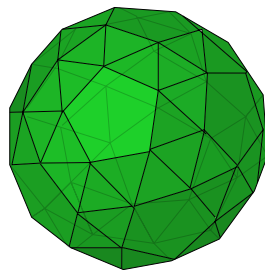


Projects in Mathematics and Applications

SPECTRAL CLUSTERING

Ngày 15 tháng 8 năm 2021

Vũ Phan Thăng Long ^{*} [†]Nguyễn Bá Khôi Nguyễn
Võ Minh Quân [‡] [§]Vòng Vĩnh Toàn



^{*}Đại học École Polytechnique

[†]Đại học Khoa học Tự nhiên TP.HCM

[‡]Trường THPT chuyên Hùng Vương

[§]Trường THPT chuyên Hùng Vương

Lời cảm ơn

Trước hết, chúng tôi xin gửi lời cảm ơn trân trọng đến với ban tổ chức trại hè Toán học và Ứng dụng PiMA, và các đơn vị tài trợ đã tạo điều kiện tốt nhất cho chúng tôi trong suốt quãng thời gian hai tuần tham gia trại. Mặc dù là năm đầu tiên tổ chức online do các nguyên nhân khách quan, nhưng các anh chị trong ban tổ chức đã cố gắng hết mình để trại hè có thể diễn ra thành công và suôn sẻ. Những kiến thức mà chúng tôi tích lũy được trong 2 tuần tại trại thực sự có ý nghĩa rất lớn đối với mỗi thành viên của nhóm, và là hành trang quý báu cho bản thân mỗi bạn trên con đường theo đuổi khoa học sau này.

Chúng tôi cũng gửi lòng biết ơn chân thành nhất đến các anh chị mentor đã nhiệt tình chỉ dạy và dẫn dắt chúng tôi trên những bước đầu trong công việc nghiên cứu. Chúng tôi cũng muốn gửi lời cảm ơn đến các giảng viên khách mời đã mang đến cho tất cả các bạn trại sinh những kinh nghiệm thiết thực về vấn đề nghiên cứu cũng như cơ hội việc làm sau này.

Cuối cùng xin cảm ơn tất cả các bạn trại sinh đã cùng nhau đồng hành với chúng mình trong kỳ trại vừa qua.

Do thời gian học tập cũng như nghiên cứu có hạn nên những thiếu sót là không thể tránh khỏi. Vì vậy nhóm chúng em rất mong nhận được sự đóng góp của các anh chị mentor cũng như các bạn để đề tài có thể được hoàn chỉnh hơn.

Tóm tắt nội dung

Bài báo cáo chủ yếu xoay quanh ý nghĩa toán học của thuật toán Unnormalized spectral clustering, các phiên bản cải tiến của nó và các vấn đề liên quan như Similarity matrix và Eigengap heuristic, đồng thời đưa ra một vài ví dụ cụ thể để minh họa cách vận hành các phương pháp này.

Mục lục

1	Mở đầu	1
1.1	Đặt vấn đề	1
1.2	Một số kí hiệu đồ thị	1
2	Unnormalized spectral clustering	2
2.1	Xử lý dataset	2
2.2	Mô hình hóa bài toán	5
2.3	Thuật toán Unnormalized Spectral Clustering	7
2.4	Chọn dữ liệu đầu vào	8
3	Áp dụng mô hình	9
4	Phiên bản cải tiến	11
5	Kết luận	13

1 Mở đầu

1.1 Đặt vấn đề

Ở section này, chúng ta sẽ bàn luận về vấn đề cần thuật toán Spectral Clustering giải quyết.

Unsupervised learning

Trong môn Máy học nói chung có hai phương pháp học chính đó là *unsupervised learning* và *supervised learning*. Thuật toán Spectral Clustering được trình bày trong bài báo cáo này thuộc lớp phương pháp học unsupervised learning.

Unsupervised learning là một phương pháp học với mục đích tìm ra cấu trúc dữ liệu khi không có đầu ra hay nhãn của dữ liệu đó. Thuật toán Spectral Clustering nằm trong dạng phân nhóm dữ liệu sẽ được nói ở phần tiếp đây.

Bài toán Clustering

Như đã nói ở phần trước, Spectral Clustering thuộc dạng phân nhóm dữ liệu (Clustering) là dạng bài toán gom nhóm một tập các đối tượng thỏa mãn các đối tượng trong cùng một nhóm (gọi là cụm, tên tiếng Anh: Cluster) sẽ có tính giống nhau hơn so với những đối tượng khác nhóm.

Một thuật toán phân cụm phổ biến là thuật toán k-means clustering cũng thuộc dạng phân cụm dữ liệu nhưng có một số nhược điểm nhất định:

- Thuật toán đưa ra một giả định có ảnh hưởng mạnh đến kết quả: các cluster có dạng một khối cầu (Với dataset là 2 vòng tròn đồng tâm thì k-means không thể cho ra cluster như ý).
- Kết quả đầu ra bị ảnh hưởng mạnh mẽ bởi những điểm khởi tạo ban đầu khiến cho thuật toán phải thực hiện lại nhiều lần để tìm ra đáp án tối ưu.

Thuật toán Spectral Clustering có những tính chất ưu việt hơn thuật toán k-means và có thể giải quyết những nhược điểm trên. Tiếp theo trong phần báo cáo sẽ đi sâu vào thuật toán Spectral Clustering và cách mà nó hoạt động.

1.2 Một số kí hiệu đồ thị

Trong toàn bài báo cáo, chúng tôi sử dụng một số kí hiệu đồ thị chung được nêu dưới đây sử dụng cho toàn bài báo cáo.

Cho một đồ thị đơn vô hướng có trọng số $G = (V, E)$ với tập các đỉnh $V = \{v_1, \dots, v_n\}$. Giữa 2 đỉnh v_i và v_j bất kì có cạnh nối giữa chúng, cạnh nối đó được đánh bằng một giá trị thực không âm $w_{ij} \geq 0$. Từ các giá trị w_{ij} ta dựng một *ma trận kề* (*adjacency matrix*) $W = (w_{ij})_{i,j=1,\dots,n}$ là ma trận trọng số giữa các đỉnh của đồ thị. Nếu $w_{i,j} = 0$ thì giữa 2 đỉnh v_i và v_j được xem là không có cạnh nối. Vì G là đơn đồ thị vô hướng nên ma trận kề W là ma trận đối xứng có đường chéo bằng không.

