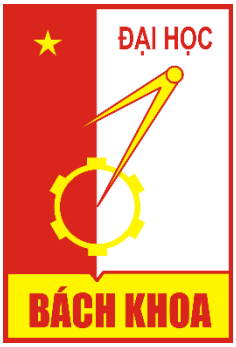
ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường công nghệ thông tin và truyền thông



BÁO CÁO NGHIÊN CỨU TỐT NGHIỆP 1

The Fuzzy C-Means Clustering &

Semi-supervised Fuzzy C-Means Clustering

Giảng viên hướng dẫn: Trần Đình Khang

Sinh viên thực hiện: Bùi Thế Hiếu - 20215047

Contents

1. Tóm tắt

- **Mục đích nghiên cứu:**
  Nghiên cứu này nhằm mục đích tìm hiểu và đánh giá hiệu quả của hai kỹ thuật phân cụm mờ: Fuzzy C-Means (FCM) và Supervised Fuzzy C-Means (SFCM) trong việc phân cụm dữ liệu.
  - FCM là một phương pháp phân cụm không giám sát
  - SFCM là một biến thể có giám sát, sử dụng thông tin nhãn để nâng cao độ chính xác của quá trình phân cụm.
- **Phương pháp nghiên cứu :**
  Nghiên cứu này sử dụng cả hai phương pháp FCM và SFCM để phân cụm một bộ dữ liệu mẫu. Quá trình nghiên cứu bao gồm các bước chuẩn bị, tiền xử lý dữ liệu và thực hiện phân cụm. Để đánh giá hiệu quả của các phương pháp, các tiêu chí như độ chính xác, tính nhất quán và hiệu suất tính toán được xem xét. Ngôn ngữ được sử dụng là Python và các thư viện hỗ trợ đã được sử dụng để triển khai và đánh giá.
- **Kết quả chính:**
  Nghiên cứu cho thấy rằng khi phân cụm dữ liệu có nhãn, SFCM đạt được độ chính xác cao hơn và tính nhất quán tốt hơn so với FCM. Tuy nhiên, để đạt được những kết quả này, SFCM cần sử dụng nhiều tài nguyên tính toán hơn và có thời gian thực thi lâu hơn so với FCM.
- **Kết luận chính:**
  Nghiên cứu kết luận rằng SFCM là phương pháp phân cụm hiệu quả hơn FCM khi có thông tin nhãn, đặc biệt phù hợp cho các ứng dụng đòi hỏi độ chính xác cao. Tuy nhiên, cần xem xét chi phí tính toán và thời gian thực thi khi lựa chọn phương pháp phân cụm cho từng ứng dụng cụ thể.

2. Giới thiệu

- **Bối cảnh:**
  Fuzzy C-Means (FCM) là một phương pháp phân cụm mờ nổi tiếng, cho phép mỗi điểm dữ liệu thuộc về nhiều cụm với mức độ khác nhau. Tuy nhiên, FCM là phương pháp không giám sát và không sử dụng thông tin nhãn trong quá trình phân cụm. Supervised Fuzzy C-Means (SFCM), một biến thể của FCM, tích hợp thông tin nhãn để nâng cao độ chính xác và hiệu quả của quá trình phân cụm. Việc đánh giá và so sánh hiệu quả giữa FCM và SFCM có vai trò quan trọng trong việc chọn lựa phương pháp phù hợp cho các ứng dụng cụ thể.

- **Mục tiêu:**
  Nghiên cứu này đặt mục tiêu so sánh hiệu quả của FCM và SFCM trong việc phân cụm dữ liệu, nhằm xác định phương pháp nào phù hợp hơn cho các tập dữ liệu cụ thể.

