BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC DUY TÂN**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙡🕮🙣



**BÁO CÁO ĐỀ TÀI**

**MÔN HỌC: Machine Learning 1 - DS 321 A**

**Đề tài: *Tìm K tối ưu cho thuật toán phân cụm Kmeans.***

**Sinh viên thực hiện: Lê Văn Hiếu**

**Mã số sinh viên: 24211906618**

**MỤC LỤC**

[DANH MỤC CÁC KÍ HIỆU 3](#_Toc61114560)

[DANH MỤC CÁC HÌNH MÌNH HỌA 4](#_Toc61114561)

[Chương I. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 5](#_Toc61114562)

[I.1. Lý do chọn đề tài: 5](#_Toc61114563)

[I.2. Mục tiêu và phương pháp nghiên cứu 5](#_Toc61114564)

[I.3. Bố cục báo cáo 6](#_Toc61114565)

[Chương II. CƠ SỞ LÝ THUYẾT 7](#_Toc61114566)

[II.1. Giới thiệu lý thuyết về Machine Learning 7](#_Toc61114567)

[II.2. Giới thiệu về ngôn ngữ lập trình. 7](#_Toc61114568)

[II.3. Một số vấn đề về tập dữ liệu kiểm thử. 8](#_Toc61114569)

[Chương III. THUẬT TOÁN PHÂN CỤM KMEANS 9](#_Toc61114570)

[III.1. Thuật toán phân cụm K-means 9](#_Toc61114571)

[III.2. Kmeans++ 10](#_Toc61114572)

[III.3. Vấn đề chọn số lượng tâm với Kmeans. 11](#_Toc61114573)

[Chương IV. XÂY DỰNG CÁC PHƯƠNG PHÁP TÌM K TỐI ƯU 11](#_Toc61114574)

[IV.1. Xây dựng hàm Kmeans 11](#_Toc61114575)

[IV.2. Phương pháp Elbow 12](#_Toc61114576)

[IV.3. Phương pháp Silhouette 13](#_Toc61114577)

[Chương V. VẬN DỤNG THỰC TẾ 16](#_Toc61114578)

[V.1. Tập dữ liệu 1 16](#_Toc61114579)

[V.2. Tập dữ liệu 2 16](#_Toc61114580)

[Chương VI. KẾT LUẬN 16](#_Toc61114581)

[VI.1. So sánh các thuật toán. 16](#_Toc61114582)

[VI.2. Các kết quả đạt được 16](#_Toc61114583)

[VI.3. Các hạn chế 16](#_Toc61114584)

[LIÊN KẾT VÀ TÀI LIỆU THAM KHẢO 17](#_Toc61114585)

DANH MỤC CÁC KÍ HIỆU

Các kí hiệu:

|  |  |
| --- | --- |
| Kí hiệu | Ý nghĩa |
| Kmeans | Thuật toán phân cụm dữ liệu Kmeans. |
| K/k | Số lượng tâm/cụm trong thuật toán phân cụm Kmeans |
| **C, c** | Lần lượt là tập ma trận các tâm của Kmeans và vector tâm thuộc **C** |
| **X, x** | Lần lượt là tập ma trận các điểm của tập dữ liệu đưa vào và vector điểm thuộc **X** |
| D(a,b) | Khoảng cách giữa ma trận hoặc vector a tới ma trân hoặc vector b. |
| d | Dimension. Số chiều hay cột của mảng/Tập dữ liệu |
| N | Numer. Số lượng phần tử của mảng/Tập dữ liệu |
| **L** | Labels. Tập ma trận các nhãn sinh từ Kmeans. |
| Min | Giá trị nhỏ nhất thuộc một tập nào đó. |
| Max | Giá trị lớn nhất thuộc một tập nào đó. |
| TWSS | Total Within Sum of Square, Giá trị dùng để đánh xem độ tượng đồng của một cụm như thế nào. |

DANH MỤC CÁC HÌNH MÌNH HỌA

[Hình 1. Bài toán phân cụm với số K=3 7](#_Toc60785491)

[Hình 2. Đầu ra của bài toán phân cụm với mỗi sự lựa chọn tâm khác nhau 8](#_Toc60785492)

[Hình 3. Ví dụ một mô hình Elbow method với tâm thích hợp là 5 13](#_Toc60785493)

[Hình 4. Biểu đồ mô ta các giá trị Silhouette của các cụm và Giá trị Silhouette trung bình 15](#_Toc60785494)

1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI
   1. Lý do chọn đề tài:

Kmeans là thuật toán phân cụm thuộc nhóm học không giám sát (Unsupervised Learning). Cách hình thành các cụm dựa trên việc tính khoảng cách các tâm và các điểm (Thường dùng khoảng cách Euclid). Sau đó điều chỉnh các tâm sao cho các tâm không còn thay đổi hoặc sự khác nhau không quá lớn. Sau khi tìm được các tâm thì ta có thể biết được các cụm.

