|  |  |  |
| --- | --- | --- |
|  | BAN CƠ YẾU CHÍNH PHỦ  “HỌC VIỆN KỸ THUẬT MẬT MÃ” | Mẫu 2 |

BÁO CÁO CHUYÊN ĐỀ SỐ 2.1

“Nghiên cứu một số thuật toán phân cụm tiêu biểu K-means, DBSCAN, FP-growth”

NHIỆM VỤ: “Nghiên cứu và ứng dụng nền tảng học sâu để xây dựng hệ thống phát hiện mã độc trực tuyến”.

Mã số: 06/2022/CB.

Cơ quan chủ trì: Học viện Kỹ thuật Mật mã

Chủ nhiệm: ThS. Lê Đức Thuận

Hà Nội - 2022

|  |  |  |
| --- | --- | --- |
|  | BAN CƠ YẾU CHÍNH PHỦ  “HỌC VIỆN KỸ THUẬT MẬT MÃ” |  |

BÁO CÁO CHUYÊN ĐỀ SỐ 2.1

“Nghiên cứu một số thuật toán phân cụm tiêu biểu K-means, DBSCAN, FP-growth”

NHIỆM VỤ: “Nghiên cứu và ứng dụng nền tảng học sâu để xây dựng hệ thống phát hiện mã độc trực tuyến”.

Mã số: 06/2022/CB.

Cơ quan chủ trì: Học viện Kỹ thuật Mật mã

Chủ nhiệm: ThS. Lê Đức Thuận

|  |  |
| --- | --- |
| **Người thực hiện chuyên đề** | **Cơ quan chủ trì** |
| *(Họ tên và chữ ký)* | *(Họ tên và chữ ký)* |

Hà Nội - 2022

MỤC LỤC

[MỤC LỤC 1](#_Toc115870730)

[DANH MỤC HÌNH ẢNH 2](#_Toc115870731)

[1. GIỚI THIỆU CHUNG VỀ CÁC THUẬT TOÁN PHÂN CỤM 3](#_Toc115870732)

[1.1 Các vấn đề của việc phân tích cụm 4](#_Toc115870733)

[1.2 Yêu cầu của phân cụm trong khai phá dữ liệu 4](#_Toc115870734)

[1.3 Phân loại các phương pháp phân cụm 5](#_Toc115870735)

[2. THUẬT TOÁN K-MEANS 9](#_Toc115870736)

[2.1. Giới thiệu thuật toán 9](#_Toc115870737)

[2.2 Giá trị khởi đầu cho thuật toán K-means 10](#_Toc115870738)

[2.3 Các cách làm cho thuật toán K-means có thể mở rộng hơn 11](#_Toc115870739)

[2.4 Điểm mạnh và điểm yếu của thuật toán K-means 11](#_Toc115870740)

[2.5 Độ phức tạp không gian và thời gian 13](#_Toc115870741)

[3. THUẬT TOÁN FP-GROWTH 14](#_Toc115870742)

[3.1 Giới thiệu thuật toán 14](#_Toc115870743)

[3.2 Giải thuật xây dựng FP tree 14](#_Toc115870744)

[3.3 Các thách thức của thuật toán phân cụm dựa trên tần xuất mẫu 15](#_Toc115870745)

[4. THUẬT TOÁN DBSCAN 19](#_Toc115870746)

[4.1 Ưu điểm và nhược điểm của DBSCAN 20](#_Toc115870747)

[4.2 Độ phức tạp không gian và thời gian của DBSCAN 20](#_Toc115870748)

[TÀI LIỆU THAM KHẢO 22](#_Toc115870749)

# DANH MỤC HÌNH ẢNH

[Hình 1.1: Phân loại các thuật toán phân cụm (Vẽ lại cho đủ nhóm) 7](#_Toc115870724)

[Hình 2.1: Thuật toán K-means 10](#_Toc115870725)

# 1. GIỚI THIỆU CHUNG VỀ CÁC THUẬT TOÁN PHÂN CỤM

Phân cụm (Clustering) là một kỹ thuật để nhóm các đối tượng dữ liệu theo cách mà các đối tượng dữ liệu của một nhóm tương tự với nhau trong nhóm đó và khác với các đối tượng dữ liệu của các nhóm khác. Phân cụm đang được thực hiện cho các mục đích đa dạng như phân tích dữ liệu thống kê và khám phá khai phá dữ liệu (Data mining). Các thuật toán này được áp dụng dựa trên các kiểu đối tượng dữ liệu, độ phức tạp của thuật toán và sự phù hợp với một nhóm cụ thể. Danh sách thuật toán cụm có thể bao gồm hơn một trăm thuật toán phân cụm đã xuất bản nhưng hầu hết chúng không cung cấp mô hình cho cụm. Trong bài báo [1], nhóm tác giả đã phân tích độ phức tạp của các thuật toán phân cụm được sử dụng nhiều nhất.