- **Phạm vi**:
  Nghiên cứu này tập trung vào việc áp dụng và so sánh FCM và SFCM trên mẫu dữ liệu từ UCI Machine Learning Repository

3. Cơ sở lí thuyết

- **Giới thiệu về Phân cụm (Clustering):**
  Phân cụm là một kỹ thuật quan trọng trong lĩnh vực khai thác dữ liệu và học máy, được sử dụng để tự động nhóm các đối tượng dữ liệu vào các nhóm có tính chất tương đồng nhau. Mục đích chính của phân cụm là tạo ra các nhóm (cụm) sao cho các đối tượng trong cùng một nhóm có tính chất gần nhau và đồng nhất, trong khi đối tượng ở các nhóm khác nhau có tính chất khác biệt.

  Phân cụm giúp phân tích và hiểu sâu hơn về cấu trúc của dữ liệu mà không cần sự can thiệp của con người để định nghĩa các nhóm trước. Kỹ thuật này có nhiều ứng dụng rộng rãi trong thực tế như phân tích thị trường, nhận dạng mẫu, phân tích hình ảnh, xử lý ngôn ngữ tự nhiên, và nhiều lĩnh vực khoa học khác.

- **Fuzzy C-Means (FCM)**:
  Fuzzy C-Means (FCM) là một trong những phương pháp phân cụm mờ quan trọng trong lĩnh vực khai thác dữ liệu và học máy. Đây là một phương pháp giúp nhóm các điểm dữ liệu vào các cụm dựa trên độ tương đồng của chúng với các trung tâm cụm. FCM cho phép mỗi điểm dữ liệu thuộc về một hoặc nhiều cụm với mức độ thuộc về mỗi cụm được biểu thị bằng giá trị mờ (fuzzy membership)

  Nguyên lý hoạt động của FCM:
  - Xác định số lượng cụm (clusters) cần phân chia trước khi áp dụng thuật toán.
  - Khởi tạo các trung tâm cụm (cluster centers) ban đầu.
  - Cập nhật giá trị mờ cho từng điểm dữ liệu: Mỗi điểm dữ liệu được gán một giá trị mờ (membership degree) đại diện cho mức độ thuộc về từng cụm dựa trên khoảng cách tới các trung tâm cụm. Cụm nào có trung tâm gần hơn, điểm dữ liệu đó sẽ có giá trị mờ cao hơn.
  - Cập nhật lại trung tâm cụm: Sau khi cập nhật giá trị mờ cho các điểm dữ liệu, các trung tâm cụm được tính toán lại dựa trên các điểm dữ liệu và giá trị mờ của chúng.

- Lặp lại quá trình cập nhật: Quá trình cập nhật giá trị mờ và trung tâm cụm được lặp lại cho đến khi tiêu chuẩn dừng được đáp ứng, chẳng hạn như khi độ biến thiên giữa các lần cập nhật nhỏ hơn một ngưỡng cho trước.

- **Supervised Fuzzy C-Means (SFCM)**:
  Supervised Fuzzy C-Means (SFCM) là một biến thể của phương pháp Fuzzy C-Means (FCM), được sử dụng để phân cụm dữ liệu khi có sẵn thông tin nhãn từ trước. SFCM tích hợp thông tin nhãn vào quá trình phân cụm để cải thiện độ chính xác và tính nhất quán của kết quả phân cụm.

  Nguyên lý hoạt động của SFCM:

  - Thông tin nhãn (label information): SFCM sử dụng các nhãn (labels) được gán cho từng điểm dữ liệu trong quá trình huấn luyện. Nhãn này thường chỉ ra rằng mỗi điểm dữ liệu nên thuộc về cụm nào.

  - Mục tiêu hàm (objective function): Mục tiêu của SFCM là tối ưu hóa một hàm mục tiêu sao cho đồng thời hài hòa với cả thông tin từ các nhãn và tính mờ mịn của phân cụm. Hàm mục tiêu này thường bao gồm hai thành phần chính:
    - Độ chính xác (accuracy): Đảm bảo các điểm dữ liệu được phân vào cụm thích hợp với nhãn đã biết từ trước.
    - Tính mờ (fuzziness): Cho phép mỗi điểm dữ liệu có một giá trị mờ (membership degree) để xác định mức độ thuộc về từng cụm.
  - Tiến trình tối ưu hóa: SFCM thực hiện lặp lại quá trình cập nhật giá trị mờ và trung tâm cụm giống như FCM, nhưng với sự bổ sung của thông tin nhãn để hướng đến kết quả phân cụm chính xác hơn.