Thuật toán Kmeans có thể giúp ích dược rất có thể rất hữu ích cho nhiều lĩnh vực khác nhau trong cuộc sống. Các lĩnh vực đó có thể là toán học, kinh tế, văn bản, quy hoạch đô thị, cứu nạn cứu trợ,… Một vài ứng dụng trong số đó có thể kể ra như là phân khúc khách hàng, nhận diện chữ viết, phân nhóm khu vực dựa vào phân bố dân cư, tách vật thể trong ảnh,…

Tuy vậy để Kmeans hoạt động được thì phải cần biết trước số cụm đưa vào(K). Đây là vấn đề thường gặp trong bài toán phân cụm nói chung. Với các bài toán đơn gian thì ta có thể nhận diện được số K và đưa vào thuật toán ngay. Nhưng đối với các bài toàn phức tạp hơn thì số K này là vấn đề lớn. Việc phân cụm mà số cụm chưa biết rõ thì ta cần phải tính toán để có được số cụm K tối ưu.

Để giải quyết vấn đề này, nhóm sẽ phân tích và xây dựng các phương pháp để tìm được số K tối ưu và áp dùng vào một số bài toán cụ thể. Các phương pháp: elbow method, silthoutte method sẽ giúp tìm được số tâm tối ưu. Tăng hiệu quả bài toán.

* 1. Mục tiêu và phương pháp nghiên cứu
     1. Mục tiêu nghiên cứu

Mục tiêu của đề tài này là nghiên cứu tối ưu thuật toán Kmean và tìm được các phương pháp hiệu quả đề tìm được số k tối ưu. Khi tìm được phương pháp thì nghiên cứu, áp dụng vào thuật toán Kmeans. Phân tích các ưu nhược điểm của các phương pháp và đánh giá nó. Đồng thời sử dụng các phương pháp đó đề tìm số cụm tối ưu cho một số bài toàn cụ thể. Hiểu thêm về các thuật toán máy học, cách sử dụng ngôn ngữ python và các công cụ kèm theo.

* + 1. Phương pháp nghiên cứu.

Nghiên cứu về ý nghĩa thuật toán, công thức toán học của thuật toán phân cụm Kmeans và thực hiện trên ngôn ngữ lập trình python.

Nghiên cứu triển khai các thuật toán về tìm K tối ưu cho thuật toán phân cụm Kmeans. Áp dụng các công cụ, thư viện trong ngôn ngữ lập trình python để thực hiện, mô tả các phương pháp nói trên bằng biểu đồ.

Thử nghiệm trên một số mẫu dữ liệu có sẵn. Lấy giá trị K có được nhờ các phương pháp trên. Áp dụng phân cụm Kmeans và đánh giá kết quả.

* 1. Bố cục báo cáo

Nội dụng báo cáo bao gồm các chương sau:

* Chương I. Giới thiệu tổng quan đề tài: Giới thiệu tổng quan, mục tiêu và phương pháp nghiên cứu đề tài.
* Chương II. Thuật toán phân cụm Kmeans và các cộng cụ hỗ trợ: Giới thiệu về thuật toán phân cụm Kmeans và các công hỗ trợ lập trình như python, numpy,…
* Chương III. Xây dựng phương pháp tìm K tối ưu: Xậy dựng hàm Kmeans. Giới thiệu, nêu thuật toán của các phương pháp và áp dụng các phương pháp đó vào hàm Kmeans.
* Chương IV. Vân dụng thực tế: Sử dụng hàm Kmeans và các phương pháp đã đượng xây dựng ở Chương 3 vào một số tập dữ liệu cụ thể.
* Chương V. Kết luận: Đưa ra các đánh giá về các phương pháp. Nêu ra những kết quả đạt được và hạn chế của đề tài.

