TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THỰC NGHIỆM
HỌC PHẦN: HỌC MÁY

ĐỀ TÀI: TÌM HIỂU VỀ THUẬT TOÁN PHÂN CỤM MỜ THEO KHẢ NĂNG VÀ ỨNG DỤNG VÀO THỰC TẾ

Giảng viên hướng dẫn: ThS. Phạm Việt Anh

Lớp : 20242IT6047001

Nhóm thực hiện : Nhóm 2

Nguyễn Thành Công - 2022606702

Nguyễn Mạnh Cường - 2022605592

Phạm Thị Hồng Duyên - 2022606581

Tống Đăng Quang - 2022603783

Bùi Hoàng Linh - 2022602573

Nguyễn Công Thành - 2022600386

Hà Nội – Năm 2025

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến giảng viên giảng dạy học phần Học máy – ThS. Phạm Việt Anh đã dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học của thầy, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc cũng như rèn luyện những kỹ năng cần thiết. Thầy cũng tạo động lực cho chúng em hoàn thành tốt nhiệm vụ của mình. Bên cạnh đó, chúng em cũng xin cảm ơn các bạn học viên trường Công nghệ thông tin và Truyền thông đã đóng góp ý kiến giúp chúng em thực hiện đề tài đạt hiệu quả hơn.

Bài tiểu luận này đã giúp chúng em rèn luyện kỹ năng tư duy nghiên cứu và phân tích thuật toán học máy và trình bày thông tin một cách có logic và rõ ràng. Chúng em hi vọng rằng những kiến thức và kinh nghiệm thu thập từ đề tài này sẽ tiếp tục hỗ trợ chúng em trong tương lai, không chỉ trong học tập mà còn trong sự nghiệp và cuộc sống.

Nhóm chúng em xin trân trọng cảm ơn!

Nhóm sinh viên thực hiện

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	FCM	Fuzzy C-Means – Thuật toán phân cụm mờ.
2	PCM	Possibilistic C-Means – Thuật toán phân cụm khả năng.
3	PFCM	Possibilistic Fuzzy C-Means – Thuật toán phân cụm mờ kết hợp khả năng.
4	K-Means	Một thuật toán phân cụm cứng truyền thống.
5	DBI	Davies-Bouldin Index – Chỉ số đánh giá độ phân tách cụm.
6	DI	Dunn Index – Chỉ số đánh giá độ chặt và tách biệt của cụm.
7	PC	Partition Coefficient – Hệ số phân hoạch, đánh giá mức độ "mờ" của cụm.

LỜI MỞ ĐẦU

Trong thời đại bùng nổ dữ liệu hiện nay, việc khai thác và xử lý dữ liệu một cách hiệu quả đóng vai trò quan trọng trong nhiều lĩnh vực như y tế, tài chính, thương mại điện tử, giao thông và trí tuệ nhân tạo. Một trong những kỹ thuật quan trọng trong khai phá dữ liệu là phân cụm, giúp nhóm các đối tượng có đặc điểm tương đồng lại với nhau nhằm phục vụ cho việc phân tích và ra quyết định. Trong đó, phân cụm mờ (Fuzzy Clustering) là một hướng tiếp cận nổi bật, cho phép mỗi đối tượng có thể thuộc về nhiều cụm với các mức độ khác nhau, thay vì chỉ thuộc một cụm duy nhất như trong phân cụm truyền thống.

Phân cụm mờ đặc biệt hữu ích trong các bài toán mà ranh giới giữa các cụm không rõ ràng, chẳng hạn như phân tích hành vi khách hàng, nhận dạng ảnh, chẩn đoán y khoa và xử lý ngôn ngữ tự nhiên. Trong số các thuật toán phân cụm mờ, Fuzzy C-Means (FCM) là một phương pháp phổ biến và hiệu quả, được sử dụng rộng rãi trong thực tế. Tuy nhiên, FCM cũng tồn tại một số hạn chế như nhạy cảm với điểm nhiễu, dễ rơi vào cực tiểu cục bộ và phụ thuộc vào số cụm được chọn trước. Do đó, nhiều nghiên cứu đã đề xuất các cải tiến nhằm khắc phục những nhược điểm này, bao gồm việc kết hợp với các thuật toán tối ưu hóa hoặc mô hình học sâu để nâng cao hiệu suất.

Báo cáo này tập trung vào việc khảo sát, phân tích và đánh giá các thuật toán phân cụm mờ, từ nền tảng lý thuyết đến ứng dụng thực tế. Nội dung nghiên cứu sẽ trình bày về lý thuyết tập mờ, các thuật toán phân cụm mờ phổ biến, cũng như những cải tiến mới nhằm nâng cao độ chính xác và hiệu quả phân cụm. Cấu trúc của báo cáo này được chia thành 3 chương chính:

Chương 1 giới thiệu tổng quan đề tài, mục tiêu và phạm vi nghiên cứu, cũng như phương pháp tiếp cận mà nhóm sử dụng. Nội dung chương này nhằm định hướng rõ ràng cho toàn bộ báo cáo và khẳng định tính cấp thiết, thực tiễn của việc nghiên cứu phân cụm mờ.

Chương 2 trình bày cơ sở lý thuyết về các thuật toán phân cụm mờ, bao gồm Fuzzy C-Means (FCM), Possibilistic C-Means (PCM) và thuật toán kết hợp PFCM. Tập trung phân tích nguyên lý hoạt động, hàm mục tiêu, công thức

cập nhật, cùng với so sánh ưu – nhược điểm của từng phương pháp, tạo nền tảng vững chắc cho phần thực nghiệm.

Chương 3 là phần trọng tâm của báo cáo, trong đó nhóm tiến hành thực nghiệm trên các loại dữ liệu khác nhau, đánh giá và phân tích kết quả thu được bằng các chỉ số khách quan. Đồng thời cũng mở rộng thực nghiệm để thấy sự linh hoạt khi thay đổi các tham số đầu vào của các thuật toán.

Thông qua việc thực hiện đề tài này, chúng em đã tích lũy được nhiều kiến thức bổ ích về lĩnh vực học máy, đặc biệt là về phân cụm và phân cụm mờ. Hy vọng rằng nghiên cứu này sẽ mang lại những đóng góp hữu ích cho cộng đồng nghiên cứu và ứng dụng phân cụm mờ trong thực tế. Nhóm thực hiện cũng xin gửi lời cảm ơn đến thầy cô, bạn bè và các chuyên gia trong lĩnh vực trí tuệ nhân tạo và khai phá dữ liệu, những người đã hỗ trợ và đóng góp ý kiến quý báu trong quá trình thực hiện nghiên cứu này.

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Trong bối cảnh dữ liệu ngày càng phức tạp và đa dạng, các phương pháp phân cụm truyền thống như K-means thường gặp khó khăn trong việc xử lý các dữ liệu không có ranh giới rõ ràng và bị ảnh hưởng bởi dữ liệu nhiễu. Thuật toán phân cụm mờ khả năng - Possibilistic Fuzzy C-Means (PFCM) ra đời nhằm khắc phục hạn chế này, cho phép một điểm dữ liệu có thể thuộc về nhiều cụm với mức độ khác nhau, đồng thời giảm sự ảnh hưởng của nhiễu đến chất lượng kết quả phân cụm. Điều này đặc biệt hữu ích trong các bài toán thực tế như phân tích dữ liệu y tế, xử lý ảnh, khai thác dữ liệu và trí tuệ nhân tạo…

1.2. Mục tiêu và nhiệm vụ

- **Mục tiêu**

Mục tiêu của đề tài bao gồm ba nội dung chính. Thứ nhất, tiến hành nghiên cứu các thuật toán phân cụm mờ, trong đó đặc biệt tập trung vào thuật toán phân cụm mờ và các biến thể của nó. Thứ hai, đánh giá những ưu điểm và hạn chế của các thuật toán phân cụm mờ khi so sánh với các phương pháp phân cụm truyền thống. Cuối cùng, áp dụng thuật toán phân cụm mờ vào một bài toán thực tế nhằm kiểm chứng hiệu quả và khả năng ứng dụng của phương pháp này trong thực tiễn…

- **Nhiệm vụ**

Nhiệm vụ của đề tài bao gồm ba nội dung chính. Thứ nhất, tìm hiểu lý thuyết về tập mờ và cách ứng dụng trong phân cụm dữ liệu nhằm xử lý các tình huống không chắc chắn. Thứ hai, nghiên cứu ba thuật toán gồm Fuzzy C-Means (FCM), phân cụm mở khả năng (PCM) và thuật toán PFCM là sự kết hợp giữa FCM và PCM, qua đó nắm rõ cơ chế và hướng cải tiến. Cuối cùng, đánh giá hiệu suất của các thuật toán này trên các bộ dữ liệu thực tế và ảnh viễn thám để kiểm chứng khả năng áp dụng vào các bài toán cụ thể.

1.3. Phương pháp nghiên cứu

- **Nghiên cứu tài liệu:** Thu thập và tổng hợp kiến thức từ các bài báo khoa học, giáo trình và tài liệu chuyên ngành.

- **Thực nghiệm trên dữ liệu thực tế:** Cài đặt và kiểm tra thuật toán trên các bộ dữ liệu khác nhau để đánh giá hiệu quả.
- **Phân tích và so sánh:** So sánh thuật toán phân cụm mờ với các phương pháp truyền thống để xác định ưu nhược điểm.

1.4. Đối tượng và phạm vi nghiên cứu

- **Đối tượng nghiên cứu:**

- **Dữ liệu:** Bộ dữ liệu các loài hoa Iris, hạt đậu khô Dry Bean và ảnh viễn thám đa phổ.
- **Thuật toán:** Các thuật toán phân cụm và phân cụm mờ như K-means, FCM, PCM, PFCM.

- **Phạm vi nghiên cứu:**

- Nghiên cứu tập trung vào thuật toán phân cụm mờ, đặc biệt là ba thuật toán FCM, PCM và PFCM, bao gồm nguyên lý hoạt động, ma trận thành viên, ma trận tâm cụm và các cải tiến nhằm nâng cao hiệu quả phân cụm.
- Nghiên cứu cũng xem xét ứng dụng của phân cụm mờ trong ảnh viễn thám, đồng thời so sánh hiệu suất của các thuật toán với các biến thể khác nhau.

1.5. Ý nghĩa khoa học và ý nghĩa thực tiễn

- **Ý nghĩa khoa học**

Nghiên cứu góp phần làm rõ lý thuyết về phân cụm, đặc biệt là ba thuật toán FCM, PCM và PFCM và các cải tiến của nó. Kết quả nghiên cứu giúp mở rộng hiểu biết về cách áp dụng tập mờ trong phân tích dữ liệu, tạo nền tảng cho các nghiên cứu nâng cao như phân cụm mờ kết hợp học sâu hoặc tối ưu hóa.

- **Ý nghĩa thực tiễn**

Trong nhiều bài toán thực tế, chẳng hạn như phân loại ảnh, chẩn đoán bệnh hay phân tích hành vi người dùng, dữ liệu không luôn phân chia rõ ràng thành các nhóm rời rạc. Phân cụm mờ cho phép một đối tượng thuộc về nhiều cụm với các mức độ khác nhau (membership), phản ánh đúng bản chất mơ hồ của dữ liệu. Đặc biệt, phân cụm mờ có ứng dụng quan trọng trong phân tích ảnh viễn thám đa phổ, giúp phân loại lớp phủ bề mặt Trái Đất với độ chính xác

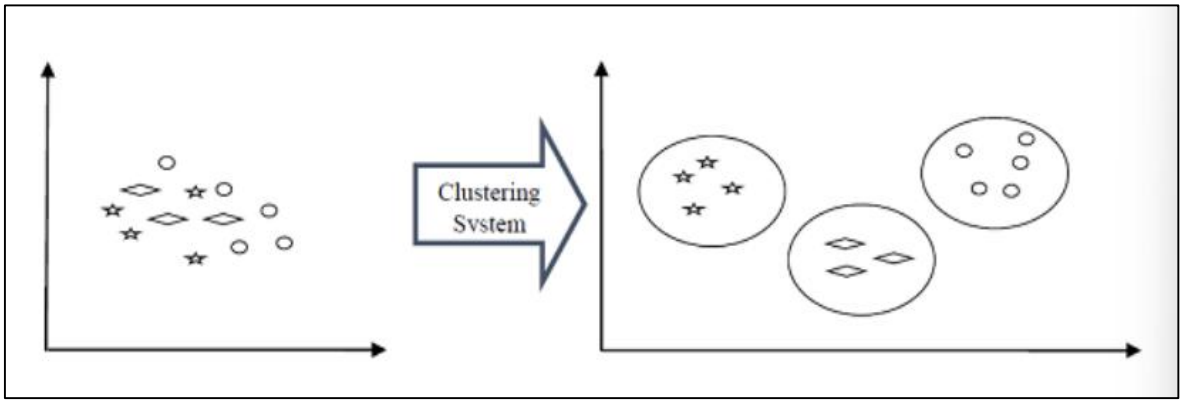
cao hơn so với phân cụm cứng. Việc áp dụng thuật toán này hỗ trợ giám sát tài nguyên thiên nhiên, quản lý đất đai, dự báo môi trường và nhiều lĩnh vực quan trọng khác. Ngoài ra, việc cải tiến và tối ưu thuật toán giúp tăng tốc độ xử lý, giảm nhiễu và nâng cao chất lượng phân loại trong ảnh viễn thám.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT PHÂN CỤM MỜ

2.1. Tổng quan về phân cụm [1]

2.1.1. Khái niệm

Phân tích cụm hay phân cụm là một kỹ thuật phân tích dữ liệu dùng để nhóm một tập hợp các đối tượng vào cùng một nhóm/cụm (group/cluster) mà ở đó các đối tượng có các đặc trưng tương tự nhau ở một phương diện nào đó được thể hiện qua các thuộc tính của các đối tượng. Đây là một trong những nhiệm vụ cốt lõi của phân tích dữ liệu và là kỹ thuật phổ biến trong nhiều lĩnh vực như nhận dạng mẫu (pattern recognition), phân tích ảnh (image analysis), truy xuất thông tin (information retrieval), học máy (machine learning) và nhiều lĩnh vực khác. Phân cụm đề cập đến một nhóm các thuật toán phục vụ cho hoạt động phân tích cụm hơn là đề cập đến một thuật toán cụ thể nào đó. Hoạt động này có thể đạt được bởi nhiều thuật toán khác nhau, mỗi thuật toán sẽ phù hợp với các bối cảnh khác nhau của dữ liệu do đó sử dụng các kỹ thuật khác nhau và cho ra các kết quả khác nhau về mặt nào đó.



Hình 2.1. Minh họa quá trình phân cụm

Một số kỹ thuật phân cụm phổ biến có thể kể đến như:

Phân cụm theo trung tâm: Chia dữ liệu thành *k* nhóm sao cho khoảng cách trung bình từ các điểm dữ liệu đến các tâm cụm là tối ưu. Phương pháp này yêu cầu xác định trước số lượng cụm, thông qua phân tích chuyên sâu, trực giác hoặc bằng các thuật toán hỗ trợ lựa chọn số cụm phù hợp, mang lại kết quả phân cụm tốt nhất. Đây cũng sẽ là kỹ thuật chính được nhắc đến xuyên suốt trong bài báo cáo này.

Phân cụm phân cấp: Xây dựng một cấu trúc cây (*dendrogram*) dựa trên việc lần lượt gộp các phần tử lại với nhau (phân cụm tích tụ) hoặc tách chúng ra (phân cụm phân chia).

2.1.2. Các thuật toán phân cụm phổ biến

- **K-means**

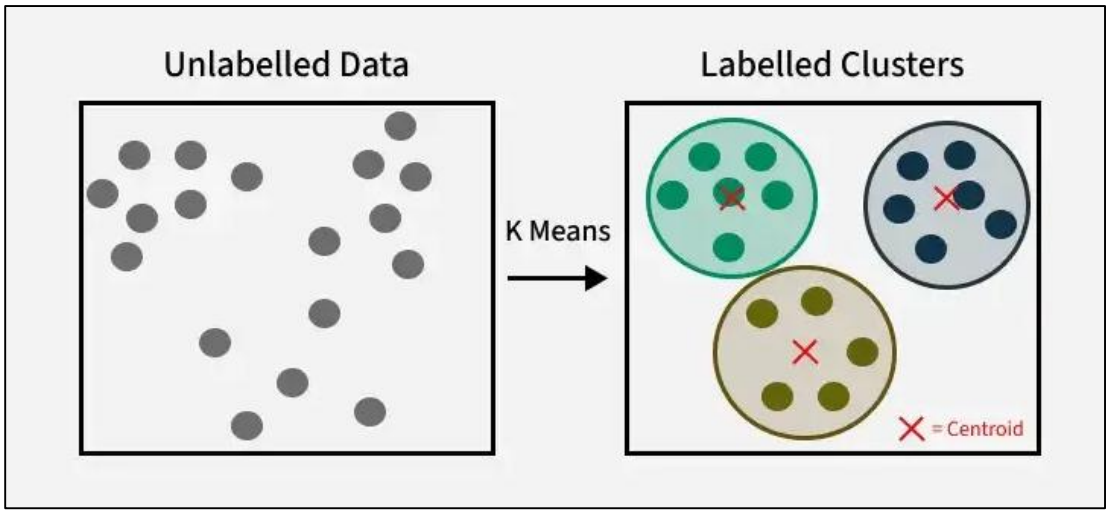
K-means là một trong những thuật toán phân cụm phổ biến nhất, được sử dụng để phân chia một tập hợp gồm N điểm dữ liệu thành C cụm sao cho tổng bình phương khoảng cách giữa mỗi điểm dữ liệu và tâm cụm gần nhất là nhỏ nhất. Ưu điểm nổi bật của K-means là thuật toán đơn giản, dễ hiểu, dễ cài đặt, tốc độ xử lý nhanh, phù hợp với các dữ liệu có phân bố tròn và có thể phân tách tuyến tính. Tuy nhiên, K-means cũng có một số nhược điểm là kết quả bị phụ thuộc vào khởi tạo ngẫu nhiên ban đầu (tâm cụm ban đầu) và việc chọn số lượng cụm C, ngoài ra nó còn bị nhạy cảm với nhiễu và ngoại lệ, khiến cho tâm cụm dễ bị kéo lệch.