- **So sánh giữa FCM và SFCM**:
  FCM (Fuzzy C-Means) và SFCM (Supervised Fuzzy C-Means) là hai phương pháp quan trọng trong lĩnh vực phân cụm dữ liệu, nhưng có những điểm khác biệt đáng chú ý:
  Đặc điểm cơ bản:

  FCM: Là phương pháp phân cụm mờ không giám sát. Nó dựa trên khoảng cách giữa các điểm dữ liệu và các trung tâm cụm để gán mỗi điểm vào các cụm với một mức độ (giá trị mờ) khác nhau.

  SFCM: Là biến thể có giám sát của FCM. Ngoài việc sử dụng khoảng cách, SFCM còn tích hợp thông tin nhãn từ các điểm dữ liệu đã biết trước để cải thiện độ chính xác và tính nhất quán của quá trình phân cụm.

  - Sử dụng thông tin nhãn:

        - FCM: Không sử dụng thông tin nhãn trong quá trình phân cụm, do đó thường áp dụng trong các trường hợp mà không có sẵn thông tin nhãn hoặc khi không cần độ chính xác cao nhất.

        - SFCM: Sử dụng thông tin nhãn để hướng dẫn quá trình phân cụm. Điều này giúp cải thiện đáng kể độ chính xác của kết quả phân cụm, đặc biệt là trong các tập dữ liệu có nhãn rõ ràng và tin cậy.

    - Hiệu suất và tính toán:

        - FCM: Thường có tính toán đơn giản hơn so với SFCM vì không phải xử lý thông tin nhãn. Điều này có thể dẫn đến thời gian thực thi nhanh hơn và yêu cầu ít tài nguyên tính toán hơn.

        - SFCM: Yêu cầu nhiều tài nguyên tính toán hơn do phải tích hợp và xử lý thông tin nhãn. Việc này có thể làm gia tăng thời gian thực thi và yêu cầu bộ nhớ lớn hơn so với FCM.

    - Tính chất của kết quả phân cụm:

        - FCM: Cho phép mỗi điểm dữ liệu có mức độ thuộc về nhiều cụm khác nhau, biểu thị bằng giá trị mờ. Điều này phù hợp với dữ liệu có tính mờ mịn và không rõ ràng.

        - SFCM: Tính chất này vẫn được giữ lại từ FCM, nhưng kết quả phân cụm thường có tính nhất quán cao hơn và chính xác hơn do sử dụng thông tin nhãn.

- **Tổng kết:**
  FCM và SFCM đều có vai trò quan trọng trong phân cụm dữ liệu, mỗi phương pháp có ưu điểm và hạn chế riêng. Lựa chọn phương pháp phù hợp sẽ phụ thuộc vào tính chất của dữ liệu và mục đích cụ thể của bài toán, có thể là cải thiện độ chính xác phân cụm hoặc giảm thiểu chi phí tính toán.

4. Phương pháp nghiên cứu

- **Thiết kế nghiên cứu:**
  Nghiên cứu này nhằm so sánh hiệu quả của hai phương pháp phân cụm mờ: Fuzzy C-Means (FCM) và Supervised Fuzzy C-Means (SFCM). Quá trình nghiên cứu bao gồm các giai đoạn chính: thu thập dữ liệu, tiền xử lý dữ liệu, thực hiện phân cụm bằng FCM và SFCM, sau đó đánh giá kết quả dựa trên các tiêu chí như độ chính xác, tính nhất quán và hiệu suất tính toán.
- **Dữ liệu và nguồn dữ liệu**
  Bộ dữ liệu Iris về các loài hoa (Iris - UCI Machine Learning Repository)
  Đây là một trong những bộ dữ liệu sớm nhất được sử dụng trong văn xuôi về các phương pháp phân loại và rất phổ biến trong thống kê và học máy. Bộ dữ

liệu này bao gồm 3 lớp, mỗi lớp có 50 ví dụ, trong đó mỗi lớp tương ứng với một loại cây hoa Iris. Một lớp có thể phân tách tuyến tính với 2 lớp còn lại; hai lớp còn lại thì không thể phân tách tuyến tính với nhau.