Bậc của một đỉnh $v_i \in V$ là một số thực d_i được định nghĩa bằng tổng trọng số các cạnh nối giữa đỉnh v_i và các đỉnh khác trong đồ thị. Vì ta đã định nghĩa nếu giữa 2 đỉnh v_i và v_j không có cạnh nối thì $w_{ij} = 0$ nên ta có thể viết lại công thức toán học bậc của đỉnh v_i như sau:

$$d_i = \sum_{j=1}^n w_{ij}$$

Ma trận bậc của đồ thị $G = (V, E)$ là một ma trận đường chéo với các giá trị bậc d_1, d_2, \dots, d_n nằm trên đường chéo:

$$D = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_n \end{bmatrix}$$

Cho một tập con các đỉnh của đồ thị $A \subset V$, ta kí hiệu phần bù $V \setminus A$ là \bar{A} . Để tiện trong phần viết, chúng tôi kí hiệu $i \in A$ cho $i \in \{i | v_i \in A\}$.

Chúng tôi định nghĩa “trọng số” giữa 2 tập con $A, B \subset V$ như sau:

$$W(A, B) := \sum_{i \in A, j \in B} w_{ij}$$

Thêm vào đó, chúng tôi định nghĩa thêm hai cách để đo lường “độ lớn” của một tập con $A \subset V$ như sau:

$$|A| := \text{số đỉnh trong tập con } A$$

$$\text{vol}(A) := \sum_{i \in A} d_i$$

2 Unnormalized spectral clustering

2.1 Xử lý dataset

Dữ liệu đầu vào của thuật toán là các data points cùng với một số hyperparameter cụ thể tùy thuộc vào loại mô hình người dùng lựa chọn. Thuật toán làm việc trên đồ thị vì thế nên ta phải chuyển các điểm dữ liệu thành một đồ thị để thuật toán làm việc trên đồ thị đó.

Với một dataset cho trước, có khá nhiều cách phổ biến để chuyển dataset từ các điểm có mối quan hệ khoảng cách hoặc giống nhau giữa hai điểm thành một đồ thị, điểm chung giữa các cách chuyển đổi là xem các điểm dữ liệu là đỉnh của đồ thị và các cạnh thể hiện mối quan hệ giữa các đỉnh đó. Mục đích của việc chuyển đổi đồ thị là thể hiện mối quan hệ gần kề giữa các điểm dữ liệu.

Khi nhắc đến khái niệm “độ giống nhau” giữa hai điểm dữ liệu, ta có thể định nghĩa bằng bất kì hàm nào thỏa mãn các điểm càng gần nhau thì độ giống nhau càng lớn, hàm được định nghĩa theo cách này được gọi là similarity function. Một similarity function phổ biến là Gaussian similarity function được định nghĩa như sau:

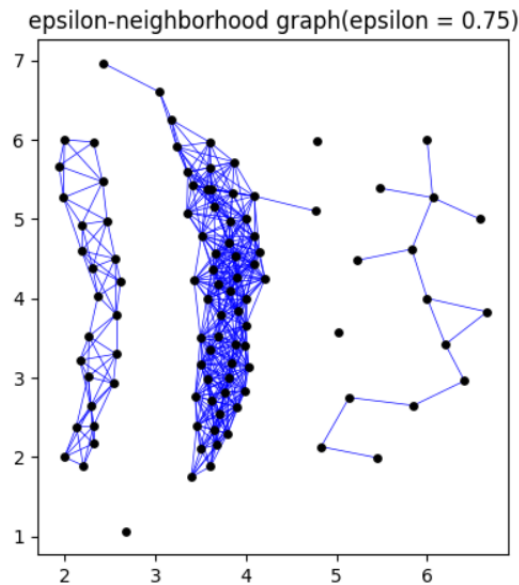
$$s(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Trong Gaussian similarity function có một siêu tham số σ dùng để tùy chỉnh mức độ giống nhau. Thêm vào đó norm ở đây cũng có thể xem là siêu tham số vì có thể sử dụng bất kì loại norm nào tùy ý, thường sẽ sử dụng 2-norm vì có ý nghĩa hình học là khoảng cách giữa 2 điểm. Bình phương của norm còn có ý nghĩa tăng độ tương phản lớn nhỏ cho độ giống nhau, norm càng lớn thì độ giống nhau sẽ càng tiến về đến 0.

Sau đây là một số similarity graph phổ biến:

The ϵ -neighborhood graph

Trong ϵ -neighborhood graph ta nối những đỉnh có khoảng bé hơn ϵ lại với nhau. Vì các cạnh được nối có khoảng cách có thể xem như cùng tỉ lệ (đều bé hơn ϵ) nên việc gán trọng số cho các cạnh không mang lại bất kì thông tin hữu ích nào cho thuật toán. Vì thế, đồ thị ϵ -neighborhood thường là đồ thị không trọng số.

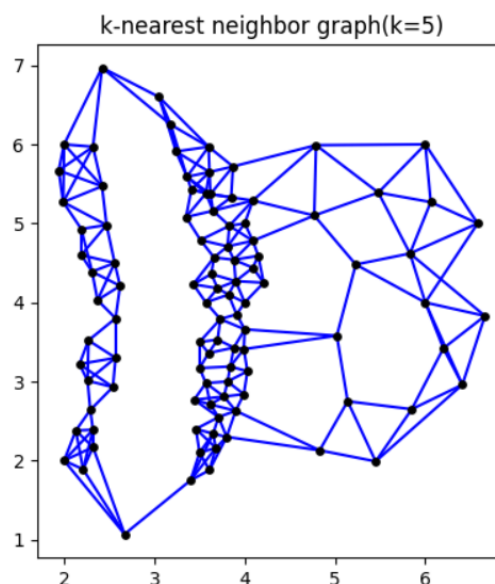


Đặc điểm: Graph ở hình trên là một ϵ -neighborhood graph trên tập dữ liệu tự sinh ngẫu nhiên. Ta có thể thấy rằng việc chọn hyperparameter ϵ một cách hợp lý và hữu dụng với dataset đã cho là một việc rất khó. Với lựa chọn $\epsilon = 0.75$ ở dataset trong hình ảnh trên, vùng các điểm nằm ở giữa được kết nối một cách chặt chẽ với nhau nhưng vùng phía bên phải thì kết nối với nhau một cách rất thưa thớt. Vấn đề này luôn xảy ra với một dataset gồm nhiều vùng có “mật độ” khác nhau.

