Bài toán gom cụm

Biên soạn: **ThS. Vũ Đình Ái***(aivd@huflit.edu.vn)*

Nội dung

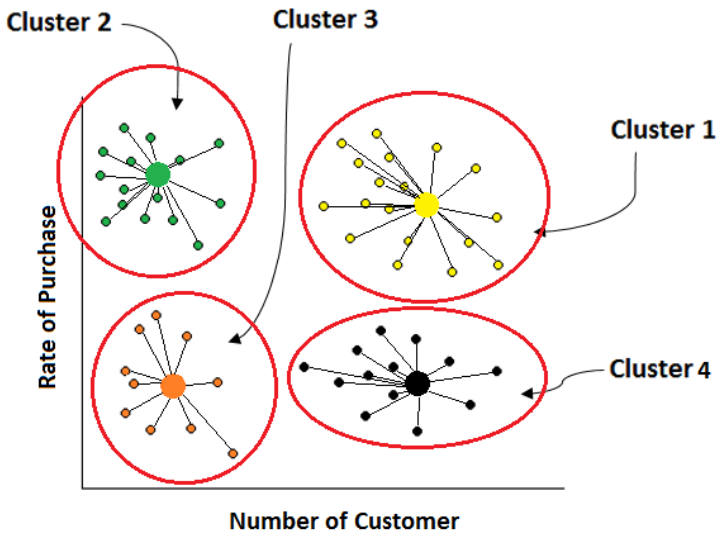
- Giới thiệu bài toán gom cụm
- Thuật toán K-mean
- Thuật toán DBScan

Giới thiệu bài toán gom cụm

- Bài toán phân cụm là 1 nhánh ứng dụng chính của lĩnh vực Unsupervised Learning (Học không giám sát), trong đó dữ liệu được mô tả trong bài toán không được dán nhãn (tức là không có đầu ra).
- Trong trường hợp này, thuật toán sẽ tìm cách phân cụm - chia dữ liệu thành từng nhóm có đặc điểm tương tự nhau, nhưng đồng thời đặc tính giữa các nhóm đó lại phải càng khác biệt càng tốt.
- Có rất nhiều định nghĩa khác nhau về kỹ thuật này, nhưng về bản chất ta có thể hiểu phân cụm là các qui trình tìm cách nhóm các đối tượng đã cho vào các cụm (clusters), sao cho các đối tượng trong cùng 1 cụm tương tự (similar) nhau và các đối tượng khác cụm thì không tương tự (Dissimilar) nhau

Giới thiệu bài toán gom cụm

- Mục đích của phân cụm là tìm ra bản chất bên trong các nhóm của dữ liệu.
- Các thuật toán phân cụm (Clustering Algorithms) đều sinh ra các cụm (clusters). Tuy nhiên, không có tiêu chí nào là được xem là tốt nhất để đánh hiệu của của phân tích phân cụm, điều này phụ thuộc vào mục đích của phân cụm như: data reduction, “natural clusters”, “useful” clusters, outlier detection