- **Quy trình phân cụm:**
  - Bước 1: Chuyển đổi dữ liệu từ định dạng CSV sang ma trận thuộc tính.
  - Bước 2: Thực hiện phân cụm bằng Fuzzy C-Means (FCM):
    - Khởi tạo các tâm cụm ban đầu.
    - Tính toán các giá trị thành viên cho mỗi điểm dữ liệu.
    - Cập nhật các tâm cụm dựa trên giá trị thành viên.
    - Lặp lại quá trình cho đến khi đạt điều kiện hội tụ.
  - Bước 3: Thực hiện phân cụm bằng Supervised Fuzzy C-Means (SFCM):
    - Sử dụng thông tin nhãn để điều chỉnh các giá trị thành viên.
    - Khởi tạo các tâm cụm ban đầu theo dạng có giám sát.
    - Tính toán và cập nhật các giá trị thành viên có giám sát.
    - Lặp lại quá trình cho đến khi đạt điều kiện hội tụ.
  - Bước 4: Đánh giá kết quả:
    - Sử dụng các tiêu chí như độ chính xác, tính nhất quán và thời gian tính toán để so sánh kết quả của FCM và SFCM.
    - Phân tích và trình bày kết quả thông qua các bảng biểu và hình vẽ.

- **Các công nghệ và phần mềm sử dụng:**
  - **Python:** Ngôn ngữ lập trình chính được sử dụng cho việc phân cụm và phân tích dữ liệu.
  - **NumPy và Pandas:** Hỗ trợ xử lý và phân tích dữ liệu.
  - **Matplotlib:** Tạo biểu đồ đánh giá kết quả.
  - **Streamlit:** Tạo giao diện.

5. Thực nghiệm và kết quả

- **Mô tả dữ liệu sử dụng:**

  Bộ dữ liệu Iris đã được sử dụng trong bài báo kinh điển năm 1936 của R.A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems", và cũng có thể được tìm thấy trên Kho dữ liệu Học máy của Đại học California Irvine (UCI).

  Bộ dữ liệu bao gồm ba loài hoa Iris, mỗi loài có 50 mẫu cùng với một số thuộc tính về từng loài hoa. Một loài hoa có thể phân tách tuyến tính với hai loài hoa còn lại, nhưng hai loài hoa còn lại không thể phân tách tuyến tính với nhau.

  Các cột trong bộ dữ liệu này bao gồm:
  - Id
  - SepalLengthCm (Chiều dài đài hoa, đơn vị: cm)
  - SepalWidthCm (Chiều rộng đài hoa, đơn vị: cm)
  - PetalLengthCm (Chiều dài cánh hoa, đơn vị: cm)

- PetalWidthCm (Chiều rộng cánh hoa, đơn vị: cm)
- Species (Loài hoa)

- **Các bước tiền xử lí dữ liệu**
  - Đọc flie dữ liệu
  - Loại bỏ các cột không cần thiết (id), khởi tạo ma trận ū dựa trên cột Species

```
def prepare_iris(path, C):
    columns = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm', 'Species']

    df = pd.read_csv(path).drop('Id', axis=1)
    N = len(df)
    x = df.drop(columns[-1], axis=1).to_numpy(float)

    indices = pd.factorize(df[columns[-1]])[0]
    U_bar = np.zeros([N, C])
    for row, idx in zip(U_bar, indices):
        row[idx] = 1

    return x, U_bar

def unsupervisedPrepare_iris(path, C):
    columns = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm', 'Species']

    df = pd.read_csv(path).drop('Id', axis=1)
    x = df.drop(columns[-1], axis=1).to_numpy(float)
    return x
```