Hàm mục tiêu của thuật toán có dạng như sau (trong đó u là ma trận 2 chiều chứa các mức độ thành viên – membership value của từng điểm dữ liệu vào từng cụm):

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij} \left| x_i - v_j \right|_2^2$$

Trong đó:

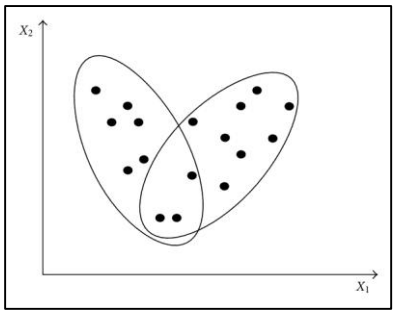
- x_i: điểm dữ liệu thứ i
- v_j: tâm cụm thứ j
- u_{ij}: giá trị nhị phân thể hiện mức độ thành viên của x_i đối với cụm j (bằng 1 nếu x_i thuộc cụm j, ngược lại bằng 0)
- $\left| x_i - v_j \right|_2^2$: bình phương khoảng cách Euclid giữa điểm dữ liệu và tâm cụm.



Hình 2.2. Minh họa phân cụm K-means

- **Fuzzy clustering - Phân cụm mờ**

Đây là một phiên bản cải tiến của thuật toán K-means truyền thống. Trong K-means, mỗi điểm dữ liệu được gán cố định vào một cụm duy nhất - cụm có tâm gần nhất với điểm đó. Tuy nhiên, điều này có thể không hợp lý trong trường hợp điểm dữ liệu nằm gần như đều giữa hai (hoặc nhiều) cụm, nhưng vẫn bị buộc phải thuộc về cụm gần hơn dù chỉ một khoảng cách nhỏ. Điều này khiến cho thuật toán trở nên cứng nhắc đặc biệt là đối với dữ liệu lớn, nơi mà các điểm dữ liệu phân bổ phức tạp, cần có sự linh hoạt trong hoạt động phân cụm. Ưu điểm nổi bật của thuật toán này là khả năng khắc phục được sự cứng nhắc của K-means. Nó cho phép mỗi điểm dữ liệu đồng thời thuộc về nhiều cụm với các mức độ thành viên khác nhau, phản ánh tính linh hoạt và khả năng thích ứng tốt hơn với đặc điểm phân bố thực tế của dữ liệu, đây chính là tính '*mờ*' được nhắc đến trong tên thuật toán. Tuy nhiên, thuật toán này vẫn có những hạn chế tương tự như K-means, điển hình là sự nhạy cảm với nhiễu và phụ thuộc vào việc khởi tạo ngẫu nhiên ban đầu. Đây sẽ một trong những là thuật toán chính được đề cập trong bài báo cáo này. Trong các chương sau của quyển báo cáo, hàm mục tiêu, cách thức thực hiện, luồng hoạt động của thuật toán sẽ được nhắc tới chi tiết hơn, bên cạnh đó là các cách thức cải thiện những nhược điểm chính của thuật toán.



Hình 2.3. Minh họa phân cụm mờ đơn giản

2.1.3. Ứng dụng

Thuật toán phân cụm có rất nhiều ứng dụng thực tiễn trong các lĩnh vực khác nhau. Dưới đây là một số ứng dụng phổ biến nhất:

Phân loại đối tượng khách hàng: Phân cụm khách hàng dựa trên các đặc điểm như hành vi mua sắm, độ tuổi, tần suất mua hàng,... giúp doanh nghiệp hiểu rõ từng nhóm khách hàng để từ đó xây dựng các chiến lược tiếp thị, chăm sóc và bán hàng phù hợp. Việc phân nhóm hiệu quả sẽ góp phần tối ưu hóa doanh thu và tăng trải nghiệm khách hàng.

Nhận dạng mẫu: Trong lĩnh vực xử lý ảnh, phân cụm được sử dụng để nhận diện các đối tượng có đặc điểm tương đồng trong ảnh, như các vùng có kết cấu hoặc màu sắc giống nhau. Kết quả phân cụm có thể đóng vai trò là đầu vào cho các thuật toán học sâu, đặc biệt là các mạng nơ-ron tích chập (CNN – Convolutional Neural Network).

Phân tích ảnh vệ tinh: Phân cụm ảnh vệ tinh chụp từ trên cao cho phép giám sát và phát hiện những thay đổi theo thời gian, ví dụ như biến động của đất nông nghiệp, diện tích rừng, hay mức nước trên sông, hồ. Từ đó, có thể đưa ra các chiến lược quản lý tài nguyên hiệu quả hơn, cũng như dự báo và ứng phó kịp thời với các hiện tượng bất thường như hạn hán hay lũ lụt.

Bên cạnh các lĩnh vực trên, phân cụm còn được ứng dụng rộng rãi trong y tế (như phân nhóm bệnh nhân, chẩn đoán hình ảnh) và xử lý ngôn ngữ tự nhiên (như phân cụm tài liệu, trích xuất chủ đề). Trong phạm vi bài báo cáo này, nhóm sẽ tập trung ứng dụng thuật toán phân cụm vào phân tích ảnh vệ tinh (còn gọi là ảnh viễn thám). Các nội dung chi tiết về quá trình triển khai và kết quả phân tích sẽ được trình bày cụ thể trong các chương tiếp theo.

2.1.4. Các chỉ số dùng để đánh giá

Trong phân cụm, việc đánh giá chất lượng kết quả là yếu tố quan trọng nhằm lựa chọn thuật toán phù hợp và xác định số lượng cụm tối ưu. Các chỉ số đánh giá thường được chia thành hai nhóm: đánh giá nội bộ (không cần nhãn) và đánh giá ngoại vi (dựa trên nhãn thật). Dưới đây là ba chỉ số nội bộ phổ biến nhất:

Davies-Bouldin Index (DB Index): Chỉ số Davies-Bouldin (sau đây sẽ được viết tắt là "DBI"), được đề xuất bởi David L. Davies và Donald W. Bouldin vào năm 1979, là một thước đo nội bộ dùng để đánh giá chất lượng của phân cụm dựa trên độ chặt (compactness) và độ tách biệt (separation) giữa các cụm. Một kết quả phân cụm tốt cần đảm bảo các điểm dữ liệu trong cùng cụm phải gần nhau, trong khi các cụm khác nhau phải cách xa nhau. Ý tưởng chính là một phân cụm tốt sẽ có các cụm riêng rẽ với các điểm dữ liệu gần nhau bên trong từng cụm, đồng thời các cụm này cũng cần cách xa nhau. DBI cho phép so sánh chất lượng giữa các kết quả phân cụm khác nhau hoặc giữa các giá trị số cụm c, từ đó lựa chọn giá trị c sao cho DBI nhỏ nhất. Giá trị DBI càng nhỏ chứng tỏ các cụm càng chặt chẽ và tách biệt tốt, do đó kết quả phân cụm càng chất lượng. Tuy nhiên, DBI có một số hạn chế như nhạy cảm với ngoại lai và giả định cụm có dạng hình cầu, khiến nó kém hiệu quả khi các cụm thực tế có hình dạng phức tạp hoặc kích thước không đồng đều.

Partition Coefficient: Partition Coefficient (sau đây sẽ được viết tắt là "PC") là một chỉ số dùng để đánh giá mức độ "sắc nét" hay "mờ" của ma trận phân hoạch trong phân cụm mờ (fuzzy clustering), đặc biệt là thuật toán Fuzzy C-Means. Nó đo mức độ rõ ràng của việc gán các điểm dữ liệu vào các cụm: nếu các giá trị membership u_{ik} gần 0 hoặc 1 thì phân cụm rõ ràng, còn nếu các giá trị gần 1/c thì biểu thị mức độ mờ cao. PC thường được chuẩn hóa về khoảng [0, 1] nhằm so sánh giữa các kết quả phân cụm có số lượng cụm khác nhau. Giá trị PC càng cao thì ma trận phân hoạch càng rõ ràng. Chỉ số PC rất phổ biến trong đánh giá phân cụm mờ vì tính đơn giản và khả năng phản ánh trực tiếp mức độ "mờ" của ma trận phân hoạch, nhưng nó không cung cấp thông tin về sự tách biệt hay độ chặt của các cụm.

Dunn Index: Chỉ số Dunn (sau đây sẽ được viết tắt là "DI") được đề xuất bởi Joseph C. Dunn năm 1974, là một thước đo chất lượng phân cụm nội bộ

không cần nhãn, với mục tiêu khuyến khích các cụm dữ liệu vừa chặt chẽ vừa tách biệt tốt. Chỉ số này được tính bằng tỷ số giữa khoảng cách nhỏ nhất giữa hai cụm bất kỳ và độ rộng lớn nhất trong tất cả các cụm.

$$\text{Dunn Index} = \frac{\min\limits_{i \neq j} \text{dist}(C_i, C_j)}{\max\limits_{k} \text{diameter}(C_k)}$$

Trong đó độ rộng của một cụm được định nghĩa là khoảng cách lớn nhất giữa hai điểm trong cùng cụm, còn khoảng cách giữa hai cụm được định nghĩa là khoảng cách ngắn nhất giữa một điểm ở cụm này và một điểm ở cụm kia. Giá trị Dunn càng lớn thì cụm càng rõ ràng – tức là vừa nhỏ gọn vừa tách biệt tốt. Ngược lại, chỉ số thấp cho thấy các cụm có thể bị chồng lấn hoặc một cụm có thể trải rộng quá mức. Tuy nhiên, chỉ số này có nhược điểm là rất nhạy cảm với cụm "xấu", chỉ cần một cụm có độ rộng lớn hoặc hai cụm gần nhau quá mức sẽ làm giảm giá trị Dunn chung. Tuy nhiên, chỉ số này rất nhạy cảm với cụm "xấu" – chỉ cần một cụm có độ rộng lớn hoặc hai cụm quá gần nhau sẽ làm giảm giá trị chung. Ngoài ra, chi phí tính toán DI cao do cần tính tất cả khoảng cách giữa các cặp điểm hoặc cụm, nên kém hiệu quả với tập dữ liệu lớn. Mặc dù vậy, DI vẫn là một chỉ số được sử dụng phổ biến nhờ khả năng diễn giải dễ hiểu và trực quan.

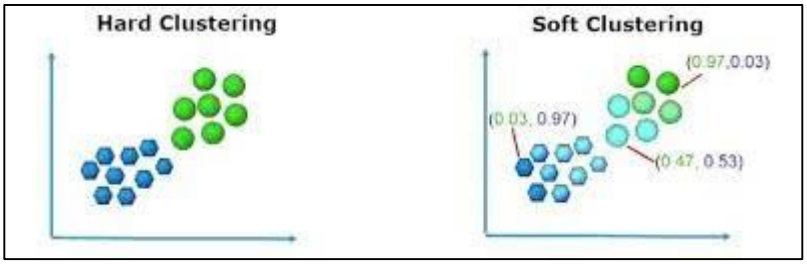
2.2. Thuật toán phân cụm mờ (Fuzzy C-Means - FCM)

2.2.1. Khái niệm

Trong các phương pháp phân cụm truyền thống, dữ liệu được phân chia dựa trên phân hoạch cứng (hard partition), tức là mỗi điểm dữ liệu chỉ được gán duy nhất vào một cụm. Một phân hoạch cứng với C cụm trên tập dữ liệu gồm N điểm là một tập các cụm rời nhau, không rỗng, sao cho hợp của tất cả các cụm bằng đúng tập dữ liệu ban đầu. Tuy nhiên, trong thực tế, ranh giới giữa các nhóm dữ liệu thường không rõ ràng, dẫn đến sự hạn chế của phân hoạch cứng - vì nó không thể phản ánh mức độ không chắc chắn hay khả năng "thuộc về một phần" của một điểm dữ liệu đối với nhiều cụm.

Để khắc phục hạn chế này, lý thuyết tập mờ (fuzzy set theory) được giới thiệu cho phép mỗi điểm dữ liệu có thể thuộc về một tập nào đó với mức độ thành viên (membership degree) nằm trong khoảng [0, 1]. Ứng dụng ý tưởng

của tập mờ vào phân cụm dẫn đến sự ra đời của phân hoạch mờ (fuzzy partition), trong đó một điểm dữ liệu có thể "thuộc một phần" vào nhiều cụm khác nhau với các mức độ khác nhau. Điều này giúp mô hình hóa tốt hơn tính không chắc chắn vốn có trong dữ liệu thực tế. Tổng các giá trị membership cho mỗi điểm vẫn được ràng buộc bằng 1, đảm bảo toàn bộ membership của một điểm được phân bố hết cho tất cả các cụm. Phân cụm mờ là một phương pháp phân cụm mờ nổi bật, thuộc nhóm học máy không giám sát. Thuật toán cho phép nhóm các điểm dữ liệu dựa trên mức độ tương đồng mà không yêu cầu mỗi điểm phải được gán cố định vào một cụm duy nhất như trong phân cụm cứng (K-means). Thay vào đó, mỗi điểm có thể thuộc vào tất cả cụm với mức độ khác nhau, thông qua đại lượng gọi là mức độ thành viên (membership value), phản ánh rõ hơn bản chất không chắc chắn của dữ liệu thực tế.



Hình 2.4. Minh họa kết quả phân cụm mờ bằng thuật toán FCM

Bảng 2.1. So sánh K-means và Fuzzy C-Means

Tiêu chí	**K-Means**	**Fuzzy C-Means**
Mỗi điểm thuộc vào	Một cụm duy nhất	Tất cả các cụm với các mức độ khác nhau
Đầu ra	Ma trận thành viên với các giá trị 0 và 1 thể hiện 2 giá trị trái ngược nhau 'thuộc' và 'không thuộc' vào một cụm. Tâm cụm	Ma trận thành viên với các giá trị nằm trong khoảng [0, 1] thể hiện mức độ thuộc vào một cụm của một đối tượng dữ liệu. Tâm cụm
Phù hợp với	Dữ liệu rõ ràng	Dữ liệu mơ hồ, chồng lấn cụm.

2.2.2. Hàm mục tiêu

Hàm mục tiêu của thuật toán Fuzzy C-Means có dạng:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} {u_{ij}}^{m} \left| x_i - v_j \right|_2^2 \qquad (1)$$

Các thành phần trong công thức:

- ***U*** là ma trận thành viên chứa các phần tử u_{ij} (membership values) có kích thước n x c (n là số lượng các đối tượng dữ liệu, c là số lượng cụm), với giá trị của phần tử u_{ij} thể hiện độ thuộc của điểm dữ liệu thứ i vào cụm thứ j. ***U*** có một ràng buộc đó là với mọi i bất kì thì $\sum_{j=1}^{c} u_{ij} = 1$. Điều này thể hiện được tính đóng của tập mờ khi mà toàn bộ điểm dữ liệu đều chắc chắn thuộc về các cụm mà không có một xác suất nào gây ra việc một điểm dữ liệu không thuộc về một cụm nào.
- ***m*** là hệ số mờ (fuzziness): thể hiện độ mờ trong thuật toán. Khi m càng lớn, mức độ thành viên giữa các cụm càng được phân phối đều nhau (không một điểm dữ liệu nào có độ thuộc vào một cụm lớn hơn hẳn so với các cụm khác), ngược lại với m = 1 tính mờ mất đi và thuật toán sẽ trở về K-Means truyền thống.
- ***V*** là ma trận tâm cụm (cluster centroids): có kích thước c x d (c là số lượng cụm, d là số chiều của dữ liệu) với v_i là tâm cụm thứ i.

Giải thích hàm mục tiêu:

Từ công thức (1) khi ta cố định giá trị của i và chỉ có j là đại lượng thay đổi, lúc này công thức có thể được viết gọn lại như sau: $J_i = \sum_{j=1}^{c} {u_j}^{m} |d_j|_2^2$. Với u_j là mức độ thành viên của điểm dữ liệu x_i vào cụm j và d_j là khoảng cách Euclid từ điểm x_i đến tâm cụm v_j . Cùng với ràng buộc $\sum_{j=1}^{c} u_j = 1$, bài toán tối ưu tương đương với việc lựa chọn các giá trị u_j sao cho hàm J_i đạt giá trị nhỏ nhất. Để đạt được điều này, cần gán trọng số lớn hơn (tức giá trị u_j lớn hơn) cho những cụm mà x_i gần hơn (tức có $|d_j|_2^2$ nhỏ hơn). Tuy nhiên, do có tham số mờ m, nên nghiệm tối ưu của bài toán không còn là các giá trị 0 hoặc 1 như trong K-means, mà là các giá trị liên tục trong khoảng [0, 1]. Điều này giúp mô hình linh hoạt hơn và thể hiện được mức độ không chắc chắn.

Đôi khi vẫn sẽ xảy ra các giá trị 0 và 1 trong u_j nhưng điều này chỉ xảy ra khi điểm dữ liệu i đã chọn đó nằm trùng với 1 tâm cụm j, điều này không thường xuyên xảy ra và có thể dẫn đến lỗi tính toán nên ta cần thiết phải xử lý riêng trường hợp này khi phát hiện ra (sẽ được trình bày ở phần cập nhật thuật toán). Lặp lại quá trình tối ưu cho tất cả các điểm x_i trong tập dữ liệu, ta thu được hàm mục tiêu tổng quát như công thức (1).

2.2.3. Công thức cập nhật chỉ số

Công thức cập nhật u:

$$u_{ij} = \left(\sum_{k=1}^{c} \left(\frac{|x_i - v_j|_2}{|x_i - v_k|_2} \right)^{\frac{2}{m-1}} \right)^{-1} \qquad (2)$$

Được rút ra từ việc đạo hàm hàm mục tiêu theo u_{ij} với ràng buộc như đã đề cập. Có thể nhận ra rằng, trong trường hợp x_i trùng với v_j, nghĩa là $|x_i - v_j|_2 = 0$, công thức trên sẽ bị lỗi vì 0^{-1} là một giá trị không xác định. Trong trường hợp xét đến giới hạn số $\lim_{x \to 0} x^{-1} = \infty$ có thể hiểu là mức độ thành viên của điểm dữ liệu i vào cụm j là ở mức tối đa, mà tối đa ở đây ta chỉ có 1 nên từ đó ta sẽ đặt cho $u_{ij} = 1$ và $u_{ij'} = 0$ $(j' \neq j)$

Công thức cập nhật v:

$$v_j = \frac{\sum_{i=1}^{n} {u_{ij}}^m x_i}{\sum_{i=1}^{n} {u_{ij}}^m} \qquad (3)$$

Được rút ta từ việc đạo hàm hàm mục tiêu theo v_j. Công thức này có thể được hiểu là trung bình có trọng số của các điểm dữ liệu x_i trong cụm j, với trọng số là lũy thừa cấp m của mức độ thành viên u_{ij}.

