2.3.2.Thuật toánFuzzy C-means (FCM)

*2.3.2.1Giới thiệu thuật toán*

Nếu như K-means là thuật toán PCDL rõ thì FCM là thuật toán phân cụm mờ tương ứng, hai thuật toán này cùng sử dụng chung một chiến lược phân cụm dữ liệu. Thuật toán FCM đã được áp dụng thành công trong giải quyết một số lớn các bài toán PCDL như trong nhận dạng mẫu (nhận dạng vân tay, ảnh), xử lý ảnh (phân tách các cụm ảnh màu, cụm màu), y học (phân loại bệnh, phân loại triệu chứng), … Tuy nhiên, nhược điểm lớn nhất của thuật toán FCM là tập dữ liệu lớn, tập dữ liệu nhiều chiều, nhạy cảm với các nhiễu và phần tử ngoại lai trong dữ liệu, nghĩa là các trung tâm cụm có thể nằm xa so với trọng tâm thực của cụm.

Kỹ thuật này phân hoạch một tập n vectơ đối tượng dữ liệu  thành c các nhóm mờ dựa trên tính toán tối thiểu hóa hàm mục tiêu để đo chất lượng của phân hoạch và tìm trọng tâm cụm trong mỗi nhóm, sao cho chi phí hàm đo độ phi tương tự là nhỏ nhất. Một phân hoạch mờ vectơ điểm dữ liệu  là đặc trưng đầu vào được biểu diễn bởi ma trận sao cho điểm dữ liệu đã cho chỉ có thể thuộc về một số nhóm với bậc được xác định bởi mức độ thuộc giữa [0, 1].

Như vậy, ma trận U được sử dụng để mô tả cấu trúc cụm của X bằng cách giải thích  như bậc thành viên  với cụm i.

Cho là phân hoạch mờ C



Dunn định nghĩa hàm mục tiêu mờ như sau:



Bezdek khái quát hóa hàm mục tiêu mờ bằng cách đưa ra trọng số mũ , là số thực nào đó bất kỳ như sau :

 (2.1)

Trong đó :

 là nửa dưới vector mẫu dữ liệu tập con thực d chiều trong không gian vector .

 là trọng số mũ hay còn gọi là tham số mờ.

 là trọng tâm của cụm thứ i

là khoảng cách theo thước đo Euclide giữa mẫu dữ liệu  với trọng tâm cụm thứ i ,

là bậc hay độ thuộc của dữ liễu mẫu  với cụm thứ i

 là ma trận biểu diễn các giá trị tâm của cụm

Để thuận tiện, coi mảng đối tượng dữ liệu  là các cột trong ma trận đối tượng dữ liệu . Ma trận phân hoạch U được sử dụng để mô tả cấu trúc cụm trong dữ liệu .

Định nghĩa 2.1: Họ các tập mờ  trong không gian vũ trụ  đ­ược gọi là phân hoạch mờ của X nếu bậc của dữ liệu mẫu thỏa mãn điều kiện :

 (2.2)

Dễ nhận thấy:  tức là 

Như vậy mỗi phân hoạch mờ cũng có biểu diễn bằng một ma trận c hàng và n cột để biể diễn phân hoạch n đối tượng thành c cụm dữ liệu trong không gian  được viết gọn như sau :

 (2.3)

 là không gian của tất cả các ma trận thực cấp .

Tập  có thể là tập vô hạn, tức là ta không thể xây dựng được công thức tính số phương án phân hoạch 

Thông thường ta gọi bài toán phân cụm mờ là bài toán tìm các độ thuộc  nhằm tối thiểu hóa hàm mục tiêu (2.1) với các điều kiện sau :

Định lý 2.1: Nếu m và c là các tham số cố định, và Ik là một tập đ­ược định nghĩa nh­ư sau:

 (2.4)

Thì hàm mục tiêu (2.1) đạt min khi và chỉ khi:

 (2.5)

 (2.6)

Định lý này đã đ­ược Bezdek chứng minh là đúng nếu 

Một phân hoạch tối ưu, nghĩa là hàm mục tiêu (2.1) đạt giá trị tối thiểu, mà chủ yếu dựa trên đó độ tương tự giữa xk và trọng tâm cụm vi, điều này tương đương với hai điều kiện (2.5) và (2.6) phải thỏa mãn các ràng buộc.

