Xác suất mua hàng của nhóm người dùng này khá cao.

Như vậy, việc phân 6 cụm cho thấy sự đa dạng người dùng lớn hơn và sau khi khảo sát các câu hỏi bên dưới, trả lời được hầu như toàn bộ các câu hỏi. Nhóm chọn $k = 6$.

STT	Home	Products	Search	ProdA	ProdB	ProdC	Cart	Purchase
0	0.931	0.7241	0.7241	0.5862	0.3448	0.6552	0.5172	0.1379
1	0.9583	0.875	0.0833	0.9167	0.3333	0.0833	0.7083	0.5833
2	1	0	1	1	1	0.75	0.75	0.5
3	1	0.1667	1	0.1667	0	0.3333	1	0.8333
4	0	1	0.3214	0	1	0.6786	0.5	0.3214
5	0	0.1111	0.1111	1	0.5556	0	0.6667	0.5556

Bảng 2: Các tâm cụm với $k = 6$

2. Gọi S là người dùng truy cập lần lượt các trang `Home => Search => Prod_B`. Xem người dùng S này là một điểm dữ liệu mới, có vector $\vec{s} = (1, 0, 1, 0, 1, 0, 0, 0)$. Ta sẽ phân cụm điểm mới này dựa trên khoảng cách đến mỗi tâm khi $k = 6$. Bảng 3 cho biết khoảng cách từ điểm dữ liệu mới này đến từng tâm cụm.

Tâm	0	1	2	3	4	5
Khoảng cách	1.44704	1.93427	1.5411	1.69147	1.8095	1.93731

Bảng 3: Khoảng cách từng tâm cụm đến $\vec{s} = (1, 0, 1, 0, 1, 0, 0, 0)$

Nhìn vào bảng số liệu, ta thấy rằng, cụm 0 có khoảng cách gần nhất, do đó ta phân người dùng này vào cụm 0:

$$\vec{c_0} = (0.931, 0.7241, 0.7241, 0.5862, 0.3448, 0.6552, 0.5172, 0.1379).$$

Nhận thấy những người dùng thuộc cụm 0 có xác suất truy cập trang `Prod_C` (0.6552) cao hơn so với `Prod_A` (0.5862) nên ta sẽ ưu tiên gợi ý sản phẩm C cho người dùng (mặc dù sản phẩm A vẫn có thể là một lựa chọn hợp lý do xác suất không quá cách biệt).

3. Làm tương tự như câu 2, ta có điểm dữ liệu mới là $\vec{a} = (0, 1, 0, 0, 0, 1, 0, 0)$, với bảng số liệu được cho trong Bảng 4.

Tâm	0	1	2	3	4	5
Khoảng cách	1.52811	1.89066	2.42384	2.20478	1.24896	1.9658

Bảng 4: Khoảng cách từng tâm cụm đến $\vec{s} = (0, 1, 0, 0, 0, 1, 0, 0)$

Dựa vào bảng số liệu, ta thấy cụm 4 có khoảng cách gần nhất, ta chọn cụm 4.

$$\vec{c_4} = (0, 1, 0.3214, 0, 1, 0.6786, 0.5, 0.3214)$$

Do loại người dùng trong cụm này không truy cập trang `Prod_A` (0), nên ta sẽ gợi ý cho người dùng sản phẩm ở trang `Prod_C` (0.6786).

4. Dựa vào nhận xét ở trên, ta có một vài quan sát sau:

- Nhóm người dùng thông thường: nghĩa là người dùng có xu hướng "dạo chơi" qua các trang. Ta có thể thấy rằng, cụm 0 là nhóm người dùng gần giống nhất với mô tả này nhất. Do xác suất truy cập trang mua hàng thấp (0.1379) nên nhóm người dùng thông thường này có xu hướng mua hàng thấp.

  Ví dụ người dùng thứ 24: $\vec{x_{24}} = (1, 1, 0, 1, 1, 1, 1, 0)$ hay người dùng thứ 35: $\vec{x_{35}} = (1, 1, 1, 1, 1, 1, 0, 0)$ đều cho thấy xu hướng truy cập nhiều trang (xem mọi sản phẩm) nhưng không mua hàng.

