HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Thị Ngọc Diễm

SO SÁNH MỘT SỐ THUẬT TOÁN PHÂN CỤM DỮ LIỆU

Chuyên ngành: Khoa học máy tính

Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ

HÀ NỘI - 2014

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **PGS.TS Trần Đình Quế**

Phản biện 1: ……

Phản biện 2: ……

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng .... năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông

MỞ ĐẦU

Phân cụm dữ liệu là quá trình nhóm các đối tượng tương tự nhau trong tập dữ liệu vào các cụm sao cho các đối tượng cùng cụm là tương đồng. Phân cụm chính là nhiệm vụ chính trong khai phá dữ liệu và là một kỹ thuật phổ biến để phân tích số liệu thông tin, các hệ trợ giúp quyết định, các thuật toán nhận dạng mẫu và phân lớp mẫu, v.v và đặc biệt là khai phá dữ liệu. Ngày nay có rất nhiều các kỹ thuật phân cụm được sử dụng, nhưng không phải kỹ thuật phân cụm nào cũng có thể giải quyết tốt tất cả các vấn đề của quá trình phân cụm. Trong khuôn khổ luận văn này sẽ khảo sát bốn thuật toán phân cụm tiêu biểu gồm có: phân cụm phân hoạch K-Means, phân cụm phân cấp Hierarchical Clustering, phân cụm theo mật độ DBSCAN, phân cụm mô hình EM.

Trước hết luận văn trình bày một cách tổng quan về phân cụm dữ liệu, và xem xét, so sánh, đánh giá các thuật toán trên. Các thuật toán sẽ được tiến hành trên phần mềm khái thác dữ liệu Weka với bộ dữ liệu chuẩn Bank.arff và Glass.arff. Tiêu chí so sánh các thuật toán là thời gian thực hiện, chất lượng cụm và độ đồng đều giữa các cụm.

Cấu trúc của luận văn gồm 3 chương:

Chương 1: Tổng quan về phân cụm dữ liệu: chương này sẽ trình bày một cách tổng quan các kiến thức về phân cụm dữ liệu.

Chương 2: Một số thuật toán phân cụm dữ liệu tiêu biểu: chương này sẽ đi sâu vào tìm hiểu bốn thuật toán phân cụm dữ liệu K-Means, HC, EM và DBSCAN.

Chương 3: So sánh một số thuật toán phân cụm dữ liệu: chương này sẽ giới thiệu về phần mềm Weka cùng bộ dữ liệu gốc Bank.arff và Glass.arff. Từ đó sẽ tiến hành thử nghiệm với các thuật toán phân cụm nhằm mục đích so sánh, đánh giá các thuật toán phân cụm này.

CHƯƠNG 1: TỔNG QUAN VỀ PHÂN CỤM DỮ LIỆU

1.1 Khái niệm phân cụm dữ liệu

Phân cụm là một trong những hành vi nguyên thủy nhất của con người nhằm nắm giữ lượng thông tin khổng lồ họ nhận được hằng ngày vì xử lý mọi thông tin như một thực thể đơn lẻ là không thể. Phân cụm là một kỹ thuật được sử dụng để kết hợp các đối tượng quan sát thành các cụm sao cho mỗi cụm có cùng một số đặc điểm tương đồng ở một số đặc điểm đang xét. Ngược lại các đối tượng trong các nhóm khác nhau thì độ tương đồng khác nhau (ít tương đồng hơn) ở một số đặc điểm đang xét .

1.2 Ứng dụng của phân cụm dữ liệu

Phân cụm dữ liệu đã được sử dụng trong một lượng lớn các ứng dụng cho một loạt các chủ đề, các lĩnh vực khác nhau như phân đoạn ảnh, nhận dạng đối tượng, ký tự và các chuyên ngành cổ điển như tâm lý học, kinh doanh, v.v. Một số ứng dụng cơ bản của phân cụm dữ liệu bao gồm:

- Thương mại
- Sinh học
- Phân tích dữ liệu không gian
- Lập quy hoạch đô thị
- Địa lý
- Khai phá Web
- …

1.3 Các yêu cầu đối với kỹ thuật phân cụm dữ liệu

Việc xây dựng, lựa chọn một thuật toán phân cụm là bước then chốt cho việc giải quyết vấn đề phân cụm. Sự lựa chọn này phụ thuộc vào đặc tính dữ liệu cần phân cụm, mục đích ứng dụng thực tế hoặc xác định độ ưu tiên giữa chất lượng của các cụm hay tốc độ thực hiện thuật toán. Hầu hết các nghiên cứu về phát triển thuật toán phân cụm dữ liệu đều nhằm thỏa mãn các yêu cầu cơ bản gồm có:

- Có khả năng mở rộng.

- Thích nghi với các kiểu dữ liệu khác nhau.
- Khám phá ra các cụm với hình thức bất kỳ.
- Ít nhạy cảm với thứ tự của dữ liệu vào.
- Khả năng giải quyết dữ liệu nhiễu.
- Ít nhạy cảm với tham số đầu vào.
- Thích nghi với dữ liệu đa chiều.
- Dễ hiểu, dễ cài đặt và khả thi.

1.4 Một số kỹ thuật phân cụm dữ liệu

Các kỹ thuật có rất nhiều cách tiếp cận và các ứng dụng trong thực tế, nhưng chung quy lại thì nó đều hướng đến hai mục tiêu đó là chất lượng của các cụm tìm được và tốc độ thực hiện thuật toán.

1.4.1 Phương pháp phân cụm theo phân hoạch

Ý tưởng chính của kỹ thuật này là phân hoạch một tập hợp dữ liệu có n phần tử cho trước thành k nhóm dữ liệu sao mỗi phần tử dữ liệu chỉ thuộc về một nhóm dữ liệu có tối thiểu ít nhất một phần tử dữ liệu. Số các cụm được thiết lập là các đặc trưng được lựa chọn trước.

Phương pháp này là tốt cho việc tìm các cụm hình cầu trong không gian Euclidean. Ngoài ra, phương pháp này cũng phụ thuộc vào khoảng cách cơ bản giữa các điểm để lựa chọn các điểm dữ liệu nào có quan hệ là gần nhau với mỗi điểm khác và các điểm dữ liệu nào không có quan hệ hoặc có quan hệ là xa nhau so với mỗi điểm khác.

1.4.2 Phương pháp phân cụm theo phân cấp

Phương pháp này xây dựng một phân cấp trên cơ sở các đối tượng dữ liệu đang xem xét. Nghĩa là sắp xếp một tập dữ liệu đã cho thành một cấu trúc có dạng hình cây, cây phân cấp này được xây dựng theo kỹ thuật đệ quy. Cây phân cụm có thể được xây dựng theo hai phương pháp sau: hòa nhập nhóm, thường được gọi là tiếp cận *từ dưới lên* và phân chia nhóm, thường được gọi là tiếp cận *từ trên xuống*.

1.4.3 Phương pháp phân cụm theo mật độ

Phương pháp này nhóm các đối tượng theo hàm mật độ xác định. Mật độ xác định được định nghĩa như là số các đối tượng lân cận của một đối tượng dữ liệu theo một ngưỡng nào đó.

1.4.4 Phương pháp phân cụm trên lưới

Kỹ thuật phân cụm dựa trên mật độ không thích hợp với dữ liệu nhiều chiều, để giải quyết cho đòi hỏi này, người ta đã sử dụng phương pháp phân cụm dựa trên lưới. Đây là phương pháp dựa trên cấu trúc dữ liệu lưới để phân cụm dữ liệu, phương pháp này chủ yếu tập trung áp dụng cho lớp dữ liệu không gian. Thí dụ như dữ liệu được biểu diễn dưới dạng cấu trúc hình học của đối tượng trong không gian cùng với các quan hệ, các thuộc tính, các hoạt động của chúng.