*2.3.2.2. Thuật toán FCM*

Thuật toán FCM cung cấp một quá trình lặp qua lại giữa phương trình (2.5) và (2.6) để xấp xỉ cực tiểu hàm mục tiêu (2.1) dựa trên độ đo tương tự có trọng số giữa xk và trọng tâm cụm vi, sau mỗi vòng lặp, thuật toán tính toán và cập nhật các phần tử u trong ma trận phân hoạch U. Phép lặp sẽ dừng khi  trong đó  là chuẩn kết thúc nằm trong khoảng  trong khi k là các bước lặp. Thủ tục này hội tụ tới cực tiểu cục bộ hay điểm yên ngựa của . Thuật toán FCM tính toán ma trận phân hoạch U và kích thước của các cụm để thu được các mô hình mờ từ ma trận này. Các bước thực hiện của thuật toán FCM như sau:

**Input: Số cụm *c* và tham số mũ *m* cho hàm mục tiêu J, sai số **

**Output: c cụm dữ liệu sao cho hàm mục tiêu (2.1) đạt giá trị cực tiểu**

Begin

Bước 1. Khởi tạo

Nhập tham số *c (1<c<n), m(1<m<+∞),*

Khởi tạo ma trận 

Bước 2. Tính ma trận phân hoạch U và cập nhật lại trọng tâm cụm V

2.1. j=j+1

2.2. Tính ma trận phân hoạch mờ  theo công thức (2.5)

2.3 Cập nhật các trọng tâm cụm  theo công thức (2. 6) và 

Bước 3: Kiểm tra điều kiện dừng. Nếu  chuyển sang bước 4, ngược lại quay lại bước 2.

Bước 4. Đưa ra các cụm kết quả.

**Nhận xét***:*

Việc chọn các tham số cụm rất ảnh hưởng đến kết quả phân cụm, tham số này thường được chọn theo phương pháp ngẫu nhiên hoặc theo Heuristic.

Đối với  thì thuật toán FCM trở thành thuật toán rõ

Đối với  thì thuật toán FCM trở thành thuật toán phân cụm mờ với . Chưa có quy tắc nào nhằm chọn lựa tham số m đảm bảo cho phân cụm hiệu quả, thông thường chọn m=2.

Ta có thể tiến hành đánh giá việc lựa chọn số tâm cụm tối ưu:

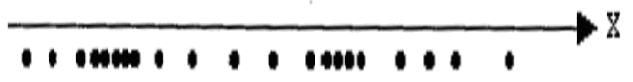


Trong đó: 

*2.3.2.3 So sánh FCM với K-means*

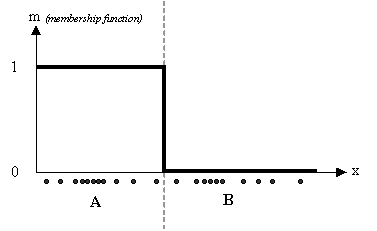
Để so sánh có thể xét ví dụ sau: Cho một tập các đối tượng dữ liệu một

chiều được biểu thị như hình 2.2 dưới đây.



Hình 2.2 Mô tả tập dữ liệu một chiều

Bằng quan sát dễ nhận thấy có hai cụm trong tập dữ liệu trên đặt tên tương ứng là "A" và "B". Với thuật toán K-means thì hàm tính độ phụ thuộc giữa đối tượng dữ liệu và trọng tâm cụm của nó được thể hiện như trong đồ thị hình 2.3 dưới đây:

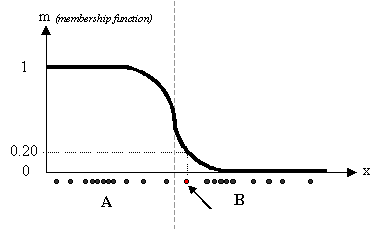


Hình 2.3 Hàm thuộc với trọng tâm của cụm A trong K-means

Dựa vào hình rút ra nhận xét rằng, các đối tượng trong cụm A có giá trị hàm thuộc với trọng tâm của cụm A là bằng 1 và bằng 0 với trọng tâm cụm B.