- Nhóm người dùng tập trung: nghĩa là người dùng biết trước được mình sẽ mua sản phẩm nào. Hình dung là những người dùng này biết về sản phẩm mình cần mua nhưng không biết nhiều mặt hàng, thương hiệu nổi tiếng của sản phẩm, thay vì dùng tìm kiếm, họ vào trang sản phẩm (`Products`) để xem trang bán hàng có những gì. Những người này có xu hướng tập trung nhiều vào một số mặt hàng. Như vậy, cụm 1 và cụm 4 thể hiện nhóm người dùng này, trong đó cụm 1 tập trung hơn vào sản phẩm A (và một chút sản phẩm B), còn cụm 4 tập trung vào sản phẩm B (và một chút sản phẩm C). Nhóm người dùng này có xu hướng mua hàng trung bình (0.3 - 0.5).

  Trong cụm 1, lấy ví dụ người dùng 18: $\vec{x_{18}} = (1, 1, 0, 1, 1, 0, 0, 0)$, nhận thấy người dùng này quan tâm đến sản phẩm A và B nhưng không mua hàng. Mặt khác người dùng 58: $\vec{x_{58}} = (1, 1, 0, 1, 0, 0, 1, 1)$ quan tâm đến sản phẩm A và có mua hàng.

  Trong cụm 4, lấy ví dụ người dùng $\vec{x_{39}} = (0, 1, 0, 0, 1, 0, 0, 0)$ vào xem sản phẩm và vào xem B nhưng không mua hàng. Tuy nhiên vẫn có người dùng $\vec{x_{46}} = (0, 1, 1, 0, 1, 1, 1, 1)$ vào xem sản phẩm B, C và có đặt mua.

- Nhóm người tìm kiếm: nghĩa là người dùng vào trang tìm kiếm để tìm sản phẩm cần mua. Cụm 2 và cụm 3 thể hiện loại người dùng này, và nhóm người dùng này có xác suất mua sản phẩm rất cao (0.5 và 0.8333).

  Trong cụm 2, ví dụ người dùng thứ 4, $\vec{x_4} = (1, 0, 1, 1, 1, 0, 1, 1)$ có vào trang tìm kiếm và có mua sản phẩm. Trong cụm 3 cũng có người dùng giống vậy, $\vec{x_{12}} = (1, 0, 1, 1, 0, 0, 1, 1)$.

5. Quan sát bảng số liệu, ta có thể thấy loại người dùng ở cụm 1, 2, 4 và 5 thể hiện rõ sở thích hay xu hướng mua hàng của mình:

   - Cụm 1: Người dùng có sở thích xem sản phẩm A nhiều hơn với xác suất truy cập trang sản phẩm A rất cao (0.9167) so với những sản phẩm khác. Nhóm người dùng này cũng thường vào trang `Products` để xem. Tuy nhiên khả năng mua hàng của những người này khoảng 58%. Ví dụ $\vec{x_{18}} = (1, 1, 0, 1, 1, 0, 0, 0)$, $\vec{x_{58}} = (1, 1, 0, 1, 0, 0, 1, 1)$.

   - Cụm 2: Người dùng có sở thích với cả 3 sản phẩm A, B và C. Những người này là những người dùng tìm kiếm và có xác suất mua hàng trung bình. Ví dụ $\vec{x_4} = (1, 0, 1, 1, 1, 0, 1, 1)$ xem qua cả 3 sản phẩm.

   - Cụm 4: Người dùng cũng có sở thích xem sản phẩm B và C nhiều hơn so với các sản phẩm khác. Nhóm người dùng này cũng thường xuyên vào trang `Products` để xem. Khả năng mua hàng của nhóm người này khá thấp, chỉ 1 trong 3 người sẽ mua hàng. Ví dụ $\vec{x_{39}} = (0, 1, 0, 0, 1, 0, 0, 0)$.