2.2.4. Luồng hoạt động của thuật toán

Bước 1: Khởi tạo tham số

Chọn số lượng cụm c. Chọn hệ số mờ $m > 1$ (thường lấy m = 2). Chọn ngưỡng hội tụ $\varepsilon > 0$ (thường chọn $\varepsilon =$ 1e-4). Khởi tạo giá trị cho ma trận tâm cụm v hoặc cho ma trận thành viên u (chỉ khởi tạo 1 trong 2).

Bước 2: Thuật toán lặp

2.1: Lưu lại ma trận tâm cụm $\boldsymbol{V}$ ban đầu, ký hiệu là v_{old}

2.2: Cập nhật ma trận thành viên bằng công thức (2)

2.3: Cập nhật ma trận tâm cụm bằng công thức (3)

2.4: Kiểm tra điều kiện hội tụ theo công thức: $|v_{old} - v|_2 < \varepsilon$

Nếu điều kiện thỏa mãn, dừng thuật toán. Ngược lại, quay lại 2.1.

Lưu ý: Thứ tự của 2.1 và 2.2 có thể đổi chỗ cho nhau phụ thuộc vào việc ta khởi tạo ma trận tâm cụm trước hay ma trận thành viên trước. Nếu khởi tạo ma trận thành viên trước thì ta thực hiện thuật toán với thứ tự như trên, nếu khởi tạo ma trận tâm cụm trước ta đổi vị trí của bước 2.1 và 2.2 cho nhau.

Bước 3: Giải mờ

Sau khi hội tụ, từ ma trận thành viên thu được ta sẽ giải mờ để tìm ra ma trận thành viên dạng cứng để xác định điểm dữ liệu nào thuộc vào cụm nào. Có thể thực hiện theo 2 cách giải mờ sau:

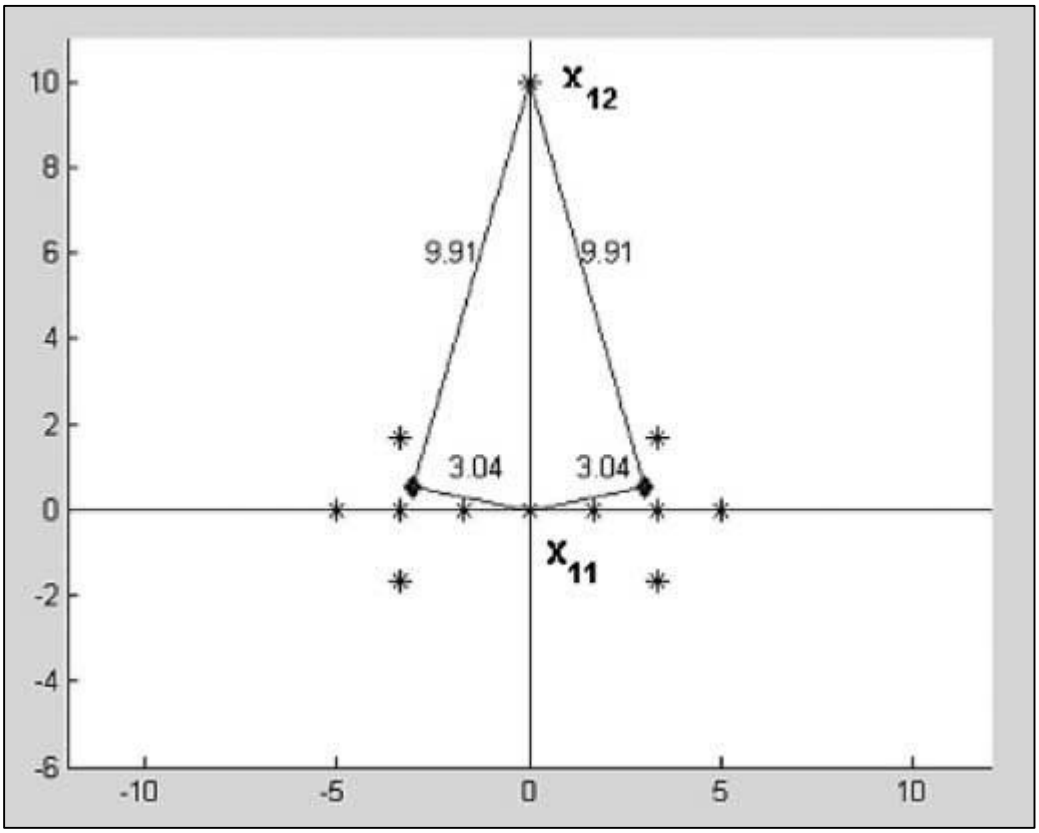
- *Cách 1:* Với mỗi điểm dữ liệu i và bộ các u_{ij} tương ứng. Ta đặt giá trị 1 cho u_{ij} có giá trị lớn nhất và 0 cho các $u_{ij'}(j' \neq j)$, từ đây ta có được một bộ ma trận thành viên dạng cứng.
- *Cách 2:* Với mỗi điểm dữ liệu i và bộ các u_{ij} tương ứng. Ta đặt giá trị 1 cho u_{ij} có giá trị lớn nhất và lớn hơn ngưỡng $\frac{3}{5c}$ với c là số tâm cụm, đặt giá trị 0 cho các $u_{ij'}(j' \neq j)$ còn lại. Các bộ u_i không thỏa mãn điều kiện này được coi là chưa thể quyết định được nó thuộc vào cụm nào, cần phải có sự cân nhắc thêm. Có thể tách các điểm dữ liệu chưa chắc chắn này ra để phân cụm lần 2, sau đó so sánh.

2.3. Thuật toán phân cụm khả năng (PCM)

2.3.1. Khái niệm

Phân cụm mờ đã chứng minh được nhiều ưu điểm so với phân cụm cứng, bởi vì nó không buộc phải gán hoàn toàn mỗi điểm dữ liệu vào một cụm duy nhất mà cho phép các điểm có mức độ "chia sẻ" giữa nhiều cụm. Một trong những thuật toán phân cụm mờ phổ biến nhất là FCM, được sử dụng rộng rãi nhờ tính hiệu quả và khả năng ứng dụng cao trong nhiều bài toán như phân loại mẫu hoặc phân đoạn ảnh. Tuy nhiên, FCM vẫn tồn tại một hạn chế lớn: ràng buộc tổng độ thuộc (membership) của mỗi điểm dữ liệu vào tất cả các cụm phải

bằng 1. Ràng buộc này khiến cho độ thuộc chỉ mang ý nghĩa tương đối, không phản ánh đúng mức độ "đặc trưng" (typicality) hay "mức độ phù hợp" (compatibility) thực sự của điểm dữ liệu với từng cụm. Vấn đề này đặc biệt rõ rệt trong môi trường có nhiễu (outliers), nơi các thuật toán như K-means hoặc FCM thường gặp khó khăn do tâm cụm có thể bị kéo lệch về phía các điểm nhiễu. Do đó, cần một cách tiếp cận mới để tạo ra độ thuộc mang ý nghĩa "tuyệt đối" về mức độ tương thích, thay vì chỉ thể hiện mức độ chia sẻ tương đối như trong FCM.



Hình 2.5. Hình ảnh mô hình PCM

Thuật toán phân cụm khả năng (Possibilistic Clustering – PCM) được đề xuất nhằm khắc phục hạn chế của thuật toán phân cụm mờ FCM, vốn yêu cầu tổng độ thuộc của mỗi điểm dữ liệu vào tất cả các cụm phải bằng 1. Ràng buộc này khiến độ thuộc chỉ mang ý nghĩa tương đối và không phản ánh đúng mức độ tương thích thực sự giữa điểm và từng cụm, đặc biệt trong môi trường có nhiễu, nơi các điểm ngoại lai có thể làm lệch tâm cụm. Ngược lại, PCM loại bỏ ràng buộc đó, cho phép độ thuộc của một điểm vào từng cụm vẫn nằm trong khoảng [0,1] nhưng không cần tổng bằng 1, miễn là không đồng thời bằng 0 cho tất cả các cụm. Nhờ vậy, độ thuộc vào mỗi cụm trở nên độc lập, không bị

ảnh hưởng bởi các cụm khác, giúp các điểm nhiễu có thể nhận giá trị rất thấp mà không kéo lệch tâm cụm. Nó cho phép các điểm outliers có độ thuộc rất nhỏ vào các cụm, làm cho tâm cụm giờ đây không còn bị lệch về phía outliers quá nhiều.

Dù bản thân thuật toán cũng thể hiện được tính mờ và còn có thể khắc phục được nhược điểm lớn của FCM nhưng thuật toán này thường không được sử dụng độc lập mà thường được dùng để tích hợp vào hàm mục tiêu của một thuật toán khác, do nó có một nhược điểm là nhạy cảm với việc khởi tạo ngẫu nhiên. Nếu tâm cụm khởi tạo quá gần hoặc bị trùng với điểm nhiễu, nó hoàn toàn có thể coi điểm nhiễu đó là một tâm cụm, còn các điểm dữ liệu đáng lẽ ra phải thuộc một cụm riêng biệt thì lại bị xem là các outliers. Một trường hợp tệ hơn đó là các tâm cụm bị trùng nhau, điều này xảy ra nếu không may 3 tâm cụm bị khởi tạo gần nhau. Cách khắc phục điều này thường là sử dụng một thuật toán khác chạy trong phase 1 để tạo ra một ma trận tâm cụm phù hợp, làm khởi tạo ngẫu nhiên cho thuật toán sau đó phase 2 của thuật toán mới chạy PCM.

Bảng 2.2. Bảng so sánh giữa FCM và PCM

Tiêu chí	**FCM**	**PCM**
Mức độ gán (Membership)	$u_{ij} \in [0,1]; \sum_{j=1}^{c} u_{ij} = 1$	$t_{ij} \in [0,1]$; *Không có ràng buộc*
Loại membership	Fuzzy membership (ràng buộc chặt chẽ)	Possibilistic membership (mềm dẻo, độc lập giữa các cụm)
Ý nghĩa của membership	Thể hiện mức độ tương đối của điểm dữ liệu thuộc vào từng cụm	Thể hiện mức độ tuyệt đối của mỗi điểm dữ liệu vào từng cụm
Nhạy cảm với nhiễu	Có nhạy cảm với nhiễu	Không nhạy cảm với nhiễu
Nhạy cảm với khởi tạo ngẫu nhiên	Ít bị ảnh hưởng của khởi tạo ngẫu nhiên nếu cấu trúc các cụm và số lượng cụm được chọn phù hợp	Bị ảnh hưởng lớn bởi khởi tạo ngẫu nhiên
Tính phổ biến	Được sử dụng nhiều trong các bài toán học không giám sát	Chủ yếu được sử dụng khi phát hiện dữ liệu có nhiễu hoặc có thể tồn tại điểm dữ liệu không thuộc về bất kỳ cụm nào

2.3.2. Hàm mục tiêu

Hàm mục tiêu của thuật toán PCM có dạng:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} {t_{ij}}^{m} \left| x_i - v_j \right|_2^2 + \sum_{j=1}^{c} \gamma_j \sum_{i=1}^{n} (1 - t_{ij})^m \qquad (4)$$

Các thành phần trong công thức:

- ***T*** là ma trận khả năng chứa các phần tử t_{ij} (Possibilistic values) có kích thước là n x c trong đó n là số lượng các điểm dữ liệu, c là số cụm đã chọn ban đầu. Từng phần tử t_{ij} có giá trị nằm trong khoảng [0, 1] thể hiện khả năng một điểm dữ liệu thứ i có khả năng thuộc về cụm thứ j là bao nhiêu, khi giá trị càng gần 0 thì điểm dữ liệu có khả năng thuộc về cụm tương ứng là càng thấp, ngược lại khi giá trị này càng gần 1 khả năng này càng cao.
- ***m*** là hệ số mờ (membership value) có vai trò như đã đề cập đến trong thuật toán FCM. Nó thể hiện tính mờ cho thuật toán.
- ***V*** là ma trận tâm cụm chứa các v_i(cluster centroid) có kích thước c x d (với c là số lượng cụm, d là số thuộc tính của mỗi điểm dữ liệu).
- $\boldsymbol{\gamma}$ là một vector chứa các phần tử γ_j dùng để điều chỉnh mức độ tác động của thành phần cân bằng chỉ số của t_{ij}.

Giải thích hàm mục tiêu:

Hàm mục tiêu hoạt động tương tự như FCM khi ta cần tìm ra ma trận ***T*** và ***V*** sao cho nó trả về được giá trị nhỏ nhất của hàm mục tiêu. Ta sẽ gọi $\sum_{i=1}^{n} \sum_{j=1}^{c} {t_{ij}}^{m} \left| x_i - v_j \right|_2^2$ là thành phần 1, $\sum_{j=1}^{c} \gamma_j \sum_{i=1}^{n} (1 - t_{ij})^m$ là thành phần 2. Điểm khác biệt của PCM so với FCM ở chỗ nó tồn tại thành phần 2 được gọi là thành phần điều chỉnh đóng vai trò điều chỉnh nghiệm tối ưu của thuật toán. Thành phần này có ý nghĩa tránh cho toàn bộ nghiệm ***T*** của thuật toán rơi vào nghiệm tầm thường (đều có giá trị là 0) vì nó không chỉ tối ưu theo khoảng cách mà cần phải cân nhắc mức độ phạt của thành phần điều chỉnh này. Mỗi giá trị t_{ij} sẽ cần phải có một giá trị hợp lý để tối thiểu tổng của 2 thành phần. Giá trị của γ_j thể hiện mức độ tác độngmục của thành phần 2. Khi giá trị của nó càng lớn thì t_{ij} sẽ càng ít chú trọng vào việc tối thiểu thành phần 1 hơn mà tập trung để giảm độ lớn của thành phần 2, và đây là điều ta không mong

muốn khi mà ta cần giá trị t_{ij} phụ thuộc chủ yếu vào khoảng cách của điểm dữ liệu i đến cụm j từ đó nó thể hiện được điểm dữ liệu nào đang ở xa cụm nào và mức độ xa thế nào so với các cụm khác. Nhưng cũng không thể để γ_j quá nhỏ vì điều này làm cho t_{ij} sẽ chú trọng vào việc tối thiểu thành phần 1 hơn thành phần 2, gây ra việc giá trị của t_{ij} trở nên quá nhỏ, dù vẫn thể hiện được đúng mức độ xa của điểm dữ liệu tới tâm cụm này so với các tâm cụm khác nhưng việc giá trị của nó quá nhỏ cũng sẽ có ảnh hưởng đến sai số tính toán, đặc biệt là khi kết hợp PCM với các thuật toán khác, giá trị của t_{ij} nhỏ sẽ làm giảm tác động của thành phần khả năng trong hàm mục tiêu kết hợp khiến cho vai trò xử lý nhiễu của nó không còn được rõ ràng nữa. Nên việc chọn γ_j phù hợp là điều cần thiết.

2.3.3. Công thức cập nhật chỉ số

Công thức cập nhật t:

$$t_{ij} = \left(1 + \left(\frac{|x_i - v_j|_2{}^2}{\gamma_i}\right)^{\frac{1}{m-1}}\right)^{-1} \qquad (5)$$

Được rút ra từ việc đạo hàm hàm mục tiêu theo t.

Công thức cập nhật v:

$$v_j = \frac{\sum_{i=1}^{n} t_{ij}{}^m x_i}{\sum_{i=1}^{n} t_{ij}{}^m} \qquad (6)$$

Được rút ra từ việc đạo hàm hàm mục tiêu theo v. Có thể hiểu là trung bình có trọng số của các điểm dữ liệu trong các cụm.

Công thức cập nhật γ:

$$\gamma_j = \frac{\sum_{i=1}^{n} t_{ij}{}^m |x_i - v_j|_2{}^2}{\sum_{i=1}^{n} t_{ij}{}^m} \qquad (7)$$

Được tính dựa trên trung bình khoảng cách của từng điểm dữ liệu đến các cụm, ý nghĩa của công thức là muốn đưa ra một bộ các giá trị γ_j sao cho 2 thành phần của hàm mục tiêu được cân bằng với nhau.

2.3.4. Luồng hoạt động của thuật toán

Bước 1: Khởi tạo tham số

Khởi tạo tham số: Chọn số lượng cụm c. Chọn hệ số mờ $m > 1$ (thường là 2). Chọn ngưỡng hội tụ $\varepsilon > 0$ (thường là 1e-4). Khởi tạo giá trị cho ma trận tâm cụm v hoặc cho ma trận khả năng $\boldsymbol{T}$ (chỉ khởi tạo 1 trong 2).

Bước 2: Thuật toán lặp

2.1: Tính ma trận γ bằng công thức (7)

2.2: Lưu lại ma trận tâm cụm $\boldsymbol{V}$ ban đầu đặt là $\boldsymbol{V_{old}}$

2.3: Cập nhật ma trận khả năng bằng công thức (5)

2.4: Cập nhật ma trận tâm cụm bằng công thức (6)

2.5: Kiểm tra điều kiện dừng $|\boldsymbol{V_{old}} - \boldsymbol{V}|_2 < \varepsilon$

Nếu điều kiện dừng tại 2.5 thỏa mãn ta dừng thuật toán. Còn không ta trở lại bước 2.1.

Thứ tự của 2.2 và 2.3 có thể đổi chỗ cho nhau phụ thuộc vào việc ta khởi tạo ma trận tâm cụm trước hay ma trận khả năng trước. Nếu khởi tạo ma trận khả năng trước thì ta thực hiện thuật toán với thứ tự như trên, nếu khởi tạo ma trận tâm cụm trước ta đổi vị trí của bước 2.2 và 2.3 cho nhau.