Điều này ngược lại với các đối tượng ở trong cụm B.

Đối với thuật toán FCM thì hàm thuộc của các đối tượng dữ liệu với các trọng tâm cụm dữ liệu được minh họa như trong đồ thị hình 2.4 dưới đây:

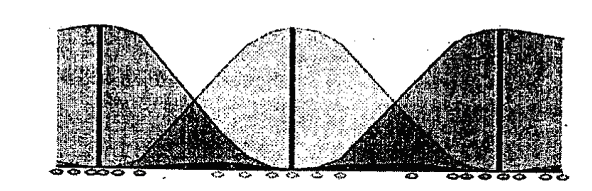


Hình 2.4 Hàm thuộc với trọng tâm cụm A trong FCM

Dựa vào hình có thể nhận xét rằng, các đối tượng dữ liệu có giá trị hàm thuộc với các trọng tâm của cụm A nằm trong khoảng [0, l], hàm thuộc lúc này là một đường cong trơn. Điểm có mũi tên chỉ đến có nhiều khả năng thuộc về lớp B hơn là lớp A do giá trị hàm thuộc của nó vào lớp A là nhỏ (=0.2). Có thể biểu diễn các giá trị hàm thuộc trên bằng ma trận cho cả hai trường hợp như sau:

 và 

Số dòng và số cột phụ thuộc vào số các đối tượng dữ liệu n và số các cụm k. Một số ví dụ mô phỏng về kết quả các cụm khám phá được của thuật toán phân cụm mờ FCM như hình 2.5 dưới đây.



Hình 2.5 Các cụm được khám phá bởi thuật toán FCM

Độ phức tạp của thuật toán FCM tương đương với độ phức tạp của thuật toán K-means trong trường hợp số đối tượng của tập dữ liệu cần phân cụm là rất lớn.

Tóm lại, thuật toán phân cụm mờ FCM là một mở rộng của thuật toán K-means nhằm để khám phá ra các cụm chồng lên nhau. Tuy nhiên, FCM vẫn chứa đựng các nhược điểm của thuật toán K-means trong việc xử lý đối với các phần tử ngoại lai và nhiễu trong dữ liệu. Thuật toán εFCM được trình bày phần sau là một mở rộng của thuật toán FCM nhằm khắc phục các nhược điểm này.

2.3.3.Thuật toán FCM cải tiến.

*2.3.3.1 Cơ sở thuật toán.*

Ta có thể tăng tốc độ tính toán của thuật toán FCM bằng cách giảm các phép toán thực hiện. Từ công thức (2.6) ta biết rằng mỗi tâm cụm vi được tính bằng trung bình của các mẫu dữ liệu trong cụm thứ i. Trong thuật toán FCM chuẩn vi được tính toán bằng cách duyệt qua toàn bộ tập dữ liệu và ma trận phân hoạch.

Tiếp theo đây chúng tôi mô tả thuật toán cải tiến mà ở đó việc tính các tâm cụm được thực hiện theo trình tự cập nhật ma trận phân hoạch. Hiệu quả của sự cải tiến nàyloại trừ một lần duyệt qua toàn bộ tập dữ liệu trong mỗi lần lặp. Kết quả là không giảm số lượng các vòng lặp được yêu cầu để hội tụ, nhưng giảm thời gian trong mỗi lần lặp.

Ta duy trì hai cấu trúc mở rộng, một ma trận , và một véc tơ *q* có chiều dài là *c*. Giá trị khởi tạo ban đầu của P và q được lấy từ tử số và mẫu số của công thức (2.6) khi ma trận thành viên phân hoạch được tạo ra. Lúc này, công thức (2.6) được viết lại như sau:

vi = pi / qi (2.7)

với pi là một véc tơ có chiều dài n, và qi là một giá trị vô hướng.

Mỗi lần một phần tử uki của ma trận thành viên được so sánh là một lần tử số của công thức (2.6) tăng lên, pi tăng lên một lượng:

 (2.8)

và mẫu số của công thức một cũng tăng, qi tăng lên một lượng:

 (2.9)

Những gia tăng này được tích lũy vào P và q theo thứ tự ma trận phân hoạch được cập nhật. Bắt đầu vòng lặp tiếp theo các tâm chùm mới được tính lại theo công thức (2.7).