- Cụm 5: Người dùng có sở thích xem sản phẩm A nhiều, tiếp đó là sản phẩm B. Nhóm người dùng này hiếm vào xem `Products` và `Search` mà vào trực tiếp trang sản phẩm (có thể do đường dẫn từ trang khác, như quảng cáo, hoặc giới thiệu của bạn bè). Xu hướng mua hàng của những người dùng này chỉ trên trung bình một chút. Ví dụ $\vec{x_{47}} = (0, 0, 1, 1, 0, 0, 1, 1), \vec{x_{79}} = (0, 0, 0, 1, 1, 0, 0, 0)$.

6. Cũng quan sát bảng số liệu, cụm 5 cho thấy người dùng không hề vào `Home` và rất ít vào `Product` hay `Search` (cùng là 0.111). Như vậy, khả năng là người dùng đã bị thu hút bởi quảng cáo (đặt ở một trang bên ngoài) nên đã vào thẳng trang xem sản phẩm.

   Xác suất truy cập trang `Prod_A` cao nhất, sau đó là trang `Prod_B` với xác suất lần lượt là 1 và 0.5556. Trong khi đó, trang `Prod_C` có xác suất truy cập là 0, tức hầu như không ai vào. Có thể thấy điều này hợp với sự thật là chỉ có quảng cáo của A và B là được đặt.

   Như vậy, chiến dịch quảng cáo cho sản phẩm A thành công hơn sản phẩm B.

2 Nội dung thực hiện cài đặt

2.1 Hướng dẫn sử dụng

Mã nguồn chương trình nằm ở tập tin `1612838_1612849.py`. Để chạy chương trình cần cài đặt Python 3 và thư viện numpy:

```
pip3 install numpy
```

Chương trình có thể chạy bằng dòng lệnh Terminal

```
python3 1612838_1612849.py [input] [output_model] [output_asgn] [k]
```

Trong đó

- `[input]`: đường dẫn đến tập tin .csv
- `[output_model]`: đường dẫn đến nơi lưu `model.txt`
- `[output_asgn]`: đường dẫn đến nơi lưu `assignment.csv`
- `[k]`: số cụm mong muốn

2.2 Ví dụ

```
python3 1612838_1612849.py test.csv model.txt assignments.csv 2
```

Dòng lệnh trên sẽ lấy điểm dữ liệu trong tập tin `test.csv` ở cùng thư mục, phân thành 2 cụm bằng thuật toán `k-means` và lưu mô hình ở tập tin `model.txt` và phân cụm ở `assignments.csv`, cả hai đều trong cùng thư mục.

Ví dụ tập tin `test.csv` sau.

```
A,B
1,1
2,1
1,2
4,6
3,5
4,4
4,5
```

Khi chạy dòng lệnh trên sẽ tạo ra tập tin `model.txt` như sau.

```
Within cluster sum of squared errors: 4.083333333333333
Cluster centroids:
                           Cluster #
Attribute                      0              1
                             (3)            (4)
=============================================
A                           1.3333         3.7500
B                           1.3333         5.0000
```

Kèm theo tập tin `assignments.csv` như sau.

```
A,B,Cluster
1.0,1.0,0
2.0,1.0,0
1.0,2.0,0
4.0,6.0,1
3.0,5.0,1
4.0,4.0,1
4.0,5.0,1
```

Trong quá trình chạy, màn hình cũng hiển thị

```
Generated random seed. Seed is: 5707993357160619285

-- Summary --
Size of data: 7
Dimension: 2
Number of clusters: 2
-- End Summary --

Finding initial centroid...
-- kmeans++ --
Centroids #0: [4. 6.]
Centroids #1: [1. 1.]
-- End kmeans++ --
Done, took 0.002627 (s).
Initial centroids:
 [[4. 6.]
 [1. 1.]]

-- Begin Loop --
Iteration #0... Done, took 0.000185 (s). SSE = 4.083333.
Iteration #1... Done, took 0.000158 (s). SSE = 4.083333.
-- End Loop --
Loop took 0.000560 (s).

Calculating SSE...Done, took 0.000048 (s).
Generating model.txt...Done, took 0.000731 (s).
Generating assignments.csv...Done, took 0.000388 (s).
```

2.3 So sánh với kết quả của Weka

Trên tập dữ liệu `sessions.csv`, việc chạy `k-means` tự cài đặt cho kết quả trong bảng 5.