Bước 3: Giải mờ

Từ ma trận khả năng thu được ta sẽ giải mờ để tìm ra ma trận thành viên dạng cứng, từ ma trận thành viên dạng cứng này có thể quyết định được điểm dữ liệu nào thuộc vào cụm nào. Với mỗi điểm dữ liệu i và bộ các t_{ij} tương ứng. Ta đặt giá trị 1 cho t_{ij} có giá trị lớn nhất và 0 cho các $t_{ij'}(j' \neq j)$, từ đây ta có được một bộ ma trận thành viên dạng cứng.

2.4. Thuật toán phân cụm mờ khả năng

2.4.1. Khái niệm

Thuật toán kết hợp giữa FCM và PCM được xây dựng nhằm tận dụng những ưu điểm nổi bật của cả hai phương pháp, đồng thời khắc phục những hạn chế cố hữu mà mỗi thuật toán riêng lẻ gặp phải. Cụ thể, FCM nổi bật với khả năng phân cụm mềm, cho phép một điểm dữ liệu thuộc về nhiều cụm với

các mức độ khác nhau, nhưng lại nhạy cảm với nhiễu và khởi tạo tâm cụm ban đầu. Ngược lại, PCM khắc phục tốt ảnh hưởng của nhiễu nhờ mô hình khả năng, nhưng lại dễ rơi vào tình trạng phân cụm kém khi khởi tạo tâm cụm không phù hợp. Khi kết hợp hai thuật toán này, thuật toán mới có khả năng giảm thiểu đáng kể tác động tiêu cực từ khởi tạo ngẫu nhiên nhờ FCM, đồng thời tăng khả năng chống nhiễu nhờ PCM. Tuy nhiên, một thách thức quan trọng trong thuật toán kết hợp này là việc lựa chọn trọng số giữa ma trận thành viên (từ FCM) và ma trận khả năng (từ PCM). Trọng số này quyết định mức độ ảnh hưởng của từng thành phần đến quá trình phân cụm.

2.4.2. Hàm mục tiêu

Hàm mục tiêu của thuật toán có dạng:

$$J = \sum_{i=1}^{n}\sum_{j=1}^{c}(au_{ij}{}^{m} + bt_{ij}{}^{m})|x_i - v_j|_2{}^{2} + \sum_{j=1}^{c}\gamma_j \sum_{i=1}^{n}\left(1 - t_{ij}\right)^{m} \quad (8)$$

Các thành phần trong công thức:

- ***U*** là ma trận thành viên chứa các phần tử u_{ij} (membership values): có kích thước là n x c (n là số lượng các đối tượng dữ liệu, c là số lượng cụm), với giá trị của phần tử u_{ij} thể hiện độ thuộc của điểm dữ liệu thứ i vào cụm thứ j. ***U*** có một ràng buộc đó là với mọi i bất kì thì $\sum_{j=1}^{c} u_{ij} = 1$. Chức năng và vai trò của nó như đã đề cập đến trong thuật toán FCM, nó đại diện cho sự tác động của thành phần FCM vào trong thuật toán.
- ***T*** là ma trận khả năng chưa các phần tử t_{ij} (Possibilistic values) có kích thước là n x c (trong đó n là số lượng các điểm dữ liệu, c là số cụm đã chọn ban đầu). Từng phần tử t_{ij} có giá trị nằm trong khoảng [0, 1] thể hiện khả năng một điểm dữ liệu thứ i có khả năng thuộc về cụm thứ j là bao nhiêu. Chức năng và vai trò tương tự như đã đề cập đến trong thuật toán PCM, nó đại diện cho sự tác động của thành phần PCM vào trong thuật toán.
- ***m*** là hệ số mờ (membership value) có vai trò như đã đề cập đến trong thuật toán FCM. Nó thể hiện tính mờ cho thuật toán.
- ***V*** là ma trận tâm cụm chứa các v_i(cluster centroid) có kích thước c x d (với c là số lượng cụm, d là số thuộc tính của mỗi điểm dữ liệu).

- $\boldsymbol{\gamma}$ là một vector chứa các phần tử γ_j dùng để điều chỉnh mức độ tác động của thành phần cân bằng chỉ số của t_{ij}.
- $\boldsymbol{a}, \boldsymbol{b}$ là trọng số điều chỉnh mức độ tác động của thành phần FCM và PCM vào hàm mục tiêu.

Giải thích hàm mục tiêu:

Gọi thành phần 1 là $\sum_{i=1}^{n}\sum_{j=1}^{c}(au_{ij}{}^{m} + bt_{ij}{}^{m})|x_i - v_j|_2{}^{2}$, thành phần 2 là $\sum_{j=1}^{c}\gamma_j\sum_{i=1}^{n}(1-t_{ij})^{m}$. Theo quan sát có thể thấy, thuật toán đơn thuần là tổng của 2 hàm mục tiêu FCM và PCM với thành phần 1 gồm có u_{ij} và t_{ij} được nhân thêm một trọng số phía trước đó là a và b nhằm điều chỉnh mức độ tác động của 2 thành phần. Nếu trọng số a được điều chỉnh lớn hơn b điều này có nghĩa là thuật toán sẽ chú trọng hơn vào việc điều chỉnh tâm cụm theo FCM, FCM giờ đây "có tiếng nói hơn" khiến cho vai trò của PCM bị giảm đi, tác động ít hơn đến quá trình tối ưu của thuật toán. Ngược lại nếu trọng số b lớn hơn a, thành phần PCM sẽ có tác động lớn hơn, giúp cho việc phân cụm tập trung vào việc giải quyết các điểm nhiễu nhiều hơn.

Rất khó để có thể quyết định a lớn hơn b là tốt hay không tốt vì tùy thuộc vào cách phân bố của từng bộ dữ liệu thì việc dồn sự tác động lớn hơn của FCM hay PCM sẽ cho kết quả khác nhau đáng kể, từ đó mang lại kết quả tốt xấu khác nhau. Trong trường hợp dữ liệu không có quá nhiều nhiễu hoặc thậm chí không có, thành phần FCM nên có trọng số lớn hơn vì vai trò của nó đang phù hợp hơn. Ngược lại nếu có nhiễu thì nên đặt trọng số lớn hơn vào PCM để nó thực hiện công việc của mình. Việc chọn trọng số a và b nên được rút ra từ quan sát thực nghiệm, từ đó rút ra được một bộ trọng số a và b phù hợp với từng kiểu dữ liệu và từng bài toán.

Thành phần 2 trong hàm mục tiêu có vai trò tương tự như thành phần 2 trong thuật toán PCM. Là một thành phần điều chỉnh tránh cho toàn bộ các phần tử trong ma trận $\boldsymbol{T}$ có giá trị là 0 (nghiệm tầm thường).

2.4.3. Công thức cập nhật chỉ số

Công thức cập nhập u:

$$u_{ij} = \left(\sum_{k=1}^{c} \left(\frac{|x_i - v_j|_2}{|x_i - v_k|_2} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (9)$$

Giống hệt với công thức cập nhật u_{ij} của FCM, do khi đạo hàm theo u_{ij} toàn bộ mọi phần có liên quan đến t_{ij} đều được coi là hằng số và sẽ có đạo hàm bằng 0. Không có tác động gì đến việc cập nhật u_{ij}.

Công thức cập nhập t:

$$t_{ij} = \left(1 + \left(\frac{b|x_i - v_j|_2{}^2}{\gamma_i} \right)^{\frac{1}{m-1}} \right)^{-1} \quad (10)$$

Được rút ra từ đạo hàm hàm mục tiêu theo t_{ij}.

Công thức cập nhật v:

$$v_j = \frac{\sum_{i=1}^{n} (a u_{ij}{}^m + b t_{ij}{}^m) x_i}{\sum_{i=1}^{n} (a u_{ij}{}^m + b t_{ij}{}^m)} \quad (11)$$

Trong công thức cập nhật v này, ta có thể thấy được rõ ràng nhất sự tác động của trong số a và b vào việc quyết định xem thành phần của FCM hay PCM có vai trò lớn hơn vào việc cập nhật tâm cụm. Đây chính là ý nghĩa của việc kết hợp 2 thuật toán. Việc cập nhật u_{ij} và t_{ij} sẽ thực hiện độc lập với nhau không bên nào tác động đến bên nào, nhưng đến khi cập nhật v_j cả u_{ij} và t_{ij} sẽ cùng tham gia để thể hiện vai trò của mình trong việc điều chỉnh tâm cụm sao cho hợp lý.

2.4.4. Luồng hoạt động của thuật toán

Bước 1: Khởi tạo tham số

Chọn số lượng cụm c, chọn hệ số mờ $m > 1$ (thường m = 2), chọn ngưỡng hội tụ $\varepsilon > 0$ (thường là 1e-4), khởi tạo giá trị cho ma trận tâm cụm $\boldsymbol{V}$ hoặc cho ma trận thành viên $\boldsymbol{U}$ cùng với ma trận khả năng $\boldsymbol{T}$ (chỉ khởi tạo 1 trong 2).

Bước 2: Thuật toán lặp

2.1: Lưu lại ma trận tâm cụm $\boldsymbol{V}$ ban đầu đặt là $\boldsymbol{V_{old}}$

2.2: Cập nhật ma trận thành viên và khả năng bằng công thức (9) và (10)

2.3: Cập nhật ma trận tâm cụm bằng công thức (11)

2.4: Kiểm tra điều kiện dừng $|\boldsymbol{V_{old}} - \boldsymbol{V}|_2 < \varepsilon$

Nếu điều kiện dừng tại 2.4 thỏa mãn ta dừng thuật toán. Còn không ta trở lại bước 2.1.

Thứ tự của 2.1 và 2.2 có thể đổi chỗ cho nhau phụ thuộc vào việc ta khởi tạo ma trận tâm cụm trước hay ma trận thành viên và ma trận khả năng trước. Nếu khởi tạo ma trận thành viên và ma trận khả năng trước thì ta thực hiện thuật toán với thứ tự như trên, nếu khởi tạo ma trận tâm cụm trước ta đổi vị trí của bước 2.1 và 2.2 cho nhau.

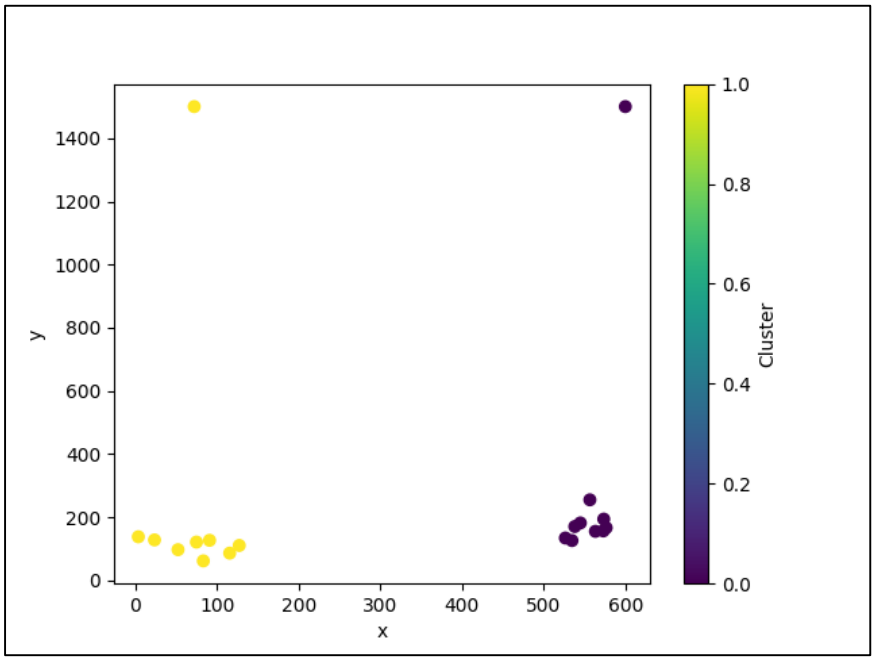
Bước 3: Giải mờ

Từ ma trận thành viên thu được ta sẽ giải mờ để tìm ra ma trận thành viên dạng cứng, từ ma trận thành viên dạng cứng này có thể quyết định được điểm dữ liệu nào thuộc vào cụm nào. Có nhiều cách giải mờ khác nhau như:

- *Cách 1:* Với mỗi điểm dữ liệu i và bộ các u_{ij} tương ứng. Ta đặt giá trị 1 cho u_{ij} có giá trị lớn nhất và 0 cho các $u_{ij'}(j' \neq j)$, từ đây ta có được một bộ ma trận thành viên dạng cứng
- *Cách 2:* Mới mỗi điểm dữ liệu i và bộ các u_{ij} tương ứng. Ta đặt giá trị 1 cho u_{ij} có giá trị lớn nhất và lớn hơn ngưỡng $\frac{3}{5c}$ với c là số tâm cụm, đặt giá trị 0 cho các $u_{ij'}(j' \neq j)$ còn lại. Các bộ u_i không thỏa mãn điều kiện này được coi là chưa thể quyết định được nó thuộc vào cụm nào, cần phải có sự cân nhắc thêm. Có thể tách các điểm dữ liệu chưa chắc chắn này ra để phân cụm lần 2, sau đó so sánh.

2.4.5. Mô phỏng trực quan thuật toán

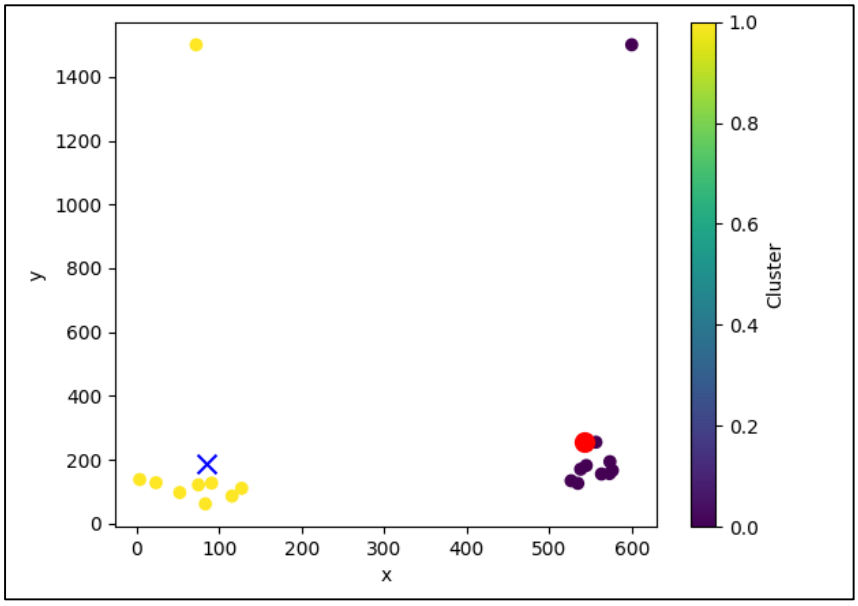
Giả sử ta có bộ dữ liệu như sau:



Hình 2.6. Mô phỏng trực quan bộ dữ liệu

Có thể nhận thấy rằng bộ dữ liệu này đang được chia làm 2 cụm và trong mỗi cụm đó đang có một điểm dữ liệu nhiễu có chỉ số y vượt trội hoàn toàn so với các điểm dữ liệu còn lại thuộc cùng cụm với mình.

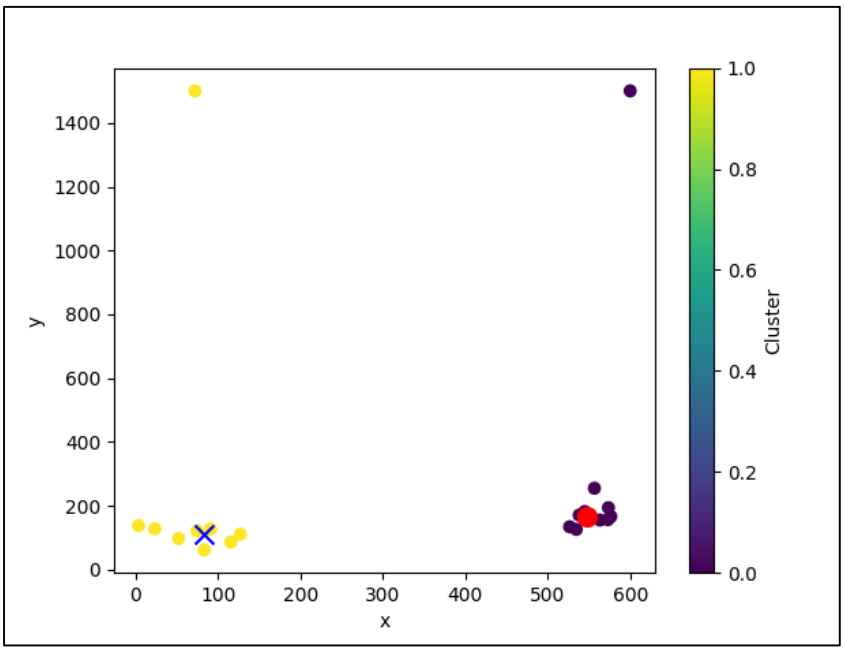
Khi chạy thuật toán FCM với các tham số m=2, số cụm k=2 trên bộ dữ liệu này ra có được một kết quả như hình sau:



Hình 2.7. Kết quả khi chạy FCM

Có thể nhận xét rằng, điểm dữ liệu nhiễu đã kéo lệch tâm cụm về phía của chúng một đoạn tương đối, dù hầu như nó không ảnh hưởng quá nhiều đến bộ dữ liệu đang được sử dụng ở đây nhưng trong các trường hợp khác như số cụm nhiều hơn, số điểm dữ liệu nhiều hơn, điểm dữ liệu nhiễu có chỉ số thuộc tính khác biệt hơn thì có thể gây ra các sai sót không mong muốn. Đây là vấn đề chính của thuật toán FCM và K-Means, bị ảnh hưởng bởi nhiễu.