- **Tiến hành phân cụm**
  - **FCM:**
    + Khởi tạo ma trận thành viên và tâm cụm:

```
  def __init__(self, N, C, D, m, A=...):
      self.U = np.random.randn(C, N)
      self.U = softmax(self.U, dim=0)
      self.v = np.random.randn(C, D)
      self.A = A if A != ... else np.identity(D)
      self.m = m
```

+ Cập nhật tâm cụm:

```python
def update_centeroids(self, y):
    um = self.U ** self.m
    scale_factor = um.sum(axis=-1, keepdims=True)  # Cx1
    self.v = np.dot(um, y) / scale_factor
```

+ Cập nhật ma trận thành viên:

```python
def update_membership_matrix(self, y):
    s = y[:, None, ...] - self.v         # NxCxD
    d = np.einsum('ncad, dd, ncda -> nc', s[..., None, :], self.A, s[..., None]).T ** 0.5

    t = 2 / (self.m - 1)
    self.U = 1 / ((d[:, None, ...] / d) ** t).sum(axis=1)                    # CxCxN
```

+ Tính khoảng cách từ ma trận thành viên mới đến tâm cụm mới

```python
def fit(self, y, eps):
    losses = []

    while True:
        self.update_centeroids(y)
        U_old = self.U
        self.update_membership_matrix(y)
        loss = self.compute_J(y).item()
        losses.append(loss)

        print(f'loss: {loss:.4f}', end='\r')

        d = np.linalg.norm(self.U - U_old, 'fro')
        if d <= eps:
            break
    print()
    return losses
```

+ Lặp lại cho đến khi hội tụ:

```
def fit(self, y, eps):
    losses = []

    while True:
        self.update_centeroids(y)
        U_old = self.U
        self.update_membership_matrix(y)
        loss = self.compute_J(y).item()
        losses.append(loss)

        print(f'loss: {loss:.4f}', end='\r')

        d = np.linalg.norm(self.U - U_old, 'fro')
        if d <= eps:
            break
    print()
    return losses
```

- **SFCM:**
  + Khởi tạo ma trận thành viên và tâm cụm:

```
def __init__(self, N, C, D, m):
    U = np.random.randn(N, C)
    self.U = softmax(U, dim=1)
    self.v = np.random.randn(C, D)
    self.m = m
```

+ Cập nhật ma trận thành viên:

```
def update_U(self, x, U_bar):
    e = 1 / (1 - self.m)
    d = np.sum((x[:, None, ...] - self.v) ** 2, axis=-1)

    multiplier = np.sum(1 - U_bar, axis=-1, keepdims=True)
    num = d ** e
    den = np.sum(num, axis=-1, keepdims=True)

    self.U = U_bar + multiplier * num / den
```

+ Cập nhật tâm cụm:

```
def update_centeroids(self, x, U_bar):
    t = np.abs(self.U - U_bar) ** self.m
    num = np.sum(t[..., None] * x[:, None, ...], axis=0)
    den = np.sum(t, axis=0)[..., None]
    self.v = num / den
```