1. CƠ SỞ LÝ THUYẾT
   1. Giới thiệu lý thuyết về Machine Learning
   2. Giới thiệu về ngôn ngữ lập trình.
      1. Python và phần mềm Jupyter lab

Python là một ngôn ngữ lập trình bậc cao sử dụng cho nhiều mục đích khác nhau do Guido Van Rossum tạo ra và ra mắt lần đầu tiên vào năm 1991. Đặc điểm của Python là một ngôn ngữ mạnh mẽ, dễ học, dễ sử dụng. Nó chứa các cấu trúc câu lệnh tối giản, đơn giản và rõ ràng giúp việc học tập và trao đổi lập trình trở nên dễ dàng.

Python là cộng cụ tuyệt vời cho việc lập trình các thuật toán máy học. Ngôn ngữ này cho phép việc lập trình trở nên ngắn ngọn và dễ hiểu. Đồng thời có chưa nhiều các hàm và thư viện được xây dựng sẵn trong Python. Trong khi đó, máy học và trí tuệ nhân tạo chứa nhiều các thuật toán phức tạp và lượng dữ liệu lớn nhằm phục vụ nó. Sự đơn giản của Python giúp lập trình viên tập trung vào các vấn đề của máy học thay vì là các kĩ thuật sử dụng ngôn ngữ lập trình.

Python chứa nhiều các thư viện hỗ trợ việc tính toán và nghiên cứu máy học. Việc thực thi các thuật toán máy học và sử dụng dữ liệu lớn có thể sẽ phức tạm và tốn nhiều thời gian. Trong khi đó, Python cung cấp môi trường làm việc, thư viện hỗ trợ đã được lập trình và vận hành mạnh mẽ. Việc này sẽ giúp tiết kiệm được nhiều vấn đề khác nhau và giúp quá trình lập trình diễn ra được tốt nhất.

Project Jupyter là một dữ án nhằm để phát triển phần mềm mã nguồn mở, các tiêu chuẩn mở và các dịch vụ cho tính toán tương tác trên nhiều ngôn ngữ lập trình khác nhau. Jupyter Notebook là một ứng dụng trên web mã nguồn mở cho phép tạo và chia sẽ các tài liệu có chưa mã lập trình trực tiếp, các phương trình, hình ảnh và văn bản tường thuật. Jupyter lab là một môi trường phát triển có tính tương tác cao trên web dành cho Jypyter Notebook, lập trình và dữ liệu. JupyterLab có tính linh hoạt cao, giao điện người dùng đã được tinh chỉnh và sắp xếp lại để hỗ trợ cho nhiều quy trình làm việc khác nhau trong khoa học dữ liệu, khoa học máy tính và máy học. JupyterLab có thể được mở rộng và tích hợp nhiều các tính năng phụ trợ khác nhau tuy theo nhu cầu của người dùng.

Anaconda là một phần mềm miễn phí, giúp dễ dàng tải về và quản lý các gói tài nguyên, môi trường làm việc và các phiên bản Python. Anaconda cung cấp JupyterLab miễn phí và dễ dàng tải về các thư viện mà người dùng cần.

Đề tài này sẽ sử dụng Anaconda JupyterLab với Python phiên bản 3.8.5 để chạy hầu hết các phần lập trình.

* + 1. Thư viện numpy

Numpy là một thư viện mã nguồn mở thuộc Python dùng để làm việc với các mảng và ma trận lớn, nhiều chiều, cùng với số lượng lớn các hàm toán học cấp cao để hoạt động trên các mảng này. Nó hỗ trợ nhiều hàm phục vụ cho việc thực hiện các phép biến đổi, tinh toán trong đại số tuyến tính và ma trận. Numpy được xây dựng bởi Tavis Oliphant vào năm 2005. NumPy là viết tắt của Numerical Python.

Một đặc điểm của Numpy là các mảng Numpy được lưu trữ liên tục lại một nơi trong bộ nhớ, làm cho tốc độ truy cập và xữ lý trở nên hiệu quả. Tóc độ sử lý này nhanh hơn nhiều so với kiểu dữ liệu List được viết sẳn của python, kiểu dữ liệu List lưu trữ dữ liệu rời rạc trên bộ nhớ. Việc này rất hữu ích cho tiệc tính toán các phép tính có khối lượng dữ liệu lớn và nhiều chiều.