The k-nearest neighbor graphs

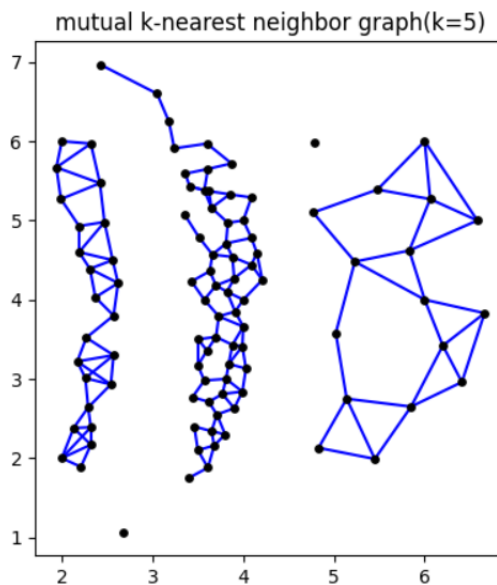
Trong k-nearest neighbor graphs, ta sẽ nối cạnh dựa trên k điểm gần điểm đang xét nhất, nhưng điều này sẽ có thể dẫn đến đồ thị tạo nên có hướng, để giải quyết vấn đề có hướng, ta đề xuất 2 loại k-nearest neighbor graphs:

- k-nearest neighbor graph: Loại này là loại thông thường. Ta sẽ nối 2 đỉnh v_i và v_j lại với nhau nếu v_j nằm trong k đỉnh gần v_i nhất hoặc v_i nằm trong k đỉnh gần v_j nhất.



Đặc điểm: Graph ở hình trên là một k -nearest neighbor graph với $k = 5$ trên cùng tập dữ liệu. Khác với ϵ -neighborhood graph, k -nearest neighbor graph có thể làm việc trên một dataset có nhiều vùng dữ liệu có mật độ khác nhau. Mặt khác, graph được tạo theo cách này có thể thể hiện mối tương quan giữa những vùng dữ liệu có mật độ khác nhau, ví dụ như ở dataset ở hình ảnh trên, vùng dữ liệu ở bên phải có một số cạnh nối với vùng dữ liệu ở giữa cũng như vùng dữ liệu ở giữa với vùng dữ liệu bên trái. Tính chất chung này của k -nearest neighbor graph sẽ rất hữu dụng trong việc áp dụng thuật toán. Đôi khi k -nearest neighbor graph còn có thể tách các vùng dữ liệu có mật độ dày đặc thành các thành phần liên thông nếu chúng ở khoảng cách vừa đủ xa nhau.

- Mutual k -nearest neighbor graph: Ta chỉ nối cạnh v_i và v_j khi v_j nằm trong k đỉnh gần v_i nhất và v_i nằm trong k đỉnh gần v_j nhất.

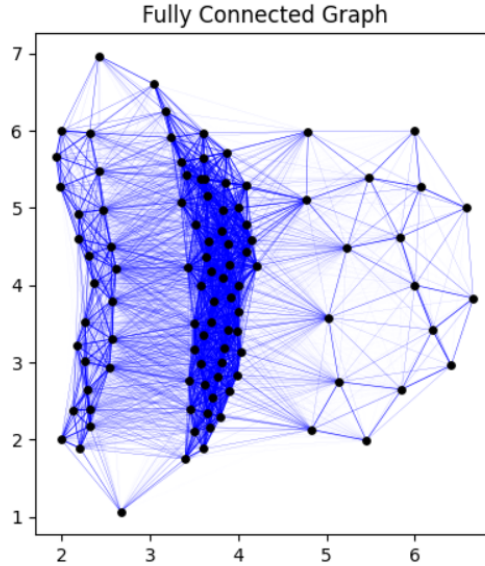


Đặc điểm: Graph ở hình trên là một mutual k -nearest neighbor graph với $k = 5$ trên cùng tập dữ liệu. Một mutual k -nearest neighbor graph có thể được xem là một sự kết hợp giữa ϵ -neighborhood graph và k -nearest neighbor graph khi nó chỉ có thể nối cạnh các điểm nằm trong một vùng dữ liệu có “mật độ” nhất định nhưng có thể sử dụng trên một dataset gồm nhiều vùng có “mật độ” khác nhau. Mutual k -nearest neighbor graph sẽ là một lựa chọn tốt nếu ta muốn các cluster có dạng “mật độ” khác nhau giữa các cluster.

Trong cả 2 loại k -nearest neighbor graphs, đồ thị sinh ra đều là đồ thị có trọng số và trọng số giữa một cặp đỉnh có cạnh nối là độ giống nhau giữa chúng có giá trị bằng similarity function đã chọn.

The fully connected graph

Như tên gọi của mình, trong fully connected graph, mọi cặp đỉnh đều có một cạnh vô hướng nối giữa chúng ($\frac{n(n-1)}{2}$ cặp đỉnh). Mỗi cạnh của fully connected graph được đánh trọng số bằng độ giống nhau của cặp đỉnh đang xét.



Đặc điểm: Thông thường, fully connected graph sẽ thường đo lường sự giống nhau bằng Gaussian similarity function: $s(x_i, x_j) = \exp(-||x_i - x_j||^2 / 2\sigma^2)$ với hyperparameter σ giữ vai trò tương tự ϵ trong ϵ -neighborhood graph. Giữa những điểm gần kề nhau sẽ có cạnh nối có trọng số tương đối lớn trong khi những đỉnh xa nhau sẽ có cạnh nối có trọng số dương nhưng dường như không đáng kể. Thế nhưng một điểm trừ khá lớn là ma trận kề tạo ra từ fully connected graph không phải là một ma trận thưa, dẫn đến việc tính toán trên ma trận bị tăng độ phức tạp làm giảm độ hiệu quả về thời gian và có thể là bộ nhớ trong thuật toán.