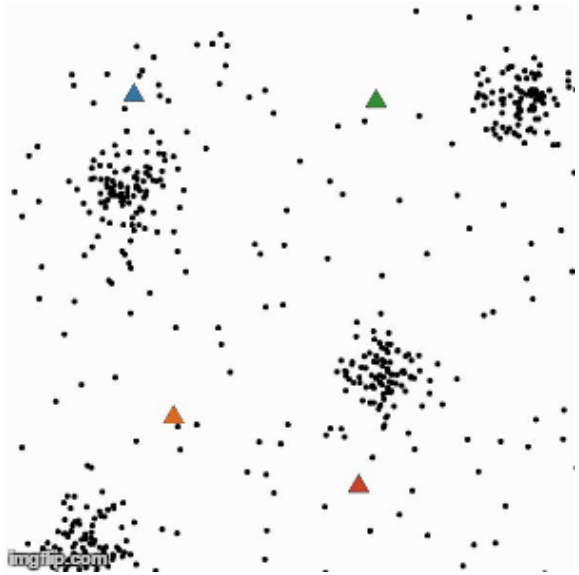
Giới thiệu bài toán gom cụm

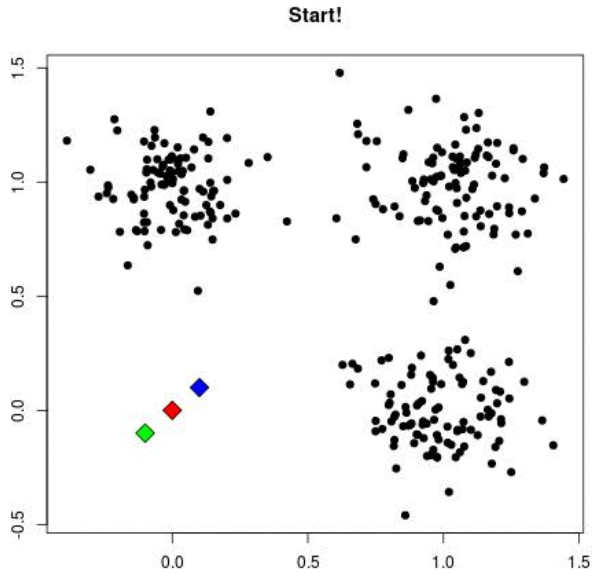
- Kỹ thuật phân cụm có thể áp dụng trong rất nhiều lĩnh vực như:
  - Marketing: Xác định các nhóm khách hàng (khách hàng tiềm năng, khách hàng giá trị, phân loại và dự đoán hành vi khách hàng,…) sử dụng sản phẩm hay dịch vụ của công ty để giúp công ty có chiến lược kinh doanh hiệu quả hơn;
  - Biology: Phận nhóm động vật và thực vật dựa vào các thuộc tính của chúng;
  - Insurance, Finance: Phân nhóm các đối tượng sử dụng bảo hiểm và các dịch vụ tài chính, dự đoán xu hướng (trend) của khách hàng, phát hiện gian lận tài chính (identifying frauds);...

Giới thiệu bài toán gom cụm

Thuật toán K-mean

- Thuật toán phân cụm K-means được giới thiệu năm 1957 bởi Lloyd K-means và là phương pháp phổ biến nhất cho việc phân cụm, dựa trên việc phân vùng dữ liệu
- Thuật toán K-Means là một trong những thuật toán phân cụm dữ liệu dựa trên học không giám sát được sử dụng nhiều trong các học máy nói chung và trong khai phá dữ liệu nói riêng.
- Thuật toán K-mean là một loại của nhóm thuật toán Unsupervised Learning. Trong đó các dữ liệu ban đầu được phân thành cụm dựa trên vị trí tương đối của chúng so với nhau.





Thuật toán K-mean

- Biểu diễn dữ liệu: $D = \{x_1, x_2, x_3, \ldots, x_n\}$ với là vector n chiều trong không gian Euclidean. K-means phân cụm D thành K cụm dữ liệu:
  - Mỗi cụm dữ liệu có một điểm trung tâm gọi là centroid.
  - K là một hằng số cho trước.

Thuật toán K-mean

- Các bước trong thuật toán K-Means
  - **Đầu vào**: Cho tập dữ liệu D, với K là số cụm, phép đo khoảng cách giữa 2 điểm dữ liệu là d(x,y)
  - **Khởi tạo**: Khởi tạo K điểm dữ liệu trong D làm các điểm trung tâm (centroid)
  - **Lặp lại** các bước sau đến khi **hội tụ**:
    - ***Bước 1***: Với mỗi điểm dữ liệu, gán điểm dữ liệu đó vào cluster có khoảng cách đến điểm trung tâm của cluster là nhỏ nhất.
    - ***Bước 2:*** Với mỗi cluster, xác định lại điểm trung tâm của tất cả các điểm dữ liệu được gán vào cluster đó.

