Lời giải phần Tự nghiên cứu

Phạm Tiến Sơn

5/2021

1 Phần câu hỏi chính - Phân cụm K-means

1.1 Câu hỏi lý thuyết

LT1 Thuật toán K-means chia tập dữ liệu đầu vào không có nhãn (không được phân nhóm trước) thành K nhóm nhỏ phân biệt (cụm) dựa vào sự liên quan giữa các điểm dữ liệu, từ đó chúng ta có thể dán nhãn cho từng nhóm và sử dụng trong bài toán cụ thể. [1]

- **Đầu vào:** Dữ liệu đầu $\mathbf{X}$ vào không có nhãn và số lượng cụm K mong muốn.
- **Đầu ra:** Trọng tâm $\mathbf{m}_j$ của các cụm C_j $(1 \leqslant j \leqslant K)$ và cụm của từng điểm dữ liệu $\mathbf{x}_i$ trong X.
- **Mục tiêu:** Chia tập dữ liệu vào các cụm sao cho tổng của độ khác nhau giữa các điểm dữ liệu trong cùng một cụm của K cụm là nhỏ nhất. Độ khác nhau giữa các điểm dữ liệu trong cùng một cụm thường được định nghĩa là:

$$W(C_j) = \frac{1}{|C_j|} \sum_{\mathbf{x}_i, \mathbf{x}_{i'} \in C_j} \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2^2 \tag{1}$$

Hàm $W(C_j)$ trên còn có thể được viết dưới dạng tổng bình phương khoảng cách của các điểm dữ liệu với trọng tâm của cụm - giá trị trung bình cộng của các điểm trong cụm:

$$W(C_j) = \frac{1}{|C_j|} \sum_{\mathbf{x}_i, \mathbf{x}_{i'} \in C_j} \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2^2 = 2 \sum_{\mathbf{x}_i \in C_j} \left\|\mathbf{x}_i - \mathbf{m}_j\right\|_2^2 \tag{2}$$

$$\text{trong đó} \quad \mathbf{m}_j = \frac{1}{|C_k|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$$

Vậy bài toán tối ưu trong K-means là làm tối thiểu hóa hàm mất mát: [2]

$$\underset{C_1, C_2, \ldots, C_K}{\operatorname{argmin}} \left\{ \sum_{j=1}^{K} W(C_j) \right\} = \underset{C_1, C_2, \ldots, C_K}{\operatorname{argmin}} \left\{ 2 \sum_{j=1}^{K} \sum_{\mathbf{x}_i \in C_j} \left\|\mathbf{x}_i - \mathbf{m}_j\right\|_2^2 \right\} \tag{3}$$

LT2 Mô tả thuật toán phân cụm K-means: [2]

Algorithm 1 K-means

Input: Tập dữ liệu $\mathbf{X}$, số cụm mong muốn K

Output: Tập trọng tâm $\mathbf{M}$ và tập label vector $\mathbf{Y}$

1: Chọn ngẫu nhiên K điểm làm các trọng tâm của các cụm ban đầu.
2: **Phân cụm:** Phân mỗi điểm dữ liệu vào cụm có trọng tâm gần nhất với nó (*gần nhất* ở đây có thể được định nghĩa theo khoảng cách Euclidean $\|\cdot\|_2$)
3: **Cập nhật:** Tìm trọng tâm mới của mỗi cụm trong K cụm bằng cách lấy trung bình cộng của các điểm dữ liệu trong cùng cụm đó.
4: Thực hiện lại các bước 2 và 3 cho đến khi không có sự thay đổi nào được thực hiện ở bước phân cụm.

Giải thích:

- Với những trọng tâm cho trước, bước 2 tối thiểu giá trị của hàm (2):