Phân loại và phân cụm là kỹ thuật quan trọng phân vùng các đối tượng có nhiều thuộc tính thành các nhóm con rời rạc có ý nghĩa để các đối tượng trong mỗi nhóm giống nhau hơn về giá trị của các thuộc tính của chúng so với các đối tượng trong nhóm khác. Có một sự khác biệt lớn giữa phân tích cụm và phân loại. Trong phân loại có giám sát, các lớp được xác định, người dùng đã biết có những lớp nào và một số dữ liệu đào tạo đã được gắn nhãn bởi thành viên lớp của họ có sẵn để đào tạo hoặc xây dựng một mô hình. Trong phân tích cụm, người ta không biết những lớp hoặc cụm nào tồn tại và vấn đề cần giải quyết là nhóm dữ liệu đã cho thành cụm có ý nghĩa. Cũng giống như ứng dụng của phân loại có giám sát, phân tích cụm có ứng dụng trong nhiều lĩnh vực khác nhau như trong marketing, y học, kinh doanh. Các ứng dụng thực tế của phân tích cụm cũng đã được tìm thấy trong nhận dạng ký tự, phân tích web và phân loại tài liệu, phân loại dữ liệu thiên văn và phân loại các đối tượng được tìm thấy trong một nghiên cứu khảo cổ học. Công việc phân cụm đã được thực hiện trong thống kê trong nhiều năm. Hầu hết các thuật toán được phát triển dựa trên một số khái niệm về sự giống nhau hoặc khoảng cách để các nhóm đối tượng giống nhau hoặc gần nhau và khác nhau hoặc xa các đối tượng khác có thể được đặt cùng nhau trong một cụm. Kỹ thuật này có vẻ đơn giản nhưng có thể khá khó khăn vì để tìm các đối tượng giống nhau giữa một tập hợp lớn các đối tượng đòi hỏi phải so sánh từng đối tượng với mọi đối tượng khác, điều này có thể rất tốn kém đối với các tập hợp lớn. Ngoài ra, chi phí tính toán khoảng cách giữa các đối tượng tăng lên khi số lượng các thuộc tính tăng lên. Cuối cùng, việc tính toán khoảng cách giữa các đối tượng có thuộc tính số sẽ dễ dàng hơn nhưng tính khoảng cách giữa các thuộc tính phân loại thì khó hơn. Các thuật toán phân cụm cổ điển được phát triển trong thống kê giả định các tập dữ liệu nhỏ nhưng với sự ra đời của máy tính, mã vạch và web, người dùng muốn áp dụng phân tích cụm cho các tập dữ liệu lớn. Do đó, các phương pháp cũ đã được sửa đổi để quản lý các tập dữ liệu lớn và các phương pháp mới đang được phát triển.

## 1.1 Các vấn đề của việc phân tích cụm

Các vấn đề đặt ra sau đây của việc phân tích cụm vẫn đang được nghiên cứu, phát triển:

* Tìm các tính năng mong muốn của phân tích cụm
* Mức độ tương đồng nên được đo lường như thế nào dựa trên các bộ dữ liệu đang được phân tích.
* Tìm phương pháp phân tích cụm tốt nhất cho tập dữ liệu đã cho
* Làm thế nào để một tập dữ liệu rất lớn được phân nhóm một cách hiệu quả
* Làm thế nào để dữ liệu với một số lượng lớn các thuộc tính được nhóm một cách hiệu quả
* Kết quả của phân tích cụm nên được đánh giá như thế nào

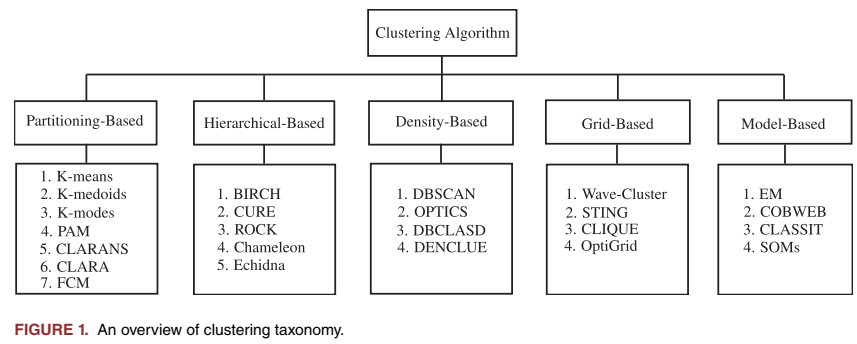
## 1.2 Yêu cầu của phân cụm trong khai phá dữ liệu

* Khả năng mở rộng
* Khả năng đối phó với các loại thuộc tính khác nhau
* Khám phá các cụm có hình dạng tùy ý
* Yêu cầu tối thiểu về kiến thức miền để xác định các tham số đầu vào
* Có khả năng đối phó với tiếng ồn và các yếu tố ngoại lai
* Không nhạy cảm với thứ tự của các bản ghi đầu vào
* Số chiều cao (High dimensionality)
* Kết hợp các ràng buộc do người dùng chỉ định
* Khả năng diễn giải (Interpretability) và khả năng sử dụng

## 1.3 Phân loại các phương pháp phân cụm

Các phương pháp phân cụm có thể được chia thành các loại sau đây:

* **Phương pháp phân vùng (Partitional method)**: Phương pháp phân vùng thu được một phân vùng cấp duy nhất của các đối tượng. Các phương pháp này thường dựa trên heuristics tham lam được sử dụng lặp đi lặp lại để có được giải pháp tối ưu cục bộ. Thuật toán phân cụm theo từng phần bao gồm Cây mở rộng tối thiểu (Minimum Spanning Tree), Thuật toán phân cụm lỗi bình phương (Squared Error Clustering Algorithm), phân cụm K-means, Thuật toán lân cận gần nhất (Nearest neighbor algorithm), thuật toán PAM (Partitioning Around Medoids), Thuật toán năng lượng liên kết (Bond Energy Algorithm), Phân cụm với thuật toán di truyền (Clustering with Genetic Algorithm), Phân cụm với mạng nơron (Clustering with Neural network).
* **Phương pháp phân cấp (Hierarchical method)**: Các phương thức phân cấp thu được một phân vùng lồng nhau của các đối tượng dẫn đến một cây các cụm. Phương pháp này hoặc bắt đầu với một cụm và sau đó chia thành các cụm nhỏ hơn (được gọi là chia hoặc từ trên xuống) hoặc bắt đầu với từng đối tượng trong một cụm riêng lẻ và sau đó cố gắng hợp nhất các cụm tương tự thành các cụm lớn hơn và lớn hơn (được gọi là kết hợp hoặc từ dưới lên). Có một số phân cụm khác thuộc loại này như BIRCH (Giảm lặp lại cân bằng và phân cụm bằng cách sử dụng phân cấp), ROCK (Thuật toán phân cụm thứ bậc cho các thuộc tính phân loại). Chameleon (Một thuật toán phân cụm phân cấp sử dụng mô hình động). Phương pháp dựa trên mật độ: Trong lớp phương pháp này, thường đối với mỗi điểm dữ liệu trong một cụm, ít nhất một số điểm tối thiểu phải tồn tại trong một bán kính nhất định. Các phương pháp dựa trên mật độ bao gồm DBSCAN (Phương pháp phân cụm dựa trên mật độ trên các vùng được kết nối với mật độ đủ cao), OPTICS (Điểm thứ tự để xác định cấu trúc phân cụm), DENCLUE (Phân cụm dựa trên chức năng phân bổ mật độ) [2].
* **Phương thức dựa trên lưới (Grid-based method):** Trong lớp phương thức này, không gian đối tượng chứ không phải dữ liệu được chia thành lưới. Phân vùng lưới dựa trên các đặc điểm của dữ liệu và các phương pháp như vậy có thể xử lý dữ liệu phi số dễ dàng hơn. Các phương pháp dựa trên lưới không bị ảnh hưởng bởi thứ tự dữ liệu. STING (Lưới thông tin thống kê - Statistical Information Grid), Cụm sóng (Phân cụm sử dụng chuyển đổi Wavelet - Wavelet Transformation), CLIQUE là một số ví dụ về phương pháp dựa trên lưới [2].
* **Phương pháp dựa trên mô hình (Model-based method):** Một mô hình được giả định, có lẽ dựa trên phân phối xác suất. Về cơ bản, thuật toán cố gắng xây dựng các cụm có mức độ tương tự cao bên trong chúng và mức độ tương tự thấp giữa chúng. Phép đo độ tương tự dựa trên các giá trị trung bình và thuật toán cố gắng giảm thiểu hàm lỗi. Kỳ vọng-Tối đa hóa (Expectation-Maximization), Phân cụm khái niệm (Conceptual Clustering) và Phương pháp tiếp cận Mạng Nơ-ron (Neural Network Approach) là ví dụ về các phương pháp dựa trên mô hình [3].
* **Phân cụm dữ liệu cao chiều (Clustering High-Dimensional Data):** CLIQUE (Phương pháp phân cụm không gian con tăng trưởng theo thứ nguyên - A Dimension-Growth Subspace Clustering Method), PROCLUS (Phương pháp phân cụm không gian con theo thứ nguyên-giảm - A Dimension-Reduction Subspace Clustering Method), Phương pháp phân nhóm dựa trên tần xuất mẫu (Frequent Pattern-Based Clustering Methods) được sử dụng cho dữ liệu cao chiều.
* **Phân tích cụm dựa trên ràng buộc (Constraint-Based Cluster Analysis):** Phân nhóm dựa trên ràng buộc tìm các cụm thỏa mãn các sở thích hoặc ràng buộc do người dùng chỉ định. Tùy thuộc vào bản chất của các ràng buộc, phân cụm dựa trên ràng buộc có thể áp dụng các cách tiếp cận khá khác nhau. Một số loại ràng buộc bao gồm ràng buộc đối với các đối tượng riêng lẻ, ràng buộc về việc lựa chọn tham số phân cụm, ràng buộc về khoảng cách hoặc các hàm tương tự [3].



Hình 1.1: Phân loại các thuật toán phân cụm (Vẽ lại cho đủ nhóm)

Phân tích cụm đã là một lĩnh vực nghiên cứu trong vài thập kỷ và có quá nhiều phương pháp khác nhau để tất cả được đề cập trong thời gian ngắn. Nhiều phương pháp mới vẫn đang được phát triển. Trong phần này, chúng tôi trình về một số thuật toán phân cụm phổ biến và được sử dụng nhiều nhất và trình bày độ phức tạp, ưu điểm và nhược điểm của chúng.

* **K-means:** K-means là phương pháp phân cụm cổ điển đơn giản và phổ biến nhất, dễ thực hiện. Cổ điển chỉ có thể được sử dụng nếu dữ liệu về tất cả các đối tượng nằm trong bộ nhớ chính. Phương thức này được gọi là K-means vì mỗi cụm K được biểu diễn bằng giá trị trung bình của các đối tượng bên trong nó [2].
* **DBSCAN** (Phương pháp phân cụm dựa trên mật độ trên các vùng được kết nối với mật độ đủ cao): DBSCAN (Density –Based Spatial Clustering of Applications with Noise - Phân cụm không gian dựa trên mật độ của các ứng dụng có nhiễu) là một thuật toán phân cụm dựa trên mật độ. Thuật toán phát triển các vùng có mật độ đủ cao thành các cụm và phát hiện ra các cụm có hình dạng tùy ý trong cơ sở dữ liệu không gian có nhiễu. Nó định nghĩa một cụm là một tập hợp tối đa các điểm được kết nối với mật độ [3]. Phương pháp này được thiết kế cho cơ sở dữ liệu không gian nhưng có thể được sử dụng trong các ứng dụng khác. Nó yêu cầu hai tham số đầu vào: kích thước của vùng lân cận (R) và các điểm tối thiểu trong vùng lân cận (N). Về cơ bản, hai tham số này xác định mật độ trong các cụm mà người dùng sẵn sàng chấp nhận vì chúng chỉ định có bao nhiêu điểm phải nằm trong vùng n. Số lượng điểm không chỉ xác định mật độ của các cụm có thể chấp nhận được mà nó còn xác định các đối tượng nào trong vùng lân cận của chúng. Tham số kích thước R xác định kích thước của các cụm được tìm thấy. Nếu R đủ lớn, sẽ có một cụm lớn và không có ngoại lai. Nếu R nhỏ, sẽ có các cụm nhỏ dày đặc và có thể có nhiều vùng ngoại lai.
* FP-Growth: Thuật toán FP-Growth xây dựng cấu trúc FP tree để lưu trữ toàn bộ cơ sở dữ liệu và khai phá FP tree để tìm các mẫu phổ biến. Việc nén tập dữ liệu vào cấu trúc cây (Frequent Pattern tree – FP-tree) làm giảm chi phí cho toàn tập dữ liệu trong quá trình khai phá, các infrequent items bị loại bỏ sớm mà vẫn đảm bảo kết quả khai phá không bị ảnh hưởng và tránh được việc tạo ra các tập dự tuyển trong mỗi lần kiểm tra dữ liệu.