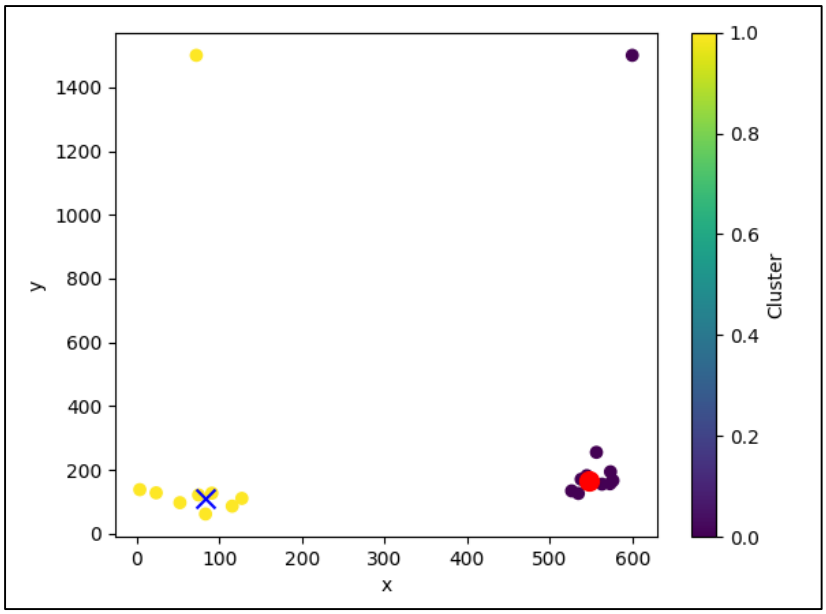
Khi chạy thuật toán PCM với các tham số mờ m=2, số cụm k=2 trên bộ dữ liệu này ra có được một kết quả như hình sau:



Hình 2.8. Kết quả khi chạy PCM

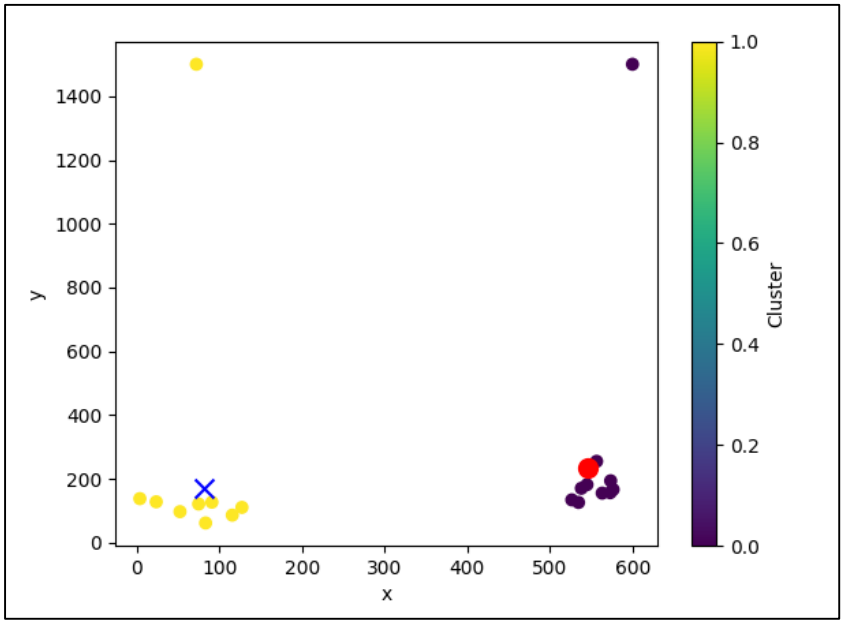
Tại đây ta có thể thấy được rõ khả năng xử lý nhiễu của thuật toán này. Các tâm cụm gần như đã được đưa về đúng vị trí phù hợp, không còn bị nhiễu kéo lệch nữa. Giúp cho hoạt động phân cụm cho ra kết quả tốt hơn mà không cần phải loại bỏ điểm dữ liệu nhiễu.

Khi chạy thuật toán PFCM với các tham số mờ m=2, số cụm k=2. Ta sẽ chia ra làm 2 trường hợp trọng số, thứ nhất là a=1 và b=2 lúc này có thể hiểu rằng PCM sẽ gây tác động lớn hơn tức sẽ có khả năng xử lý nhiễu tốt hơn cho thuật toán và kết quả đạt được như sau:



Hình 2.9. Kết quả khi chạy PFCM với a nhỏ hơn b

Kết quả của nó khá tương tự với PCM khi chạy độc lập, điều này là do PCM đã có tác động lớn hơn đến thuật toán với bộ trọng số a và b đã chọn. Ngược lại khi chọn a=2, b=1 ta có quan sát sau:

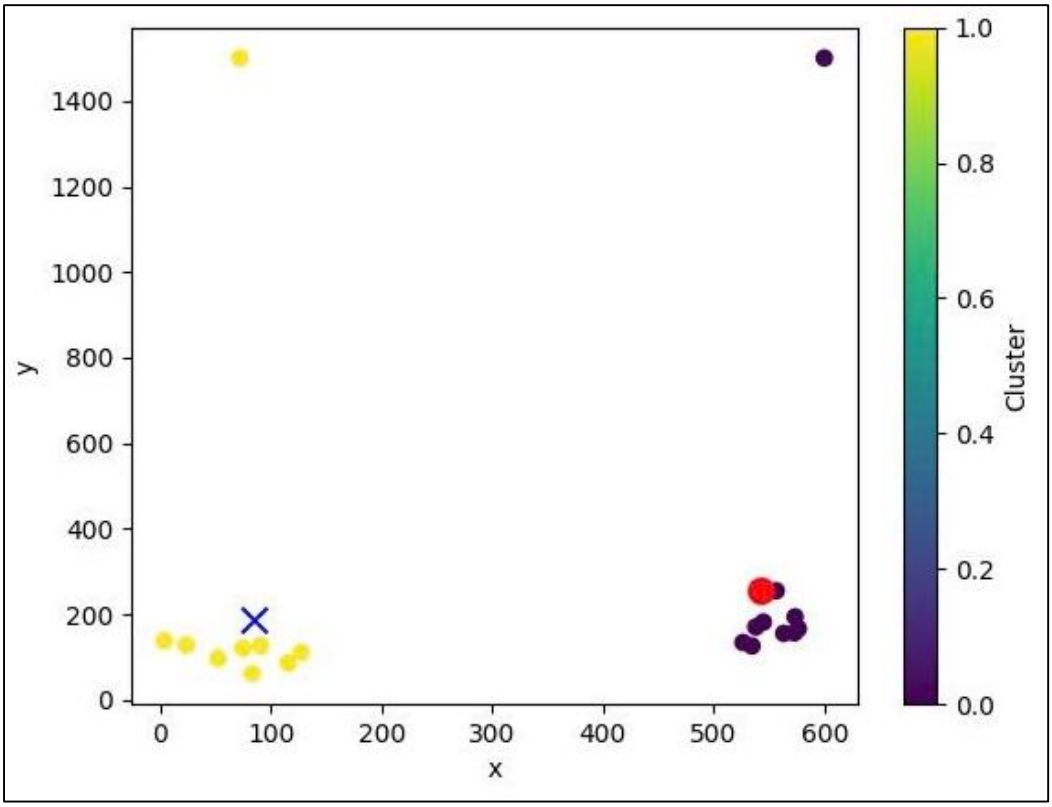


Hình 2.10. Kết quả khi chạy PFCM với a lớn hơn b

Kết quả của nó khá tương đồng với FCM, điều này là do FCM đã có tác động lớn hơn đến thuật toán với bộ trọng số a và b đã chọn. Tùy thuộc vào mong muốn ta có thể điều chỉnh các mức độ tác động của ma trận thành viên và ma trận khả năng, chỉnh b lớn hơn khi ta mong muốn xử lý nhiễu và kết quả

phân cụm sẽ như hình 1, chỉnh a lớn hơn khi ta mong muốn thuật toán phân cụm chấp nhận nhiễu và kết quả sẽ thu được giống như hình 2.

Nhận thấy rằng, khi a=b kết quả nhận được tương tự với khi a>b, điều này cho thấy khoảng giá trị của các phần tử trong ma trận ***U*** đang lớn hơn khoảng giá trị của các phần tử trong ma trận ***T*** nên ngay cả khi chúng có mức độ tác động như nhau thì FCM vẫn được chú trọng để tối ưu hơn.



Hình 2.11. Kết quả khi chạy PFCM với a bằng b

2.5. Kết luận chương 2

Chương 2 đã cung cấp nền tảng lý thuyết toàn diện về phân cụm mờ, từ các khái niệm cơ bản đến những thuật toán cốt lõi như FCM, PCM và PFCM. Qua việc phân tích chi tiết từng hàm mục tiêu, công thức cập nhật và luồng hoạt động, nhóm đã làm rõ cơ chế phân cụm cũng như khả năng mô hình hóa dữ liệu mơ hồ của từng phương pháp. Đặc biệt, sự kết hợp giữa FCM và PCM trong thuật toán PFCM được đánh giá là hướng tiếp cận tiềm năng, khi dung hòa được độ mềm dẻo của FCM và khả năng chống nhiễu của PCM. Những kiến thức lý thuyết này không chỉ giúp nhóm hiểu sâu bản chất của các thuật toán, mà còn là cơ sở vững chắc để bước sang giai đoạn thực nghiệm và ứng dụng.

CHƯƠNG 3. THỰC NGHIỆM

3.1. Thực nghiệm trên bộ dữ liệu Iris

3.1.1. Giới thiệu về bộ dữ liệu

- **Nguồn gốc bộ dữ liệu**

Bộ dữ liệu Iris, được công bố bởi nhà thống kê và sinh học Ronald Fisher vào năm 1936, là một tập hợp dữ liệu đa biến kinh điển được sử dụng rộng rãi trong các nghiên cứu về phân loại và phân cụm trong lĩnh vực học máy và thống kê. Tính phổ biến và cấu trúc rõ ràng của nó đã biến Iris thành một tập hợp chuẩn để đánh giá hiệu suất của các thuật toán học máy.

- **Cấu trúc dữ liệu**

Số lượng mẫu (Instances): Bộ dữ liệu bao gồm 150 mẫu quan sát, mỗi mẫu đại diện cho một bông hoa Iris.

Số lượng lớp (Classes): Các mẫu được phân loại vào ba loài hoa Iris riêng biệt, với 50 mẫu cho mỗi loài: *Iris setosa, Iris versicolor, Iris virginica*. Sự phân bố đồng đều giữa các lớp (50 mẫu mỗi lớp) làm cho bộ dữ liệu này trở thành một ví dụ tốt cho các bài toán phân loại cân bằng.

Đặc trưng định lượng (Features): Mỗi mẫu được mô tả bằng bốn đặc trưng định lượng, đo bằng centimet: sepal length (chiều dài đài hoa), sepal width (chiều rộng đài hoa), petal length (chiều dài cánh hoa), petal width (chiều rộng cánh hoa). Các đặc trưng này cung cấp thông tin hình thái học cần thiết để phân biệt các loài.

- **Đặc điểm nổi bật và ứng dụng**

Bộ dữ liệu Iris nổi tiếng với khả năng phân tách tuyến tính rõ ràng của loài *Iris setosa* khỏi hai loài còn lại, trong khi *Iris versicolor* và *Iris virginica* có sự chồng lấn nhất định trong không gian đặc trưng, tạo ra một thách thức phân loại vừa phải. Điều này làm cho Iris trở thành một công cụ lý tưởng để:

Kiểm định thuật toán: Được sử dụng làm benchmark để so sánh và kiểm định hiệu suất của các thuật toán phân loại (ví dụ: Support Vector Machines, Cây quyết định, Logistic Regression) và phân cụm (ví dụ: K-Means, Fuzzy C-Means).

Minh họa khái niệm: Giúp minh họa các khái niệm cơ bản trong học máy như trích chọn đặc trưng, giảm chiều dữ liệu (ví dụ: PCA), và đánh giá mô hình.

Nghiên cứu và giáo dục: Phục vụ như một tập dữ liệu khởi đầu cho sinh viên và nhà nghiên cứu trong việc tìm hiểu và phát triển các phương pháp học máy mới.

3.1.2. Thực thi và đánh giá

Kết quả chạy bộ dữ liệu Iris qua các thuật toán với tham số đầu vào m=2, c=3, a=1 và b=1.

Phương pháp	Số vòng lặp	Thời gian (s)	DB Index	PC Index	DI Index
FCM	23	0.0113	0.6692	0.6751	0.0886
PCM	151	0.0275	0.3828	0.1704	0.0355
PFCM	32	0.0470	0.6922	0.6495	0.0678
K-Means	4	0.0047	0.7894	None	0.0530

Hình 3.1. Kết quả chạy bộ dữ liệu Iris qua các thuật toán

- **Hiệu quả hội tụ và thời gian tính toán**

Số lần lặp (Iters): K-Means vẫn thể hiện tốc độ hội tụ nhanh nhất với chỉ 4 lần lặp, điều này là đặc trưng của thuật toán phân cụm cứng. FCM (23 lần lặp) và PFCM (32 lần lặp) yêu cầu số lần lặp cao hơn nhưng vẫn ở mức hiệu quả. PCM có số lần lặp cao nhất (151 lần lặp), cho thấy quá trình hội tụ của nó chậm hơn đáng kể, có thể do bản chất tối ưu hóa của thuật toán tập trung vào tính điển hình ít bị ràng buộc hơn.

Thời gian chạy (Time): Với dữ liệu cập nhật, K-Means (0.0047 s) là thuật toán nhanh nhất, khẳng định hiệu quả tính toán vượt trội của nó trên bộ dữ liệu Iris. Đây là kết quả phù hợp với lý thuyết về K-Means. FCM (0.0113 s), PCM (0.0275 s) và PFCM (0.0470 s) mặc dù chậm hơn K-Means nhưng vẫn rất nhanh, chỉ vài phần trăm giây, cho thấy chúng đều là các lựa chọn hiệu quả về mặt thời gian cho bộ dữ liệu có kích thước tương tự.

- **Chất lượng Phân cụm (Chỉ số DB và DI):**

Chỉ số Davies-Bouldin (DB): PCM đạt chỉ số DB thấp nhất (0.3828), cho thấy nó tạo ra các cụm có độ nén cao và khả năng phân tách tốt nhất so với các thuật toán còn lại. Điều này đặc biệt quan trọng nếu bộ dữ liệu chứa nhiễu hoặc ngoại lai, vì PCM có khả năng xử lý chúng hiệu quả hơn. FCM (0.6693) và PFCM (0.6922) cũng cho kết quả DB tốt, trong khi K-Means có chỉ số DB cao nhất (0.7894), cho thấy chất lượng cụm kém tối ưu hơn về mặt này.

Chỉ số Dunn (DI): FCM đạt chỉ số DI cao nhất (0.0886), chứng tỏ khả năng tạo ra các cụm nhỏ gọn và được phân tách rõ ràng. PFCM (0.0678) đứng thứ hai, cũng cho thấy hiệu suất tốt về phân tách cụm. K-Means (0.0530) có DI thấp hơn FCM và PFCM. PCM có chỉ số DI thấp nhất (0.0355), điều này có thể liên quan đến bản chất của thuật toán cho phép sự chồng lấn hoặc các cụm kém chặt chẽ hơn nếu các điểm ngoại lai được xem xét là tiêu biểu.

- **Đặc điểm Phân cụm Mờ (Chỉ số PC):**

Chỉ số Partition Coefficient (PC): K-Means không có giá trị PC (None) do bản chất phân cụm cứng của nó. Đối với các thuật toán mờ, FCM (0.6751) và PFCM (0.6495) có giá trị PC tương đối cao, phản ánh rằng phân hoạch của chúng có độ "mờ" thấp, tức là các điểm dữ liệu có xu hướng thuộc về một cụm chính rõ ràng. Ngược lại, PCM có giá trị PC rất thấp (0.1704). Giá trị PC thấp của PCM không nhất thiết là một nhược điểm mà là hệ quả từ việc thuật toán tập trung vào mức độ "khả năng" (typicality), cho phép một điểm có thể có khả năng cao với nhiều cụm hoặc thấp với tất cả, điều này tăng cường khả năng xử lý nhiễu của nó nhưng làm giảm độ rõ ràng của phân hoạch theo tiêu chí PC.

- **Kết luận**

Phân tích kết quả thực nghiệm trên bộ dữ liệu Iris cho thấy:

K-Means, thuật toán nhanh nhất về thời gian chạy và có tốc độ hội tụ số lần lặp cao nhất. Tuy nhiên, về chất lượng cụm nội tại (DB và DI), K-Means cho thấy hiệu suất kém hơn so với các thuật toán phân cụm mờ khả năng trên bộ dữ liệu này.

PCM, nổi bật về khả năng tạo ra các cụm có cấu trúc nội tại tốt nhất (DB thấp nhất), cho thấy tiềm năng trong các bộ dữ liệu có nhiễu hoặc ngoại lai.

Tuy nhiên, nó có thời gian hội tụ chậm nhất và giá trị PC thấp, phản ánh đặc tính của thuật toán.

FCM, thể hiện sự cân bằng rất tốt giữa tốc độ, chất lượng cụm (DB và DI tốt) và độ rõ ràng của phân hoạch mờ (PC cao). Đây là một lựa chọn mạnh mẽ cho các ứng dụng phân cụm mờ trên dữ liệu như Iris.

PFCM, với vai trò là sự kết hợp của FCM và PCM, cung cấp hiệu suất đáng tin cậy trên hầu hết các chỉ số, cho thấy khả năng dung hòa các ưu điểm của cả hai phương pháp.

3.2. Thực nghiệm trên bộ dữ liệu Dry Bean

3.2.1. Giới thiệu về bộ dữ liệu

- **Nguồn gốc về bộ dữ liệu**

  Tên bộ dữ liệu: Dry Bean Dataset.

  Nguồn tải dữ liệu: https://www.muratkoklu.com/datasets/

Bộ dữ liệu *Dry Bean* (Hạt đậu khô) là một bộ dữ liệu được xây dựng nhằm phục vụ các nghiên cứu liên quan đến nhận dạng và phân loại các loại hạt đậu dựa trên đặc trưng hình học. Bộ dữ liệu này được công bố công khai trên nền tảng UCI Machine Learning Repository – một trong những kho dữ liệu học máy uy tín và được sử dụng rộng rãi trong cộng đồng nghiên cứu. Bộ dữ liệu được thu thập và chia sẻ bởi các tác giả M. K. Koklu và M. Ozkan từ Đại học Selcuk, Thổ Nhĩ Kỳ, như một phần của nghiên cứu về phân loại hạt đậu bằng các kỹ thuật học máy truyền thống và hiện đại (Koklu & Ozkan, 2020).