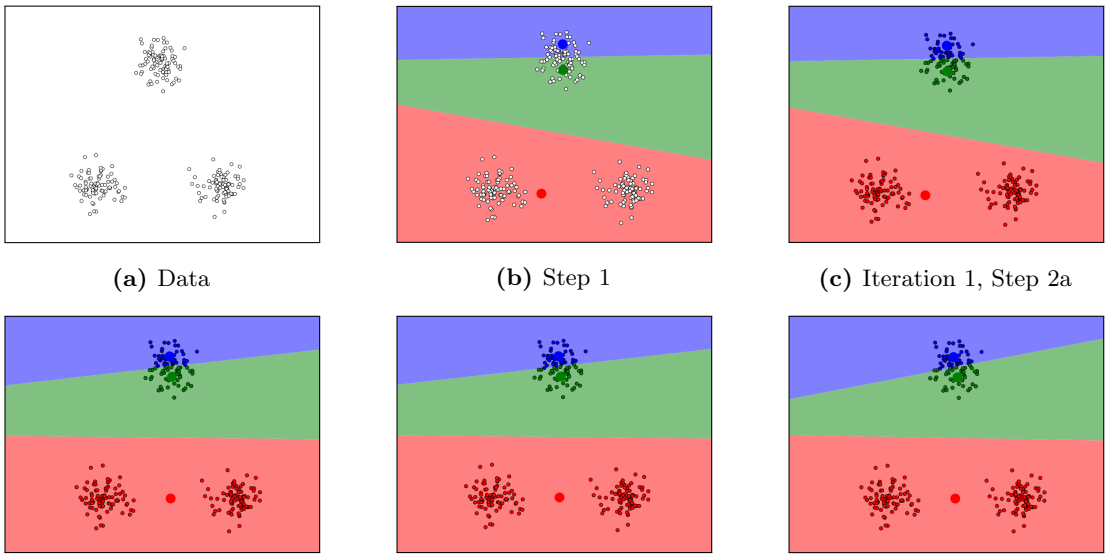
$$\arg\min_j \left\|\mathbf{x}_i - \mathbf{m}_j\right\|_2^2$$

  Điều này phù hợp với trực giác của chúng ta: Các điểm dữ liệu trong một cụm sẽ nằm gần trọng tâm của cụm đó.

- Sau khi các điểm trong cụm bị thay đổi, Bước 3 cũng tối thiểu hóa giá trị hàm (2):

$$\arg\min_{m_j} \left\|\mathbf{x}_i - \mathbf{m}_j\right\|_2^2 = \frac{1}{|C_k|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$$

- Thuật toán sẽ chạy đến khi không có sự thay đổi nào được thực hiện và hàm (3) hội tụ về một giá trị cực tiểu.



(a) Data **(b)** Step 1 **(c)** Iteration 1, Step 2a

(d) Iteration 1, Step 2b **(e)** Iteration 2, Step 2a **(f)** Final Results

Hình 1: *Quá trình chạy của thuật toán K-means với tập dữ liệu "Gaussian Mixture", số cụm là 3 [3]: Nếu các trọng tâm được khởi tạo trong cùng một cụm thực (1b) thì kết quả cuối cùng không tối ưu (1f).*

LT3 Với một nhóm các trọng tâm được khởi tạo ngẫu nhiên, thuật toán K-means có thể đưa chúng vào vị trí trung tâm của những nhóm các điểm dữ liệu gần chúng nhất. Tuy nhiên, giới hạn của K-means là chúng thường không thể tối ưu vị trí trọng tâm một cách toàn cục. [4]

Như ở Hình 3, việc có các trọng tâm được đặt vào cùng một cụm thực khiến cho K-means đưa ra một kết quả rất phản trực giác. Các bước phân điểm dữ liệu về gần trọng tâm (Bước 2) và cập nhật trọng tâm mới (Bước 3) cho thấy chúng không thể di chuyển một trọng tâm sang cụm khác khi khoảng cách đến cụm đó quá lớn hoặc bị những cụm dữ liệu khác cản trở.

Do đó kết quả của thuật K-means phụ thuộc rất lớn đến vị trí khởi tạo ban đầu của các trọng tâm.

LT4 k-means ++ là thuật toán khởi tạo các điểm trọng tâm cho K-means được sử dụng phổ biến.