Các mục sau đây sẽ trình bày chi tiết về ba thuật toán phân cụm này.

# 2. THUẬT TOÁN K-MEANS

Nhiều kết quả nghiên cứu sử dụng kỹ thuật phân cụm K-means được đưa ra [4, 5, 6], cho thấy mức độ ứng dụng của kỹ thuật này trong nhiều lĩnh vực: Y học (Phát hiện khối u não bằng cách sử dụng K-means dựa trên màu sắc) [6], Wirless network (phân cụm các cảm biến nhằm tối ưu năng lượng trong hoạt động và định tuyến ) [4], marketing (Xác định các nhóm khách hàng, khách hàng tiềm năng, khách hàng giá trị, phân loại và dự đoán hành vi khách hàng,v.v. sử dụng sản phẩm hay dịch vụ của công ty để giúp công ty có chiến lược kinh doanh hiệu quả hơn); Biology (Phân nhóm động vật và thực vật dựa vào các thuộc tính của chúng); Libraries (Theo dõi độc giả, sách, dự đoán nhu cầu của độc giả); Insurance, Finance (Phân nhóm các đối tượng sử dụng bảo hiểm và các dịch vụ tài chính, dự đoán xu hướng (trend) của khách hàng, phát hiện gian lận tài chính (identifying frauds)); WWW (Phân loại tài liệu (document classification); phân loại người dùng web (clustering weblog)); v.v. Tuy nhiên, để giải quyết những bài toán này với K-means đòi hỏi thêm độ phức tạp tính toán và các phần mềm chức năng.

## 2.1. Giới thiệu thuật toán

K-Means là thuật toán rất quan trọng và được sử dụng phổ biến trong kỹ thuật phân cụm. Tư tưởng chính của thuật toán K-Means là tìm cách phân nhóm các đối tượng (objects) đã cho vào K cụm (K là số các cụm được xác đinh trước, K nguyên dương) sao cho tổng bình phương khoảng cách giữa các đối tượng đến tâm nhóm (centroid) là nhỏ nhất.

Thuật toán K-Means được mô tả ở Hình 2.2, được thực hiện qua các bước chính sau:

1. Chọn số lượng cụm. Gọi con số này là K

2. Chọn K hạt làm tâm của K cụm. Các hạt giống có thể được chọn ngẫu nhiên trừ khi người dùng có một số thông tin chi tiết về dữ liệu.

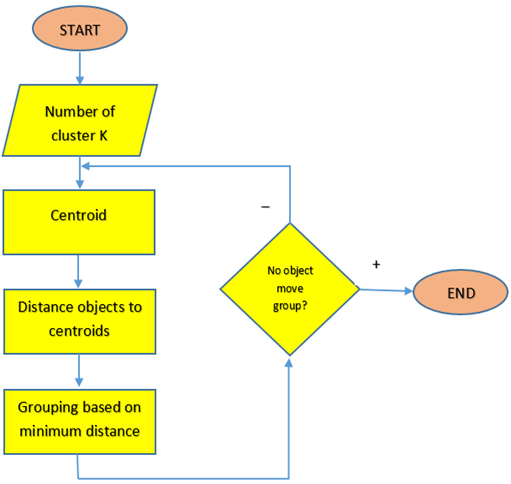
3. Tính khoảng cách Euclid của mỗi đối tượng trong tập dữ liệu từ mỗi tâm.

4. Phân bổ từng đối tượng cho cụm mà nó gần nhất để dựa trên tính toán khoảng cách ở bước trước.

5. Tính toán trọng tâm của các cụm bằng cách tính toán các giá trị thuộc tính của các đối tượng trong mỗi cụm.

6. Kiểm tra xem tiêu chí dừng đã được đáp ứng chưa (ví dụ: tư cách thành viên cụm không thay đổi) nếu có, hãy chuyển sang bước 7. Nếu không, hãy chuyển sang bước 3.

7. [tùy chọn] Người ta có thể quyết định dừng ở giai đoạn này hoặc tách một cụm hoặc kết hợp hai cụm theo phương pháp phỏng đoán cho đến khi đáp ứng được tiêu chí dừng.



Hình 2.1: Thuật toán K-means

## 2.2 Giá trị khởi đầu cho thuật toán K-means

Thường thì người dùng có ít cơ sở để chỉ định số lượng cụm và hạt giống khởi đầu. Vấn đề này có thể được khắc phục bằng cách sử dụng phương pháp lặp lại. Ví dụ, trước tiên người ta có thể chọn ba cụm và chọn ngẫu nhiên ba hạt khởi đầu. Khi đã thu được các cụm cuối cùng, quá trình này có thể được lặp lại với một nhóm hạt khác. Cần cố gắng chọn những hạt càng xa nhau càng tốt.