1.4.5 Phương pháp phân cụm dựa trên mô hình

Phương pháp này cố gắng khám phá các phép xấp xỉ tốt của các tham số mô hình sao cho khớp với dữ liệu một cách tốt nhất. Chúng có thể sử dụng chiến lược phân cụm phân hoạch hoặc phân cụm phân cấp, dựa trên cấu trúc hoặc mô hình mà chúng giả định về tập dữ liệu và cách chúng hiệu chỉnh các mô hình này để nhận dạng ra các phân hoạch.

1.4.6 Phương pháp phân cụm có dữ liệu ràng buộc

Hiện nay các phương pháp phân cụm này đã và đang phát triển và áp dụng nhiều trong các lĩnh vực khác nhau và đã có một số nhánh nghiên cứu được phát triển trên cơ sở các phương pháp đó như:

- Phân cụm thống kê
- Phân cụm khái niệm
- Phân cụm mờ

1.5 Kết luận

Chương này đã trình bày khái niệm cơ bản về phân cụm dữ liệu. Trong bài toán phân cụm dữ liệu cũng đã trình bày những những ứng dụng, yêu cầu cơ bản, các kỹ thuật đối với phân cụm dữ liệu. Chương sau sẽ đi sâu hơn về các thuật toán phân cụm dữ liệu từ đó có thể cài đặt các thuật toán phân cụm dữ liệu vào chương tiếp theo.

CHƯƠNG 2: MỘT SỐ THUẬT TOÁN PHÂN CỤM DỮ LIỆU TIÊU BIỂU

2.1 Thuật toán phân cụm K-Means

Phân cụm dựa trên phân nhóm dữ liệu thường cố gắng tạo ra k phân chia dữ liệu từ cơ sở dữ liệu gồm n đối tượng, trong đó mỗi cụm sẽ tối ưu một tiêu chuẩn phân cụm, chẳng hạn cực tiểu hóa tổng bình phương khoảng cách từ tâm của các cụm. Trong phần này luận văn đi sâu tìm hiểu thuật toán K-Means.

2.1.1 Ý tưởng

Thuật toán K-Means được đề xuất bởi MacQueen là một trong những thuật toán học không giám sát thông dụng nhất trong phân nhóm dữ liệu. Với mục tiêu chia tập gồm n đối tượng của cơ sở dữ liệu thành k cụm ($k \le n$, k là số nguyên, dương) sao cho các đối tượng trong cùng một vùng có khoảng cách bé còn các đối tượng khác vùng thì có khoảng cách lớn hơn nhiều.

2.1.2 Thuật toán

Đầu tiên, xác định K tâm cụm, trong đó K là một tham số mà người dùng đưa vào. Với $x = \{x_1, x_2, ..., x_N\}$ là tập dữ liệu đầu vào và $C = \{C_1, C_2, ..., C_K\}$ là tập K tâm cụm.

Đầu vào: $X = \{x_1, x_2, ..., x_N\}$ (Tập dữ liệu đầu vào)

K (Số lượng tâm cụm)

MaxIters *(*Số vòng lặp tối đa*)*

Đầu ra: $C = \{c_1, c_2, ..., c_K\}$(Tập các cụm)

2.1.3 Độ phức tạp thuật toán

Thuật toán K-Means có độ phức tạp theo thời gian của nó là $O(n \times k \times l)$

2.1.4 Ưu nhược điểm

Ưu điểm của thuật toán là một phương pháp đơn giản, hiệu quả, tự tổ chức.

Nhược điểm của thuật toán là số cụm k phải được xác định trước, chỉ áp dụng được khi xác định được giá trị trung bình, không thể xử lý nhiễu, không thích hợp nhằm khám

phá các dạng không lồi hay các cụm có kích thước khác nhau, đây là thuật toán độc lập tuyến tính.

2.2 Phân cụm phân cấp (Hierarchical Clustering)

2.2.1 Ý tưởng

Phân cụm phân cấp sắp xếp một tập dữ liệu đã cho thành một cấu trúc có hình dạng cây. Cây phân cụm có thể được xây dựng theo hai phương pháp tổng quát phương pháp phân cấp từ trên xuống và phương pháp phân cấp vun đống từ dưới lên.

2.2.2 Thuật toán

Đối với phương pháp phân cụm phân cấp từ dưới lên giải thuật được mô tả như sau, với $x = \{x_1, x_2, ..., x_N\}$ là tập các đối tượng. Gọi $C = \{c_1, c_2, ..., c_K\}$ là tập các cụm với μ_i là tâm cụm của cụm c_i và n_i là số đối tượng trong cụm c_i. Ma trận $D_{N\times N}$ được gọi là ma trận khoảng cách với $D_{i\times j} = d(c_i, c_j)$. Thuật toán ban đầu sẽ gán mỗi đối tượng là một cụm chẳn hạn chúng ta có N đối tượng thì chúng ta sẽ có N cụm. Sau đó tiến hành hai cụm gần nhau nhất lại cho đến khi nào số cụm còn lại một thì dừng lại. Chúng ta sẽ sử dụng mảng hai chiều $\mu_{N\times N}$ để lưu tâm cụm.

Đầu vào: $X = \{x_1, x_2, ..., x_N\}$ (Tập dữ liệu đầu vào).

Đầu ra: Các tâm cụm và các đối tượng thuộc mỗi cụm.

Mã giả thuật toán phân cụm phân cấp từ dưới lên

---

1. $C \leftarrow$ *initCluster () // Khởi tạo với mỗi cụm*
2. $D \leftarrow$*initMatricDistance(C) // Khởi tạo ma trận khoảng cách*
3. *repeat*
4. $(C_p, C_{p'}) \leftarrow$*caculatorDistanceMin(D); //Tính toán cặp cụm cần gom.*
5. $C \leftarrow$ *merge* $(C_p, C_{p'})$ *// Nhóm hai cụm* $C_p, C_{p'}$ *với nhau*

---

6. *D ← updateMatricDistance(C); // Cập nhật ma trận khoảng cách*
7. *until (length(C)>1).*

Ngược lại đối với phân cụm phân cấp từ trên xuống thì thuật toán phân cụm từ trên xuống sẽ chọn cụm cần phân tách, sau đó với cụm được chọn sẽ phân tách cụm đó thành hai cụm con dựa vào độ đo tương đồng giữa hai cụm. Đến khi nào không còn cụm nào còn có thể tách được nữa thì dừng lại.

2.2.3 Độ phức tạp thuật toán

Để tính toán ma trận khoảng cách thì độ phức tạp tính toán là $O(n^2)$. Sau đó ở mỗi bước thì số lượng tâm cụm giảm đi một $(n-1)$, nếu vị trí gom cụm là vị trí thứ i thì cần $O\left((m-i-1)^2\right)$ để cập nhật hai cụm lại thành một. Để cập nhật ma trận khoảng cách thì cần $O(m-i-1)$. Nói tóm lại độ phức tạp tính toán của thuật toán là $O\left(m^3\right)$, nếu dữ liệu độ tương đồng giữa các cụm được sắp xếp thì độ phức tạp tính toán là $O\left(n^2 \log n\right)$.

2.2.4 Ưu nhược điểm

Ưu điểm của thuật toán đây là một phương pháp phân cụm đơn giản, mềm dẻo, linh hoạt, dễ cài đặt, so với phương pháp k-means thì số cụm là không cần biết trước. Các cụm sinh ra thường thể hiện tốt sự phân bố của dữ liệu đầu vào, tuy nhiên lại gặp phải vấn đề độ phức tạp tính toán cao $O\left(n^3\right)$ (có thể tối ưu $O\left(n^2 \log n\right)$) với n số đối tượng cần phân cụm. Vì lý do đó, việc áp dụng trực tiếp phương pháp này với tập dữ liệu đầu vào lớn là không khả khi.