2.2 Mô hình hóa bài toán

2.2.1 Bài toán tối ưu đồ thị

Sau khi chọn một loại similarity graph và biến dataset thành một đồ thị có trọng số, ta có thể nhìn bài toán phân cụm dưới góc nhìn của bài toán phân hoạch đồ thị. Một cách hình dung đơn giản là ta chia các điểm theo sự “giống nhau” của chúng. Tức là ta cần 2 điểm ở 2 cụm thì khác nhau (có trọng số thấp), còn 2 đỉnh trong cùng 1 cụm thì giống nhau (có trọng số cao). Cụ thể, với cách định nghĩa $W(A, B)$ trong mục 1.2, bài toán phân cụm trở thành: Cho đồ thị $G = (V, E)$ và k là số cụm, ta cần tìm một phân hoạch $(A_i)_{i=1}^k$ của V để tối thiểu biểu thức

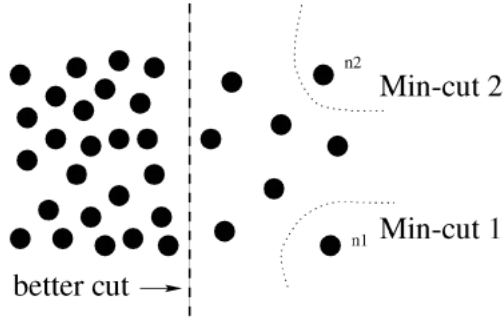
$$\sum_{i=1}^k W(A_i, \bar{A}_i).$$

Ở đây ta gặp một vấn đề chính đó là, do bản chất của hàm W , bài toán phân hoạch trên thường dẫn đến kết quả không như ý muốn, cụ thể là nó sẽ chia đồ thị của mình thành những điểm riêng lẻ như trong Hình 1.

Hagen và Kahng [2] đã đề xuất một cách để giải quyết vấn đề này, đó là yêu cầu các tập A_i phải gồm nhiều điểm, hay nói cách khác là các $|A_i|$ đủ lớn. Cụ thể ta cần phải tìm một phân hoạch $(A_i)_{i=1}^k$ của V để tối thiểu đại lượng

$$\mathcal{P} = \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{|A_i|}. \quad (1)$$

Đây là bài toán tối ưu đồ thị.



Hình 1: Ví dụ về phân hoạch không tối ưu [6]

2.2.2 Bài toán tối ưu ma trận

Bây giờ ta sẽ đưa biểu thức (1) về dạng tối ưu ma trận. Để làm được vậy thì ta cần giới thiệu hai loại ma trận đặc biệt.

Định nghĩa 2.1. Cho đồ thị $G = (V, E)$ với D, W lần lượt là ma trận bậc và ma trận kề của G . Khi đó unnormalized Laplacian matrix L của G được định nghĩa là

$$L = D - W.$$

Mọi Laplacian matrix từ đây trở về sau đều hiểu là unnormalized Laplacian matrix trừ khi có giải thích thêm.

Định nghĩa 2.2. Với mỗi phân hoạch $(A_i)_{i=1}^k$ của V , ta định nghĩa k vector $h_i = (h_{1,i}, \dots, h_{n,i})^T$ với

$$h_{j,i} = \begin{cases} \frac{1}{\sqrt{|A_i|}}, & v_j \in A_i \\ 0, & \text{TH khác} \end{cases}, \quad (2)$$

và H là ma trận nhận h_i làm cột thứ i .

Do các h_i mang đầy đủ thông tin của phân hoạch $(A_i)_{i=1}^k$ nên việc tìm một phân hoạch của V trở thành việc tìm ma trận H . Vì vậy tiếp theo ta sẽ tìm hiểu một vài tính chất của ma trận H (và L).

Mệnh đề 2.3. Các vector h_i tạo thành hệ vector trực chuẩn. Nói cách khác, $H^T H = I_k$.

Chứng minh. Ta có $h_i \cdot h_j = 0$ do A_i, A_j không giao nhau với $i \neq j$ và $h_i \cdot h_i = \sum_{j \in A_i} \frac{1}{|A_i|} = 1$. \square

Mệnh đề 2.4. Với mọi vector $v = (v_1, \dots, v_n)^T \in \mathbb{R}^n$ ta có:

$$v^T L v = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (v_i - v_j)^2.$$

Chứng minh. Ta có biến đổi sau

$$\begin{aligned} v^T L v &= v^T D v - v^T W v = \sum_{i=1}^n v_i^2 d_{i,i} - \sum_{i,j=1}^n v_i v_j w_{i,j} \\ &= \frac{1}{2} \left(\sum_{i=1}^n v_i^2 d_{i,i} - \sum_{i,j=1}^n 2v_i v_j w_{i,j} + \sum_{j=1}^n v_j^2 d_{j,j} \right) \\ &= \frac{1}{2} \left(\sum_{i=1}^n v_i^2 \sum_{j=1}^n w_{i,j} - \sum_{i,j=1}^n 2v_i v_j w_{i,j} + \sum_{j=1}^n v_j^2 \sum_{i=1}^n w_{j,i} \right) \\ &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (v_i - v_j)^2. \end{aligned}$$

\square

Như vậy với mọi $v \in \mathbb{R}^n$ thì $v^T L v \geq 0$. Kết hợp với tính chất đối xứng của L ta có kết quả sau.

Hệ quả 2.5. L có n giá trị riêng là số thực không âm $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.

Mệnh đề tiếp theo cho phép chúng ta liên hệ biểu thức (1) với vết của một ma trận.

Mệnh đề 2.6.

$$\frac{W(A_i, \bar{A}_i)}{|A_i|} = h_i^T L h_i = [H^T L H]_{i,i}.$$

Từ đó suy ra $\mathcal{P} = \text{trace}(H^T L H)$.

Chứng minh. Theo mệnh đề 2.4 ta có

$$\begin{aligned} h_i^T L h_i &= \frac{1}{2} \sum_{j,k=1}^n w_{j,k} (h_{j,i} - h_{k,i})^2 = \frac{1}{2} \sum_{\substack{j \in A_i \\ k \in \bar{A}_i}} w_{j,k} \left(\frac{1}{\sqrt{|A_i|}} \right)^2 + \frac{1}{2} \sum_{\substack{j \in A_i \\ k \in A_i}} w_{j,k} \left(-\frac{1}{\sqrt{|A_i|}} \right)^2 \\ &= \frac{W(A_i, \bar{A}_i)}{|A_i|}. \end{aligned}$$

Hơn nữa ta cũng có

$$h_i^T L h_i = \sum_{j,k=1}^n h_{j,i} h_{k,i} [L]_{j,k} = \sum_{j,k=1}^n [H^T]_{i,j} [L]_{j,k} [H]_{k,i} = [H^T L H]_{i,i}.$$

Từ hai biến đổi trên ta có đpcm. □

Như vậy bài toán tối ưu có thể được viết lại là

$$\min_{A_1, \dots, A_k} \text{trace}(H^T L H), H \text{ được định nghĩa như ở (2)}.$$

Bài toán tối ưu mới thu được tương đương với bài toán tối ưu đồ thị, và vẫn là một bài toán khó. Tuy nhiên nếu bỏ đi điều kiện 2 mà chỉ đòi hỏi H thỏa tính chất ở 2.3 thì ta thu được bài toán *tối ưu ma trận*

$$\min_{H \in \mathbb{R}^{n \times k}} \text{trace}(H^T L H), H^T H = I_k. \quad (3)$$

Định lý sau đây cho chúng ta biết được nghiệm của bài toán (3).