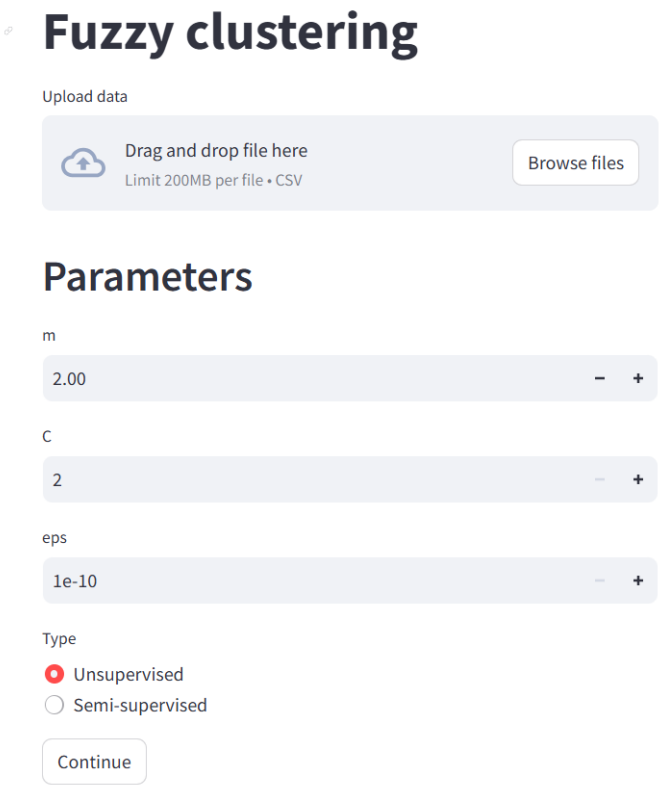
+ Tính khoảng cách từ ma trận thành viên mới đến tâm cụm mới:

```
def compute_J(self, x, U_bar):
    t1 = np.abs(self.U - U_bar) ** self.m
    t2 = np.sum((x[:, None, ...] - self.v) ** 2, axis=-1)
    return np.sum(t1 * t2)
```

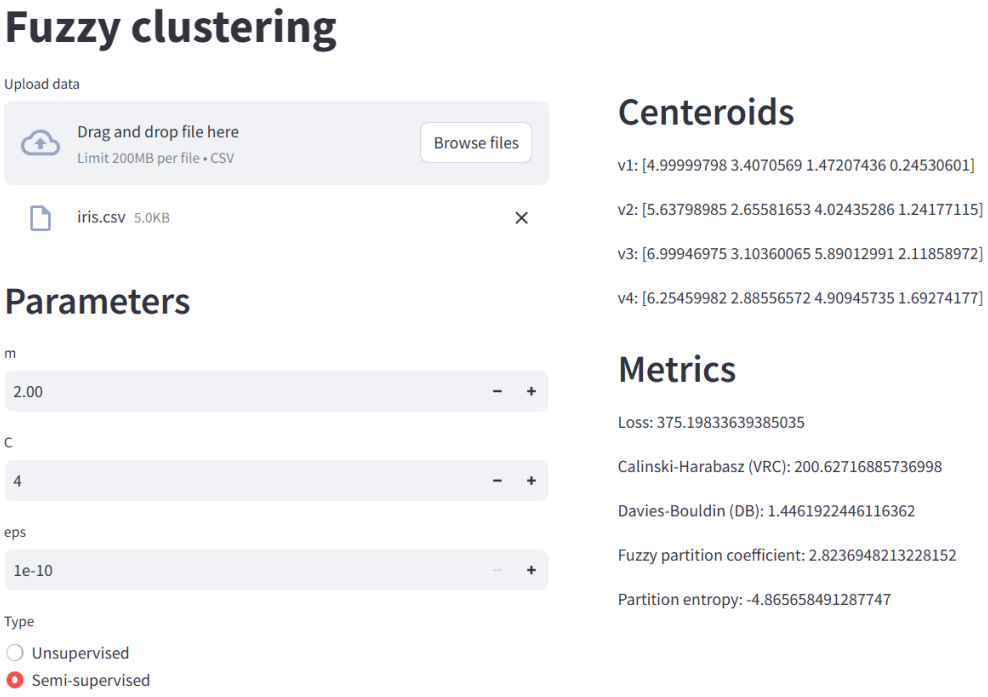
+ Lặp lại cho đến khi hội tụ:

```
def fit(self, x, U_bar, eps):
    losses = []

    while True:
        self.update_centeroids(x, U_bar)
        U_old = self.U
        self.update_U(x, U_bar)
        loss = self.compute_J(x, U_bar).item()
        losses.append(loss)

        print(f'loss: {loss:.4f}', end='\r')

        d = np.linalg.norm(self.U - U_old, 'fro')
        if d <= eps:
            break
    print()
    return losses
```