2.3 Phân cụm theo mật độ DBSCAN

2.3.1 Ý tưởng

Ý tưởng cơ bản của phân cụm dựa trên mật độ như sau: Đối với mỗi đối tượng của một cụm, láng giềng trong một bán kính cho trước (Eps) phải chứa ít nhất một số lượng tối thiểu các đối tượng ($MinPts$).

Thuật toán DBSCAN gom cụm các đối tượng trong cơ sở dữ liệu không gian ứng với thông số Eps, $MinPts$ cho trước, DBSCAN xác định một cụm thông qua 2 bước:

1) Chọn đối tượng bất kỳ thỏa mãn điều kiện đối tượng lõi làm đối tuợng hạt giống;

2) Tìm các đối tượng tới được theo mật độ từ đối tượng hạt giống.

2.3.2 Thuật toán

Thuật toán phân cụm dữ liệu dựa DBSCAN kiểm soát thông số Eps của mỗi điểm dữ liệu. Nếu như số Eps của một điểm p chứa nhiều hơn $MinPts$ thì một cụm mới với điểm p nòng cốt được thiết lập. Sau đó lặp lại việc tập hợp các đối tượng trực tiếp từ đối tượng nòng cốt này. Thuật toán dừng khi không còn điểm mới nào được thêm vào trong bất kỳ cụm nào.

2.3.3 Độ phức tạp thuật toán

Độ phức tạp của thuật toán DBSCAN là $O(n \times$thời gian tìm các đối tượng $Eps)$. Trong đó n là số đối tượng cần phân cụm. Trong trường hợp xấu nhất thì độ phức tạp sẽ là $O(n^2)$.

2.3.4 Ưu nhược điểm

Thuật toán DBSCAN đã khắc phục được vấn đề độ phức tạp tính toán cao và dữ liệu nhiễu. Nhưng để có thể tìm ra cụm các đối tượng thì người ta vẫn phải chọn tham số Eps và $MinPts$ để tìm ra cụm chính xác. Các thiết lập tham số như vậy thường khó xác định, đặc biệt trong thế giới thực, khi sự thiết lập có sự khác biệt nhỏ có thể dẫn đến sự phân chia cụm là rất khác nhau.

2.4 Phân cụm mô hình EM

2.4.1 Ý tưởng

Thuật toán phân cụm EM được Dempster, Laird và Rubin công bố năm 1977. Thuật toán này tìm ra sự ước lượng về khả năng lớn nhất của các tham số trong mô hình xác suất. Nó được xem là thuật toán dựa trên mô hình hoặc là mở rộng của thuật toán k-means. Thuật toán EM gán các đối tượng cho các cụm dữ liệu đã cho theo xác suất phân phối thành phần của đối tượng đó. Phân phối xác suất thường được sử dụng là phân phối Gaussian với mục

đích là khám phá lặp các giá trị tốt cho các tham số của nó bằng hàm tiêu chuẩn là logarit khả năng của đối tượng dữ liệu, đây là hàm tốt để mô hình xác suất cho đối tượng dữ liệu.

2.4.2 Thuật toán

Thuật toán gồm có ba bước cơ bản là bước khởi tạo, tiếp đến là bước E bước này sẽ gán nhãn dữ liệu, cuối cùng là bước M là bước đánh giá các tham số của mô hình. Bước E và bước M sẽ được lặp lại khi mà thỏa mãn tiêu chuẩn hội tụ.

2.4.3 Ưu nhược điểm

Thuật toán EM có khả năng khám phá ra nhiều hình dạng cụm khác nhau. Tuy nhiên do thời gian lặp của thuật toán nên chi phí tính toán của thuật toán là cao. Đã có một số cải tiến cho thuật toán EM dựa trên tính chất của dữ liệu: có thể nén, sao lưu bộ nhớ, hủy bỏ. Trong các cải tiến này thì có thể nén khi không bị loại bỏ và thuộc về cụm quá lớn so với bộ nhớ, đối tượng được hủy bỏ khi biết chắc chắn nhãn của cụm, chúng sẽ được lưu lại trong các trường hợp còn lại.

2.5 Kết luận

Chương này đã trình bày bốn thuật toán phân cụm cơ bản là thuật toán K-Means, thuật toán Phân cụm phân cấp Hierarchical Clustering, thuật toán phân cụm theo mật độ DBSCAN, thuật toán phân cấp theo mô hình EM.

CHƯƠNG 3: SO SÁNH MỘT SỐ THUẬT TOÁN PHÂN CỤM DỮ LIỆU

3.1 Phần mềm sử dụng WEKA

3.1.1. Giới thiệu về Weka và lịch sử phát triển

- Weka là phần mềm khai phá dữ liệu do các nhà khoa học thuộc trường Đại học Waikato, New Zealand khởi xướng và xây dựng. Weka là phần mềm mã nguồn mở, với mục tiêu xây dựng một công cụ hiện đại nhằm phát triển các kỹ thuật học máy và áp dụng chúng vào bài toán khai thác dữ liệu trong thực tế. Weka cung cấp nhiều giải thuật khác nhau với nhiều phương thức cho quá trình xử lý để ước lượng kết quả bằng sơ đồ cho bất kì một dữ liệu nào.

3.1.2 Các chức năng chính, thuật toán, dữ liệu của WEKA

- Chức năng chính

+ Khảo sát dữ liệu

+ Thực nghiệm mô hình

+ Biểu diễn trực quan dữ liệu bằng nhiều dạng đồ thị khác nhau.

- Cung cấp rất nhiều thuật toán phân lớp, được gom thành các nhóm dựa trên cơ sở lý thuyết hoặc chức năng.

- Cung cấp các thuật toán gom nhóm phổ biến: DBSCAN, EM, K-Means

- Cung cấp các thuật toán khai thác luật kết hợp: Apriori, PredictiveApriori

- Dữ liệu có thể được nhập vào từ một tập tin có khuôn dạng: ARFF, CSV hoặc cũng có thể được đọc vào từ một địa chỉ URL, hoặc từ một cơ sở dữ liệu thông qua JDBC.

3.1.3 Môi trường chính

- Explorer: Môi trường cho phép sử dụng tất cả các khả năng của WEKA để khám phá dữ liệu.

- Experimenter: Môi trường cho phép tiến hành các thí nghiệm và thực hiện các kiểm tra thống kê giữa các mô hình học máy.

- KnowledgeFlow: Môi trường cho phép bạn tương tác đồ họa kiểu kéo/thả để thiết kế các bước (các thành phần) của một thí nghiệm.

- Simple CLI Giao diện đơn giản kiểu dòng lệnh (như MS-DOS).

Hình 3.1: Giao diện khởi đầu của WEKA

3.2 Giới thiệu về bộ dữ liệu gốc trong WEKA

Bộ dữ liệu sử dụng để phân cụm trong luận văn này là bộ dữ liệu chuẩn Bank.arff và Glass.arff của phần mềm Weka.

Bảng 3.1: Số thuộc tính và đối tượng của các bộ dữ liệu.

Bộ dữ liệu	Số thuộc tính	Số đối tượng
Bank.arff	11	600
Glass.arff	10	214

Tệp dữ liệu Bank.arff

Như đã giới thiệu ở trên về khuôn dạng của tệp .arff, luận văn phân tích dữ liệu Bank.arff có 11 thuộc tính và 600 khách hàng gồm có ba phần:

- Phần định nghĩa quan hệ:

@relation bank => Định nghĩa mối quan hệ về ngân hàng (bank).