Định lý 2.7 ([3, Hệ quả 4.3.39]). Cho $L \in \mathbb{R}^{n \times n}$ đối xứng có n giá trị riêng $0 \leq \lambda_1 \leq \dots \leq \lambda_n$ ứng với các vector riêng u_1, \dots, u_n và $1 \leq k \leq n$. Khi đó

$$\min_{H^T H = I_k} \text{trace}(H^T L H) = \lambda_1 + \dots + \lambda_k.$$

Dấu bằng đạt tại $H_1 = [u_1 \dots u_k]$.

Ma trận H_1 chính là nghiệm của bài toán tối ưu ma trận, và cũng là nghiệm bài toán đồ thị đã giảm điều kiện. Hay nói cách khác nó là phiên bản giảm điều kiện của ma trận H cần tìm.

2.3 Thuật toán Unnormalized Spectral Clustering

Ta đã đưa bài toán tối ưu đồ thị về bài toán tối ưu ma trận bằng cách đơn giản bớt điều kiện. Như vậy ta có thể tiếp cận bằng cách sử dụng kết quả của bài toán tối ưu ma trận là ma trận H_1 và “xấp xỉ ngược lại” để thu được kết quả của bài toán ban đầu là ma trận H . Đây cũng là ý tưởng cốt lõi của thuật toán Spectral Clustering.

(Unnormalized) Spectral Clustering

Dữ liệu đầu vào là similarity matrix $S \in \mathbb{R}^{n \times n}$, số cụm $k \in \mathbb{N}$.

1. Tính Laplacian matrix L .
2. Tính k vector riêng u_1, \dots, u_k ứng với k giá trị riêng nhỏ nhất của L . Gọi $U \in \mathbb{R}^{n \times k}$ là ma trận nhận u_i làm cột thứ i .
3. Dùng thuật toán k -means để phân các y_i thành các cụm C_1, \dots, C_k với y_i là hàng thứ i của U .

Dữ liệu đầu ra là A_1, \dots, A_k với $A_i = \{v_j | y_j \in C_i\}$.

Ma trận U mà thuật toán tính ở bước 2 chính là ma trận H_1 xấp xỉ H mà ta cần tìm. Từ định nghĩa của ma trận H , ta có thể thấy rằng 2 đỉnh v_i, v_j thuộc cùng một cụm nếu và chỉ nếu dòng thứ i và j của H bằng nhau, hay khoảng cách Euclidean của chúng là 0. Như vậy việc sử dụng k -means bước thứ 3 chính là bước “xấp xỉ ngược lại”, nhằm chia các dòng của U theo khoảng cách Euclidean, tức là các điểm thuộc cùng một cụm thì khoảng cách Euclidean nhỏ. Hay nói cách khác, ta phải biến đổi ngược lại ma trận thực U thành dạng phân hoạch rời rạc của H cần tìm.

2.4 Chọn dữ liệu đầu vào

Từ thuật toán ta cũng thấy rằng chỉ hai yếu tố ảnh hưởng tới kết quả của thuật toán là similarity matrix S và số cụm k . Và đây cũng là hai yếu tố mà người cài đặt thuật toán phải quyết định trước khi tiến hành sử dụng thuật toán.

Nên chọn loại similarity graph nào ?

Câu trả lời cho câu hỏi này tùy thuộc vào mong muốn của người cài đặt thuật toán, nếu người cài đặt muốn có được một số tính chất đặc điểm nhất định của cluster thì họ sẽ lựa chọn loại similarity graph phù hợp với mong muốn của mình. Nhưng một cách tổng quát thì nếu bạn chưa biết nên sử dụng loại similarity graph nào thì đề xuất tốt nhất đó là sử dụng k -nearest neighbor graph làm lựa chọn đầu tiên vì những đặc điểm ưu việt mà nó mang lại, k -nearest neighbor graph rất dễ để làm việc với và ma trận kề tạo ra từ nó là một ma trận thưa sẽ giúp ích cho việc tìm các eigenvalues sau này. Theo các kinh nghiệm thực tế đúc kết được, việc lựa chọn hyperparameter cụ thể là số k cho k -nearest neighbor graph là đơn giản nhất trong số tất cả các similarity graph còn lại.

Chọn số cụm k

Chọn số cụm là một vấn đề chung của hầu hết các thuật toán phân cụm. Đã có nhiều phương pháp được nghiên cứu như phương pháp gap statistic [7], phương pháp average silhouette [4]. Và tất nhiên tất cả các phương pháp này đều có thể áp dụng được cho Spectral Clustering. Tuy nhiên có một phương pháp được phát triển dành cho Spectral Clustering là eigengap heuristic. Mục tiêu của phương pháp này là chọn k sao cho $\lambda_1, \dots, \lambda_k$ nhỏ còn λ_{k+1} lớn, hay hiệu $\lambda_{k+1} - \lambda_k$ lớn hơn hẳn các hiệu trước đó. Động lực chính của việc chọn k như vậy là dựa vào kết quả quan trọng sau của spectral graph theory.

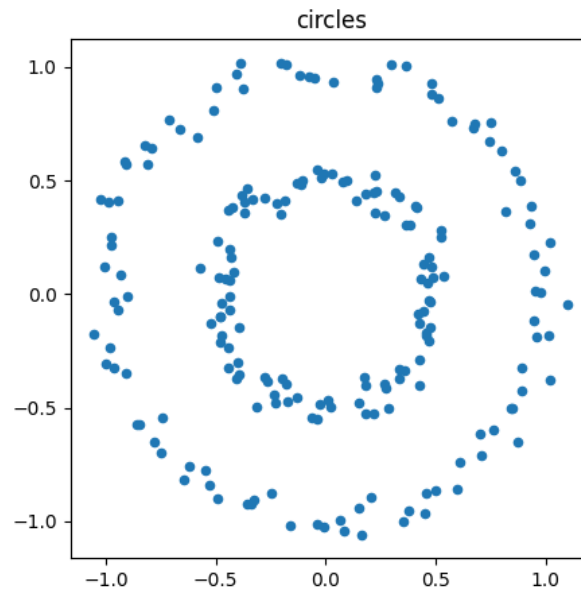
Định lý 2.8. Cho $G = (V, E)$ là đồ thị vô hướng và các cạnh mang trọng số không âm. Khi đó số bội k của giá trị riêng 0 của Laplacian matrix L cũng chính là số thành phần liên thông của G .