Thuật toán K-mean

- Điều kiện hội tụ (điều kiện dừng thuật toán)
- Ta sẽ xác định điều kiện dừng thuật toán theo một số cách như sau:
  - Tại 1 vòng lặp: có ít các điểm dữ liệu được gán sang cluster khác hoặc
  - Điểm trung tâm (centroid) không thay đổi nhiều hoặc
  - Giá trị hàm mất mát không thay đổi nhiều:

$$\text{Error} = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, m_i)^2$$

  - Trong đó:
    - ✓ C_i là cụm thứ i
    - ✓ m_i là điểm trung tâm cụm C_i tương ứng

Thuật toán K-mean

- Điểm trung tâm

$$m_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

- Trong đó:
  - C_i là cụm thứ i
  - m_i là điểm trung tâm cụm C_i tương ứng

Thuật toán K-mean

- Phép đo khoảng cách
- Trong K-means để đánh giá **mức độ giống nhau** hay **khoảng cách giữa 2 điểm dữ liệu** ta có thể sử dụng các phép đo khoảng cách khác nhau. Ngoài khoảng cách **Euclidean**, tuỳ thuộc vào từng bài toán có thể sử dụng phương pháp đo khác (**cosine**, **manhattan**…)

$$d(x, m_i) = \sqrt{(x_1 - m_{i1})^2 + \ldots + (x_n - m_{in})^2}$$

Thuật toán K-mean

- Xét ví dụ : tiến hành phân tập dữ liệu thành 3 cụm

X1	X2
2	4
2	6
5	6
4	7
8	3
6	6
5	2
5	7
6	3
4	4

Thuật toán K-mean

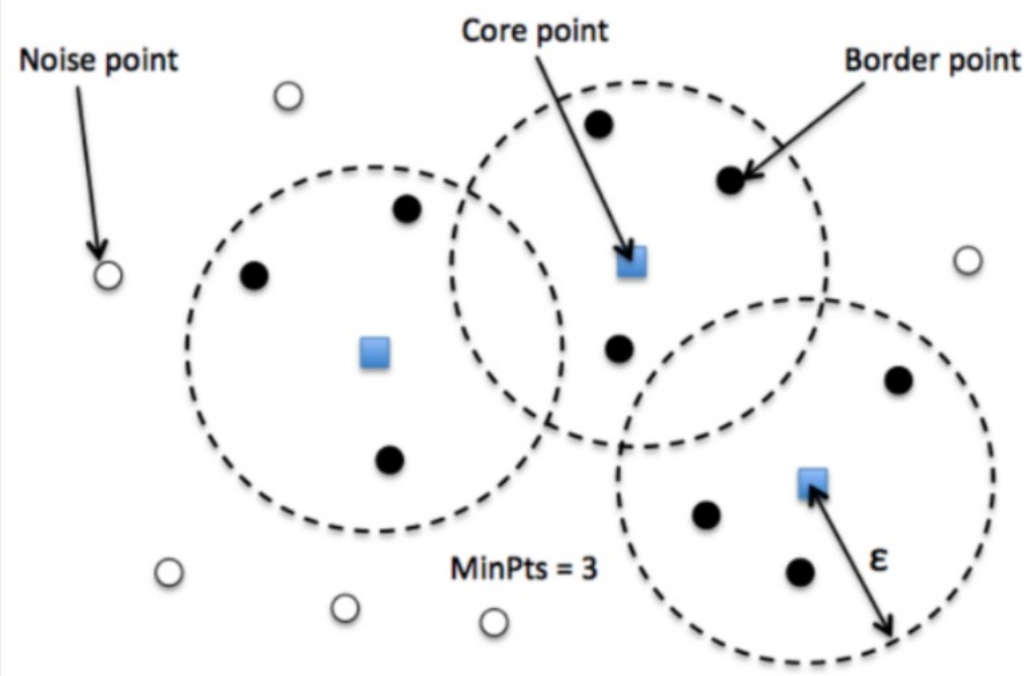
				Distance to			Cluster
iter5	X1	X2	C1=(2,5)	C2 = (5.75,3)	C3=(5.33,6.3		number
C1=(2,5)	2	4	1	3.88	4.06		C1
C2 = (5.75,3)	2	6	1	4.8	3.35		C1
C3=(5.33,6.33)	5	6	3.16	3.009	0.47		C3
	4	7	2.83	4.37	1.49		C3
	8	3	6.32	2.25	4.27		C2
Dừng thuật toán do	6	6	4.12	3.01	0.75		C3
không có sự thay đổi điểm giữa	5	2	4.24	1.25	4.34		C2
các cụm	5	7	3.61	4.07	0.75		C3
	6	3	4.47	0.25	3.4		C2
	4	4	2.24	2.02	2.68		C2