*2.3.3.2 Thuật toán FCM cải tiến*

Thuật toán FCM cải tiến thực hiện các bước như sau:

Input: Số cụm *c* và tham số mũ *m* cho hàm mục tiêu J, sai số 

Output: c cụm dữ liệu sao cho hàm mục tiêu (2.1) đạt giá trị cực tiểu

Begin

Bước 1. Khởi tạo

Nhập tham số *c (1<c<n), m(1<m<+∞),*

Khởi tạo ma trận 

Khởi tạo các cấu trúc dữ liệu P và q sử dụng công thức (2.6).

Bước 2. Tính ma trận phân hoạch U và cập nhật lại trọng tâm cụm V

2.1. j=j+1

2.2. Tính ma trận phân hoạch mờ  theo công thức (2.5) và tăng các phần tử tương ứng trong P và q dùng công thức (2.8) và (2.9)

2.3 Cập nhật các trọng tâm cụm  theo công thức (2.7)

Bước 3: Kiểm tra điều kiện dừng. Nếu  chuyển sang bước 4, ngược lại quay lại bước 2.

Bước 4. Đưa ra các cụm kết quả.

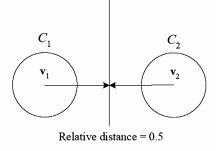
CHƯƠNG 3 - KỸ THUẬT PHÂN CỤM DỮ LIỆU MỜ LOẠI 2

## 3.1. Thuật toán phân cụm mờ loại hai khoảng (IT2FCM)

### 3.1.1 Cơ sở thuật toán IT2FCM.

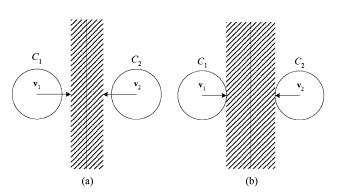
Nếu như thuật toán FCM là một giải thuật lặp gắn các giá trị hàm thuộc mờ cho dữ liệu mẫu, sau đó cập nhật các tham số trọng tâm cụm theo các giá trị hàm thuộc đã gán. Các giá trị hàm thuộc này đóng vai trò như là các giá trị trọng số, chúng thể hiện mức độ ảnh hưởng của một dữ liệu mẫu với các tham số cụm.

Trong thuật toán phân cum FCM ta thấy sẽ xuất hiện các vùng độ thuộc mờ cực đại (The maximally fuzzy membership locations), ở đó các mẫu dữ liệu ở giữa hai trọng tâm cụm, khoảng cách tương đối giữa một mẫu và mỗi tâm cụm bằng 0.5. Như hình 3.1 dưới đây, cùng độ thuộc mờ cực đại chính là đường thẳng đứng ở giữa.



Hình 3.1 Mô tả vị trí độ thuộc mờ cực đại

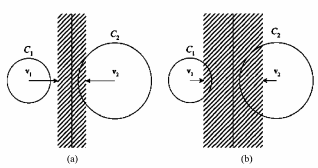
Ta thấy đường thẳng đứng trong hình 3.1 đóng vai trò như là ranh giới xác định những mẫu dữ liệu nào ở bên trái (phải) ranh giới sẽ thuộc về cụm C1(C2). Ranh giới này có thể mở rộng bởi tham số mờ *m* như hình 3.2 dưới đây. Nếu tham số mờ tăng thì ranh giới độ mờ cực đại sẽ mở rộng hơn. Độ rộng của vùng ranh giới chỉ phạm vi mong muốn gán độ thuộc mờ cực đại cho các mẫu.



Hình 3.2 Vùng độ thuộc mờ cực đại cho hai cụm cùng kích thước với tham số mờ m. (a). Tham số m nhỏ (b). Tham số m lớn

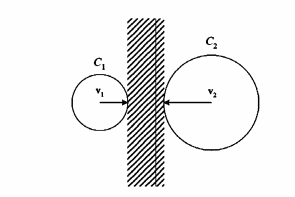
Ta nhận thấy rằng hầu như không có sự thay đổi kết quả phân cụm bất kể ta hay đổi tham số mờ m. Chính vì vậy, thuật toán FCM thực hiện khá tốt trên các mẫu dữ liệu chứa các cụm có cùng thể khối hình cầu và mật độ. Tuy nhiên, nếu các cụm trong tập dữ liệu mẫu khác nhau về kích thước cụm hoặc tỷ lệ số mẫu thì kết quả thực hiện của FCM có thể thay đổi đáng kể tùy thuộc vào tham số mờ m. Do vậy, việc lựa chọn m tốt lien quan đến sự phân bố của các dữ liệu mẫu trong tập mẫu.