Ngoài ra, trong quá trình lặp lại nếu hai cụm được tìm thấy gần nhau, có thể mong muốn hợp nhất chúng. Ngoài ra, một cụm lớn có thể được chia làm hai nếu phương sai trong cụm lớn hơn một số giá trị ngưỡng. Một cách tiếp cận khác liên quan đến việc tìm kiếm centroid của toàn bộ tập dữ liệu và sau đó xoay vòng giá trị centroid này để tìm hạt giống. Tuy nhiên, một cách tiếp cận khác khuyến nghị sử dụng phương pháp phân cấp như phương pháp tổng hợp (agglomerative method) trên dữ liệu trước tiên, vì phương pháp đó không yêu cầu giá trị bắt đầu và sau đó sử dụng kết quả của phương pháp đó làm cơ sở để chỉ định số lượng cụm và hạt bắt đầu. Tuy nhiên, một cách tiếp cận khác dựa trên việc sử dụng một số mẫu nhỏ của dữ liệu đã cho và sử dụng các hạt ngẫu nhiên cho mỗi mẫu đơn giản. Kết quả phân tích cụm của các mẫu này có thể giúp tìm kiếm hạt giống cho toàn bộ dữ liệu [2].

## 2.3 Các cách làm cho thuật toán K-means có thể mở rộng hơn

Một cách tiếp cận gần đây để mở rộng thuật toán K-means dựa trên ý tưởng xác định ba loại vùng trong dữ liệu: vùng có thể nén, vùng phải được duy trì trong bộ nhớ chính và vùng có thể loại bỏ. Một đối tượng có thể bị loại bỏ nếu tư cách thành viên của nó trong một cụm được xác định chắc chắn. Một đối tượng có thể nén được nếu nó không thể bị loại bỏ nhưng thuộc về một cụm con chặt chẽ. Cấu trúc dữ liệu được gọi là một đặc trưng phân cụm (clustering feature) được sử dụng để tóm tắt các đối tượng đã bị loại bỏ hoặc nén. Nếu một đối tượng không thể loại bỏ cũng như không thể nén được, thì nó nên được giữ lại trong bộ nhớ chính. Để đạt được khả năng mở rộng, thuật toán phân cụm lặp (iterative clustering algorithm) có thể chỉ bao gồm các tính năng phân cụm của các đối tượng có thể nén và các đối tượng phải được giữ lại trong bộ nhớ chính, do đó biến thuật toán dựa trên bộ nhớ phụ thành thuật toán dựa trên bộ nhớ chính. Một cách tiếp cận thay thế để mở rộng thuật toán K-means khám phá ý tưởng phân cụm vi mô, trước tiên nhóm các đối tượng lân cận thành các cụm vi mô và sau đó thực hiện phân cụm K-mean trên các cụm vi mô [2].

## 2.4 Điểm mạnh và điểm yếu của thuật toán K-means

K-means rất đơn giản và có thể được sử dụng cho nhiều loại dữ liệu khác nhau. Nó cũng khá hiệu quả mặc dù thường thực hiện nhiều lần chạy. Một số biến thể bao gồm cả K-means phân đôi thậm chí còn hiệu quả hơn và ít bị các vấn đề khởi tạo hơn. Kmeans là một phương pháp tham lam lặp đi lặp lại (iterative greedy method). Thường cần một số lần lặp để hội tụ và do đó tập dữ liệu được xử lý một số lần.

Nếu dữ liệu rất lớn và không thể chứa trong bộ nhớ chính, quá trình này có thể trở nên kém hiệu quả. Mặc dù phương pháp K-means được biết đến và sử dụng rộng rãi nhất, nhưng có một số vấn đề liên quan đến phương pháp này cần được hiểu rõ.

1. Phương pháp K-means cần tính khoảng cách Euclide và phương tiện (means) của các giá trị thuộc tính của các đối tượng trong một cụm. Do đó, thuật toán cổ điển này chỉ phù hợp với dữ liệu liên tục. Các biến thể K-means xử lý dữ liệu phân loại có sẵn nhưng không được sử dụng rộng rãi.

2. Phương pháp K-means giả định ngầm định các phân phối xác suất hình cầu.

3. Kết quả của phương pháp K-means phụ thuộc nhiều vào những phỏng đoán ban đầu của các hạt giống.

4. Phương pháp K-means có thể nhạy cảm với các ngoại lệ. Nếu một cây ngoại lai được chọn làm hạt giống khởi đầu, nó có thể kết thúc thành một cụm của chính nó.

Ngoài ra, nếu một ngoại lệ di chuyển từ cụm này sang cụm khác trong quá trình lặp lại, nó có thể có tác động lớn đến các cụm vì phương tiện của hai cụm có thể thay đổi đáng kể.

5. Mặc dù một số giải pháp tối ưu cục bộ(the local optimum solutions) được phát hiện bởi phương pháp K-means là thỏa đáng, nhưng thường thì phương pháp tối ưu cục bộ không tốt bằng phương pháp tối ưu toàn cục (the global optimum solutions).

6. Phương pháp K-means không xem xét kích thước của các cụm. Một số cụm có thể lớn và một số rất nhỏ.

7. K-means không đối phó (deal) với các cụm chồng chéo.

## 2.5 Độ phức tạp không gian và thời gian

Yêu cầu về không gian cho K-means là khiêm tốn vì chỉ có các điểm dữ liệu và trung tâm được lưu trữ. Cụ thể, bộ nhớ được yêu cầu là O ((m + k) n), trong đó m là số điểm và n là số thuộc tính.

Các yêu cầu về thời gian đối với K-means cũng rất khiêm tốn - về cơ bản là tuyến tính về số lượng điểm dữ liệu. Cụ thể, thời gian cần thiết là O (I \* k \* m \* n), trong đó I là số lần lặp lại cần thiết để hội tụ, như đã đề cập, I thường nhỏ và thường có thể được ràng buộc an toàn (safely bound) vì hầu hết các thay đổi thường xảy ra trong vài lần đầu tiên các lần lặp lại. Do đó K-mean là tuyến tính theo m, số điểm và hiệu quả cũng như đơn giản với điều kiện K, số lượng cụm nhỏ hơn đáng kể so với m [7].