Phiên bản tự cài đặt này dùng công thức tính SSE là

$$SSE = \sum_{\vec{x} \in \mathcal{X}} \min_{\vec{c} \in \mathcal{C}} \|\vec{x} - \vec{c}\|^2$$

Hay nói đơn giản nó là tổng bình phương khoảng cách của mỗi điểm đến tâm gần nhất (nói cách khác là tâm cụm mà nó thuộc về) của nó.

Nhìn chung, kết quả gom cụm thu được thường có SSE xấp xỉ nhau qua nhiều lần chạy với các hạt giống khác nhau.

Ngoài ra, việc phân cụm cũng tạo ra các cụm khá khác nhau. Đơn cử với $k = 5$, một số cụm tìm được là

- Cụm 0: Người đi dạo quanh tất cả các trang với khả năng cao sẽ mua hàng, giống cụm 4 của bản Weka.
- Cụm 2: Người đi dạo quanh tất cả các trang với khả năng thấp sẽ mua hàng, giống cụm 1 của bản Weka.
- Cụm 3: Người vào Product và ưu tiên B, C hơn A, khá giống với cụm 5 của phiên bản Weka.

Các cụm còn lại thể hiện các hành vi khác. Ví dụ cụm 4 gồm những người thích A nhất và thích B, C ngang nhau, trong khi gần nhất với điều này là cụm 4 trong Weka cho thấy những người cũng thích A nhất nhưng lại ít thích C). Hoặc cụm 1 thể hiện xu hướng biết chắc sản phẩm cần tìm, chỉ vào trang sản phẩm A, B, C mà hầu như không vào Home, Product hay Search.

Thực tế thì kết quả của kmeans trong trường hợp này bị ảnh hưởng chủ yếu bởi cách khởi tạo tâm cụm. Về bản chất, việc so sánh giữa Weka và bản tự cài đặt có lẽ chỉ đơn giản là đang nhìn vào 2 kết quả khả dĩ khác nhau của thuật kmeans với cụm khởi tạo khác nhau.

Nhóm cũng đã thử chạy trên tập dữ liệu `letter.arff` từ Lab 03 với 26 cụm. Kết quả SSE của nhóm là khoảng 490k trong khi của Weka là 2.2k (với cùng tâm cụm khởi tạo). Có vẻ như khi kích thước lớn, Weka sẽ chia tỉ lệ khi tính SSE để tránh tràn số (mặc dù không rõ cách chia tỉ lệ này là gì).