Trong hình 3.3 dưới đây, vùng độ thuộc mờ cực đại có thể gây ra các kết quả phân cụm không mong muốn nếu tham số mờ m không được xác định đúng vì sự khác nhau về kích thước giữa hai cụm.



Hình 3.3 Vùng độ thuộc mờ cực đại cho hai cụm có kích thước khác nhau với tham số m.(a) Tham số m nhỏ (b) Tham số m lớn

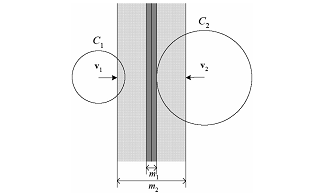
Trong hình 3.3 (a) ta thấy các mẫu dữ liệu bên trái vùng độ mờ cực đại là một phần của cụm C2 song trực quan dường như nó có ảnh hưởng đến cụm C1 hơn là cụm C2 do vậy cần phải điều chình tham số m để đạt kết quả chính xác hơn như hình 3.3 (b), tuy nhiên tâm v1 của cụm C1 có xu hướng di chuyển đến cụm C2, nó sẽ bị chệch đi so với tâm cụm lý tưởng, tương tự ta có thể có tâm cụm không mong muốn với cụm C2. Do vậy, trong trường hợp này ta cần có một vùng độ mờ cực đại lý tưởng với miền bên trái rộng và miền bên phải hẹp như hình 3.4 dưới đây.



Hình 3.4. Vùng độ thuộc mờ cực đại mong muốn

Ta thấy rằng vùng độ thuộc mờ mong muốn như trong hình 3.7 không thể thực hiện bởi thuật thuật toán FCM vì sự thay đổi tham số m ảnh hưởng đến các cụm là bằng nhau. Do đó thuật toán FCM thực hiện không tốt khi kích thước, hay mật độ của các cụm không tương đồng nhau.

Vì vậy, Hwang và Rhee đã đề xuất thuật toán IT2FCM sử dụng IT2FS với hai tham số mờ  và  có thể giải quyết vấn đề trên để đạt kết quả phân cụm mong muốn như hình 3.5



Hình 3.5 Vùng độ thuộc mờ cực đại với hai tham số mờ m1 và m2 (m1­<m2)

### 3.1.2 Thuật toán IT2FCM

Một tập mờ loại hai khoảng được biểu diễn như sau:

 (3.1)

 (3.2)

Để định nghĩa hàm thuộc khoảng cho một mẫu ta định nghĩa các hàm thuộc khoảng trên dưới (3.2) sử dụng hai giá trị tham số mờ  và . Việc sử dụng các tham số mờ dẫn đến các hàm mục tiêu khác nhau được cực tiểu hóa là:

 (3.3)

Trong đó :

 là nửa dưới vector mẫu dữ liệu tập con thực d chiều trong không gian vector .

 là trọng số mũ hay còn gọi là tham số mờ.

là khoảng cách theo thước đo Euclide giữa mẫu dữ liệu  với trọng tâm cụm thứ i ,

 là trọng tâm của cụm thứ i

là bậc hay độ thuộc của dữ liễu mẫu  với cụm thứ i

 là ma trận biểu diễn các giá trị tâm của cụm

Với hàm thuộc chính được tính theo công thức:

 (3.4)

 (3.5)

Chúng ta sẽ tính các hàm thuộc chính cho một mẫu với hai tham số mờ  và  sau đó thực hiện hai thủ tục chính là cập nhật trọng tâm cụm và giải mờ để có kết quả phân cụm cuối cùng (trong 1 bước lặp).