* + 1. Thư viện pandas

Pandas là một thư viện mã nguồn mở của Python. Đây là một cộng cụ mạnh mẽ, nhanh nhẹn, linh hoạt và dễ dàng sử dụng để thực hiện phân tích và xữ lý dữ liệu. Pandas hỗ trợ nhiều hàm giúp phân tích, làm sạch, đọc và xữ lý dữ liệu. Pandas được tạo ra bởi Wes McKinney vào năm 2008. Pandas là công cụ tuyệt vời trong ngành khoa học dữ liệu

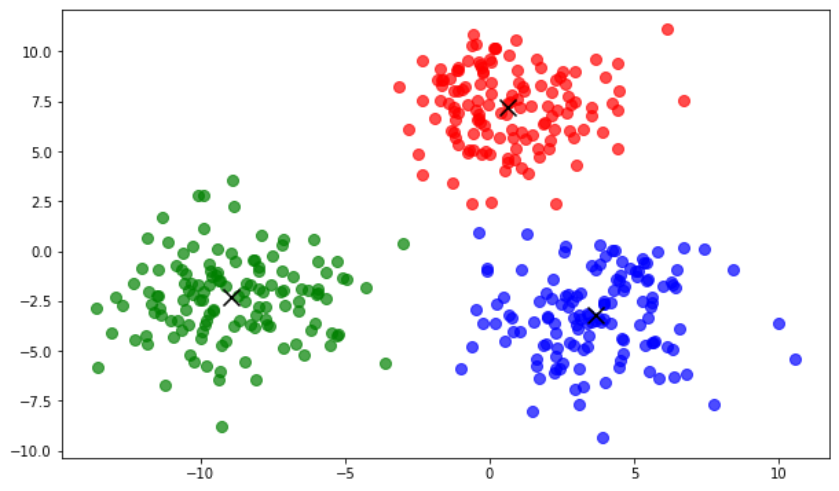
* + 1. Thư viện matplotlib

Matplotlib là một thư viện mã nguồn mở Python giúp hỗ trợ vẽ đồ thị. Đây là một cộng cụ rất hữu ích khi làm việc với Python và numpy. Module được sử dụng nhiều nhất của Matplotib là Pyplot cung cấp giao diện như MATLAB.

* 1. Một số vấn đề về tập dữ liệu kiểm thử.

1. THUẬT TOÁN PHÂN CỤM KMEANS
   1. Thuật toán phân cụm K-means
      1. Giới thiệu thuật toán phân cụm Kmeans

Phân cụm là nhiệm vụ phân chia dân số hoặc các điểm dữ liệu thành một số nhóm sao cho các điểm dữ liệu trong cùng một nhóm giống với các điểm dữ liệu khác trong cùng một nhóm hơn các điểm dữ liệu trong các nhóm khác. Nói cách đơn giản, mục đích là để tách các nhóm có đặc điểm giống nhau và gán chúng thành các cụm. Mục tiêu của thuật toán Kmean là tìm các nhóm trong dữ liệu, với số lượng nhóm được đại diện bởi biến K. thuật toán hoạt động lặp đi lặp lại để gán mỗi điểm dữ liệu cho một trong k nhóm dựa trên các tính năng được cung cấp. Trong hình ảnh tham khảo bên dưới, k = 3, và có ba cụm được xác định từ tập dữ liệu nguồn.