- **Cấu trúc bộ dữ liệu**

Bảy loại đậu khô khác nhau đã được sử dụng trong nghiên cứu bộ dữ liệu này, có tính đến các đặc điểm như hình thức, hình dạng, loại và cấu trúc theo tình hình thị trường. Một hệ thống thị giác máy tính đã được phát triển để phân biệt bảy giống đậu khô đã đăng ký khác nhau với các đặc điểm tương tự để có được phân loại hạt giống thống nhất. Bộ dữ liệu bao gồm 13.611 hạt của 7 loại đậu khô đã đăng ký khác nhau được chụp bằng máy ảnh độ phân giải cao. Hình ảnh đậu thu được bằng hệ thống thị giác máy tính đã trải qua các giai đoạn phân đoạn và trích xuất tính năng, và tổng cộng có 16 đặc điểm (thuộc tính) gồm 12 kích thước và 4 dạng hình dạng, thu được từ các hạt.

Các thuộc tính của bộ dữ liệu:

- Area (A): Diện tích của vùng hạt đậu, tương ứng với số lượng pixel nằm bên trong ranh giới của nó.
- Perimeter (P): Chu vi của hạt đậu, được định nghĩa là độ dài đường viền bao quanh hạt.
- MajorAxisLength (L): Khoảng cách giữa hai điểm xa nhất trên hạt đậu theo trục chính – đường thẳng dài nhất có thể kẻ qua hạt.
- MinorAxisLength (l): Độ dài của đường thẳng dài nhất có thể kẻ qua hạt đậu theo phương vuông góc với trục chính.
- AspectRatio (K): Tỷ lệ giữa trục chính và trục phụ, thể hiện mức độ thon dài của hạt, tức là L/l.
- Eccentricity (Ec): Độ lệch tâm của elip có cùng các mô-men với vùng hạt, phản ánh mức độ kéo dài hoặc tròn đều.
- ConvexArea (C): Số lượng pixel nằm trong hình đa giác lồi nhỏ nhất có thể bao trọn toàn bộ vùng hạt đậu.
- EquivDiameter (Ed): Đường kính của hình tròn có diện tích bằng với diện tích thực tế của hạt đậu.
- Extent (Ex): Tỷ lệ giữa diện tích của hạt và diện tích của hình chữ nhật nhỏ nhất có thể bao trọn nó (hộp giới hạn).
- Solidity (S): Còn được gọi là độ lồi (convexity), là tỷ lệ giữa số pixel trong phần bao lồi và số pixel trong hạt đậu thực tế.
- Roundness (R): Chỉ số độ tròn được tính bằng công thức: $(4\pi \times A) / (P^2)$.
- Compactness (CO): Đo lường mức độ tròn của hạt thông qua tỷ lệ Ed/L.
- ShapeFactor1-4: Các hệ số hình dạng được xây dựng từ các đặc trưng hình học nhằm phản ánh hình dáng tổng quát của hạt đậu.
- Class: Nhãn phân loại của hạt đậu, bao gồm 7 loại: Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, và Sira. [2]

Ba mươi dòng đầu tiên của bộ dữ liệu:

	Area	Perimeter	MajorAxisLength	MinorAxisLength	AspectRation	Eccentricity	ConvexArea	EquivDiameter	Extent	Solidity	roundness	Compactness	ShapeFactor1	ShapeFactor2	ShapeFactor3	ShapeFactor4	Class
1	28395	610,291	208,1781167	173,888747	1,197191424	0,549812187	28715	190,1410973	0,763922518	0,988855999	0,958027126	0,913357755	0,007331506	0,003147289	0,834222388	0,998723889	SEKER
2	28734	638,018	200,5247957	182,7344194	1,097356461	0,411785251	29172	191,2727505	0,783968133	0,984985603	0,887033637	0,953860842	0,006978659	0,003563624	0,909850506	0,998430331	SEKER
3	29380	624,11	212,8261299	175,9311426	1,209712656	0,562727317	29690	193,4109041	0,778113248	0,989558774	0,947849473	0,908774239	0,00724391 2	0,003047733	0,825870617	0,999066137	SEKER
4	30008	645,884	210,557999	182,5165157	1,153638059	0,498615976	30724	195,4670618	0,782681273	0,976695743	0,903936374	0,928328835	0,007016729	0,003214562	0,861794425	0,994198849	SEKER
5	30140	620,134	201,8478822	190,2792788	1,06079802	0,333679658	30417	195,896503	0,773098035	0,99089325	0,984877069	0,970515523	0,00669701	0,003664972	0,941900381	0,999166059	SEKER
6	30279	634,927	212,5605564	181,5101816	1,171066849	0,52040066	30600	196,3477022	0,775688485	0,989509804	0,943851783	0,923725952	0,007020065	0,003152779	0,853269634	0,999235781	SEKER
7	30477	670,033	211,0501553	184,0390501	1,146768336	0,489477894	30970	196,9886332	0,762401501	0,984081369	0,853079869	0,933373552	0,006924899	0,003242016	0,871186188	0,999048736	SEKER
8	30519	629,727	212,9967551	182,7372038	1,165590535	0,513759558	30847	197,1243203	0,770681818	0,989366875	0,967109244	0,925480392	0,006979152	0,003158285	0,856513956	0,99834456	SEKER
9	30685	635,681	213,5341452	183,1571463	1,163852108	0,51408086	31044	197,659696	0,771561479	0,988435769	0,954239808	0,925658498	0,00695891	0,00315155	0,856843654	0,998952981	SEKER
10	30834	631,934	217,2278128	180,8974686	1,2008339	0,553642225	31120	198,1390121	0,783682806	0,990809769	0,97027823	0,91212543	0,007045074	0,00300804	0,8319728	0,999061142	SEKER
11	30917	640,765	213,5600894	184,4398709	1,157884618	0,504102365	31280	198,4055115	0,770805285	0,988395141	0,94625818	0,929038343	0,006907529	0,00317422	0,863112242	0,999384389	SEKER
12	31091	638,558	210,4862549	188,3268476	1,117664622	0,446621924	31458	198,9630385	0,786377317	0,988333651	0,958172836	0,945254305	0,006770006	0,003333984	0,8935057	0,998639711	SEKER
13	31107	640,594	214,6485485	184,9692526	1,160455295	0,507365875	31423	199,0142269	0,761046142	0,989943672	0,952581757	0,927163162	0,006900329	0,003145388	0,859631529	0,997563959	SEKER
14	31158	642,626	216,4848362	183,6443122	1,178826797	0,529514251	31492	199,1773023	0,798759229	0,989394132	0,948119004	0,92005198	0,00694797	0,003071052	0,846495646	0,997871751	SEKER
15	31158	641,105	212,0669751	187,1929601	1,132879009	0,469924157	31474	199,1773023	0,781313473	0,989959967	0,952623101	0,939218858	0,006806181	0,00326701	0,882132064	0,999348898	SEKER
16	31178	636,888	212,9759252	186,5620882	1,141582018	0,482352224	31520	199,2412169	0,764110482	0,989149746	0,965899596	0,935510513	0,006830968	0,003227429	0,875179919	0,999089658	SEKER
17	31202	644,454	215,6406947	184,4716842	1,168963657	0,517871223	31573	199,3178875	0,779192888	0,988249454	0,944079243	0,924305534	0,006911118	0,003111647	0,85434072	0,998693253	SEKER
18	31203	639,782	215,067737	184,8748759	1,163315112	0,510946829	31558	199,3210815	0,762984155	0,988750871	0,957948542	0,926782809	0,006892534	0,003136682	0,858926376	0,999202033	SEKER
19	31272	638,666	212,4503189	187,535939	1,132851229	0,469883494	31593	199,5413417	0,770322199	0,989839521	0,963425036	0,939237666	0,006793627	0,003261246	0,882167393	0,999364415	SEKER
20	31335	635,011	216,7900923	184,1634403	1,177161395	0,52758671	31599	199,7422367	0,774277242	0,991645305	0,976510834	0,921362386	0,006918465	0,003075469	0,848908647	0,999302487	SEKER
21	31374	636,401	219,865394	182,0088637	1,207992784	0,56099452	31604	199,8664991	0,769196823	0,99272244	0,973459862	0,909040279	0,007007885	0,002951884	0,826354229	0,998229946	SEKER
22	31530	638,857	213,7856543	188,0664823	1,136755746	0,475535642	31791	200,3627781	0,768949371	0,991790129	0,970792739	0,937213391	0,006780389	0,003226921	0,878368941	0,99849094	SEKER
23	31573	674,103	217,3070261	185,4482507	1,171793345	0,52126839	32197	200,499357	0,756964757	0,980619312	0,873118507	0,922654737	0,006882685	0,003076767	0,851291764	0,997538024	SEKER
24	31637	656,711	229,7192546	175,510446	1,308863717	0,645190802	32045	200,702465	0,761823348	0,987267905	0,921842182	0,873685862	0,007261095	0,002609775	0,763326986	0,999090996	SEKER
25	31675	657,431	236,7526321	171,2105592	1,3828156	0,690678219	32009	200,8229633	0,740935673	0,989565435	0,92092896	0,848239622	0,007474432	0,002386889	0,719510457	0,994950284	SEKER
26	31682	646,721	210,0456816	192,2484159	1,092574316	0,402841974	32026	200,8451524	0,773184303	0,989258727	0,951893863	0,956197485	0,006629811	0,003418781	0,91431363	0,998955589	SEKER
27	31703	656,305	215,7089067	187,2724497	1,151845384	0,496263281	32093	200,9117052	0,777110501	0,987847817	0,92490856	0,931401991	0,006804053	0,003158611	0,867509669	0,999237008	SEKER
28	31748	641,826	219,7765183	184,1151053	1,193690859	0,546072628	32020	201,0542441	0,775589974	0,991505309	0,968482156	0,914812218	0,006922531	0,002990698	0,836881394	0,99897767	SEKER
29	31768	650,954	220,9594949	183,2920681	1,205504947	0,558465259	32173	201,1175623	0,777674419	0,987411805	0,94210463	0,910201041	0,006955411	0,002944773	0,828465935	0,998719784	SEKER
30	31811	642,092	223,9846829	180,9171228	1,238051321	0,589564969	32052	201,2536289	0,773877293	0,992480968	0,969600136	0,898515141	0,007041108	0,002830886	0,807329459	0,999515303	SEKER

Hình 3.2.Bộ dữ liệu Dry Bean

- **Tiền xử lý dữ liệu**

Làm sạch dữ liệu là quá trình loại bỏ các sai sót, lỗi, nhiễu và thông tin không chính xác hoặc không cần thiết khỏi tập dữ liệu ban đầu để đảm bảo dữ liệu đáng tin cậy và phù hợp cho các hoạt động phân tích, huấn luyện mô hình,… Đối với bài thực nghiệm hiện tại, sau khi khảo sát chi tiết các cột dữ liệu, việc sửa lỗi sai sót và chọn lọc đặc trưng cho bộ dữ liệu không quá quan trọng nên ta sẽ chỉ tiến hành loại bỏ dữ liệu trùng lặp và xử lý dữ liệu thiếu hụt.

Qua khảo sát có thể thấy bộ dữ liệu có 68 bản ghi trùng lặp, vì vậy cần tiến thành loại bỏ các bản ghi trùng lặp và thu được bộ dữ liệu gồm 13543 bản ghi không trùng lặp và không bị khuyến. Tuy nhiên, để phù hợp với các thuật toán phân cụm trong thực nghiệm này ta sẽ tiến hành loại bỏ đi cột “Class” trong bộ dữ liệu.

3.2.2. Thực thi và đánh giá

Kết quả chạy bộ dữ liệu Dry Bean qua các thuật toán với tham số đầu vào m=2, c=3, a=1 và b=1:

Algo	Iters	Time	DB	PC	DI
KMEANS	18	0.02000	1.18188	1.00000	0.13070
FCM	176	2.55500	0.50025	0.68700	0.00017
PCM	483	7.18818	4.13952	0.34806	0.00006
PFCM	133	10.24688	0.49991	0.47928	0.00006

Hình 3.3. Kết quả chạy bộ dữ liệu Dry Bean qua các thuật toán

- **Số vòng lặp (Iters) và thời gian thực thi (Time)**

Thuật toán K-means cho thấy khả năng hội tụ nhanh vượt trội với chỉ 18 vòng lặp và thời gian thực thi 0.02 giây, thấp nhất trong tất cả các phương pháp. Điều này phù hợp với bản chất đơn giản của K-means, vốn không yêu cầu tính toán ma trận độ thuộc hay độ điển hình. Trong khi đó, FCM và PFCM yêu cầu số vòng lặp lần lượt là 176 và 133, và thời gian thực thi tăng đáng kể (2.55500 và 10.24688 giây), phản ánh chi phí tính toán liên quan đến việc cập nhật giá trị mờ. Đặc biệt, PCM cần tới 483 vòng lặp và mất 7.18818 giây, cho thấy sự phức tạp trong việc điều chỉnh mức độ điển hình nhằm kháng nhiễu và xử lý dữ liệu không rõ ràng.

- **Davies–Bouldin (DB)**

Chỉ số DB đánh giá mức độ tách biệt giữa các cụm, với giá trị càng thấp cho thấy phân cụm càng rõ ràng và chính xác. Kết quả cho thấy PFCM (0.49991) và FCM (0.50025) đạt hiệu quả phân cụm tốt nhất theo tiêu chí này, sát nhau một cách đáng chú ý. K-means tuy nhanh nhưng có DB cao hơn đáng kể (1.18188), cho thấy các cụm tạo ra có phần chồng lấn hoặc thiếu rõ ràng. Đặc biệt, PCM có DB lên tới 4.13952, cho thấy việc dựa vào điển hình tính mà bỏ qua tính mờ dẫn đến sự phân tán cao trong cụm.

- **Hệ số phân cụm (Partition Coefficient - PC)**

Hệ số PC phản ánh mức độ "sắc nét" của sự gán nhãn phân cụm, với giá trị gần 1 thể hiện mỗi điểm dữ liệu chỉ rõ ràng thuộc về một cụm. Không bất ngờ khi K-means đạt PC tuyệt đối là 1.00000 do bản chất rạch ròi (hard clustering) của nó. FCM và PFCM có PC lần lượt là 0.68700 và 0.47928, phản ánh bản chất phân cụm mờ - cho phép một điểm dữ liệu thuộc về nhiều cụm với các mức độ khác nhau. PCM có PC thấp nhất (0.34806), điều này là hợp lý vì PCM nhấn mạnh vào mức độ điển hình, không yêu cầu các giá trị độ thuộc tuân theo tính chuẩn hóa.

- **Chỉ số Dunn (DI)**

Chỉ số Dunn được sử dụng để đánh giá mức độ phân tách giữa các cụm, trong đó giá trị cao hơn tương ứng với phân cụm tốt hơn. Trong khi K-means đạt giá trị DI là 0.13070, cao hơn nhiều so với cả FCM (0.00017), PCM (0.00006) và PFCM (0.00006). Điều này một lần nữa phản ánh rằng các phương pháp fuzzy và possibilistic, dù hiệu quả hơn trong việc phản ánh tính chất không chắc chắn, lại có xu hướng tạo ra các biên cụm chồng lấn nhiều hơn, dẫn đến chỉ số Dunn thấp.

- **Tổng kết và nhận xét**

Từ các kết quả thu được, có thể rút ra một số kết luận như sau:

- *K-means*, thuật toán hiệu quả nhất về mặt thời gian và tốc độ hội tụ, tuy nhiên lại có độ phân cụm thấp hơn khi xét trên các chỉ số đánh giá phân tách cụm như DB và DI. Nó phù hợp trong các tình huống cần tốc độ nhanh và dữ liệu có cấu trúc rõ ràng.

- *FCM*, cung cấp sự cân bằng hợp lý giữa tính mờ và hiệu quả phân cụm, đạt DB thấp và PC cao, cho thấy khả năng mô hình hóa dữ liệu không chắc chắn khá tốt.
- *PCM*, thể hiện rõ đặc tính kháng nhiễu nhưng lại bị suy giảm đáng kể hiệu quả phân cụm theo DB và DI. Điều này lý giải cho số vòng lặp lớn và thời gian xử lý cao, cùng với giá trị PC thấp.
- *PFCM*, như một sự kết hợp giữa FCM và PCM, cho thấy khả năng tổng hợp ưu điểm của cả hai: DB thấp gần bằng FCM, và PC cao hơn PCM, mặc dù thời gian xử lý vẫn còn cao.

Tóm lại, nếu mục tiêu là tốc độ và hiệu quả tính toán, K-means vẫn là lựa chọn hàng đầu. Ngược lại, khi cần xử lý các tập dữ liệu có tính không chắc chắn cao và yêu cầu độ chính xác trong biên cụm, FCM hoặc PFCM là những lựa chọn ưu việt hơn. Trong trường hợp dữ liệu có nhiễu mạnh hoặc cần khả năng mô hình hóa không chắc chắn sâu hơn, PCM có thể được xem xét với điều kiện tài nguyên xử lý cho phép.

3.3. Thực nghiệm trên dữ liệu ảnh viễn thám

3.3.1. Khái niệm [3]

Ảnh viễn thám là kết quả thu được từ các bộ cảm biến (sensor) đặt trên thiết bị thu. Nó là những bức ảnh số thể hiện các sự vật, hiện tượng trên bề mặt Trái Đất. Dữ liệu nhận từ vệ tinh cần truyền tải đến Trái Đất từ vệ tinh hoạt động quỹ đạo suốt thời gian sống của nó. Những dữ liệu ảnh viễn thám sẽ được các cơ quan có thẩm quyền quản lý và lưu trữ. Từ đó giúp cho việc khai thác và sử dụng dữ liệu hiệu quả nhất và đảm bảo theo đúng quy định của pháp luật.