Do mỗi dữ liệu có mẫu có hai độ thuộc  nên tâm cụm vj cũng sẽ có tâm cực đại  và tâm cực tiểu 

Trước tiên chúng ta sẽ trình bày thuật toán tìm tâm cụm:

Bước 1: Tính ma trận hàm thuộc , theo công thức (3.4)và (3.5)

Bước 2:

2.1 Cho tham số m bằng tùy ý: 

2.2 Tính trọng tâm  theo công thức *(2.6)* với 

2.3 Sắp xếp n mẫu trên d thuộc tính theo thứ tự tăng dần

Bước 3:

3.1 Tìm k để: 

3.2 Cập nhật lại :

+ Với  thì 

+ Với  thì 

Bước 4:

4.1 Tính  theo công thức (2.6)

4.2 So sánh  và 

- Nếu  thì 

- Ngược lại:

+ Gán 

+ Quay lại bước 3.

\* Trong trường hợp tìm tâm cực tiểu  thực hiện tương tự như trên chỉ khác ở bước 3.2:

+ Với  thì 

+ Với  thì 

Sau khi tìm được  và  để có thể sử dụng được ta phải tiến hành giải mờ cho tâm cụm, tính  bằng cách đơn giản:

 (3.6)

Và giải mờ tính giá trị hàm thuộc:

 (3.7)

Trong đó:

 với  nếu  sử dụng  để tính  và  trong trường hợp ngược lại.

 với  nếu  sử dụng  để tính  và  trong trường hợp ngược lại.

Sau đó, ta tiến hành giải mờ để xác định mẫu dữ liệu thuộc về một một cụm nào đó thong qua luật:

Nếu  với  và  thì ta thực hiện gán mẫu  thuộc cụm thứ *j* (3.8)

*Các bước thực hiện thuật toán IT2FCM:*

Bước 1. Khởi tạo

1.1 Số cụm c (1<c<n), hai tham số mờ , sai số 

1.2 Khởi tạo ma trận trọng tâm ban đầu *, j=0*

Bước 2 Tính ma trận phân hoạch mờ U và cập nhật trọng tâm cụm V

2.1. j=j+1

2.2. Tính ma trận phân hoạch mờ U(j) theo công thức (3.4),(3.5) và (3.7)

2.3. Gán mẫu dữ liệu vào một cụm theo (3.8)

2.4. Cập nhật các trọng tâm cụm  theo thuật toán tìm tâm cụm cực đại, cực tiểu và công thức (3.6)

Bước 3. Kiểm tra điều kiện dừng: Nếu  thì chuyển sang bước 4, ngược lại chuyển về bước 2.

Bước 4. Đưa ra kết quả phân cụm.

3.2.Thuật toán IT2FCM cải tiến

3.2.1. Thuật toán khởi tạo tâm cụm ban đầu.

Trong các thuật toán phân cụm, chúng ta thường gặp một vấn đề đó là là làm thế nào để khởi tạo ma trận trọng tâm *V*, bởi vì việc khởi tạo ma trận *V* có thể ảnh hưởng đến số bước thực hiện và kết quả phân cụm. Do đó, chúng ta cần một giải thuật để khởi tạo ma trận trọng tâm V giúp cho các thuật toán phân cụm ổn định và hiệu quả. Chính vì vậy,tôi đề xuất giải thuật khởi tạo tâm cụm dựa trên xác suất thống kê.

Chúng ta nhận thấy rằng tâm cụm sẽ nằm trong các mẫu dữ liệu mà mật độ các mẫu dữ liệu xung quanh (lân cận) chúng là lớn.

Vấn đề chọn lân cận :

Bước này ảnh hưởng khá lớn tới quá trình tính toán. Chúng ta có thể dùng khái niệm phương sai mẫu trong thống kê toán học để giải quyết cho vấn đề chọn lân cận của một điểm dữ liệu.

Chẳng hạn, xét tập , . Khi đó, kỳ vọng mẫu:

 (3.9)

Độ lệch tiêu chuẩn là  (3.10)

Ta xét lân cận của mỗi điểm dữ liệu là hình hộp d chiều với bán kính có thể định nghĩa theo độ lệch tiêu chuẩn là r, .