k	SSE	Cluster centroids							
			Home	Products	Search	ProdA	ProdB	ProdC	Cart	Purchase
3	135.5760	0	0.9184	0.6735	0.4694	0.8367	0.2449	0.4082	0.8163	0.4898
		1	0.0625	0.7500	0.0624	0.3125	0.8125	0.5625	1.000	0.9375
		2	0.4000	0.7714	0.5429	0.2000	0.8571	0.4571	0.1429	0.0000
4	121.7571	0	0.9535	0.7442	0.3953	0.8605	0.3023	0.4186	0.8372	0.5581
		1	0.0000	0.1111	0.1111	1.000	0.4444	0.0000	0.7778	0.6667
		2	0.7917	0.6250	0.8750	0.2500	0.5833	0.3333	0.2917	0.0000
		3	0.0000	1.0000	0.1667	0.0417	1.0000	0.7917	0.4583	0.3750
5	107.0309	0	0.9286	0.6786	0.6071	0.8214	0.3214	0.4286	1.0000	0.8571
		1	0.0000	0.1111	0.1111	1.0000	0.4444	0.0000	0.7778	0.6667
		2	0.7059	0.7647	0.9412	0.0000	0.5882	0.2353	0.4706	0.0000
		3	0.0000	1.0000	0.1739	0.0000	1.0000	0.8261	0.4783	0.3913
		4	0.9565	0.6957	0.2174	0.9130	0.3913	0.4348	0.3043	0.0000
6	98.0118	0	1.0000	0.7059	0.3529	0.8235	0.4118	0.1176	1.0000	1.0000
		1	0.0000	0.1111	0.1111	1.0000	0.4444	0.0000	0.7778	0.6667
		2	0.6667	0.7333	1.0000	0.0000	0.6667	0.2000	0.4000	0.0000
		3	0.0000	1.0000	0.1739	0.0000	1.0000	0.8261	0.4783	0.3913
		4	0.9600	0.7200	0.2400	0.8800	0.3600	0.4000	0.3600	0.0000
		5	0.8182	0.6364	1.0000	0.7273	0.1818	1.0000	1.0000	0.6364
7	97.2062	0	1.0000	0.7059	0.3529	0.8235	0.4118	0.1176	1.0000	1.0000
		1	0.0000	0.1250	0.1250	1.0000	0.5000	0.0000	0.7500	0.7500
		2	0.6667	0.7333	1.0000	0.0000	0.6667	0.2000	0.4000	0.0000
		3	0.0000	1.0000	0.1739	0.0000	1.0000	0.8261	0.4783	0.3913
		4	0.9600	0.7200	0.2400	0.8800	0.3600	0.4000	0.3600	0.0000
		5	0.8182	0.6364	1.0000	0.7273	0.1818	1.0000	1.0000	0.6364
		6	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000
8	82.5517	0	1.0000	0.7778	0.4444	1.0000	1.0000	0.7778	0.5556	0.0000
		1	0.9000	0.3000	0.9500	0.6500	0.1000	0.5000	0.8000	0.7000
		2	0.0000	1.0000	0.1538	0.0769	1.0000	0.6923	0.0000	0.0000
		3	0.3636	0.0000	0.0000	0.8182	0.3636	0.0000	0.4545	0.3636
		4	0.0000	1.0000	0.1000	0.0000	1.0000	0.9000	1.0000	0.9000
		5	0.9231	1.0000	0.3077	0.7692	0.0000	0.4615	0.5385	0.0000
		6	0.5000	0.9167	1.0000	0.0000	0.9167	0.2500	0.5000	0.0000
		7	0.9167	1.0000	0.0833	0.9167	0.5000	0.0833	1.0000	1.0000

Bảng 5: Kết quả chạy thuật toán `k-means` tự cài đặt.

3 Tìm hiểu cải tiến k-means

Báo cáo được viết dựa trên bài báo *"k-means++: The Advantages of Careful Seeding"* của nhóm tác giả David Arthur và Sergei Vassilvitskii (Đại học Stanford), được đăng trong Hội nghị thường niên ACM-SIAM về các thuật toán rời rạc lần thứ 18, năm 2007.

3.1 Bài toán k-means

Giả sử có một tập n điểm dữ liệu trong không gian $\mathbb{R}^d$ chiều.

$$\mathcal{X} = \{\vec{x} : \vec{x} \in \mathbb{R}^d\}, |\mathcal{X}| = n$$

Cho một số nguyên dương k, mục tiêu của chúng ta là tìm một tập k **tâm**, $\mathcal{C} \subset \mathbb{R}^d, |\mathcal{C}| = k$ (lưu ý các tâm không nhất thiết phải là các điểm thuộc $\mathcal{X}$), sao cho giá trị sau là tối thiểu

$$\Phi = \sum_{\vec{x} \in \mathcal{X}} \min_{\vec{c} \in \mathcal{C}} \|\vec{x} - \vec{c}\|^2$$

Với mỗi tâm $\vec{c_i}$ tìm thấy ở trên, ta định nghĩa một **cụm** $\mathcal{C}_i$ **tâm** $\vec{c_i}$, là tập các điểm dữ liệu $\vec{x}$ gần $\vec{c_i}$ hơn các tâm còn lại. Nói cách khác

$$\mathcal{C}_i = \{\vec{x} : \vec{x} \in \mathcal{X}, \|\vec{x} - \vec{c_i}\| \leq \|\vec{x} - \vec{c_j}\|, \forall j \neq i\}$$

Có thể thấy Φ đại diện cho tổng bình phương khoảng cách từ mỗi điểm đến tâm cụm mà nó thuộc về.