- Phần định nghĩa các thuộc tính:

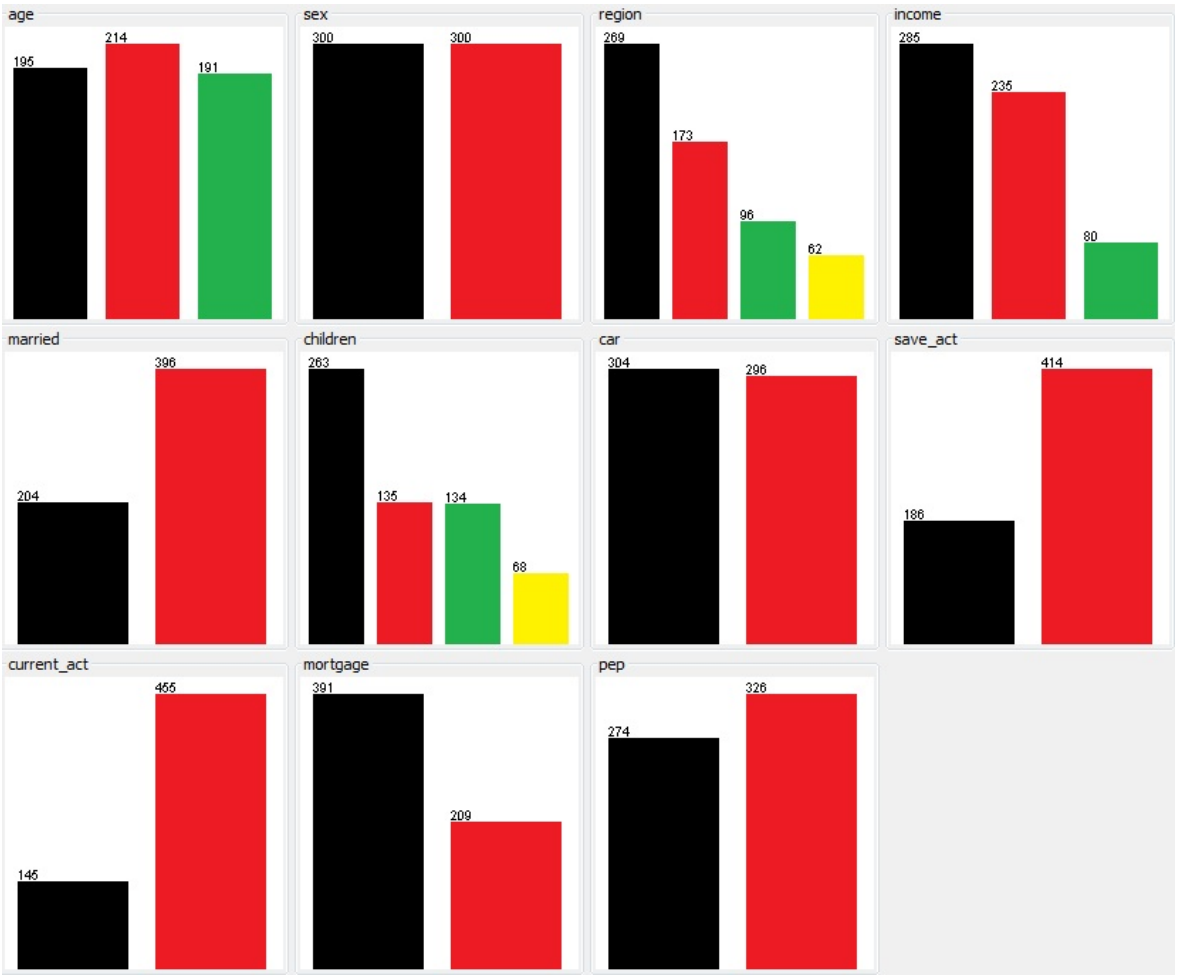
```
@relation bank

@attribute age {0_34,35_51,52_max}
@attribute sex {FEMALE,MALE}
@attribute region {INNER_CITY,TOWN,RURAL,SUBURBAN}
@attribute income {0_24386,24387_43758,43759_max}
@attribute married {NO,YES}
@attribute children {0,1,2,3}
@attribute car {NO,YES}
@attribute save_act {NO,YES}
@attribute current_act {NO,YES}
@attribute mortgage {NO,YES}
@attribute pep {YES,NO}

@data

35_51,FEMALE,INNER_CITY,0_24386,NO,1,NO,NO,NO,NO,YES
35_51,MALE,TOWN,24387_43758,YES,3,YES,NO,YES,YES,NO
52_max,FEMALE,INNER_CITY,0_24386,YES,0,YES,YES,YES,NO,NO
0_34,FEMALE,TOWN,0_24386,YES,3,NO,NO,YES,NO,NO
52_max,FEMALE,RURAL,43759_max,YES,0,NO,YES,NO,NO,NO
52_max,FEMALE,TOWN,24387_43758,YES,2,NO,YES,YES,NO,YES
0_34,MALE,RURAL,0_24386,NO,0,NO,NO,YES,NO,YES
52_max,MALE,TOWN,24387_43758,YES,0,YES,YES,YES,NO,NO
35_51,FEMALE,SUBURBAN,24387_43758,YES,2,YES,NO,NO,NO,NO
52_max,MALE,TOWN,0_24386,YES,2,YES,YES,YES,NO,NO
52_max,FEMALE,TOWN,43759_max,YES,0,NO,YES,YES,NO,NO
52_max,FEMALE,INNER_CITY,24387_43758,NO,0,YES,YES,YES,YES,NO
35_51,FEMALE,TOWN,0_24386,YES,1,NO,YES,YES,YES,YES
52_max,FEMALE,TOWN,43759_max,YES,1,YES,YES,YES,YES,YES
35_51,MALE,RURAL,0_24386,YES,0,NO,YES,YES,YES,NO
35_51,FEMALE,INNER_CITY,0_24386,YES,0,YES,YES,YES,YES,NO
```

Hình 3.2: Dữ liệu file Bank.arff



Hình 3.3: Phân bố dữ liệu Bank.arff theo các thuộc tính

Tệp dữ liệu Glass.arff

Tương tự như dữ liệu Bank.arff, dữ liệu Glass.arff thể hiện dữ liệu về các loại cốc thủy tinh. Dữ liệu này gồm 10 thuộc tính với 214 bản ghi. Cụ thể:

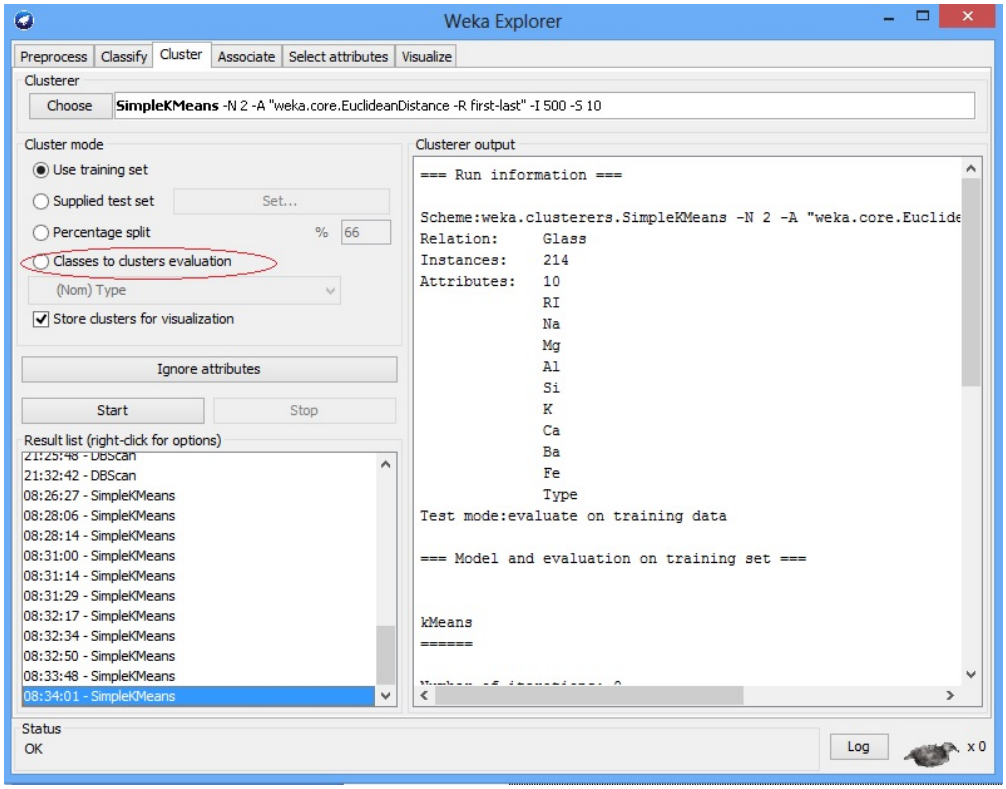
- Thuộc tính RI thể hiện chỉ số khúc xạ từ 1.5112 đến 1.5339.
- Thuộc tính Na: Phần trăm hàm lượng Natri trong cốc, từ 10.73 đến 17.38.
- Thuộc tính Mg: Phần trăm hàm lượng Magie trong cốc, từ 0 đến 4.49.
- Thuộc tính Al: Phần trăm hàm lượng Nhôm trong cốc, từ 0.29 đến 3.5.
- Thuộc tính Si: Phần trăm hàm lượng Silic trong cốc, từ 69.81 đến 75.41.
- Thuộc tính K: Phần trăm hàm lượng Kali trong cốc, từ 0 đến 6.21.
- Thuộc tính Ca: Phần trăm hàm lượng Canxi trong cốc, từ 5.43 đến 16.19.
- Thuộc tính Ba: Phần trăm hàm lượng Bari trong cốc, từ 0 đến 3.15.
- Thuộc tính Fe: Phần trăm hàm lượng Magie trong cốc, từ 0 đến 4.49.

- Thuộc tính Type: Thể hiện kiểu của loại cốc đó, gồm các giá trị: 'build wind float', 'build wind non-float', 'vehic wind float', 'vehic wind non-float', containers, tableware, headlamps.

3.3 So sánh và đánh giá kết quả

3.3.1. Đánh giá kết quả trên từng thuật toán riêng rẽ

Để kiểm tra độ chính xác của việc phân cụm, luận văn sử dụng thuộc tính cuối cùng trong 2 bộ dữ liệu để kiểm tra. Thuộc tính cuối chính là thuộc tính phân lớp mẫu có sẵn của dữ liệu. Để sử dụng chức năng kiểm tra độ chính xác này, luận văn thực hiện thao tác chọn chức năng *classes to clusters evaluation* trong vòng tròn đỏ như hình 3.4 dưới đây:



Hình 3.4: Chọn thuộc tính kiểm tra phân cụm

- **Thuật toán DBSCAN:** đầu vào của thuật toán DBSCAN bao gồm hai tham số *Eps* và *MinPts*. Luận văn sẽ tiến hành thực nghiệm với giá trị của hai tham số *Eps* và *MinPts* thay đổi để tìm ra bộ tham số tốt nhất cho bài toán này.

Bảng 3.2: Dữ liệu Bank.arff chạy thuật toán DBSCAN

STT	Tham số		Số cụm	Số điểm được phân	Số điểm không được phân cụm	Độ chính xác	Thời gian
	Eps	*MinPts*					
1	0.1 -> 1	1	543	600	0	1.2%	2.08
2	0.1 -> 1	2	532	126	474	1,2%	2.17
3	0.1 -> 1	3	58	126	474	1,2%	2.31
4	0.1 -> 1	4	8	26	574	1,2%	2.19
5	0.1 -> 1	5	1	5	595	0,7%	2.3
6	1.1->1.4	1	1	5	595	0,7%	2.55
7	1.1->1.4	2	105	600	0	43,5%	2.28
8	1.1->1.4	3	26	521	79	43,5%	2.7
9	1.1->1.4	4	8	376	224	43,5%	2.32
10	1.1->1.4	5	6	448	152	42,0%	2.18
11	1.1->1.4	6	4	407	193	39,7%	2.22
12	1.1->1.4	7	2	359	241	36,3%	2.27
13	1.1->1.4	8	6	299	301	29,3%	2.29
14	1.1->1.4	9	3	230	370	25,2%	2.23
15	1.1->1.4	10	2	198	402	22,7%	2.26
16	1.5->1.7	1	2	174	426	20,8%	2.28
17	1.5->1.7	2,3	532	126	474	1,2%	2.19
18	1.5->1.7	4	58	126	474	1,2%	2.2
19	1.5->1.7	5	8	26	574	1,2%	2.21
20	1.5->1.7	6	1	5	595	0,7%	2.19
21	1.5->1.7	7	1	5	595	0,7%	2.34
22	1.5->1.7	8	105	600	0	43,5%	2.27
23	1.5->1.7	9	26	521	79	43,5%	2.31
24	1.5->1.7	10	8	376	224	43,5%	2.25
25	> 1.7	1-10	6	448	152	42,0%	2.21

Từ bảng dữ liệu 3.2 có thể nhận thấy, với tham số *Eps* từ 0.1 đến 1 thì thuật toán DBSCAN với bộ dữ liệu này phân cụm không tốt.

Khi tăng giá trị *Eps* lên trong khoảng từ 1.1 đến 1.4 thì kết quả phân cụm tốt nhất của thuật toán với bộ dữ liệu Bank.arff.

Tiếp tục thực nghiệm với việc tăng giá trị *Eps* từ 1.5 trở đi thì độ chính xác của thuật toán với bộ dữ liệu này vẫn chỉ giữ nguyên ở mức 43.5% hoặc thấp hơn với việc xuất hiện nhiều hơn các phần tử không được phân cụm.

Tương tự như vậy, luận văn tiến hành thực nghiệm với dữ liệu Glass.arff. Bảng 3.3 đã cho thấy kết quả đầu ra tốt nhất là 19 cụm với 214 điểm dữ liệu được phân cụm, đạt độ chính xác cao nhất là 36.4%.

Bảng 3.3: Dữ liệu Glass.arff chạy thuật toán DBSCAN

STT	Tham số		Số cụm	Số điểm được phân	Số điểm không được phân cụm	Độ chính xác	Thời gian
	Eps	*MinPts*					
1	0.1	1	103	214	0	22%	0.27
2	0.1	2	17	128	86	21.9%	0.36
3	0.1	3	10	114	100	21%	0.34
4	0.1	4	7	72	142	7.5%	0.35
5	0.1	5	4	60	154	7%	0.23
6	0.1	6	3	55	159	7.5%	0.24
7	0.1	7	2	49	165	5.1%	0.35
8	0.1	>= 8	2	48	166	6.5%	0.24
9	0.5	1	19	214	0	36.4%	0.23
10	0.5	2	8	203	11	33.2%	0.36
11	0.5	>= 3	6	199	15	31.3%	0.23
12	1.1	1	3	214	0	36%	0.28
13	1.1	2	2	213	1	36%	0.28
14	1.1	3-10	1	214	0	36.4%	0.28
15	>1.2	1-10	1	214	0	35.5%	0.28

- Thuật toán K-Means:

Trong Weka, thuật toán Simple K-means chỉ hỗ trợ hai hàm để đo khoảng cách giữa các điểm là hàm Euclidean, Manhattan. Trong thực nghiệm này luận văn sử dụng hàm Euclidean. Tham số seed được sử dụng để sinh ra số ngẫu nhiên chọn các tâm cụm ban đầu để khởi tạo thuật toán. Trong thuật toán này luận văn sử dụng số seed cố định bằng 1 và thay đổi số cụm.