Ảnh viễn thám được ứng dụng trong nhiều lĩnh vực của cuộc sống hiện đại như địa lý, khả sát đất đai, giám sát biến đổi khí hậu, cùng các ngành khoa học liên quan đến trái đất…. Ngoài ra, trong quân sự, tình báo, thương mại, kinh tế… người ta cũng sử dụng ảnh viễn thám để theo dõi các biến động và đình hướng được sự phát triển trong thời gian tới. Để có được những bức ảnh viễn thám giá trị, người ta sẽ sử dụng đến vệ tinh nhân tạo có gắn các bộ cảm biến hoặc máy bay không người lái như máy bay UAV (Unmanned Aerial Vehicle) có ứng dụng thu phát tín hiệu từ hệ thống định vị toàn cầu GNSS, các thiết bị GNSS RTK để thu ảnh viễn thám.

3.3.2. Các phổ ảnh viễn thám

Phổ ảnh viễn thám là các lớp dữ liệu hình ảnh thu được từ các dải bước sóng khác nhau của bức xạ điện từ, được cảm biến gắn trên vệ tinh hoặc máy bay ghi lại. Mỗi dải phổ thể hiện thông tin vật lý khác nhau của bề mặt Trái đất như thực vật, nước, đất, đô thị... Các phổ ảnh giúp bổ sung cho nhau nhằm cung cấp cái nhìn toàn diện về đặc điểm địa hình, vật thể và môi trường.

Bảng 3.1. Bảng các phổ ảnh thường sử dụng trong ảnh viễn thám

Dải phổ	**Khoảng bước sóng**	**Nguồn thu phổ biến**	**Ứng dụng trong học máy**
Băng xanh dương (Blue band)	~450 – 520 nm	Vệ tinh Landsat, Sentinel-2,WorldView…	Đầu vào cho mô hình phân loại, phân cụm, hỗ trợ tách nước và vật thể nhân tạo
Băng xanh lục (Green band)	~520 – 600 nm	Vệ tinh Landsat (B2), Sentinel-2 (B2), WorldView	NDWI (phát hiện nước), phân loại cây trồng, theo dõi thực vật
Băng đỏ (Red band)	~630 – 690 nm	Các vệ tinh quang học	Cải thiện khả năng phân biệt đối tượng khi kết hợp đa phổ
Băng cận hồng ngoại (NIR - Near Infrared)	~760 – 900 nm	Vệ tinh Sentinel, Landsat	Thành phần dữ liệu giúp nhận diện thực vật, hỗ trợ phân cụm chính xác hơn
Băng hồng ngoại trung bình (SWIR – Short Wave Infrared)	~1550 – 1750 nm hoặc ~2080 – 2350 nm	Vệ tinh Sentinel, Landsat	Phân biệt rõ thực vật, đất, nước, tăng hiệu quả học của mô hình

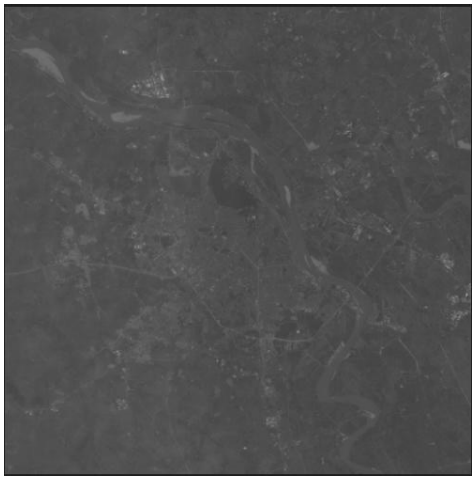
Các phổ ảnh trên được lấy từ dữ liệu vệ tinh (Landsat, Sentinel…), đóng vai trò là các kênh đặc trưng trong bài toán học máy trên ảnh viễn thám, giúp mô hình phân loại hoặc phân cụm đạt độ chính xác cao hơn.

3.3.3. Lý do sử dụng phân cụm mờ cho phân cụm ảnh viễn thám

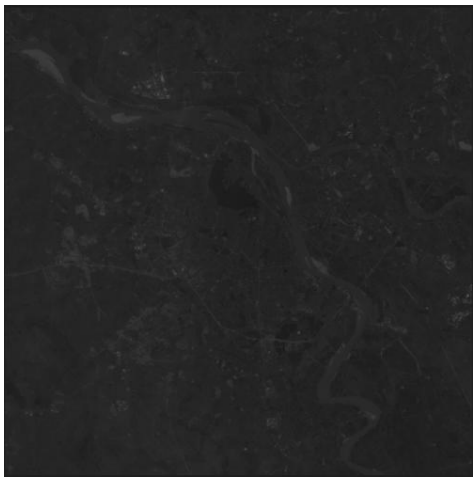
Trong ảnh viễn thám, do hạn chế về độ phân giải không gian hoặc do đặc điểm bề mặt Trái đất phức tạp, nhiều điểm ảnh thường chứa thông tin hỗn hợp từ nhiều loại đối tượng khác nhau như thực vật, đất trống, mặt nước hoặc khu dân cư. Hiện tượng này được gọi là "hiệu ứng trộn lẫn phổ", khiến việc phân cụm chính xác gặp nhiều khó khăn. Bên cạnh đó, ranh giới giữa các vùng đối tượng trong ảnh viễn thám thường không rõ ràng, có tính chuyển tiếp dần dần. Điều này khiến cho việc áp dụng các phương pháp phân cụm cứng (mỗi điểm ảnh chỉ được gán cho một cụm duy nhất) không phản ánh đúng bản chất thực tế của dữ liệu.

Phân cụm mờ là giải pháp phù hợp cho bài toán này. Phương pháp này cho phép mỗi điểm ảnh có thể thuộc về nhiều cụm với các mức độ khác nhau, thông qua giá trị thành viên. Nhờ đó, những vùng có sự chuyển tiếp hoặc trộn lẫn sẽ được mô hình hóa chính xác hơn. Đặc biệt, trong các bài toán học máy xử lý ảnh viễn thám, phân cụm mờ giúp tăng khả năng thích ứng của mô hình, hạn chế sai số ở các vùng biên, đồng thời phản ánh rõ ràng các đặc điểm không chắc chắn của dữ liệu. Đây là lý do việc sử dụng các thuật toán phân cụm mờ như Fuzzy C-Means (FCM) hoặc Possibilistic Fuzzy C-Means (PFCM) được ưu tiên khi xử lý, phân tích, phân cụm tự động ảnh viễn thám.

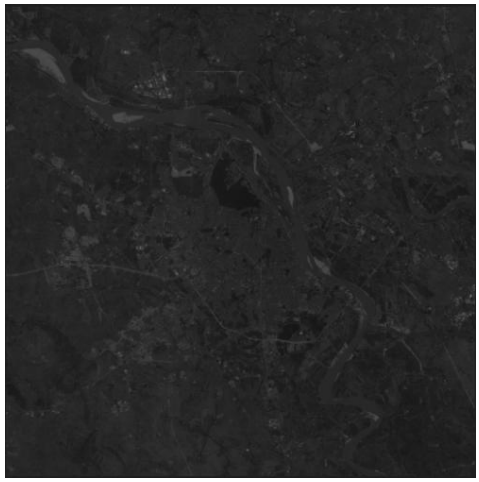
3.3.4. Thực nghiệm phân cụm ảnh viễn thám với thuật toán PFCM



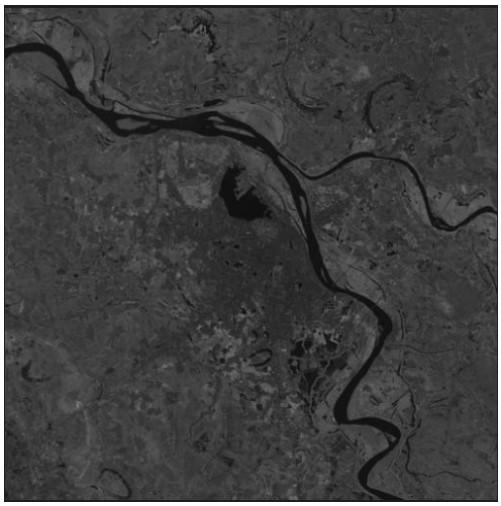
Hình 3.4. Phổ xanh lam (Blue band)



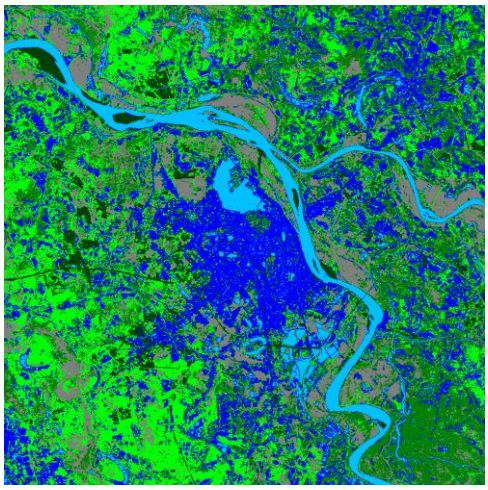
Hình 3.5. Phổ xanh lục (Green band)



Hình 3.6. Phổ Đỏ (Red Band)



Hình 3.7. Phổ cận hồng ngoại (NIR)



Hình 3.8. Ảnh kết quả phân cụm bằng PFCM

- **Các bước thực nghiệm phân cụm ảnh với PFCM:**

1. Đọc và chuẩn hóa ảnh: Đọc 4 ảnh viễn thám (Blue, Green, Red, NIR), chuẩn hóa và tạo ảnh đa phổ.
2. Tạo dữ liệu đầu vào: Ghép các ảnh thành ma trận dữ liệu có dạng (số điểm ảnh, 4 kênh phổ).
3. Phân cụm FCM và PCM: Chạy FCM để lấy membership ***U***, sau đó dùng ***U*** để khởi tạo và huấn luyện PCM (ra ***T*** và ***V***).
4. Phân cụm bằng PFCM: Dùng ***U***, ***T***, ***V*** từ FCM và PCM để khởi tạo và huấn luyện PFCM.
5. Sinh ảnh kết quả: Gán nhãn, tô màu cụm và lưu ảnh đầu ra *anhvientham.png*.

- **Nhận xét:**

- Thông qua kết quả phân cụm này, ta có thể theo dõi được mức độ thay đổi của lượng nước trên sông hoặc độ phủ của rừng, qua đó có những chiến lược quản lý tài nguyên bền vững.
- Cụm vùng nước nổi bật, hiển thị rõ ràng (xanh lam), bám sát hình dạng sông, hồ, kênh rạch.
- Các vùng còn lại được phân cụm thành nhiều mảng khác biệt, thể hiện rõ sự phân hóa phổ, có thể là các loại đất, bề mặt hoặc lớp phủ khác nhau.

- Ranh giới cụm rõ, không nhiễu, đặc biệt ở vùng giáp nước.
- Phân bố cụm đồng đều, không có cụm rỗng hay quá chênh lệch.

3.3.5. Kết luận

Kết quả phân cụm ảnh viễn thám bằng thuật toán PFCM cho thấy khả năng phân tách rõ ràng giữa các vùng có đặc trưng phổ khác nhau. Vùng nước được nhận diện rất rõ với màu xanh lam, bám sát theo hình dạng của sông và kênh rạch, cho thấy thuật toán hoạt động hiệu quả trong việc phát hiện các khu vực có phản xạ thấp. Các vùng còn lại được phân chia thành nhiều cụm khác biệt, phản ánh sự đa dạng của các loại bề mặt như đất hoặc lớp phủ khác nhau. Biên phân cụm giữa các vùng nhìn chung rõ ràng, không có hiện tượng nhiễu hoặc sai lệch lớn. Các cụm được phân bố hợp lý, không có cụm rỗng hay bị chiếm ưu thế quá mức. Tuy chưa gán nhãn cụ thể cho từng vùng nhưng kết quả vẫn cho thấy PFCM là một phương pháp hiệu quả trong việc phân tích và tách lớp phủ mặt đất dựa trên thông tin phổ từ ảnh viễn thám.

3.4. Mở rộng thực nghiệm

Thay đổi tham số: theo dõi sự thay đổi kết quả phân cụm khi thay đổi giá trị của các tham số. Bộ dữ liệu được sử dụng là bộ Iris và 2 tham số thay đổi sẽ là hệ số mờ (m), số cụm (c).

- **Thay đổi hệ số m (số tâm cụm cố định c=3 - số cụm chuẩn của bộ Iris):**

- Với m=2 (hệ số mờ chuẩn thường được sử dụng)

	Iters	**Time**	**DB**	**PC**	**DI**
fcm	23	0.00388	0.66925	0.6751	0.08855
pcm	151	0.01556	0.38275	-0.17036	0.03553
pfcm	19	0.02525	0.67297	0.67266	0.05658

- Với m=3

	Iters	**Time**	**DB**	**PC**	**DI**
fcm	20	0.00313	0.67297	0.34045	0.04643
pcm	34	0.00666	0.38275	0.0552	0.1509
pfcm	18	0.01511	0.68557	0.33088	0.03179

- Với m=4

	Iters	**Time**	**DB**	**PC**	**DI**
fcm	28	0.00638	0.67672	0.18056	0.04241
pcm	38	0.00928	1.93716	0.29294	0.0272
pfcm	98	0.03215	0.69662	0.15857	0.03232

Nhận xét:

Chỉ số DB với các thuật toán như FCM, PFCM không biến động quá nhiều cho thấy các cụm đã được phân tương đối chặt chẽ kể cả khi mức độ mờ của thuật toán được tăng lên một giá trị khá lớn. Riêng về pcm cần phải được nói riêng do với m=2 và m=3 chỉ số DB được coi là rất tốt vì chúng khá gần với 0, thể hiện sự tách biệt giữa các cụm cao nhưng khi tăng m=4, chỉ số DB đã nhảy vọt lên quá 1 điều này chứng tỏ việc có các cụm bị trùng nhau đã xảy ra, đây cũng chính là một trong các điểm yếu chí mạng của PCM khi không có một ràng buộc nào về ma trận khả năng $\boldsymbol{T}$.

Chỉ số PC là chỉ số thể hiện mức độ mờ của kết quả đầu ra, tại m=2 chỉ số này khá cao, điều này chứng tỏ rằng mức độ mờ đang thấp, các điểm dữ liệu có xu hướng thuộc vào một cụm nào đó rõ ràng hơn. Còn khi tham số m tăng lên giá trị này lại dần nhỏ đi, điều này là đúng với lý thuyết (khi m càng cao, mức độ mờ của thuật toán càng lớn).

DI càng cao càng tốt (ngược với DB). Ở đây DI nhìn chung nhỏ (<0.1), điều này cho thấy khoảng cách giữa cụm không lớn so với độ rộng cụm. FCM cho DI lớn nhất ở m=2 (0.08855) → cụm phân biệt tốt nhất ở giá trị m nhỏ. PCM có DI dao động thất thường, ở m=3 DI tăng (0.1509), nhưng ở m=4 lại giảm (0.0272) → một lần nữa cho thấy độ không ổn định. FCM giữ DI ổn định hơn, và cao nhất ở m=2, cho thấy phân cụm rõ nhất ở hệ số mờ nhỏ.

Về chi phí tính toán, có thể nhận xét rằng với hệ số m càng cao, việc hội tụ của thuật toán trở nên khó khăn hơn với 2 thuật toán FCM và PFCM do mức độ mờ làm cho tâm cụm di chuyển chậm sau mỗi lần lặp. Còn về PCM thì có xu hướng giảm đi, điều này dễ hiểu khi việc trùng lặp cụm sảy ra một số các cụm cùng tiến về phía nhau nên nhanh chóng đạt được sự hội tụ nhưng kết quả trả về lại không tốt.

- **Thay đổi tham số c (hệ số mờ cố định m=2 - hệ số mờ chuẩn)**

- Với c=2

	Iters	**Time**	**DB**	**PC**	**DI**
fcm	9	0.00204	0.40429	0.78443	0.16551
pcm	35	0.00466	0.38275	-0.58596	0.07245
pfcm	6	0.00479	0.40429	0.7846	0.16558

- Với c=3 (số cụm chuẩn của bộ Iris)

	Iters	**Time**	**DB**	**PC**	**DI**
fcm	23	0.0041	0.66925	0.6751	0.08855
pcm	151	0.01495	0.38275	-0.17036	0.03553
pfcm	19	0.01978	0.67297	0.67266	0.05658

- Với c=4

	Iters	**Time**	**DB**	**PC**	**DI**
fcm	40	0.00872	0.7814	0.60905	0.11676
pcm	128	0.02052	1.09167	-0.00178	0.03621
pfcm	28	0.02676	0.78031	0.60468	0.17575

Nhận xét:

DB đánh giá độ chồng lấn giữa các cụm, càng nhỏ càng tốt. Với c=2, DB 0.40 (FCM, PFCM) và 0.38 (PCM) chỉ số khá thấp, cho thấy cụm rõ, tách biệt tốt. Khi c=3, DB của FCM tăng lên 0.67, PFCM 0.67, PCM vẫn 0.38 riêng PCM giữ DB thấp nhưng như đã thấy PC của nó bất thường. Việc DB tăng ở FCM/PFCM phản ánh cụm chồng lấn hơn khi đúng số lớp (c=3). Ở c=4, DB tăng mạnh: FCM 0.78, PFCM 0.78, PCM lên đến 1.09 cho thấy cụm bị ép chia nhỏ gây chồng lấn nghiêm trọng, vì dữ liệu Iris vốn chỉ có 3 lớp thực. Như vậy, DB cho thấy chất lượng cụm giảm rõ rệt khi c tăng vượt số cụm chuẩn (DB tăng mạnh).

PC đánh giá độ rõ ràng của membership: càng gần 1 càng "rõ ràng" (gần hard-cluster), càng thấp/mờ thì PC giảm. Với c=2, PC rất cao: FCM 0.78, PFCM 0.78 phân tách rõ, membership gần như là phân cụm cứng. Nhưng PCM

cho PC âm (-0.58) không hợp lý điều này là do ma trận khả năng không có ràng buộc khiến cho khoảng giá trị của nó bị rơi vào bất cứ đâu trong khoảng [0, 1].