Như vậy trong trường hợp tốt nhất là G có k thành phần liên thông thì ta muốn k thành phần ấy cũng là các cụm của mình. Và lúc đó thì $\lambda_{k+1} - \lambda_k = \lambda_{k+1} > 0$ nhưng các hiệu $\lambda_{i+1} - \lambda_i = 0$

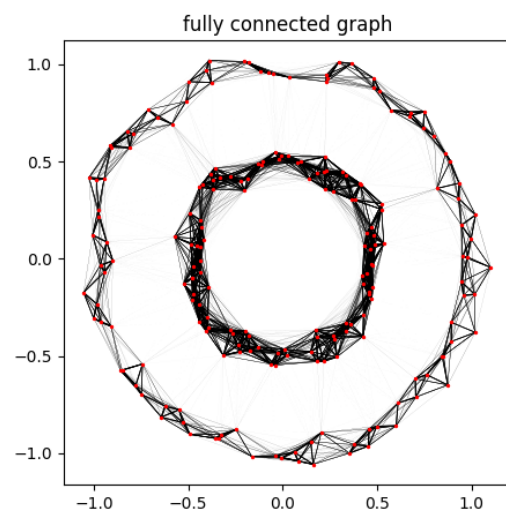
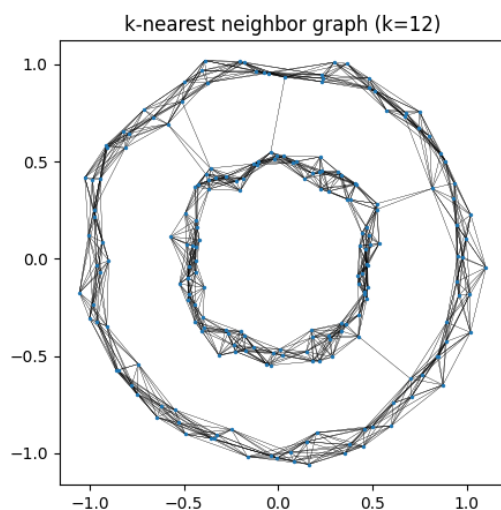
với $i < k$. Vì thế nếu chọn theo eigengap heuristic thì ta sẽ được kết quả tối ưu. Chúng tôi sẽ áp dụng minh họa thuật toán này cùng với spectral clustering vào phần tiếp theo.

3 Áp dụng mô hình

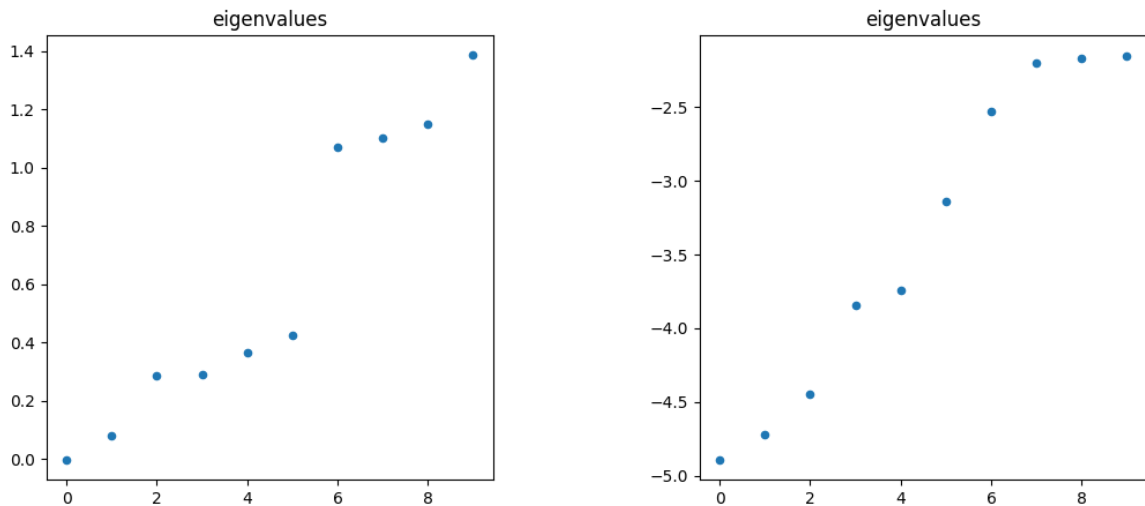
Áp dụng Spectral Clustering vào dataset circles gồm 2 hình tròn đồng tâm được lấy từ thư viện scikit-learn.



Ta xây dựng 2 loại similarity graph là k-nearest neighbor graph với $k = 12$ và fully connected graph sử dụng Gaussian similarity function với $\sigma = 0.125$ thì được như hình:

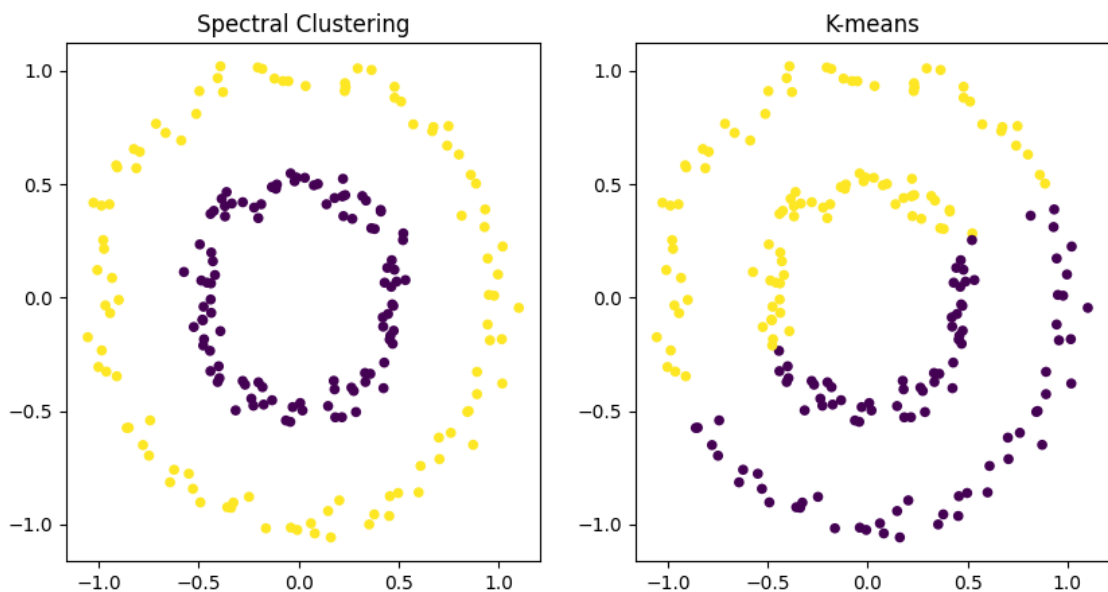


Từ 2 loại graph đã xây dựng ta tính được Laplacian matrix và tính các giá trị riêng (eigenvalues) của Laplacian matrix tương ứng:



Dựa vào các giá trị riêng của k-nearest neighbor graph, ta thấy rằng ta có thể phân dataset thành 2,6 hoặc 9 cluster và ta có thể thử các giá trị đó. Còn ở fully connected graph, eigengap giữa λ_2 và λ_3 không lớn hơn các eigengap khác quá nhiều do việc sử dụng fully connected graph khiến lát cắt giữa 2 hình tròn có giá trị tương đối lớn và eigengap heuristic không hiệu quả.

Sau khi thực hiện clustering trên fully connected graph và k-nearest neighbor graph với số lượng cluster bằng 2 ta có cùng kết quả như hình bên trái. Ta thấy rằng spectral clustering có khả năng phân cụm những cụm có hình dạng phức tạp thay vì chỉ chia được các cụm hình cầu như k -means.



Dataset MNIST là tập hợp các hình ảnh chữ số viết tay với mỗi ảnh có kích thước 28×28 pixel. Mỗi pixel có một số nguyên thuộc $[0, 255]$ tượng trưng cho độ đậm nhạt của pixel đó (số càng lớn thì màu càng đậm). Ta cho rằng tập dữ liệu gồm x_1, x_2, \dots, x_n với x_i là một ma trận

kích thước 28×28 .

Từ đó ta có similarity function:

$$s(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|_F^2}{2\sigma^2}\right),$$

với $\|x_i - x_j\|_F$ là Frobenius norm của ma trận $x_i - x_j$.

Dựa vào công thức trên để tính similarity graph và Laplacian matrix với $\sigma = 100$. Ta thực hiện clustering và được kết quả như sau (các chữ số cùng hàng thuộc cùng cluster):

6	4	9	4	4	4	4	4	9	9
4	8	9	4	4	4	9	8	8	9
1	1	4	1	1	1	1	1	1	1
2	6	1	1	1	5	1	1	8	4
0	0	0	0	0	0	0	0	0	0
6	6	6	6	6	6	6	6	6	6
2	2	2	3	3	2	2	2	3	3
6	3	0	2	6	0	5	3	0	6
7	7	7	7	7	7	2	7	7	2
3	3	8	5	5	3	3	8	3	8

Ta thấy rằng do cách tính similarity function dựa vào sự khác nhau giữa các pixel ở vị trí tương đương nhau khiến kết quả clustering chưa được chính xác (chữ số 1 được viết thẳng đứng thuộc cluster khác so với chữ số 1 viết nghiêng). Từ đó ta thấy được vai trò quan trọng của similarity function trong Spectral clustering.

4 Phiên bản cải tiến

Trong phần này ta giả sử loại fully connected graph được dùng cho similarity graph.

Nhớ lại rằng trong mục 2.2.1, một giải pháp được đưa ra để tránh việc thuật toán cắt các điểm riêng lẻ (và dẫn tới bài toán tối ưu đồ thị) là chia trọng số $W(A_i, \bar{A}_i)$ cho số lượng điểm trong một cụm, tức $|A_i|$, để yêu cầu mỗi cụm có nhiều điểm. Tuy nhiên cũng nhớ lại rằng, một trong những mục đích của việc phân hoạch là các điểm trong cùng một cụm phải có trọng số lớn, nhưng $|A_i|$ chỉ cho biết số điểm chứ không cho biết gì về thông tin này. Vì thế Shi và Malik [6] đề xuất một cách cải tiến thuật toán bằng cách sử dụng $\text{vol}(A_i)$ thay cho $|A_i|$ được định nghĩa ở mục 1.2. Nhắc lại, với mỗi phân hoạch $(A_i)_{i=1}^k$ của V ta định nghĩa

$$\text{vol}(A_i) = \sum_{j \in A_i} d_j.$$

Như vậy *bài toán đồ thị được cải tiến* là: Cho đồ thị $G = (V, E)$, ta cần tìm phân hoạch $(A_i)_{i=1}^k$ của V để tối thiểu đại lượng

$$\mathcal{P}_{\text{new}} = \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{\text{vol}(A_i)}. \quad (4)$$

Với cách làm hoàn toàn tương tự như trong mục 2.2.2, ta cũng có thể đưa bài toán tối ưu đồ thị được cải tiến về bài toán tối ưu ma trận bằng cách đơn giản bớt điều kiện. Trước hết ta định nghĩa 2 loại normalized Laplacian matrix.

Định nghĩa 4.1. Cho $G = (V, E)$ có D, W lần lượt là ma trận bậc và ma trận kề của G . Khi đó symmetric Laplacian matrix và random walk Laplacian matrix được định nghĩa là

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}, \quad (5)$$

$$L_{rw} = D^{-1} L \quad (6)$$

Hai normalized Laplacian matrix này có quan hệ chặt chẽ với nhau thông qua giá trị riêng và vector riêng của chúng.

Mệnh đề 4.2. Hai normalized Laplacian matrix thỏa mãn các tính chất sau:

1. λ là một giá trị riêng của L_{rw} ứng với vector riêng u khi và chỉ khi λ là giá trị riêng của L_{sym} ứng với vector riêng $D^{\frac{1}{2}}u$.
2. λ là một giá trị riêng của L_{rw} ứng với vector riêng u khi và chỉ khi λ và u là nghiệm của phương trình giá trị riêng tổng quát $Lu = \lambda Du$.

Chứng minh. Ý (1):

$$\begin{aligned} L_{rw}u = \lambda u &\Leftrightarrow (D^{-1}L)u = \lambda u \Leftrightarrow D^{-\frac{1}{2}}Lu = \lambda D^{\frac{1}{2}}u \Leftrightarrow (D^{-\frac{1}{2}}LD^{-\frac{1}{2}})(D^{\frac{1}{2}}u) = \lambda(D^{\frac{1}{2}}u) \\ &\Leftrightarrow L_{sym}(D^{\frac{1}{2}}u) = \lambda(D^{\frac{1}{2}}u). \end{aligned}$$

Ý (2):

$$L_{rw}u = \lambda u \Leftrightarrow D^{-1}Lu = \lambda u \Leftrightarrow Lu = \lambda Du.$$

□

Bây giờ ta bắt đầu định nghĩa lại ma trận H và các tính chất của nó.