Bảng 3.4: Kết quả của thuật toán Kmeans với hai bộ dữ liệu

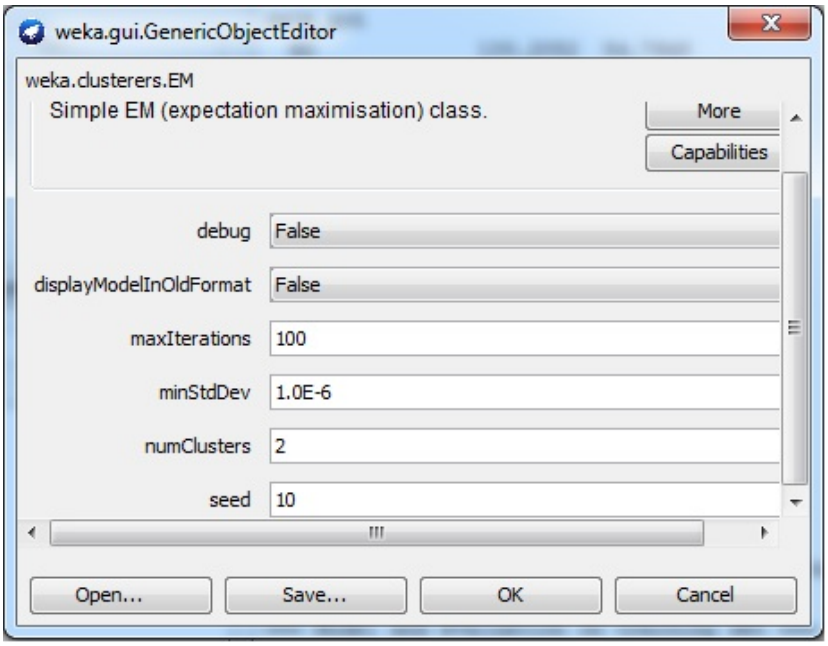
Dữ liệu / Số cụm	Bank.arff			Glass.arff		
	Bình phương sai số	Độ chính xác	Thời gian (s)	Bình phương sai số	Độ chính xác	Thời gian (s)
2	2280	53.5%	0.06	49.95	37%	0.06
3	2161	44.3%	0.02	29.14	48%	0.03
4	2051	39%	0.22	26.95	43.6%	0.03
5	1971	35.1%	0.06	24.99	41.1%	0.03
6	1886	29.3%	0.06	19.77	45%	0.07
7	1791	24%	0.1	18.97	49.1%	0.05
8	1754	22.5%	0.06	17.99	42%	0.1
9	1714	20.7%	0.05	17.03	40.7%	0.12
10	1627	18.7%	0.08	15.91	41.2%	0.07
11	1598	18%	0.07	14.38	44%	0.15
12	1543	17%	0.07	13.57	43.5%	0.12
13	1519	17%	0.05	11.85	40%	0.06

Đối với bộ dữ liệu Bank.arff thì kết quả tốt nhất khi phân cụm là 2 và đối với bộ dữ liệu Glass.arff số cụm phân chia cho độ chính xác tốt nhất là 7 cụm.

- **Thuật toán EM:**

EM cũng là một thuật toán quan trọng trong khai phá dữ liệu. Chúng ta sử dụng thuật toán này khi chúng ta không hài lòng với kết quả của thuật toán K-Means. Bản chất của thuật toán EM là một thuật toán lặp nhằm tìm ra độ đo likelihood lớn nhất hoặc tối đa ước

tính các thông số trong mô hình thống kê, nơi các mô hình phụ thuộc vào các biến tiềm ẩn không quan sát được.



Hình 3.5: Đầu vào của thuật toán EM trên weka

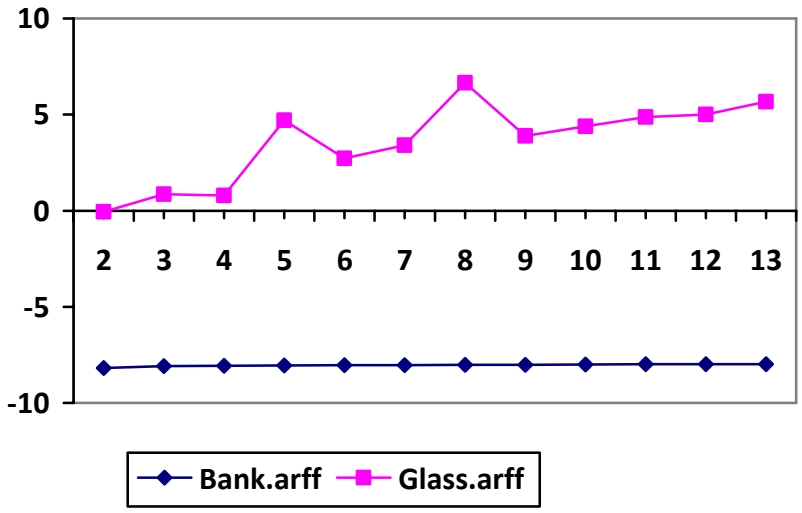
Đối với thuật toán EM, luận văn sử dụng số seed bằng 10, số cụm thay đổi, tham số minStdDev = 1.0E-6, maxIterations = 100.

Tiến hành thực nghiệm thuật toán EM trên Weka với tham số như hình 3.5 tôi thu được bảng dữ liệu sau:

Bảng 3.5: Kết quả chạy thuật toán EM với hai bộ dữ liệu

Số cụm	Bank.arff			Glass.arff		
	Likelihood	Độ chính xác	Thời gian	Likelihood	Độ chính xác	Thời gian
2	-8.1917	57.7%	0.53	-0.05201	43%	0.16
3	-8.091	47.3%	0.66	0.8598	46.7%	0.27
4	-8.0702	41.3%	0.68	0.7964	49.1%	0.23
5	-8.0553	28.5%	1.03	4.7094	44.9%	0.42
6	-8.0418	26.3%	1.67	2.7192	44.9%	0.38
7	-8.0296	24.9%	1.71	3.4097	44%	0.37
8	-8.0174	31.3%	1.75	6.6514	44.9%	0.4
9	-8.0219	22.5%	1.72	3.8852	41.6%	0.45
10	-8.0039	29.7%	2.03	4.3839	42.1%	0.5
11	-7.9863	22%	2.19	4.8702	37%	0.49
12	-7.9878	20.5%	1.76	5.0066	42%	0.54
13	-7.9866	19%	2.55	5.6651	40%	0.71

Từ bảng 3.5 có thể nhận thấy, đối với bộ dữ liệu Bank.arff, khi số cụm tăng lên thì giá trị likelihood cũng tăng theo. Tuy nhiên đối với bộ dữ liệu Glass.arff thì điều này lại không đúng.