Algorithm 2 K-means ++

Input: Tập dữ liệu $\mathbf{X}$, số cụm mong muốn K

Output: Tập các trọng tâm ban đầu $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K\}$

1: Chọn ngẫu nhiên điểm $\mathbf{m}_1$ trong tập $\mathbf{X}$, $\mathbf{M} = \{\mathbf{m}_1\}$
2: Với mỗi điểm còn lại $\mathbf{x}$ trong tập $\mathbf{X}$, tính $d(\mathbf{x}) = \min_{\mathbf{m}\in\mathbf{M}} \|\mathbf{x} - \mathbf{m}\|_2$.
3: Chọn một điểm trong tập $\mathbf{X}$ là trọng tâm mới $\mathbf{m}_j$ với xác suất chọn điểm $\mathbf{x}$ là $\frac{d(\mathbf{x})^2}{\sum_{\mathbf{x}\in\mathbf{X}} d(\mathbf{x})^2}$
$\mathbf{M} = \mathbf{M} \cup \{\mathbf{m}_j\}$
4: Lặp lại bước 2 và 3 đến khi đủ K trọng tâm được tạo.

Mặc dù thuật k-means ++ tốn nhiều thời gian hơn là đơn thuần lựa chọn ngẫu nhiên K điểm, thực nghiệm cho thấy sử dụng k-means ++ để khởi tạo các trọng tâm thường cho sự cải thiện gấp đôi về tốc độ và, với một số tập dữ liệu, cho sự cải thiện gần 1000 lần về độ chính xác.[5]

LT5 Dự đoán:

- Tập dữ liệu blobs: Thuật K-means có thể dễ dàng tìm được chính xác các cụm.
- Tập dữ liệu mouse: Thuật K-means rất khó tìm được chính xác hai cụm, dự đoán rằng ở phần lớn lần chạy K-means không thể bắt chính xác được cụm dữ liệu tô màu tím.
- Tập dữ liệu circles: Thuật K-means không thể tìm được chính xác hai cụm.

Giải thích:

Do đặc điểm của hàm đo độ khác nhau giữa các điểm trong cụm sử dụng khoảng cách Euclidean (2), K-means sẽ cho kết quả tốt nhất với các cụm thường có cụm dữ liệu có kích cỡ, hình dạng giống nhau và nằm tách biệt nhau như ở tập blobs. Với những tập dữ liệu có kích cỡ khác nhau như mouse, cụm lớn thường bị chia nhỏ và những các cụm nhỏ thì bị gộp lại với phần của cụm khác.

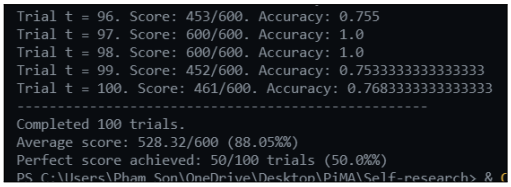
K-means cũng không thể tìm chính xác một cụm ở bên trong cụm khác như ở tập circles.

1.2 Câu hỏi lập trình

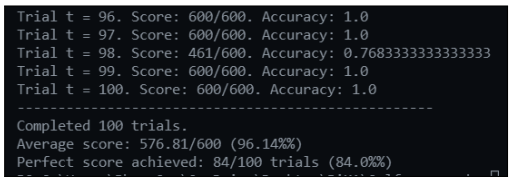
TH1 Đã hoàn thành.

TH2 Như đã nói ở LT3, thuật K-means có thể trả về kết quả rất tệ khi ngẫu nhiên có các trọng tâm được khởi tạo trong cùng một cụm thực.

Thuật toán k-means ++ được triển khai ở đây chọn một điểm dữ liệu để khởi tạo trọng tâm với xác suất tỉ lệ thuận với bình phương khoảng cách từ điểm đó đến trọng tâm gần nhất. Nó không chỉ có xu hướng chọn các trọng tâm cách xa nhau mà, với yếu tố ngẫu nhiên, các trọng tâm có khả năng được khởi tạo ở các điểm gần trung tâm của cụm hơn là khi khởi tạo ngẫu nhiên hoặc khởi tạo bằng cách chọn điểm xa nhất (*farthest-point heuristic* - không được trình bày ở đây).