Định nghĩa 4.3. Với mỗi phân hoạch $(A_i)_{i=1}^k$ của V , ta định nghĩa k vector $h_i = (h_{1,i}, \dots, h_{n,i})^T$ với

$$h_{j,i} = \begin{cases} \frac{1}{\sqrt{\text{vol}(A_i)}}, & v_j \in A_i \\ 0, & \text{TH khác} \end{cases}, \quad (7)$$

và H là ma trận nhận h_i làm cột thứ i .

Mệnh đề 4.4. Ta có đẳng thức $H^T D H = I_k$.

Mệnh đề 4.5.

$$\frac{W(A_i, \overline{A_i})}{\text{vol}(A_i)} = h_i^T L h_i = [H^T L H]_{i,i}.$$

Từ đó suy ra $\mathcal{P}_{new} = \text{trace}(H^T L H)$.

Như vậy bài toán tối ưu có thể được viết lại là

$$\min_{A_1, \dots, A_k} \text{trace}(H^T L H), H \text{ được định nghĩa như ở (7)}.$$

Tương tự ta bỏ đi điều kiện (7) mà chỉ đòi hỏi H thỏa tính chất ở mệnh đề 4.4 và thế $K = D^{\frac{1}{2}}H$ thì ta thu được bài toán *tối ưu ma trận*

$$\min_{K \in \mathbb{R}^{n \times k}} \text{trace}(K^T L_{sym} K), K^T K = I_k. \quad (8)$$

Do L_{sym} đối xứng nên theo định lý 2.7 nghiệm của bài toán này là $K_1 = [u_1 \dots u_k]$ với các u_i là các vector riêng ứng với k giá trị riêng nhỏ nhất của L_{sym} . Theo mệnh đề 4.2 thì ma trận $H_1 = D^{-\frac{1}{2}}K_1$, phiên bản giảm điều kiện của H cần tìm, gồm k vector riêng ứng với k giá trị

riêng nhỏ nhất của phương trình $Lu = \lambda Du$. Như vậy từ đây ta có được thuật toán Normalized Spectral Clustering của Shi và Malik [6].

Normalized Spectral Clustering theo Shi và Malik

Dữ liệu đầu vào là similarity matrix $S \in \mathbb{R}^{n \times n}$, số cụm $k \in \mathbb{N}$.

1. Tính Laplacian matrix L .
2. Tính k vector riêng u_1, \dots, u_k ứng với k giá trị riêng nhỏ nhất của phương trình $Lu = \lambda Du$. Gọi $U \in \mathbb{R}^{n \times k}$ là ma trận nhận u_i làm cột thứ i .
3. Dùng thuật toán k -means để phân các y_i thành các cụm C_1, \dots, C_k với y_i là hàng thứ i của U .

Dữ liệu đầu ra là A_1, \dots, A_k với $A_i = \{v_j | y_j \in C_i\}$.

Tương tự như Unnormalized Spectral Clustering, ma trận U là ma trận H_1 và bước k -means để chuyển H_1 về dạng rời rạc phân hoạch của ma trận H cần tìm.

Thuật toán cải tiến có thực sự tốt hơn ?

Ta nhắc lại 2 mục đích của việc phân cụm trên similarity graph là: trọng số giữa các cụm khác nhau thấp và trọng số trong cùng một cụm cao. Cả hai thuật toán Unnormalized và Normalized đều thực hiện mục đích đầu tiên do sử dụng hàm $W(A, \bar{A})$ trong hàm cần tối thiểu. Tuy nhiên, chỉ có Normalized spectral clustering thực hiện mục tiêu thứ hai, cụ thể

$$W(A, A) = W(A, V) - W(A, \bar{A}) = \text{vol}(A) - W(A, \bar{A}).$$

Nếu ta tối thiểu \mathcal{P}_{new} cũng đồng nghĩa là ta muốn $W(A, \bar{A})$ nhỏ còn $\text{vol}(A)$ lớn, tức là $W(A, A)$ hay trọng số trong cùng một cụm lớn. Còn đại lượng $|A|$ không đánh giá được mục tiêu này vì một cụm có thể có nhiều điểm nhưng chưa chắc các cạnh giữa chúng đã có trọng số lớn.

5 Kết luận

Như vậy, chúng ta có thể thấy Spectral Clustering là một cách tiếp cận phổ biến và hữu dụng cho bài toán phân nhóm dữ liệu vì có những ưu điểm như:

- Phân cụm được dataset dựa trên độ kết nối của các điểm trong cùng một cluster thay vì các cluster có dạng một khối cầu như k -means clustering.
- Không cần chạy nhiều lần bước khởi tạo thuật toán như k -means clustering, điều này khiến cho kết quả không bị ảnh hưởng bởi bước khởi tạo và giúp cho tốc độ thuật toán nhanh hơn.

Tuy vậy, khi sử dụng thuật toán, ta cần phải chú ý một vài vấn đề tiềm năng như:

- Độ hiệu quả của thuật toán phụ thuộc nhiều vào cách chúng ta chọn hyperparameter và similarity graph.
- Các Laplacian matrix càng dense thì việc tính các giá trị riêng và vector riêng sẽ lâu hơn, ảnh hưởng đến tốc độ của thuật toán.

Tóm lại, thuật toán Spectral Clustering là một thuật toán mạnh, có thể dùng được cho nhiều trường hợp khác nhau, tuy nhiên nó cũng cần được áp dụng một cách cẩn thận.

Tài liệu

- [1] David Benson-Putnins, Magaret Bonfardin, Meagan E. Magnoni, and Daniel Martin. Spectral clustering and visualization: A novel clustering of Fischer's Iris dataset.
- [2] L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.
- [3] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2013.
- [4] Leonard Kaufmann and Peter J. Rousseeuw. *Finding groups in data: An introduction to cluster analysis*. Wiley, 2005.
- [5] Bojan Mohar. Some applications of Laplace eigenvalues of graphs. pages 225–275, 1997.
- [6] Jianbo Shi and Jitendra Malik. Normalized cut and image segmentation. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 22(8), 2000.
- [7] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society Series B*, 63:411–423, 02 2001.
- [8] Ulrike von Luxburg. A tutorial on spectral clustering. 2007.
- [9] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its Application to image segmentation. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 15(11), 1993.