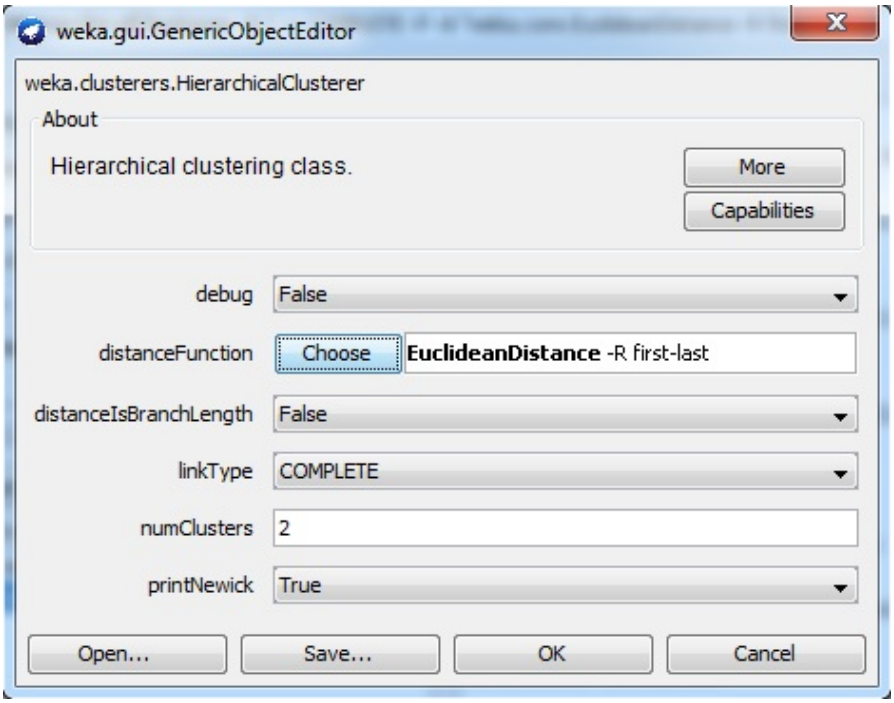
Hình 3.6: Biểu đồ giá trị likelihood với số cụm khác nhau.

So sánh với độ chính xác khi phân lớp thì số cụm cho giá trị likelihood tốt nhất chưa chắc đã cho giá trị độ chính xác tốt nhất. Độ chính xác tốt nhất của bộ dữ liệu Bank.arff tốt nhất đối với 2 cụm là 57.7% còn đối với bộ dữ liệu Glass.arff là 4 cụm với độ chính xác là 49.1%.

- Thuật toán Hierarchical Clustering (HC):

Đây là kỹ thuật phân cụm theo thứ bậc. Đầu vào của thuật toán này bao gồm số cụm cần phân, hàm khoảng cách, kiểu liên kết. Với hàm khoảng cách, luận văn vẫn sử dụng hàm khoảng cách Euclidean, còn về kiểu liên kết luận văn sử dụng các kiểu liên kết khác nhau để đánh giá thuật toán. Các kiểu liên kết trong Weka gồm có: kiểu Single, complete, average, mean, centroid, ward, adjcomplete, neighbor_joining. Tuy nhiên trong số các kiểu liên kết trên, các kiểu single, complete, average và centroid là được dùng nhiều hơn cả.

Các tham số đầu vào của thuật toán sẽ được lựa chọn thông qua hộp hội thoại của phần mềm Weka như hình 3.11.

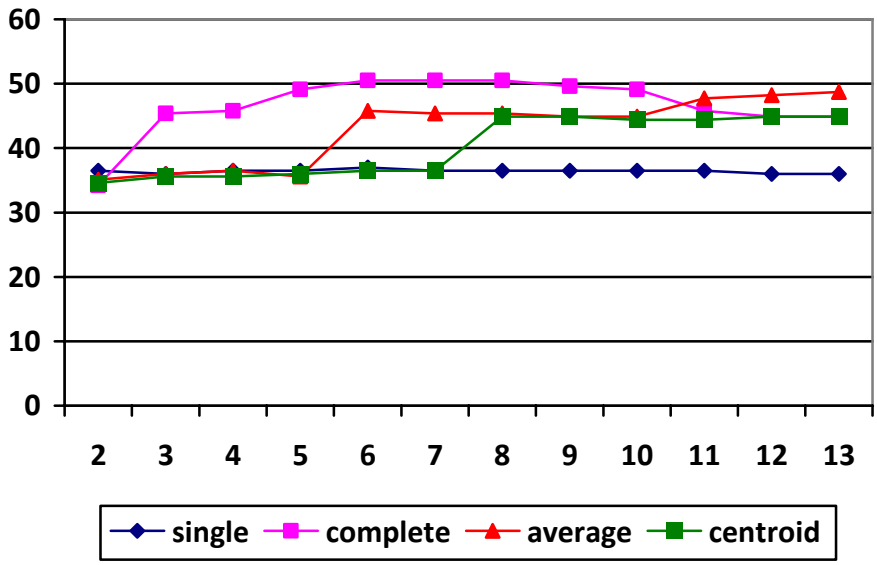


Hình 3.11: Đầu vào của thuật toán Hierarchical Clusterer

Bảng 3.6: Kết quả thực nghiệm thuật toán HC với bộ dữ liệu Glass.arff:

Liên kết / Số cụm	Single (s)	Complete (s)	Average (s)	Centroid (s)
2	36.5% (0.34)	34.2% (0.25)	35.1% (0.23)	34.6% (0.37)
3	36% (0.42)	45.4% (0.22)	36% (0.5)	35.6% (0.61)
4	36.5% (0.36)	45.8% (0.27)	36.5% (0.23)	35.6% (0.37)
5	36.5% (0.39)	49.1% (0.27)	35.6% (0.25)	36% (0.28)
6	37% (0.29)	50.5% (0.23)	45.8% (0.23)	36.5% (0.31)
7	36.5% (0.31)	50.5% (0.27)	45.4% (0.25)	36.5% (0.28)
8	36.5% (0.33)	50.5% (0.25)	45.4% (0.19)	44.9% (0.34)
9	36.5% (0.27)	49.6% (0.19)	44.9% (0.23)	44.9% (0.28)
10	36.5% (0.3)	49.1% (0.19)	44.9% (0.23)	44.4% (0.3)
11	36.5% (0.38)	45.8% (0.2)	47.7% (0.28)	44.4% (0.33)
12	36% (0.32)	44.9% (0.16)	48.2% (0. 2)	44.9% (0.36)
13	36% (0.27)	44.9% (0.23)	48.7% (0.2)	44.9% (0.58)

Bảng 3.6 thể hiện kết quả chạy thuật toán HC với bộ dữ liệu Glass.arff. Nhìn chung chất lượng cụm của cả bốn kiểu liên kết với số cụm khác nhau biến động không lớn.