Hình 1. Bài toán phân cụm với số K=3

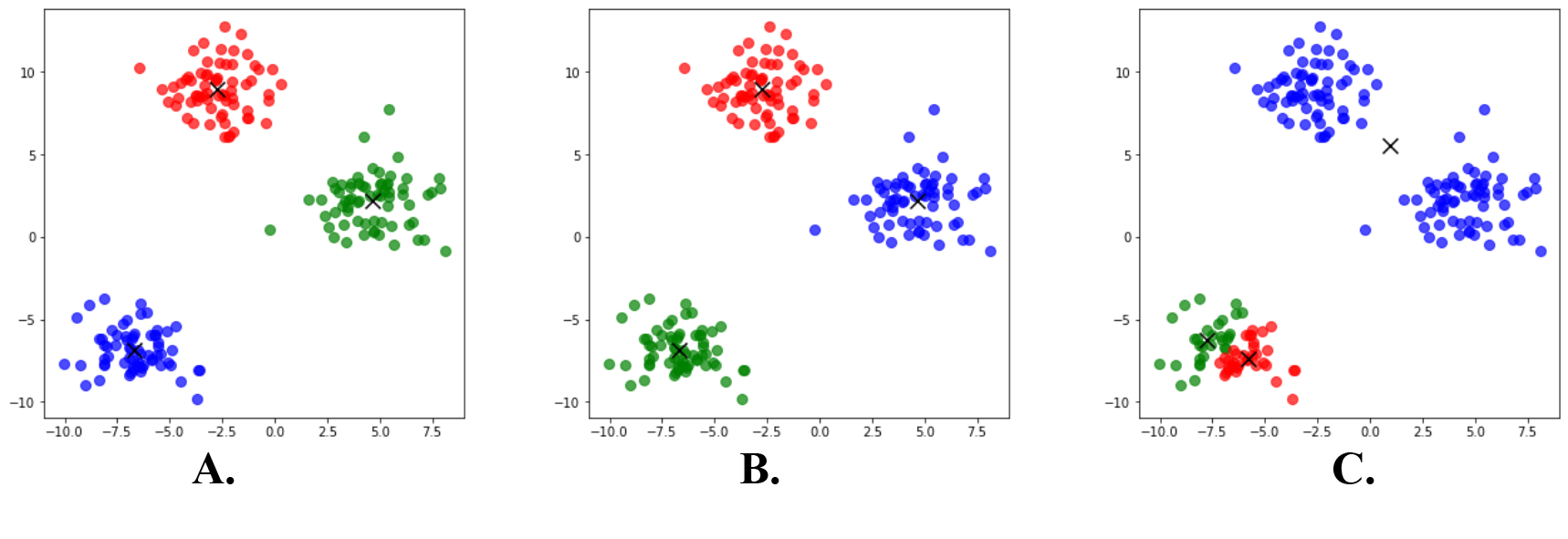
* + 1. Giải thích thuật toán

Thuật toán Kmeans có để được thực hiện như sau.

* Đầu vào: Ma trận dữ liệu và số lượng tâm K<N cần tìm
* Đầu ra: Ma trận tâm cụm và ma trận nhãn .

1. Chọn K điểm bất kì làm các tâm khởi tạo ban đầu của **C**.
2. Tính khoảng cách từ các điểm đến tâm **C**
3. Gán nhãn cho mỗi điểm dữ liệu vào cụm có tâm gần nó nhất.
4. Nếu việc phân cụm dữ liệu ở Bước 3. Không thay đổi so với vòng lặp trước thì kết thúc thuật toán
5. Cập nhật các tâm cụm bằng cách lấy trung bình cộng các điểm dữ liệu đã được gán nhãn.
6. Quay lại bước 2.
   1. Kmeans++

Trong bài toán Kmeans, ta phải lựa chọn K tâm ban đâu để thuật toán có thể điều tâm và phân cụm. Lựa chọn tâm theo cách chọn nhẫu nhiễn K điểm dữ liệu đã cho ban đầu là một cách đơn giản và nhanh chống. Tuy nhiên phương pháp này nay sinh ra một số vấn đề có thể gây ảnh hưởng đến kết quả cuối cùng cho bài toán phân cụm. Các tâm được sinh ra khác nhau khiến cho một nghiệm của bài toán là khác nhau. Trong trường hợp tệ nhất là khiên cho bài toán phân cụm bị lệch.



Hình 2. Đầu ra của bài toán phân cụm với mỗi sự lựa chọn tâm khác nhau

Trong hình trên ta có thể thấy: Hình A. B. là có sự khác nhau về màu trong 2 cụm. Riêng hình C. thì các cụm được phân bố khác đi so với hình A. và B.. Dễ thấy rằng, sự phân cụm như hình C. là có sự sai lệch. Sự lựa chọn tâm ngẫu nhiên quá gần nhau đã dẫn đến điều này.

Đề làm giảm đi sự sai lệch do khởi tao tâm ban đầu, ta có thể khởi tạo các tâm ban đầu sử dụng phương pháp Kmeans++. Đây là phương pháp được David Arthur và Sergei Vassilvitskii viết trong một bài báo nói về ưu điểm của việc gieo hạt cẩn thận(“k-means++: The Advantages of Careful Seeding”). Đây là phương pháp giúp việc lựa chọn tâm ban đầu được giàn trải hơn so với phương pháp lựa chọn tâm ngẫu nhiên cho toàn bộ các tâm.

Thuật toán được mô tả như sau:

1. Lựa chọn một tâm ngẫu nhiên là một điểm thuộc tập dữ liệu đầu vào
2. Với mỗi điểm **x**, xác định D(**x,c**) với D(**x,c**) là khoảng cách tối thiểu từ **x** đến tâm **c**
3. Chọn tâm **c** tiếp theo thuộc **X** sao cho tỉ lệ giữa khoảng cách d(x,ci)2 với tổng khoảng cách đối với tâm gần nhất là cao nhất
4. Lặp lại bước 2, 3 cho đến khi đủ số lượng tâm cần tìm.