# 3. THUẬT TOÁN FP-GROWTH

Kỹ thuật khai phá luật kết hợp sử dụng các thuật toán Apriori và các thuật toán từ Apriori hay FP-Growth không còn quá mới mẻ trong lĩnh vực khai phá dữ liệu lớn [8, 9]. Với cách duyệt dữ liệu và xây dựng cấu trúc cây, thuật toán FP-Growth có tốc độ xử lý dữ liệu nhị phân vượt trội hơn so với các thuật toán Apriori tiền nhiệm [10, 11].

## 3.1 Giới thiệu thuật toán

Thuật toán FP-Growth xây dựng cấu trúc FP tree để lưu trữ toàn bộ cơ sở dữ liệu và khai phá FP tree để tìm các mẫu phổ biến. Việc nén tập dữ liệu vào cấu trúc cây (Frequent Pattern tree – FP-tree) làm giảm chi phí cho toàn tập dữ liệu trong quá trình khai phá, các infrequent items bị loại bỏ sớm mà vẫn đảm bảo kết quả khai phá không bị ảnh hưởng và tránh được việc tạo ra các tập dự tuyển trong mỗi lần kiểm tra dữ liệu như với thuật toán Apriori. Với những đặc điểm đó, thuật toán FP-Growth có tốc độ xử lý nhanh hơn nhiều so với thuật toán Apriori.

Cấu trúc FP tree:

* Nút gốc giá trị Null (cấp độ 0)
* Thông tin mỗi nút: <name, support, parent\_node>
* Trên cùng một nhánh, nút cấp độ k có giá trị support lớn hơn hoặc bằng nút cấp độ k+1 (1<=k<=N, N là số lượng item phổ biến)

## 3.2 Giải thuật xây dựng FP tree

Input:

* D, a transaction database
* Min\_sup, the minimum support count threshold

Output: The complete set of frequent patterns

Thuận toán:

// 1. The FP-tree is constructed in the following steps:

**For each** transaction in D {

Tìm item\_i đưa vào tập item phổ biến F;

Tính support của item\_i }

//Sắp xếp F theo thứ tự giảm dần support tạo thành tập phổ biến L

L=F.sort(item, support);

//Build T tree

Initialize T; root\_node = Null; root\_node.name=Null;

**For each** transaction in D {

Select item in transaction; L.sort(item, support);

Insert\_tree([p|P],T) }

//Thủ tục Insert\_tree([p|P],T) như sau:

**If** (T có node con N & N.name=item.name)

N.count++

Else {

Tạo node N mới; N.parent=T; N.count=1 }

If (P<>rỗng)

Insert\_tree(P, N)

//2. Khám phá frequent itemsets với FP-tree được thực hiện bằng cách gọi thủ tục FP-Growth(FP\_tree, null)

Procedure FP-Growth(Tree,α) {

**If Tree** contains a single path P **then**

**For each** combination (denoted as β) of the nodes in P

{Generate pattern 𝛽 ∪α; support = minimum support of nodes in 𝛽 }

**Else for each** in the header of Tree {

Generate pattern β= ∪α; support = .support;

Construct 𝛽’s conditional pattern base;

Construct 𝛽’s conditional FP\_tree

**If** ≠ø **then** Call FP-Growth (, β) }}

## 3.3 Các thách thức của thuật toán phân cụm dựa trên tần xuất mẫu

Mục này xem xét cách các phương pháp khai phá tần xuất mẫu (frequent pattern mining) có thể được áp dụng cho phân cụm, dẫn đến phân tích cụm dựa trên tần xuất mẫu. Khai phá tần xuất mẫu, như tên của nó, tìm kiếm các mẫu (chẳng hạn như tập hợp các mục - item hoặc đối tượng - object) thường xuyên xảy ra trong các tập dữ liệu lớn. Khai phá tần xuất mẫu có thể dẫn đến việc phát hiện ra các liên kết và mối tương quan thú vị giữa các đối tượng dữ liệu. Ý tưởng đằng sau phân tích cụm dựa trên tần xuất mẫu là các tần xuất mẫu được phát hiện cũng có thể chỉ ra các cụm. Phân tích cụm dựa trên tần xuất mẫu rất phù hợp với dữ liệu cao chiều. Nó có thể được xem như một phần mở rộng của cách tiếp cận phân cụm không gian con tăng trưởng theo thứ nguyên (dimension-growth subspace clustering approach). Tuy nhiên, ranh giới của các kích thước khác nhau là không rõ ràng, vì ở đây chúng được biểu diễn bằng các tập hợp các tập mục thường xuyên. Nghĩa là, thay vì tăng thứ nguyên cụm theo thứ nguyên, người ta phát triển các nhóm tập hợp mục thường xuyên, cuối cùng dẫn đến mô tả cụm. Các ví dụ điển hình về phân tích cụm dựa trên tần xuất mẫu bao gồm phân nhóm các tài liệu văn bản có chứa hàng nghìn từ khóa riêng biệt và phân tích dữ liệu mảng vi mô (microarray) chứa hàng chục nghìn giá trị đo được hoặc “features”. Trong [1], nhóm tác giả xem xét hai hình thức phân tích cụm dựa trên tần xuất mẫu: *phân nhóm văn bản dựa trên tần xuất mẫu và phân nhóm theo sự giống nhau về mẫu trong phân tích dữ liệu mảng vi mô*.