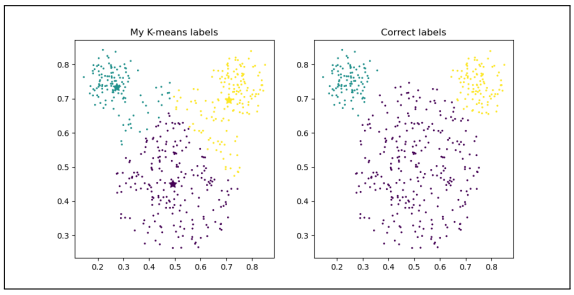
(a) Khởi tạo ngẫu nhiên



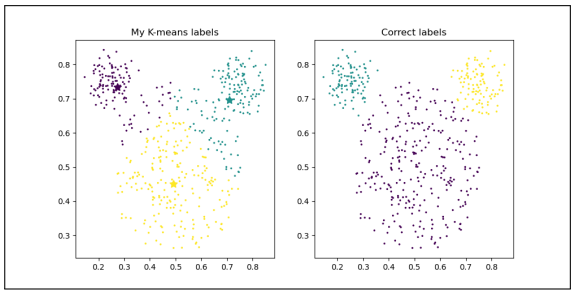
(b) Sử dụng thuật k-means ++

Hình 2: *So sánh kết quả của thuật K-means trên tập dữ liệu blob với hai cách khởi tạo trọng tâm.*

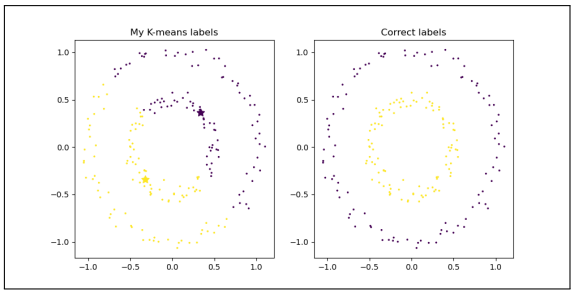
TH3 Thuật toán lựa chọn các trọng tâm ban đầu k-means ++ được sử dụng không cải thiện output của K-means trên các dataset "mouse" và "circle". Lý do là, như đã nói ở LT5, K-means không tìm được cụm chính xác là do sử dụng khoảng cách Euclidean, nên dù khi các trọng tâm có được khởi tạo ở các vị trí tốt hơn ta cũng nhận được kết quả tốt hơn.



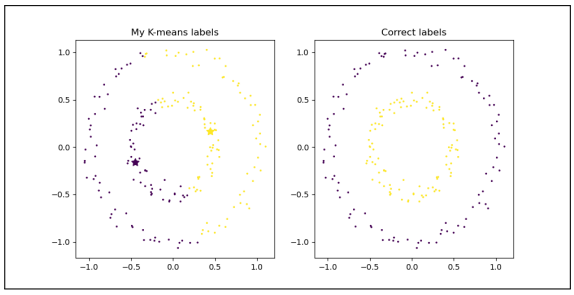
(a) Khởi tạo ngẫu nhiên trọng tâm trên tập "mouse"



(b) Sử dụng thuật k-means ++ trên tập "mouse"



(c) Khởi tạo ngẫu nhiên trọng tâm trên tập "circles"



(d) Sử dụng thuật k-means ++ trên tập "circles"

Hình 3: *So sánh kết quả của thuật K-means trên hai tập dữ liệu "mouse" và "circles" với hai cách khởi tạo trọng tâm.*

2 Câu hỏi nâng cao

NC4 K-means có thể được dùng để các thượng nghị sĩ vào 2 cụm sao cho các thành viên cùng đảng vào chung cụm nhiều nhất, do:

- Các thành viên cùng đảng thường có cách bỏ phiếu giống nhau cho các đạo luật.
- Lịch sử bỏ phiếu của mỗi nghị sĩ được đánh các số 1, 0.5 và 0 tương ứng cho phiếu Thuận, Trắng và Chống thuận lợi cho việc sử dụng thuật K-means.

NC5 Đầu vào là một ma trận với các phần tử thể hiện sự liên quan giữa các điểm dữ liệu (nhân viên) khiến cho thuật K-means không dễ dàng áp dụng được trong bài toán này.

Ở đây em đề xuất sử dụng một thuật toán phân cụm khác là *Spectral Clustering*.[6]

Tài liệu

[1] Vũ Hữu Tiệp. Bài 4: K-means clustering, 2017.

[2] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112, chapter 10.3.1 K-Means Clustering. Springer, 2013.

[3] Naftali Harris. Visualizing k-means clustering.

[4] Pasi Fränti and Sami Sieranoja. How much can k-means be improved by using better initialization and repeats? *Pattern Recognition*, 93:95–112, 2019.

[5] Wikipedia contributors. k-means++ - wikipedia, 2021. [Online; Truy cập 5-6-2021].

[6] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.