Áp dụng thuật toàn này vào nghiên cứu sẽ giúp hạn chế dược sai số trong quá trình thử nghiệm các thuật toán. Tuy vậy Kmeans++ sẽ có tốc độ chậm hơn so với việc lựa chọn ngẫu nhiên cho toàn bộ các tâm do phải thực hiện tính toán lại khoảng cách các điểm với các tâm. Điều này thậm chí còn tệ hơn nếu số K lớn.

Trong suốt bài báo cáo này, các tâm ban đầu sẽ được khởi tạo ngẫu nhiên bằng thuật toán Kmeans++ để tăng độ chính xác và hiệu suật bài toán mặc dù tốc độ thực thi có thể sẽ chậm hơn.

* 1. Vấn đề chọn số lượng tâm với Kmeans.

Muốn cho thuật toán phân cụm Kmeans phân chia được số cụm thì ta cần biết được số cụm cần thiết để đưa vào. Câu hỏi đặt ra là: với một tập dữ liệu đã có sẵn thì phân bao nhiêu cụm là hợp lý, tối ưu?

* + 1. Tính khoảng cách Euclid:

Cho a, b là 2 điểm dữ liệu, khoảng cách Euclid từ điểm a đến điểm b:

Với a, b có số chiều lớn hơn 1 ta có:

Với số chiều là 2:

Với số chiều là n:

1. XÂY DỰNG CÁC PHƯƠNG PHÁP TÌM K TỐI ƯU
   1. Phương pháp Elbow

Phương pháp Elbow(Củi chỏ) là một trong những phương pháp nổi tiếng nhất và đơn giản mà ta có thể dùng để chọn giá trị K phù hợp và tăng độ chính xác của mô hình.

Ý tưởng chính của thuật toán Kmean như sau. Với một tập dữ liệu ban đầu, phân tập dữ liệu đó thành K cụm khác nhau. Với mối cụm sẽ có một tâm gọi là Centroid, ta tính tổng khoảng cách từ tâm này đến toàn bộ các điểm dữ liệu thuộc cụm của tâm đó. Tham số này gọi là WSS (Within-cluster Sum of Square). Sau đó ta tính tiếp tục tìm WSS cho các tâm còn lại là lấy tổng của nó. Tổng này gọi là TWSS(Total Within-cluster Sum of Square). Giá trị này có thể được tính như sau:

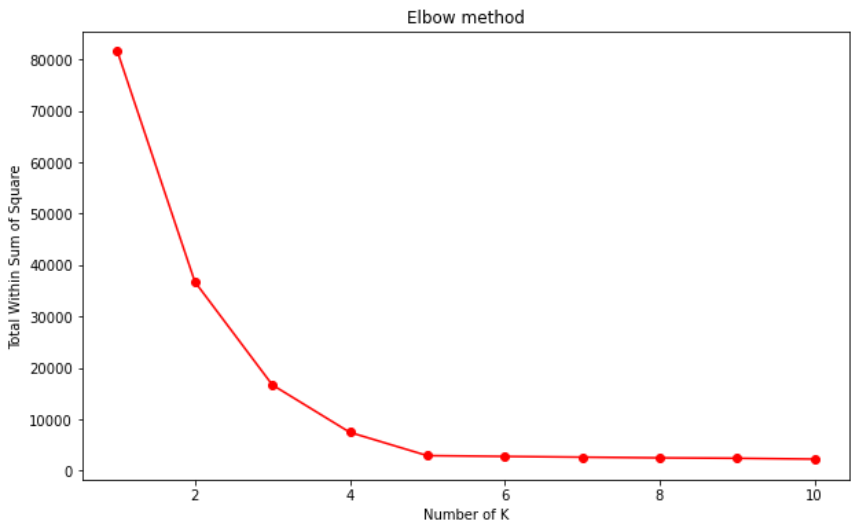
Vì mục tiêu của thuật toán phân cụm là phân chia toàn bộ các điểm của tập liệu ban đầu thành K cụm khác nhau sao cho các điểm dữ liệu thuộc cùng một cụm là giống với nhau nhất có thể, hay TWSS nhỏ nhất có thể, nên việc lựa chọn phương thức tính khoảng cách là rất quan trọng vì khoảng cách được sử dụng như là một thước đo mức độ tương đồng của các quan sát. Tuy vậy với số lượng K càng lớn thì các giá trị TWSS càng nhỏ dần và không có sự thay đổi lớn và dẫn vô nghĩa với số K rất lớn. Do đó ta nên chọn giá trị K cho TWSS tối ưu nhất.