Thuật toán DBScan

- DBSCAN (là viết tắt của cụm từ tiếng Anh: density-based spatial clustering of applications with noise, tạm dịch là phân cụm không gian dựa trên mật độ các ứng dụng với nhiễu) là một thuật toán phân tích cụm do Martin Ester, Hans-Peter Kriegel, Jörg Sander và Xiaowei Xu đề xuất vào năm 1996.
- Ý tưởng:
  - Đây là thuật toán phân cụm dựa theo mật độ phi tham số: với một tập các điểm đã cho trong một số không gian,
  - thuật toán sẽ gom nhóm các điểm với nhau (các điểm có nhiều hàng xóm lân cận với bán kính cố định (fixed-radius near neighbors)) thành một nhóm, và đánh dấu là các điểm ngoại lệ nếu chúng nằm tách biệt với các điểm đã gom nhóm ở các vùng mật độ thấp (các điểm có các hàng xóm gần nhất ở khoảng cách quá xa).

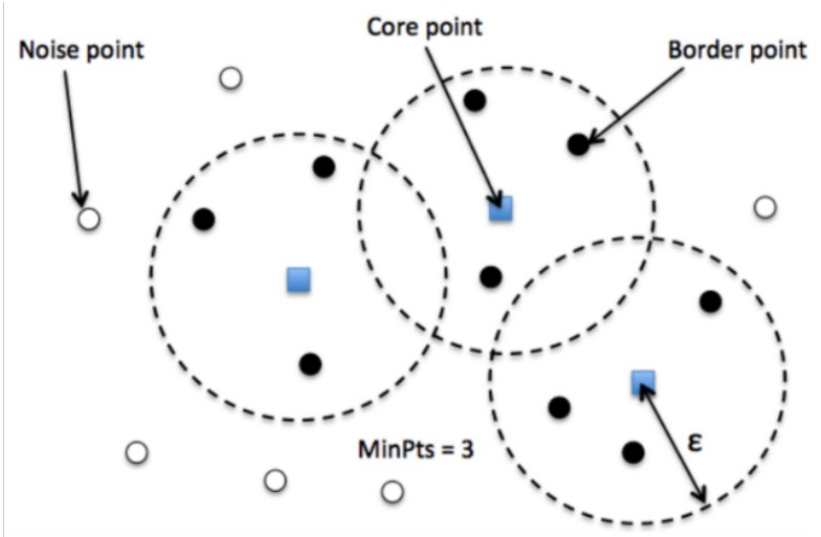
Thuật toán DBScan

- Căn cứ vào vị trí của các điểm dữ liệu so với cụm chúng ta có thể chia chúng thành ba loại:
  - Đối với các điểm nằm sâu bên trong cụm chúng ta xem chúng là **điểm lõi.**
  - Các **điểm biên** nằm ở phần ngoài cùng của cụm và **điểm nhiễu** không thuộc bất kì một cụm nào.