Về bản chất, đây là một bài toán NP-khó.

3.2 Thuật toán `k-means`

Một thuật toán xấp xỉ để giải quyết bài toán k-means được đề xuất bởi Stuart LLoyd lần đầu năm 1957. Thuật toán chạy như sau.

1. Chọn ngẫu nhiên tập k tâm $\mathcal{C}$.
2. Xác định các cụm tâm $\mathcal{C}_i$ từ tập $\mathcal{C}$ đã chọn.
3. Cập nhật tập k tâm mới: với mỗi $i = 1..k$, tâm mới $\vec{c_i'}$ là trọng tâm của mọi điểm trong $\mathcal{C}_i$, $\vec{c_i'} = |\mathcal{C}_i|^{-1} \sum_{\vec{x} \in \mathcal{C}_i} \vec{x}$.
4. Lặp 2 và 3 cho tới khi hội tụ ($\mathcal{C}$ giữ nguyên sau bước cập nhật).

Một cách cơ bản để chọn ngẫu nhiên trong bước 1 là chọn ngẫu nhiên theo phân phối đều từ các đỉnh trong $\mathcal{X}$.

Có thể chứng minh bước 2 và bước 3 của thuật toán sẽ đảm bảo giảm giá trị Φ và do đó tốt lên theo từng lần lặp. Tuy nhiên nó chỉ đạt được tối ưu cục bộ và chịu ảnh hưởng rất lớn từ việc lựa chọn tập tâm khởi đầu, gọi là **hạt giống**, hay bước 1. Việc chọn ngẫu nhiên như đề xuất cơ bản không cho một đảm bảo hiệu quả nào của giải thuật.

3.3 Thuật toán `k-means++`

k-means++ là thuật toán được đề xuất nhằm cải tiến bước 1, tức bước tìm các hạt giống. Ý tưởng của cải tiến rất đơn giản: chọn các hạt giống sao cho chúng xa rời nhau nhất có thể.

Cụ thể, gọi $D(\vec{x})$ là khoảng cách ngắn nhất từ một điểm tới tâm gần nó nhất trong các tâm đã chọn. Nói cách khác, giả sử ta đã chọn được i tâm $\mathcal{C} = \{c_1, c_2, \ldots, c_i\}$:

$$D_i(\vec{x}) = \min_{\vec{c} \in \mathcal{C}} \|\vec{x} - \vec{c}\|$$

Lưu ý khi điểm $\vec{x}$ càng gần một tâm nào đó thì $D_i(\vec{x})$ càng nhỏ, hay $D_i(\vec{x})$ sẽ lớn khi điểm $\vec{x}$ nằm xa rời so với các tâm đã có. Như vậy, có thể tính giá trị xác suất chọn điểm tỉ lệ thuận với $D_i(x)$. Cụ thể, xác suất để chọn điểm $\vec{x}$ làm tâm là

$$P_i(\vec{X} = \vec{x}) = \frac{D_i(\vec{x})^2}{\sum_{\vec{z} \in \mathcal{X}} D_i(\vec{z})^2}$$

Phần mẫu có mục đích chuẩn hóa giúp P_i là một phân phối xác suất ($\sum_{\vec{x} \in \mathcal{X}} P_i(\vec{x}) = 1$).

Từ đó, thuật toán k-means++ đề xuất cách tìm hạt giống như sau.

1. Chọn một tâm đầu tiên c_1 ngẫu nhiên theo phân phối đều từ các điểm trong $\mathcal{X}$.
2. Chọn tâm c_{i+1} từ $\mathcal{X}$ với phân phối P_i.
3. Lặp bước 2 cho đến khi tìm được đủ k tâm.

Với tập tâm hạt giống này, thực hiện cập nhật tương tự như thuật toán k-means.

Việc tìm hạt giống theo cách này sẽ đảm bảo các tâm khởi tạo sẽ nằm khá xa cách lẫn nhau, tăng khả năng các tâm xấp xỉ các cụm tối ưu và do đó giúp hội tụ nhanh và ít bị kẹt ở tối ưu cục bộ hơn. Theo bài báo, thuật toán `k-means++` cho kết quả xấu nhất (giá trị Φ) lớn gấp $\mathcal{O}(\log k)$ lần kết quả tối ưu.