Thuật toán lựa chọn các ứng viên làm tâm cụm ban đầu:

Ta xét tập dữ liệu đầu *X* vào gồm *n* mẫu dữ liệu:  và 

Bước 1: Tính kỳ vọng mẫu và độ lệch tiêu chuẩn theo công thức (3.9) và (3.10), bán kính hình hộp d chiều 

Bước 2: Tính toán mật độ  của 

 Với 

Bước 3: Tìm  với , và đưa xi vào tập kết quả 

Bước 4: Tính  và thực hiện 

Bước 5: Nếu  thì chuyển sang bước 6 ngược lại chuyển về bước 2

Bước 6 : Đưa ra tập các điểm ứng viên .

Nếu tập  mà lớn ta có thể tiếp tục tiến hành thuật toán này để làm giảm bớt số điểm ứng viên làm tâm cụm.

**Nhận xét:** Ta có thể tăng tốc độ tính toán bằng cách chia tập dữ liệu đầu vào thành các tập con, sau đó tiến hành áp dụng thuật toán trên cho các tập con đó được các tập ứng viên . Sau đó ta tiến hành hợp các tập ứng viên  =V, sau đó áp dụng tiếp thuật toán này trên tập V.

Ngoài ra, tùy từng bài toán mà ta có thể áp dụng thêm các biện pháp để giảm số ứng viên làm tâm cụm như: khi biết hình dạng của cụm ta có thể loại bớt các ứng viên theo đường thẳng, hay theo hình elip….

Việc cuối cùng là ta chỉ cần khởi tạo tâm cụm bằng cách khởi tạo lần lượt theo mật độ các ứng viên để thuật toán phân cụm ổn định và hiệu quả.

3.2.2 Thuật toán tự động xác định số cụm.

Ngoài vấn đề khó khăn trong việc khởi tạo trọng tâm cụm ban đầu, các thuật toán phân cụm còn gặp khó khăn trong việc xác định một cách chính xác số số tâm cụm, tùy vào từng bài toán mà ta có thể chọn số tâm cụm khác nhau.

Nhận xét thấy rằng quá trình phân cụm thực hiện có những cụm dữ liệu to nhỏ khác nhau, có cụm không có mẫu dữ liệu nào. Do đó tùy vào yêu cầu bài toán ta có thể loại bỏ các cụm có số phần tử bằng 0. Chính vì vậy, tôi đề xuất thuật toán tự điều chỉnh số tâm cụm, loại bỏ các cụm không cần thiết dựa trên mật độ và kích thước của các cụm. Đầu vào ta khởi tạo số cụm c, sau khi phân cụm những cụm nhỏ sẽ bị hút vào cụm lớn và bị triệt tiêu ta sẽ có kết quả số cụm đầu ra là .

Thuật toán xác định số tâm cụm thực tế  được chia làm 2 phần chính. Trong phần 1 chúng ta sử dụng thuật toán IT2FCM để phân cụm để mỗi cụm thực sẽ có ít nhất một tâm cụm. Chúng ta giả sử rằng số cụm c đưa vào lớn hơn số cụm  thực tế.

Bước 1: Khởi tạo c tâm cụm từ dữ liệu đầu vào theo thuật toán khởi tạo tâm cụm

Bước 2: Lấy ngẫu nhiên mẫu dữ liệu  từ dữ liệu ban đầu với i =1,2,...,n tính hàm thuộc theo công thức (3.4), (3.5) và (3.7). Sau đó thực hiện đưa mẫu dữ liệu vào cụm theo (3.8).

Bước 3: Cập nhật các trọng tâm cụm  theo thuật toán tìm tâm cụm cực đại, cực tiểu và công thức (3.6)

Ta sẽ lặp lại bước 2 và bước 3 cho đến khi tất cả các tâm cụm không có sự thay đổi.Tuy nhiên, tùy từng bài toán khác nhau ta sẽ xác định điểm dừng của vòng lặp này như khi các tâm cụm thay đối đến một giá trị ngưỡng nào đó hay đơn giản là xác định số vòng lặp tối đa.

Trong phần 2 chúng ta sẽ tiến hành thuật toán loại bỏ các cụm dư thừa.Trước tiên ta định nghĩa độ thuộc mới phụ thuộc vào mật độ của cụm:

 (3.11)

Với 

Ta có thể xác định  bằng cách đơn giản:

Trong đó  là số mẫu dữ liệu thuộc cụm thứ i và n là tổng số mẫu dữ liệu đầu vào.