Trong phân nhóm văn bản dựa trên tần xuất mẫu, các tài liệu văn bản được phân nhóm dựa trên các thuật ngữ thường xuyên mà chúng chứa. Sử dụng từ vựng của phân tích tài liệu văn bản, một thuật ngữ là bất kỳ chuỗi ký tự nào được phân tách với các thuật ngữ khác bằng dấu phân cách. Một thuật ngữ có thể được tạo thành từ một từ hoặc một số từ. Nói chung, trước tiên, nhóm tác giả xóa thông tin nontext (chẳng hạn như thẻ HTML và dấu chấm câu) và các từ dừng. Các thuật ngữ sau đó được trích xuất. Sau đó, một thuật toán tạo gốc (stemming algorithm) được áp dụng để giảm từng thuật ngữ về gốc cơ bản của nó. Bằng cách này, mỗi tài liệu có thể được biểu diễn dưới dạng một tập hợp các item. Mỗi bộ thường lớn. Nói chung, một bộ tài liệu lớn sẽ chứa một bộ rất lớn các thuật ngữ riêng biệt. Nếu chúng ta coi mỗi thuật ngữ là một chiều, thì không gian chiều sẽ có số chiều rất cao!

Điều này đặt ra những thách thức lớn cho việc phân tích cụm tài liệu. Không gian chiều có thể được gọi là không gian vectơ thuật ngữ, trong đó mỗi tài liệu được biểu diễn bằng một vectơ thuật ngữ. Khó khăn này có thể được khắc phục bằng cách thường xuyên phân tích theo thuật ngữ. Có thể khai thác một tập hợp các thuật ngữ thường xuyên từ tập hợp các tài liệu văn bản. Sau đó, thay vì phân cụm trên không gian vectơ số chiều cao, ta chỉ cần coi các tập hợp số hạng thường xuyên có chiều thấp là "ứng cử viên cụm" (cluster candidates).

Lưu ý rằng một tập hợp thuật ngữ thường xuyên không phải là một cụm mà là mô tả của một cụm. Cụm tương ứng bao gồm tập hợp các tài liệu chứa tất cả các thuật ngữ của tập hợp thuật ngữ thường xuyên. Một tập hợp con được chọn lọc tốt (*well-selected subset*) của tập hợp tất cả các tập thuật ngữ thường xuyên có thể được coi là một phân cụm. “*Vậy làm thế nào, chúng ta có thể chọn một tập hợp con tốt của tập hợp tất cả các tập hợp thuật ngữ thường gặp?*” Bước này rất quan trọng vì việc lựa chọn như vậy sẽ xác định chất lượng của kết quả việc phân cụm. Gọi Fi là một tập hợp các bộ thuật ngữ thường gặp và cov (Fi) là tập các tài liệu được bao hàm bởi Fi. Có nghĩa là, cov (Fi) đề cập đến các tài liệu có chứa tất cả các thuật ngữ trong Fi. Nguyên tắc chung để tìm một tập hợp con được chọn tốt, F1,. . . , Fk, của tập hợp tất cả các tập thuật ngữ thường xuyên là để đảm bảo rằng:

(1) Ski = 1cov (Fi) = D (tức là tập hợp con được chọn phải bao gồm tất cả các tài liệu được nhóm lại); và

(2) nên giảm thiểu sự chồng chéo giữa hai phân vùng Fi và Fj (đối với i6 = j).

Một phép đo chồng chéo dựa trên entropy được sử dụng để đánh giá sự chồng chéo của cụm bằng cách đo lường sự phân bố của các tài liệu hỗ trợ một số cụm so với các ứng cử viên của cụm còn lại. Một lợi thế của phân cụm văn bản dựa trên cụm từ thường xuyên là nó tự động tạo mô tả cho các cụm được tạo theo nhóm cụm từ thường xuyên của chúng. Các phương pháp phân cụm truyền thống chỉ tạo ra các cụm — mô tả cho các cụm được tạo yêu cầu một bước xử lý bổ sung.

Một cách tiếp cận thú vị khác để phân cụm dữ liệu cao chiều dựa trên sự giống nhau về mẫu giữa các đối tượng trên một tập hợp con các kích thước. Ở tài liệu [1] nhóm tác giả giới thiệu phương pháp p Cluster, phương pháp này thực hiện phân cụm theo mẫu tương tự trong phân tích dữ liệu microarray. Trong phân tích microarray DNA, mức độ biểu hiện của hai gen có thể tăng và giảm đồng bộ để đáp ứng với một loạt các kích thích hoặc điều kiện môi trường. Trong mô hình p Cluster, hai đối tượng tương tự nhau nếu chúng thể hiện một mô hình nhất quán trên một tập hợp con các kích thước. Mặc dù mức độ biểu hiện của chúng có thể không gần nhau, nhưng các mẫu chúng thể hiện có thể rất giống nhau [3].

# 4. THUẬT TOÁN DBSCAN

DBSCAN (Density Based Spatial Clustering of Applications with Noise - Phân cụm không gian dựa trên mật độ của các ứng dụng có nhiễu) là một ví dụ về phương pháp phân cụm dựa trên mật độ. Phương pháp này được thiết kế cho cơ sở dữ liệu không gian nhưng có thể được sử dụng trong các ứng dụng khác. Nó yêu cầu hai tham số đầu vào: kích thước của vùng lân cận (R) và các điểm tối thiểu trong vùng lân cận (N). Về cơ bản, hai tham số này xác định mật độ bên trong các điểm trong các cụm mà người dùng sẵn sàng chấp nhận vì chúng chỉ định có bao nhiêu điểm trong một vùng. Số lượng điểm không chỉ xác định mật độ của các cụm có thể chấp nhận được mà nó còn xác định đối tượng nào sẽ được gắn nhãn ngoại lệ hoặc nhiễu. Các đối tượng được tuyên bố là ngoại lai nếu có ít đối tượng khác trong vùng lân cận của chúng. Tham số kích thước R xác định kích thước của các cụm được tìm thấy. Nếu R đủ lớn, sẽ có một cụm lớn và không có ngoại lai. Nếu R nhỏ, sẽ có các cụm nhỏ dày đặc và có thể có nhiều điểm bất thường [2].