Cũng cần lưu ý rằng việc tìm hạt giống thế này sẽ tốn thời gian hơn do phải tìm $D_i(x)^2$ của mỗi điểm (và phải tìm k lần). Do đó nếu n, k lớn thì thời gian chạy của bước này có thể hơn nhiều lần so với sinh ngẫu nhiên thông thường. Tuy nhiên, kết quả thực nghiệm ở phần cho thấy việc này cũng không quá ảnh hưởng (bởi nó tối ưu đáng kể thời gian hội tụ).

Ví dụ 1 *Cho biết điểm từng môn trong học kì của hai sinh viên A và B trong Bảng 6.*

Ta muốn chia các môn học thành 2 nhóm dựa trên thống kê này.

Với trường hợp `k-means`, để tìm hạt giống thuật toán sẽ chọn 2 điểm với phân bố đều. Một khả năng có thể là sẽ chọn cặp $(4, 4)$ *và* $(4, 5)$ *là điểm khởi đầu, với xác suất là 1 trên* $C_7^2 = 21$ *tức khoảng 4.76%.*

Tuy nhiên, trong trường hợp `k-means++`, xác suất để mỗi điểm trong 2 điểm đó được chọn là tâm đầu tiên là 1 trên 7.

Giả sử tâm đầu là $(4, 4)$*. Các giá trị* $D_1(\vec{x})^2$ *từ các điểm còn lại đến nó được cho trong bảng 7.*

Môn	A	B
1	1	1
2	2	1
3	1	2
4	4	6
5	3	5
6	4	4
7	4	5

Bảng 6: Điểm từng môn trong học kỳ của hai sinh viên.

$\vec{x}$	$D_1(\vec{x})^2$
$(1,1)$	18
$(2,1)$	13
$(1,2)$	13
$(4,6)$	4
$(3,5)$	2
$(4,5)$	1

Bảng 7: Giá trị $D_1(\vec{x})$ ứng với $(4,4)$.

Xác suất để tâm thứ hai là $(4,5)$ *là*

$$p(4,5) = \frac{1}{18+13+13+4+2+1} = \frac{1}{51}$$

Tương tự, giả sử tâm đầu là $(4,5)$ *thì bảng giá trị* $D_1(\vec{x})^2$ *được cho trong bảng 8.*

$\vec{x}$	$D_1(\vec{x})^2$
$(1,1)$	25
$(2,1)$	20
$(1,2)$	18
$(4,6)$	1
$(3,5)$	1
$(4,4)$	1

Bảng 8: Giá trị $D_1(\vec{x})$ ứng với $(4,5)$.