Tham số E trong (3.11) là một hằng số để điều chỉnh mức độ ảnh hưởng của mật độ tới kết quả phân cụm. Qua kết quả thực nghiệm, ta thường chọn E, khoảng .

Trong đó, 

Với  là bán kính của các cụm sau khi kết thúc phần 1 của thuật toán, và *d* là khoảng cách nhỏ nhất giữa hai tâm cụm mà lớn hơn *3r*.

Trong trường hợp phân cụm mạnh ta có thể nhân đôi tham số E. Nếu tham số E nhỏ hơn mong muốn, thuật toán sẽ không loại bỏ được các cụm dư thừa, ngược lại nếu E quá lớn thì có thể loại hết các cụm.

Các bước thực hiện:

Bước 1:

+, Với mỗi mẫu dữ liệu  tính giá trị hàm thuộc theo công thức (3.4), (3.5) và (3.7).

+, Tính độ thuộc mới  với *i=1..n* và *j=1..c* theo công thức (3.11)

+, Ta đưa mẫu dữ liệu  vào cụm. Nếu  với  và  thì ta thực hiện gán mẫu  thuộc cụm thứ *j*

Bước 2: Cập nhật các trọng tâm cụm  theo thuật toán tìm tâm cụm cực đại, cực tiểu và công thức (3.6)

Ta sẽ lặp lại bước 1 và bước 2 cho đến khi tất cả các tâm cụm không có sự thay đổi. Tuy nhiên, tùy từng bài toán khác nhau ta sẽ xác định điểm dừng của vòng lặp này như khi các tâm cụm thay đối đến một giá trị ngưỡng nào đó hay đơn giản là xác định số vòng lặp tối đa. Cuối cùng ta sẽ thu được kết quả phân cụm mong muốn gồm  cụm. ().

3.2.3 Thuật toán IT2FCM cải tiến.

Trước khi trình bày thuật toán IT2FCM, chúng tôi trình bày một số định nghĩa mới.

Định nghĩa 3.1: Nếu m và c là các tham số cố định, và Ik là một tập đ­ược định nghĩa như­ sau:

 (3.12)

Với là khoảng cách theo thước đo Euclide giữa mẫu dữ liệu  với trọng tâm cụm thứ i ,

Ta xác định lại cách tính hàm thuộc dưới đây để làm giảm nhiễu khi các mẫu dữ liệu ngẫu nhiên trùng với tâm của cụm.

 (3.13)

 (3.14)

Trong đó:  là n véc tơ mẫu dữ liệu tập con thực d chiều trong không gian véctơ *Rd* gồm có N mẫu,  là trọng số mũ,  là trọng tâm cụm thứ i

Trong (3.13), (3.14) m1 và m2 là các tham số mờ biểu diễn các bậc mờ khác nhau. Chúng ta định nghĩa khoảng của một hàm thuộc chính cho một mẫu dữ liệu như là giá trị hàm thuộc cao nhất và thấp nhất cho một mẫu. Những giá trị này được ký hiệu bằng các giá trị hàm thuộc trên và dưới. Việc sử dụng các tham số mờ để biểu thị các bậc mờ khác nhau dẫn đến các hàm mục tiêu khác nhau để làm cực tiểu hóa trong FCM (3.3)

*Thuật toán Interval Type II Fuzzy C\_means cải tiến*

Input: Số cụm c và tham số mờ  cho hàm mục tiêu J, sai số 

Output:  cụm dữ liệu sao cho hàm mục tiêu (3.3) đạt giá trị cực tiểu.

Bước 1: Khởi tạo ma trận trọng tâm ban đầu  theo thuật toán khởi tạo tâm cụm ban đầu đã trình bày trong mục 3.2.1.

Bước 2: Tính ma trận phân hoạch mờ (giá trị hàm thuộc chính) U(j) theo (3.4), (3.5) và (3.7)

Bước 3: Áp dụng thuật toán xác định tâm cụm đã trình bày ở phần 3.2.2 với ma trận phân hoạch U(j) được tính ở bước 2.

Thuật toán kết thúc khi thuật toán xác định tâm cụm kết thúc, kết quả đưa ra là  cụm dữ liệu.