Ta nhắc lại định nghĩa một số khái niệm được yêu cầu trong phương pháp DBSCAN:

1. Vùng lân cận (Neighborhood): Vùng lân cận của một đối tượng y được định nghĩa là tất cả các đối tượng nằm trong bán kính R tính từ y.

2. Đối tượng lõi (Core object): Một đối tượng y được gọi là đối tượng lõi nếu có N đối tượng bên trong vùng lân cận của nó.

3. Gần nhau (Proximity): Hai đối tượng được xác định là ở gần nhau nếu chúng thuộc cùng một cụm. Đối tượng ở gần đối tượng nếu hai điều kiện được thỏa mãn:

* Các đối tượng đủ gần nhau, ví dụ, trong một khoảng cách R
* là đối tượng lõi như đã định nghĩa ở trên.

4. Khả năng kết nối (Connectivity): Hai đối tượng và được kết nối nếu có một đường dẫn hoặc chuỗi đối tượng , , … sao cho mỗi ở gần (Proximity) đối tượng

Bây giờ chúng ta phác thảo thuật toán cơ bản để phân cụm dựa trên mật độ như sau:

1. Chọn giá trị của R và N

2. Tùy ý chọn một đối tượng p.

3. Lấy tất cả các đối tượng liên kết với p, R và N cho trước.

4. Nếu p là một đối tượng lõi, một cụm được hình thành

5. Nếu p là một đối tượng biên (border object), không có đối tượng nào ở gần nó. Chọn một đối tượng khác. Chuyển đến bước 3

6. Tiếp tục quá trình cho đến khi tất cả các đối tượng đã được xử lý.

## 4.1 Ưu điểm và nhược điểm của DBSCAN

Bởi vì DBSCAN sử dụng định nghĩa dựa trên mật độ của một cụm, nó có khả năng chống nhiễu tương đối và có thể xử lý các cụm có hình dạng và kích thước tùy ý. Do đó, DBSCAN có thể tìm thấy nhiều cụm không thể tìm thấy bằng cách sử dụng K-means. Tuy nhiên, như đã chỉ ra trước đây, DBSCAN gặp khó khăn khi các cụm có mật độ khác nhau. Nó cũng gặp rắc rối với dữ liệu cao chiều vì mật độ khó xác định hơn đối với dữ liệu đó. Cuối cùng, DBSCAN có thể tốn kém khi tính toán các lân cận gần nhất yêu cầu tính toán tất cả các lân cận theo cặp, như thường xảy ra đối với dữ liệu cao chiều [12].

## 4.2 Độ phức tạp không gian và thời gian của DBSCAN

Độ phức tạp thời gian cơ bản của thuật toán DBSCAN là O (m × thời gian để tìm các điểm trong vùng lân cận Eps), trong đó m là số điểm. Trong trường hợp xấu nhất, độ phức tạp này là O(*m*2). Tuy nhiên, trong không gian thấp chiều, có các cấu trúc dữ liệu, chẳng hạn như *kd-*trees, cho phép truy xuất hiệu quả tất cả các điểm trong một khoảng cách nhất định của một điểm xác định và độ phức tạp về thời gian có thể thấp đến mức O(*m* log*m*).

Yêu cầu về không gian của DBSCAN, ngay cả đối với dữ liệu chiều cao, là O(m) bởi vì nó chỉ cần giữ một lượng nhỏ dữ liệu cho mỗi điểm, tức là nhãn cụm và việc xác định từng điểm làm lõi, đường biên, hoặc điểm nhiễu

# TÀI LIỆU THAM KHẢO

1. Sabhia Firdaus1, Md. Ashraf Uddin, *A Survey on Clustering Algorithms and Complexity Analysis,* IJCSI International Journal of Computer Science Issues, Volume 12, Issue 2, March 2015. ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
2. Gupta, G. K, Introduction to data mining with case studies, PHI Learning Pvt. Ltd., 2006.
3. Han, Jiawei, Micheline Kamber, and Jian Pei, *Data mining: concepts and techniques*, Morgan kaufmann, 2006.
4. Purnima Bholowalia, Arvind Kumar, EBK-Means, *A Clustering Technique based on Elbow Method and K-Means in WSN*, International Journal of Computer Applications, Volume 105 – No. 9, November 2014
5. Soumi Ghosh, Sanjay Kumar Dubey, *Comparative Analysis of K-Means and Fuzzy C-Means Algorithms*, International Journal of Advanced Computer Science and Applications, Vol. 4, No.4, 2013
6. Ming-Ni Wu, Chia-Chen Lin, Chin-Chen Chang, *Brain Tumor Detection Using Color-Based K-Means Clustering Segmentation*, Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)
7. <http://wwwusers.cs.umn.edu/~kumar/dmbook/ch8.pdf>
8. Coenen, F., Goulbourne, G. and Leng, P., *Tree Structures for Mining association Rules*, Journal of Data Mining and Knowledge Discovery, Vol 8, No 1, pp25-51, 2003
9. Asantha Thilina, Shakthi Attanayake, Sacith Samarakoon, Dahami Nawodya, Lakmal Rupasinghe, Nadith Pathirage,Tharindu Edirisinghe, Kesavan Krishnadeva, *Intruder Detection Using Deep Learning and Association Rule Mining*, IEEE International Conference on Computer and Information Technology, 2016
10. Ke Wang, Liu Tang, Jiawei Han, and Junqiang Liu, *Top Down FP-Growth for Association Rule Mining*,
11. J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, In MOD 2000, pp. 1-12.

Singh, Shalini S., and N. C. Chauhan, *K-means v/s K-medoids: A Comparative Study*, National Conference on Recent Trends in Engineering & Technology, 2011.