- **Thiết kế giao diện:**
  - File đầu vào
  - Các tham số: số tâm cụm, hệ số mờ, epsilon có thể điều chỉnh
  - Lựa chọn phương pháp phân cụm



- Hiển thị kết quả:



- **Đánh giá kết quả của 2 phương pháp:**
  - **Độ chính xác:** Kết quả cho thấy SFCM thường cho độ chính xác cao hơn so với FCM khi sử dụng các chỉ số VRC và DBI. Đặc biệt là khi dữ liệu có sự can thiệp từ thông tin nhãn giám sát, SFCM có xu hướng tạo ra các cụm rõ ràng hơn và tách biệt hơn
  - **Tính nhất quán (Consistency):** Kết quả cho thấy SFCM thường có độ tính nhất quán cao hơn so với FCM khi có sự can thiệp từ nhãn giám sát. Điều này được phản ánh qua việc giảm thiểu PE và tăng FPC trong SFCM
  - **Thời gian tính toán (Computation Time):** thời gian tính toán của FCM và SFCM có thể khác nhau và phụ thuộc vào nhiều yếu tố như kích thước của tập dữ liệu, độ phức tạp của thuật toán, và mức độ can thiệp từ thông tin nhãn giám sát. SFCM thường có thể yêu cầu nhiều thời gian tính toán hơn do tính phức tạp của quá trình tính toán ma trận membership và cập nhật các tâm cụm có giám sát. Tuy nhiên, sự khác biệt này có thể không đáng kể trên các tập dữ liệu nhỏ và đơn giản

6. Thảo luận

- **Ý nghĩa của kết quả:** Nghiên cứu này nhấn mạnh vai trò của thông tin nhãn giám sát trong việc cải thiện độ chính xác và tính nhất quán của phương pháp phân cụm SFCM so với FCM. Tuy nhiên, cần cân nhắc đến chi phí tính toán khi áp dụng SFCM, đặc biệt là đối với các ứng dụng yêu cầu xử lý dữ liệu lớn và thời gian thực.

- **Hạn chế của nghiên cứu:**
  - Giới hạn của các chỉ số đánh giá: Nghiên cứu sử dụng các chỉ số như VRC và DB để đánh giá độ chính xác và tính nhất quán của phương pháp phân cụm. Tuy nhiên, các chỉ số này có thể không phản ánh đầy đủ các đặc trưng của dữ liệu thực tế, và việc áp dụng chúng cần phải cân nhắc kỹ lưỡng để đảm bảo tính đáng tin cậy của kết quả.

  - Giới hạn của thông tin nhãn giám sát: Sự hiệu quả của SFCM phụ thuộc mạnh vào chất lượng và độ chính xác của thông tin nhãn giám sát được sử dụng. Nếu thông tin nhãn không đủ chính xác hoặc thiếu sót, kết quả phân cụm có thể bị sai lệch. Điều này đặt ra thách thức trong việc áp dụng SFCM cho các tập dữ liệu thực tế có độ phức tạp và đa dạng cao.

  - Khả năng mở rộng và áp dụng: Nghiên cứu có thể hạn chế trong việc áp dụng và kiểm chứng trên nhiều loại dữ liệu khác nhau từ nhiều lĩnh vực khác nhau. Việc mở rộng nghiên cứu để bao gồm nhiều bộ dữ liệu đa dạng có thể cần thiết để đánh giá tính tổng quát và ứng dụng thực tế của các phương pháp phân cụm này.

  - Thời gian tính toán và chi phí tính toán: Sử dụng SFCM có thể đòi hỏi nhiều tài nguyên tính toán hơn và thời gian thực thi lâu hơn so với FCM, đặc biệt là trên các tập dữ liệu lớn và phức tạp. Việc này có thể là một hạn chế đối với các ứng dụng yêu cầu phản hồi nhanh và xử lý dữ liệu lớn.