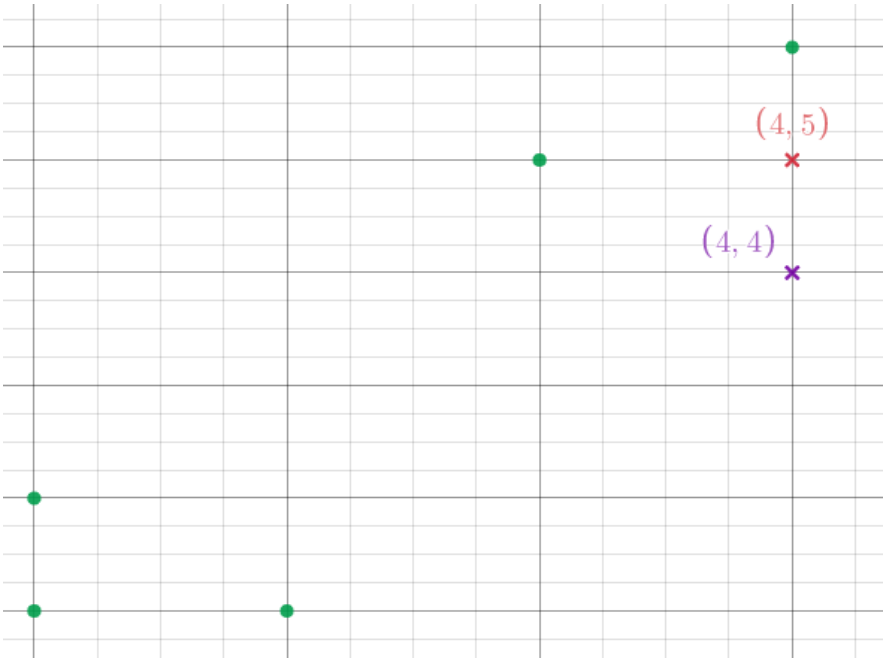
Xác suất để tâm thứ hai là $(4,4)$ *là*

$$p(4,4) = \frac{1}{25+20+18+1+1+1} = \frac{1}{66}$$

Vậy xác suất để hai tâm được chọn là $(4, 4)$ *và* $(4, 5)$ *là*

$$\frac{1}{7}\left(\frac{1}{51} + \frac{1}{66}\right) = 0.5\%$$

Có thể thấy xác suất trong trường hợp `k-means++` *nhỏ hơn rất nhiều so với trong trường hợp* `k-means`*. Hình 7 minh họa tập dữ liệu và 2 điểm này. Có thể thấy 2 điểm này không thực sự là một khởi đầu tốt.*



Hình 7: Minh họa tập dữ liệu và hạt giống.

3.4 Thực nghiệm

Trong bài báo, tác giả đã đưa ra một số bảng số liệu thực nghiệm trên các bộ dữ liệu Norm-10 (Bảng 9), Norm-25 (Bảng 10), Cloud (Bảng 11), Intrusion (Bảng 12).

Trong mỗi bảng, giá trị Φ chính là giá trị tổng bình phương nội cụm như trên, giá trị T là thời gian chạy. Các giá trị này được xét trung bình hoặc tối thiểu.

Có thể thấy trong đa số bảng thì thuật toán `k-means++` đều cho giá trị Φ và T thấp hơn hẳn so với thuật toán `k-means`. Đa số là do khi số cụm lớn, việc tìm hạt giống tốn thời gian nhiều ngang ngửa việc hội tụ (như có trình bày ở trên).

Nhóm cũng đã có cài đặt thuật toán trong mã nguồn bên trên và khi chạy thử với ví dụ phía trên thì `kmeans++` luôn cho hạt giống là 2 tâm ở 2 cụm khác nhau.

k	Average Φ		Minimum Φ		Average T	
	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	10898	5.122	2526.9	5.122	0.48	0.05
25	787.992	4.46809	4.40205	4.41158	1.34	1.59
50	3.47662	3.35897	3.40053	3.26072	2.67	2.84

Bảng 9: Kết quả thực nghiệm trên tập Norm-10 ($n = 10000, d = 5$)

k	Average Φ		Minimum Φ		Average T	
	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	135512	126433	119201	111611	0.14	0.13
25	48050.5	15.8313	25734.6	15.8313	1.69	0.26
50	5466.02	14.76	14.79	14.73	3.79	4.21

Bảng 10: Kết quả thực nghiệm trên tập Norm-25 ($n = 10000, d = 15$)

k	Average Φ		Minimum Φ		Average T	
	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	7553.5	6151.2	6139.45	5631.99	0.12	0.05
25	3626.1	2064.9	2568.2	1988.76	0.19	0.09
50	2004.2	1133.7	1344	1088	0.27	0.17

Bảng 11: Kết quả thực nghiệm trên tập Cloud ($n = 1024, d = 10$)

k	Average Φ		Minimum Φ		Average T	
	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	3.45×10^8	2.31×10^7	3.25×10^8	1.79×10^7	107.5	64.04
25	3.15×10^8	2.53×10^6	3.1×10^8	2.06×10^6	421.5	313.65
50	3.08×10^8	4.67×10^5	3.08×10^8	3.98×10^5	766.2	282.9

Bảng 12: Kết quả thực nghiệm trên tập Intrusion ($n = 494019, d = 35$)

Tài liệu

[1] Arthur, David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding." Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2007.