DI đo độ tách biệt giữa cụm: càng lớn càng tốt, cụm càng rõ. Với c=2, DI khá cao: FCM/PFCM ~ 0.1655, PCM thấp hơn 0.072, cụm phân biệt rõ ở FCM/PFCM. Khi c=3, DI giảm mạnh: FCM 0.0885, PFCM 0.0566, PCM 0.0355 các cụm khó tách biệt hơn khi phân đúng 3 lớp chuẩn hợp lý vì Iris có 3 lớp nhưng các lớp này chồng lấn một phần tự nhiên. Khi c=4, DI của FCM & PFCM tăng nhẹ lại (0.1167 và 0.1757) điều này thường do việc ép dữ liệu chia thành nhiều cụm hơn làm khoảng cách cụm ngắn hơn cục bộ (nhiều cụm nhỏ chen giữa) nhưng không phản ánh chất lượng phân cụm tốt (vì DB lại tăng). DI giảm mạnh, nhưng tăng nhẹ c=4 không hẳn là dấu hiệu tốt vì đồng thời DB tăng cao.

3.5. Kết luận chương 3

Thông qua ba bài toán thực nghiệm trên các dạng dữ liệu khác nhau là bộ dữ liệu Iris, bộ dữ liệu Dry Bean và ảnh viễn thám - chương 3 đã kiểm chứng khả năng áp dụng của các thuật toán phân cụm mờ vào thực tiễn. Các kết quả thực nghiệm cho thấy FCM, PCM và PFCM đều cho thấy tính linh hoạt và khả năng thích ứng tốt trên nhiều loại dữ liệu nhờ sự kết hợp ưu điểm của chúng. Những phân tích qua chỉ số DB, DI, PC cùng đánh giá định tính cũng cho thấy tầm quan trọng của những giá trị tham số đầu vào của thuật toán. Qua đó, chương 3 không chỉ xác nhận tiềm năng ứng dụng của phân cụm mờ mà còn mở ra hướng cải tiến và triển khai hiệu quả trong các bài toán dữ liệu thực tế.

KẾT LUẬN

Trong khuôn khổ đề tài "Tìm hiểu về thuật toán phân cụm mờ theo khả năng và ứng dụng vào thực tế", nhóm chúng em đã tập trung nghiên cứu một hướng tiếp cận quan trọng trong lĩnh vực học máy không giám sát – các thuật toán phân cụm mờ. Với mục tiêu xây dựng nền tảng lý thuyết vững chắc và kiểm nghiệm hiệu quả phân cụm trên dữ liệu thực tế, nhóm đã tìm hiểu ba thuật toán tiêu biểu: Fuzzy C-Means (FCM), Possibilistic C-Means (PCM) và thuật toán kết hợp PFCM.

Báo cáo đã được triển khai theo quy trình bài bản: từ tổng quan lý thuyết, phân tích sâu về đặc điểm và cơ chế hoạt động của từng thuật toán, đến thực nghiệm trên các bộ dữ liệu khác nhau như Iris, Dry Bean và ảnh viễn thám đa phổ. Nhờ đó, nhóm đã có cơ hội quan sát trực quan hiệu quả của từng thuật toán trong những điều kiện và đặc trưng dữ liệu khác nhau.

Mặc dù đạt được kết quả khả quan, nhóm cũng nhận thấy đề tài vẫn tồn tại một số hạn chế. Các thực nghiệm hiện tại chủ yếu tập trung vào đánh giá định lượng và trực quan, chưa mở rộng sang các bộ dữ liệu có tính động theo thời gian hoặc tính phân cụm phức tạp cao. Ngoài ra, việc lựa chọn trọng số trong mô hình PFCM vẫn đang dựa trên kinh nghiệm, chưa có một cơ chế tối ưu hóa tự động hoặc thích nghi theo đặc trưng dữ liệu.

Trong các bước tiếp theo, nhóm sẽ mở rộng phạm vi nghiên cứu bằng cách tích hợp thêm các kỹ thuật tối ưu hóa, nhằm cải thiện hiệu suất và độ ổn định của mô hình. Đồng thời, nhóm cũng dự kiến xây dựng một công cụ phân cụm trực quan, cho phép người dùng nhập dữ liệu và theo dõi kết quả phân cụm theo thời gian thực. Những định hướng này không chỉ giúp hoàn thiện mô hình lý thuyết mà còn tạo tiền đề ứng dụng trong các bài toán thực tế như phân tích ảnh vệ tinh, phân loại khách hàng, hay phát hiện bất thường trong dữ liệu cảm biến.

Với những kết quả đã đạt được, nhóm hy vọng rằng đề tài sẽ là nền tảng hữu ích cho các nghiên cứu và ứng dụng về phân cụm mờ trong tương lai, đồng thời góp phần thúc đẩy các hướng tiếp cận mới trong khai phá dữ liệu và trí tuệ nhân tạo.

PHỤ LỤC

- **Chứng minh công thức FCM**

Chứng minh công thức cập nhật u_{ij}

- Do tồn tại ràng buộc $\sum_{j=1}^{c} u_{ij} = 1$ ta có hàm Lagrange sau:
- $J_L = \sum_{i=1}^{n}\sum_{j=1}^{c} {u_{ij}}^m \left|x_i - v_j\right|_2^{\,2} - \sum_{i=1}^{N} \lambda_i \left(\sum_{j=1}^{c} u_{ij} - 1\right)$
- $Ta\ có$: $\frac{\partial J_L}{\partial u_{ij}} = m{u_{ij}}^{m-1}\left|x_i - v_j\right|_2^{\,2} - \lambda_i$
- Xét $\frac{\partial J_L}{\partial u_{ij}} = 0$ ta được:
- $m{u_{ij}}^{m-1}\left|x_i - v_j\right|_2^{\,2} - \lambda_i = 0$
- $\Leftrightarrow {u_{ij}}^{m-1} = \frac{\lambda_i}{m\left|x_i - v_j\right|_2^{\,2}}$
- $\Leftrightarrow u_{ij} = \left(\frac{\lambda_i}{m\left|x_i - v_j\right|_2^{\,2}}\right)^{\frac{1}{m-1}}$ (*)
- Lại có $\sum_{j=1}^{c} u_{ij} = 1$ từ đây ta có: $\sum_{k=1}^{c}\left(\frac{\lambda_i}{m\left|x_i - v_k\right|_2^{\,2}}\right)^{\frac{1}{m-1}} = 1$
- $\Leftrightarrow \left(\frac{\lambda_i}{m}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{k=1}^{c}\left(\frac{1}{\left|x_i - v_k\right|_2^{\,2}}\right)^{\frac{1}{m-1}}}$ (**)
- Thay ngược (**) vào (*) ta được:
- $u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\left|x_i - v_j\right|_2^{\,2}}{\left|x_i - v_k\right|_2^{\,2}}\right)^{\frac{1}{m-1}}} = \left(\sum_{j=1}^{c}\left(\frac{\left|x_i - v_j\right|_2}{\left|x_i - v_k\right|_2}\right)^{\frac{2}{m-1}}\right)^{-1}$
- $\Rightarrow u_{ij} = \left(\sum_{j=1}^{c}\left(\frac{\left|x_i - v_j\right|_2}{\left|x_i - v_k\right|_2}\right)^{\frac{2}{m-1}}\right)^{-1}$ Công thức số (2)

Chứng minh công thức cập nhật v_j

- $J = \sum_{i=1}^{n}\sum_{j=1}^{c} {u_{ij}}^m \left|x_i - v_j\right|_2^{\,2}$

- $Ta\ có: \frac{\partial J_L}{\partial v_j} = 2\sum_{i=1}^{n} {u_{ij}}^m (x_i - v_j)$
- Xét $\frac{\partial J_L}{\partial v_j} = 0$ ta được:
- $2\sum_{i=1}^{n} {u_{ij}}^m (x_i - v_j) = 0$
- $\Leftrightarrow\ v_j = \frac{\sum_{i=1}^{n} {u_{ij}}^m x_i}{\sum_{i=1}^{n} {u_{ij}}^m}$ Công thức số (3)

- **Chứng minh công thức PCM**

  Công thức cập nhật t_{ij}

  - $J = \sum_{i=1}^{n}\sum_{j=1}^{c} {t_{ij}}^m \left|x_i - v_j\right|_2^{\ 2} + \sum_{j=1}^{c} \gamma_j \sum_{i=1}^{n} (1 - t_{ij})^m$
  - $\frac{\partial J}{\partial t_{ij}} = m{t_{ij}}^{m-1} \left|x_i - v_j\right|_2^{\ 2} - m\gamma_j (1 - t_{ij})^{m-1}$
  - $Xét\ \frac{\partial J}{\partial t_{ij}} = 0\ ta\ được:$
  - $m{t_{ij}}^{m-1} \left|x_i - v_j\right|_2^{\ 2} - m\gamma_j (1 - t_{ij})^{m-1} = 0$
  - $\left(\frac{t_{ij}}{1-t_{ij}}\right)^{m-1} = \frac{\gamma_j}{\left|x_i - v_j\right|_2^{\ 2}}$
  - $\Leftrightarrow \left(\frac{1}{1-t_{ij}} - 1\right) = \left(\frac{\gamma_j}{\left|x_i - v_j\right|_2^{\ 2}}\right)^{\frac{1}{m-1}}$
  - $\Leftrightarrow 1 - t_{ij} = \frac{1}{1 + \left(\frac{\gamma_j}{\left|x_i - v_j\right|_2^{\ 2}}\right)^{\frac{1}{m-1}}}$
  - $\Leftrightarrow t_{ij} = 1 - \frac{1}{1 + \left(\frac{\gamma_j}{\left|x_i - v_j\right|_2^{\ 2}}\right)^{\frac{1}{m-1}}}$
  - $\Leftrightarrow t_{ij} = \frac{1}{\left(\frac{\left|x_i - v_j\right|_2^{\ 2}}{\gamma_j}\right)^{\frac{1}{m-1}} + 1}$
  - $\Leftrightarrow t_{ij} = \left(1 + \left(\frac{\left|x_i - v_j\right|_2^{\ 2}}{\gamma_i}\right)^{\frac{1}{m-1}}\right)^{-1}$ Công thức số (5)

Công thức cập nhật v_j

- $J = \sum_{i=1}^{n}\sum_{j=1}^{c} {t_{ij}}^{m}\left|x_i - v_j\right|_2^{\,2} + \sum_{j=1}^{c}\gamma_j\sum_{i=1}^{n}(1-t_{ij})^m$
- $Ta\ có$: $\frac{\partial J}{\partial v_j}$=$2\sum_{i=1}^{n} {t_{ij}}^{m}(x_i - v_j)$
- $Xét\ \frac{\partial J}{\partial v_j} = 0\ ta$ được:
- $2\sum_{i=1}^{n} {t_{ij}}^{m}(x_i - v_j) = 0$
- $\Leftrightarrow\ v_j = \frac{\sum_{i=1}^{n} {t_{ij}}^{m} x_i}{\sum_{i=1}^{n} {t_{ij}}^{m}}$ Công thức số (6)

- **Chứng minh công thức PFCM**

Công thức cập nhật u_{ij}

- Do tồn tại ràng buộc $\sum_{j=1}^{c} u_{ij} = 1$ ta có hàm Lagrange sau:
- $J = \sum_{i=1}^{n}\sum_{j=1}^{c}(a{u_{ij}}^{m} + b{t_{ij}}^{m})|x_i - v_j|_2^{\,2} + \sum_{j=1}^{c}\gamma_j\sum_{i=1}^{n}(1 - t_{ij})^m - \sum_{i=1}^{N}\lambda_i\left(\sum_{j=1}^{c} u_{ij} - 1\right)$
- $Ta\ có$: $\frac{\partial J_L}{\partial u_{ij}} = ma{u_{ij}}^{m-1}\left|x_i - v_j\right|_2^{\,2} - \lambda_i$
- Xét $\frac{\partial J_L}{\partial u_{ij}} = 0$ ta được:
- $ma{u_{ij}}^{m-1}\left|x_i - v_j\right|_2^{\,2} - \lambda_i = 0$
- $\Leftrightarrow\ {u_{ij}}^{m-1} = \frac{\lambda_i}{ma\left|x_i - v_j\right|_2^{\,2}}$
- $\Leftrightarrow\ u_{ij} = \left(\frac{\lambda_i}{ma\left|x_i - v_j\right|_2^{\,2}}\right)^{\frac{1}{m-1}}$ (*)
- Mà $\sum_{j=1}^{c} u_{ij} = 1$ từ đây ta có: $\sum_{k=1}^{c}\left(\frac{\lambda_i}{ma\left|x_i - v_k\right|_2^{\,2}}\right)^{\frac{1}{m-1}} = 1$
- $\Leftrightarrow\ \left(\frac{\lambda_i}{ma}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{k=1}^{c}\left(\frac{1}{\left|x_i - v_k\right|_2^{\,2}}\right)^{\frac{1}{m-1}}}$ (**)
- Thay ngược (**) vào (*) ta được:

- $u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{|x_i - v_j|_2^{\,2}}{|x_i - v_k|_2^{\,2}}\right)^{\frac{1}{m-1}}} = \left(\sum_{j=1}^{c}\left(\frac{|x_i - v_j|_2}{|x_i - v_k|_2}\right)^{\frac{2}{m-1}}\right)^{-1}$
- $\Rightarrow u_{ij} = \left(\sum_{j=1}^{c}\left(\frac{|x_i - v_j|_2}{|x_i - v_k|_2}\right)^{\frac{2}{m-1}}\right)^{-1}$ Công thức số (9)

Công thức cập nhật t_{ij}

- $J = \sum_{i=1}^{n}\sum_{j=1}^{c}\left(a{u_{ij}}^m + b{t_{ij}}^m\right)\left|x_i - v_j\right|_2^{\,2} + \sum_{j=1}^{c}\gamma_j\sum_{i=1}^{n}\left(1 - t_{ij}\right)^m$
- $\frac{\partial J}{\partial t_{ij}} = mb{t_{ij}}^{m-1}\left|x_i - v_j\right|_2^{\,2} - m\gamma_j(1 - t_{ij})^{m-1}$
- $Xét\ \frac{\partial J}{\partial t_{ij}} = 0\ ta\ được$:
- $mb{t_{ij}}^{m-1}\left|x_i - v_j\right|_2^{\,2} - m\gamma_j\left(1 - t_{ij}\right)^{m-1} = 0$
- $\left(\frac{t_{ij}}{1 - t_{ij}}\right)^{m-1} = \frac{\gamma_j}{b\left|x_i - v_j\right|_2^{\,2}}$
- $\Leftrightarrow \left(\frac{1}{1 - t_{ij}} - 1\right) = \left(\frac{\gamma_j}{b\left|x_i - v_j\right|_2^{\,2}}\right)^{\frac{1}{m-1}}$
- $\Leftrightarrow 1 - t_{ij} = \frac{1}{1 + \left(\frac{\gamma_j}{b\left|x_i - v_j\right|_2^{\,2}}\right)^{\frac{1}{m-1}}}$
- $\Leftrightarrow t_{ij} = 1 - \frac{1}{1 + \left(\frac{\gamma_j}{b\left|x_i - v_j\right|_2^{\,2}}\right)^{\frac{1}{m-1}}}$
- $\Leftrightarrow t_{ij} = \frac{1}{\left(\frac{b\left|x_i - v_j\right|_2^{\,2}}{\gamma_j}\right)^{\frac{1}{m-1}} + 1}$
- $\Leftrightarrow t_{ij} = \left(1 + \left(\frac{b|x_i - v_j|_2^{\,2}}{\gamma_i}\right)^{\frac{1}{m-1}}\right)^{-1}$ Công thức số (10)

Công thức cập nhật v_j

- $J = \sum_{i=1}^{n}\sum_{j=1}^{c}\left(au_{ij}{}^{m} + bt_{ij}{}^{m}\right)\left|x_i - v_j\right|_2^{\ 2} + \sum_{j=1}^{c}\gamma_j \sum_{i=1}^{n}\left(1 - t_{ij}\right)^{m}$
- $Ta\ có: \frac{\partial J}{\partial v_j} = 2\sum_{i=1}^{n}\left(au_{ij}{}^{m} + bt_{ij}{}^{m}\right)\left(x_i - v_j\right)$
- $Xét\ \frac{\partial J}{\partial v_j} = 0\ ta$ được:
- $2\sum_{i=1}^{n}\left(au_{ij}{}^{m} + bt_{ij}{}^{m}\right)\left(x_i - v_j\right) = 0$
- $\Leftrightarrow\ v_j = \frac{\sum_{i=1}^{n}\left(au_{ij}{}^{m} + bt_{ij}{}^{m}\right)x_i}{\sum_{i=1}^{n}\left(au_{ij}{}^{m} + bt_{ij}{}^{m}\right)}$ Công thức số (11)

- **Đường dẫn chứa mã nguồn của toàn bộ bài báo cáo:**

  https://github.com/TranThanh39/BTL-Hoc-May.git

TÀI LIỆU THAM KHẢO

[1]. Cluster analysis, https://en.wikipedia.org/wiki/Cluster_analysis, ngày truy cập gần nhất 01/07/2025.

[2]. Dry Bean Dataset, https://www.kaggle.com/datasets/muratkokludataset/dry-bean-dataset, ngày truy cập gần nhất 29/06/2025.

[3]. Ảnh viễn thám là gì?, https://viet-thanh.vn/tim-hieu-anh-vien-tham-la-gi/, ngày truy cập gần nhất 29/06/2025.

Các công thức trong bài báo cáo được tham khảo từ nguồn:

- *Bezdek, J.C., Ehrlich, R., & Full, W., "FCM: The Fuzzy c-Means Clustering Algorithm.", Computers & Geosciznces, Vol. 10, No. 2-3, pp. 191–203, 1984.*
- *Krishnapuram, R., & Keller, J.M., "A Possibilistic Approach to Clustering.", IEEE Transactions on Fuzzy Systems, Vol. 1, No. 2, pp. 98–110, May 1993.*
- *Pal, N.R., Pal, K., Keller, J.M., & Bezdek, J.C., "A Possibilistic Fuzzy c-Means Clustering Algorithm.", IEEE Transactions on Fuzzy Systems, Vol. 13, No. 4, pp. 517–530, August 2005.*