Thuật toán DBScan

- Xác định ba loại điểm bao gồm
  - điểm lõi (core) chấm vuông màu xanh,
  - điểm biên(border),
  - chấm tròn màu đen và điểm nhiễu (noise) chấm tròn màu trắng trong thuật toán DBSCAN.
- Các hình tròn đường viền nét đứt bán kính thể hiện vùng lân cận epsilon tương ứng với các điểm lõi nhằm xác định nhãn cho từng điểm. minPts=3 là số lượng tối thiểu để một điểm lõi rơi vào vùng có mật độ cao nếu xung quanh chúng có số lượng điểm tối thiểu là 3.



Thuật toán DBScan

- Trong thuật toán DBSCAN sử dụng hai tham số chính đó là:
  - **minPts:** Là một ngưỡng số điểm dữ liệu tối thiểu được nhóm lại với nhau nhằm xác định một vùng lân cận epsilon có mật độ cao. Số lượng minPts không bao gồm điểm ở tâm.
  - **epsilon (kí hiệu ε):** Một giá trị khoảng cách được sử dụng để xác định vùng lân cận epsilon của bất kỳ điểm dữ liệu nào.

Thuật toán DBScan

- Hai tham số trên giúp xác định ba loại điểm:
  - **điểm lõi (core):** Đây là một điểm có ít nhất minPts điểm trong vùng lân cận epsilon của chính nó.
  - **điểm biên (border):** Đây là một điểm có ít nhất một điểm lõi nằm ở vùng lân cận epsilon nhưng mật độ không đủ minPts điểm.
  - **điểm nhiễu (noise):** Đây là điểm không phải là điểm lõi hay điểm biên.

Thuật toán DBScan

- Các bước của thuật toán:
  - Bước 1: lựa chọn một điểm dữ liệu bất kì. Sau đó tiến hành xác định các **điểm lõi** và **điểm biên** thông qua vùng lân cận **epsilon** bằng cách lan truyền theo liên kết chuỗi các điểm thuộc cùng một cụm.
  - Bước 2: Cụm hoàn toàn được xác định khi không thể mở rộng được thêm. Khi đó lặp lại đệ qui toàn bộ quá trình với điểm khởi tạo trong số các điểm dữ liệu còn lại để xác định một cụm mới.

Thuật toán DBScan

- Xét ví dụ

ID	X	Y
1	3	7
2	4	6
3	5	5
4	6	4
5	7	3
6	6	2
7	7	2
8	8	4
9	3	3
10	2	6
11	3	5
12	2	4

minPts=4 và epsilon=1.9

Thuật toán DBScan

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0											
P2	1.41	0										
P3	2.83	1.41	0									
P4	4.24	2.83	1.41	0								
P5	5.66	4.24	2.83	1.41	0							
P6	5.83	4.47	3.16	2.00	1.41	0						
P7	6.40	5.00	3.61	2.24	1.00	1.00	0					
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0				
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0			
P10	1.41	2.00	3.16	4.47	5.83	5.66	6.40	6.32	3.16	0		
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

minPts=4 và epsilon=1.9

P1	P2, P10
P2	P1, P3, P11
P3	P2, P4
P4	P3, P5
P5	P4, P6, P7, P8
P6	P5, P7
P7	P5, P6
P8	P5
P9	P12
P10	P1, P11
P11	P2, P10, P12
P12	P9, P11

Thuật toán DBScan

minPts=4 và epsilon=1.9

P1	P2, P10
P2	P1, P3, P11
P3	P2, P4
P4	P3, P5
P5	P4, P6, P7, P8
P6	P5, P7
P7	P5, P6
P8	P5
P9	P12
P10	P1, P11
P11	P2, P10, P12
P12	P9, P11

Point	Status	
P1	Noise	Border
P2	Core	
P3	Noise	Border
P4	Noise	Border
P5	Core	
P6	Noise	Border
P7	Noise	Border
P8	Noise	Border
P9	Noise	
P10	Noise	Border
P11	Core	
P12	Noise	Border

Thuật toán DBScan

- Kết luận:
  - Cụm 1: P2, P1,P3,P11
  - Cụm 2: P5, P4,P6,P7,P8
  - Cụm 3: P11, P2,P10,P12