Theo đó ta có quy trình triển khai Elbow method như sau:

* Đầu vào: Ma trận dữ liệu và khoảng số lượng NumK<N cần tìm.
* Đầu ra: TWSS của từng giá trị K trong khoảng NumK

1. Triển khai thuật toán phân cụm với số cụm K trong khoảng NumK bắt đầu từ 1.
2. Với mỗi giá trị K, tính giá trị TWSS tương ứng.
3. Biểu thị biểu đồ Elbow Method theo các giá trị TWSS của K đã tính ở bước 2.
4. Dựa vào biểu đồ trên, lấy điểm K sao cho giá trị TWSS của K đó không thay đổi quá nhiều so với trước.

Ta lựa cọn điểm K trong biểu đồ sao cho giá trị của nó có sự giảm không đáng kể với các điểm tiếp theo nó.



Hình 3. Ví dụ một mô hình Elbow method với tâm thích hợp là 5

* 1. Phương pháp Silhouette

Phương pháp Silhouette(Hình bóng) là một phương pháp dùng để tính toán, đánh giá mức độ thích hợp của kĩ thuật phân cụm. Giá trị của có được gọi là hệ số Silhouette(Silhouette Coefficient) hay điểm Silhouette(ilhouette score). Giá trị này giao động từ trong khoảng [-1;1].

* Các giá trị càng gần 1 chứng tỏ cụm đó đã tách xa và phân biệt với các điểm thuộc cụm khác.
* Giá trị 0 mang ý nghĩa là cụm đó không có sự khác biệt giữa cụm này và cụm kia. Hoặc khoảng cách giữa các cụm là không khác biệt
* Giá trị càng về -1 thì có nghĩa là cụm đó đã bị phân cụm sai.

Chung quy lại nếu giá trị Silhouette càng lớn thì cụm đó đã được phận cụm chính xác

Giá trị Silhouette cho mỗi điểm có thể được tính như sau:

Với mỗi điểm dữ liệu (điểm dữ liệu i thuộc cụm đặt:

Với a(i) là tổng khoảng cách từ điểm dữ liệu i đến tất cả các điểm dữ liệu j thuộc sau cho i khác j chia cho số điểm dữ liệu thuộc trừ 1(không tính i vào số lượng). Ta có thể hiểu a(i) như là cách để đánh giá điểm dữ liệu i có thích hợp với cụm hay không. Giá trị a(i) càng nhỏ thì điểm dữ liệu i càng hợp với cụm và ngược lại.

Tiếp theo ta xác định sự khác biệt trung bình của điểm dữ liệu i đến các tâm tính trung bình khoảng cách giữa điểm dữ liệu i đến tất cả các điểm dữ liệu của các cụm không phải là cụm :

Giá trị nhỏ nhất ở đây tức là ta lấy trung bình tổng khoảng cách đến gần cụm nhất. Muốn làm vậy ta buộc phải tính trung bình khoảng cách của tất cả các cụm và chọn giá trị nhỏ nhất.

Sau đó ta có thể tính giá trị Silhouette bằng công thức sau:

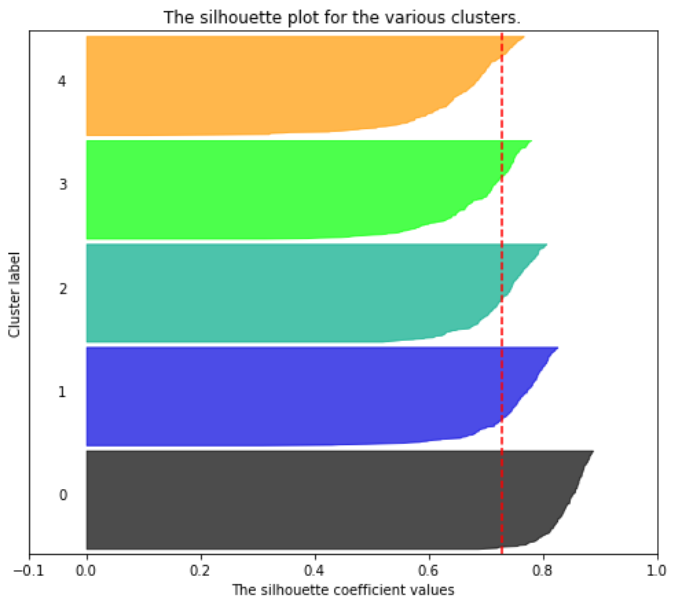
Và gắn cho nếu số lượng của cụm, có rằng buộc này để ngắn việc số lượng cụm tăng lên một cách đáng kể và tránh gặp lỗi khi số lượng nhỏ bằng 0 khiến máy tính chia cho 0:

Ta có thể viết lại rằng

Do đó cho thấy:

Muốn cho giá trị về gần 1 thì cần phải có . Do là giá trị dùng để đánh giá sự không giống nhau của một phần tử trong cụm, a(i) càng nhỏ thì càng thích hợp. Và thêm nữa, là giá trị chỉ độ không phù hợp với cụm lân cận của nó, b(i) càng lớn thì càng không thích hợp. Nếu s(i) càng về -1 thì ngược lại. Tức giá trị s(i) càng lớn thì độ thích hợp của điểm đó với cụm là càng cao.

Để có thể biểu diễn ra biểu đồ thì ta tính giá trị Silhouette cho toàn bộ các điểm và phân cụm mỗi điểm ra. Đồng thời tính giá trị Silhouette trung bình cho cụm đó và biệu thị ra biểu đồ. Thông thường thì ta sẽ chọn K nào có biểu đồ Silhouette có giá trị trung bình cao và phân bố các giá trị s(i) đồng đều, ít/không có giá trị s(i) nào bé hơn 0 ở mỗi cụm.



Hình 4. Biểu đồ mô ta các giá trị Silhouette của các cụm và Giá trị Silhouette trung bình

1. VẬN DỤNG THỰC TẾ
   1. Tập dữ liệu 1
   2. Tập dữ liệu 2
2. KẾT LUẬN
   1. So sánh các thuật toán.
   2. Các kết quả đạt được

Qua quá trình nghiên cứu thì bài báo cáo này đã đạt được môt số kết quả nhất định.

Thứ nhất, đã làm rõ được các vấn đề liên quan tới thuật toán Kmeans. Bài báo cáo đã trình bày những nét nổi bật, công dụng của thuật toán phân cụm Kmeans. Đồng thời nêu lên được thuật toán và các thực thi trên ngôn ngữ Python. Kết hợp với một số cải thiện trọng việc khởi tạo tâm ban đầu của thuật toán. Làm cho thuật toán này có phần tinh cậy hơn.

Thứ hai, bài báo cáo này đã trình bày được 2 phương pháp phổ biến và thông dụng để tìm K tối ưu đó là Elbow method và Silhouette method. Đã giải thích được thuật toán cũng như là cách sử dụng. 2 phương pháp này đã được thực thi thành công trên ngôn ngữ lập trình Python. Đồng thời đã cho ra được một số kết quả như ý muốn.

Thứ ba, với các phương pháp tìm k có sẳn. Đã có thể áp dụng được để tìm K tối ưu cho một số tập dữ liệu mẫu. Tuy rằng kết quả phân cụm có thể khác với thực tế. Những đã cho thấy rằng việc thực thi các phương pháp trên là thành công.

* 1. Các hạn chế

Kmeans là một thuật toán đơn giản và phổ biển dùng để phân cụm dữ liệu. Tuy vậy Kmeans gặp phải nhiều hạn chế khi chỉ có thể phân cụm dữ liệu mạng một số thuộc tính nhất định. Khi gặp một số cụm giữ liệu phức tạp, có kiểu hình khác nhau khí khó có thể áp dụng đúng được. Đồng thời với tập dữ liệu có số chiều D lớn thì tý lệ chính xác phần nào cũng giảm. Điều này là cho trong bào báo cáo này Kmeans vẫn chưa được điều chính về phương pháp tính khoảng cách. Chú yếu vẫn dùng phương pháp tính khoảng cách cố điển là Euclid.

Do nhiều vấn đề khác nhau nên quá trình thực thi thuật toán lên ngôn ngữ lập trình còn nhiều thiếu sót. Cho nên bài toàn có thể vẫn chưa được tối ưu. Dẫn đến thời gian thực thi còn khá chậm.

LIÊN KẾT VÀ TÀI LIỆU THAM KHẢO

Toàn bộ các tệp và dữ liệu được lưu vào:

Tài liệu tham khảo:

* [book\_ML.pdf](https://github.com/tiepvupsu/ebookMLCB)
* Kmeans++: <http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf>
* [Silhouette (clustering)](https://en.wikipedia.org/wiki/Silhouette_(clustering))