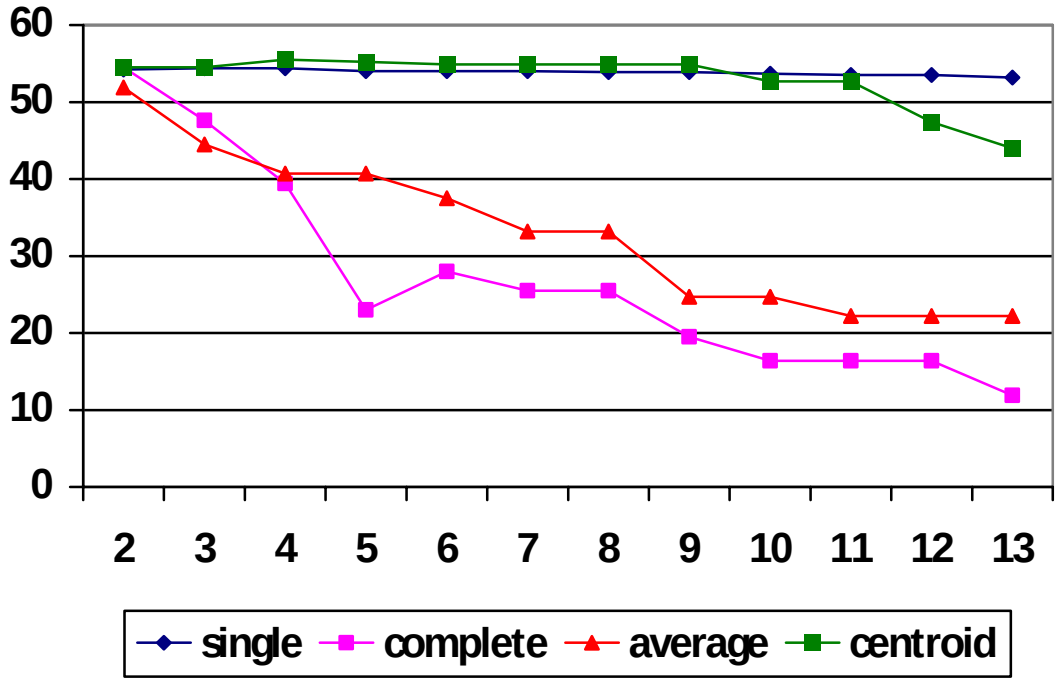


Hình 3.12: So sánh về chất lượng cụm với 4 kiểu liên kết của dữ liệu Glass.arff

Hình 3.12 chỉ ra đối với bộ dữ liệu này, kiểu liên kết single tỏ ra không hiệu quả bằng ba kiểu liên kết còn lại.

Bảng 3.7: Kết quả thực nghiệm thuật toán HC với bộ dữ liệu Bank.arff:

Liên kết / Số cụm	Single (s)	Complete (s)	Average (s)	Centroid (s)
2	54.2% (2.12)	54.5% (0.7)	51.9% (0.64)	54.5% (1.78)
3	54.4% (2.04)	47.6% (0.73)	44.5% (0.62)	54.5% (1.56)
4	54.4% (1.79)	39.4% (0.89)	40.7% (0.61)	55.5% (1.93)
5	54% (2.12)	23% (0.75)	40.7% (0.75)	55.2% (1.61)
6	54% (2.27)	28% (0.7)	37.5% (0.64)	54.9% (1.45)
7	54% (2)	25.5% (0.89)	33.2% (0.58)	54.9% (1.31)
8	53.9% (2.05)	25.5% (0.72)	33.2% (0.58)	54.9% (1.31)
9	53.9% (2.22)	19.5% (0.72)	24.7% (0.59)	54.9% (1.26)
10	53.7% (2)	16.4% (0.73)	24.7% (0.58)	52.7% (1.2)
11	53.5% (1.95)	16.4% (0.72)	22.2% (0.57)	52.7% (1.26)
12	53.5% (2.02)	16.4% (0.72)	22.2% (0.68)	47.4% (1.19)
13	53.2% (2.2)	11.9% (0.67)	22.2% (0.59)	44% (1.19)



Hình 3.13: So sánh về chất lượng cụm với 4 kiểu liên kết của dữ liệu Bank.arff

3.3.1. So sánh và đánh giá kết quả trên cả bốn thuật toán

Sau khi đánh giá riêng rẽ từng thuật toán, luận văn tiến hành đánh giá cả bốn thuật toán với nhau. Tiêu chí đánh giá của bốn thuật toán này là độ chính xác của thuật toán so với thuộc tính phân loại và thời gian thực hiện thuật toán. Về độ chính xác, luận văn sẽ chọn kết quả phân cụm cho độ chính xác tốt nhất của từng thuật toán làm giá trị để so sánh. Các kết quả chạy của bốn thuật toán được thể hiện trong bảng 3.8

Bảng 3.8: Kết quả thực nghiệm của bốn thuật toán trên hai bộ dữ liệu:

Thuật toán		DBSCAN	EM	KMEANS	HC
Bank.arff	Độ chính xác	43.5%	57.7%	53.5%	55.5%
	Số cụm	105	2	2	4
	Thời gian (s)	2.28	0.53	0.06	1.93
Glass.arff	Độ chính xác	36.4%	49.1%	49.1%	50.5%
	Số cụm	19	4	7	6,7,8
	Thời gian (s)	0.23	0.23	0.05	0.25

Rõ ràng với hai bộ dữ liệu này, thuật toán DBSCAN tỏ ra yếu thế hơn so với ba thuật toán còn lại. Thuật toán KMEANS cho thời gian chạy nhanh nhất tuy nhiên thuật toán EM lại cho độ chính xác tốt nhất đối với bộ dữ liệu Bank.arff và thuật toán HC cho kết quả phân cụm với chất lượng cụm tốt nhất đối với bộ dữ liệu Glass.arff.

3.4 Kết luận

Chương 3 đã trình bày về phần mềm Weka, bộ dữ liệu sử dụng và một số thực nghiệm trên bốn thuật toán đề xuất là K-Means, EM, Hierarchical Clusterer, DBSCAN. Đồng thời chương này cũng giới thiệu về bộ dữ liệu Bank.arff và Glass.arff đều là các bộ dữ liệu mẫu chuẩn của phần mềm Weka. Tiếp đó, luận văn tiến hành chạy thực nghiệm và đánh giá độ hiệu quả của cả bốn thuật toán này. Kết quả thực nghiệm cho thấy thuật toán DBSCAN cho kết quả phân cụm chậm nhất, thuật toán K-Means cho kết quả phân cụm nhanh nhất. Tuy nhiên thuật toán cho độ chính xác phân cụm hay chất lượng cụm tốt nhất lại thuộc về thuật toán EM với bộ dữ liệu Bank.arff và thuật toán HC với bộ dữ liệu Glass.arff.

KẾT LUẬN

Khai phá dữ liệu và đặc biệt là phân cụm dữ liệu ngày càng đóng vai trò quan trọng trong các ứng dụng ngày nay như thương mại điện tử, ngân hàng, bảo hiểm, chứng khoán, v.v. Phân cụm dữ liệu nhằm mục tiêu chính là gom các đối tượng tương đồng vào cùng một nhóm để từ đó phục vụ rút trích ra tri thức cho các hệ hỗ trợ quyết định về sau. Nội dung của luận văn tập trung chính vào tìm hiểu các kiến thức cơ bản trong phân cụm dữ liệu và đi sâu vào tìm hiểu, thực nghiệm phân cụm dựa trên công cụ khai phá dữ liệu mã nguồn mở được sử dụng phổ biến hiện nay là Weka để tiến hành so sánh đánh giá các thuật toán với nhau. Luận văn đã đạt được một số kết quả sau đây:

- Luận văn đã trình bày tổng quan về phân cụm dữ liệu với các khái niệm, các ứng dụng và một số phương pháp phân cụm dữ liệu.
- Khảo sát bốn thuật toán toán phân cụm dữ liệu được sử dụng chính hiện nay là thuật toán K-Means, DBSCAN, EM, Hierarchical Clusterer. Các thuật toán này được trình bày chi tiết từ ý tưởng, thuật toán đến độ phức tạp cũng như ưu nhược điểm.
- Luận văn đã tiến hành thực nghiệm chạy các thuật toán này trên phần mềm Weka cho hai bộ dữ liệu mẫu là Bank.arff và Glass.arff. Các kết quả thu được cho thấy thuật toán K-Means cho tốc độ tính toán nhanh nhất song thuật toán cho chất lượng cụm tốt nhất lại thuộc về EM với bộ dữ liệu Bank.arff và HC với bộ dữ liệu Glass.arff. Thuật toán DBSCAN tỏ ra kém hiệu quả đối với hai bộ dữ liệu này.

Hướng phát triển

- Luận văn sẽ tiếp tục nghiên cứu một số ứng dụng của các thuật toán phân cụm trong Weka.
- Thực nghiệm trên các tập dữ liệu mới, lớn hơn, thực tế hơn như phân cụm dữ liệu y tế, chứng khoán, tài chính v.v.