- **Các yếu tố ảnh hưởng đến kết quả:**
    - Kích thước tập dữ liệu: Kích thước của tập dữ liệu có thể ảnh hưởng đến hiệu suất của cả hai phương pháp. Các tập dữ liệu lớn thường đòi hỏi thời gian tính toán lâu hơn và tài nguyên tính toán cao hơn để xử lý.

    - Số lượng cụm: Số lượng cụm được chỉ định trước có thể ảnh hưởng đến độ chính xác của phân cụm. Việc lựa chọn số lượng cụm phù hợp là một yếu tố quan trọng để đạt được kết quả tốt nhất.

    - Hệ số mờ (fuzziness coefficient): Hệ số mờ trong FCM và SFCM ảnh hưởng đến mức độ mờ của các cụm. Giá trị của hệ số này cần được điều chỉnh phù hợp để tối ưu hóa kết quả phân cụm.

    - Chất lượng của thông tin nhãn: Đối với SFCM, chất lượng và độ tin cậy của thông tin nhãn là yếu tố quyết định đến hiệu quả phân cụm. Thông tin nhãn không chính xác có thể dẫn đến kết quả phân cụm kém chính xác.

    - Đặc điểm của dữ liệu: Tính phân tán, độ chi tiết, và độ phức tạp của dữ liệu có thể ảnh hưởng đến khả năng của thuật toán trong việc phân cụm chính xác và hiệu quả.

    - Các tham số thuật toán: Các tham số như số lần lặp, điều kiện dừng, và phương pháp khởi tạo các tâm cụm cũng có thể ảnh hưởng đến kết quả của phương pháp phân cụm.
- **Đề xuất hướng nghiên cứu tiếp theo:**
    - Tối ưu hóa SFCM: Nghiên cứu và áp dụng các kỹ thuật tối ưu hóa để giảm chi phí tính toán và thời gian thực thi của SFCM. Các phương pháp như sử dụng các thuật toán song song, phân tán, hoặc các kỹ thuật tối ưu hóa thuật toán có thể giúp cải thiện hiệu suất của SFCM trên các tập dữ liệu lớn và phức tạp.

    - Nâng cao chất lượng nhãn: Phát triển các phương pháp tiền xử lý và xử lý dữ liệu để cải thiện chất lượng của thông tin nhãn trước khi áp dụng vào SFCM. Các phương pháp này có thể bao gồm xử lý nhiễu, phân đoạn dữ liệu, và xây dựng mô hình dự đoán để tăng độ chính xác và tin cậy của thông tin nhãn.

    - Kiểm chứng trên nhiều bộ dữ liệu: Áp dụng và kiểm chứng FCM và SFCM trên nhiều bộ dữ liệu khác nhau từ nhiều lĩnh vực khác nhau. Việc đánh giá tính tổng quát và hiệu quả của các phương pháp phân cụm trên các bộ dữ liệu đa dạng sẽ giúp khẳng định và mở rộng ứng dụng của chúng trong thực tế.

    - Phát triển các phương pháp đánh giá mới: Nghiên cứu và phát triển các phương pháp đánh giá hiệu quả hơn cho các phương pháp phân cụm mờ,

đặc biệt là SFCM. Các chỉ số mới có thể cân nhắc đến cả yếu tố can thiệp của thông tin nhãn giám sát để đánh giá hiệu quả của SFCM một cách chính xác hơn.

- Áp dụng trong các lĩnh vực ứng dụng cụ thể: Nghiên cứu ứng dụng của FCM và SFCM trong các lĩnh vực như y tế, marketing, và khoa học dữ liệu để đánh giá khả năng áp dụng thực tiễn và tiềm năng của các phương pháp này trong các bối cảnh ứng dụng cụ thể.

7. Bộ dữ liệu, mã nguồn

Github: https://github.com/hiewbt/GR1