TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**NHẬP MÔN HỌC MÁY**

**ĐỀ TÀI:**

**Nhận dạng khuôn mặt dự đoán năm sinh và xu hướng tính cách**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện:** | **ĐOÀN HUY ĐỨC**  **NGUYỄN HUY KIÊN**  **TRẦN MINH HÙNG** | | **Giảng viên hướng dẫn:** | **NGÔ HOÀNG HUY** | | | **Ngành:** | **CÔNG NGHỆ THÔNG TIN** | | | **Chuyên ngành:** | **CÔNG NGHỆ PHẦN MỀM** | | | **Lớp:** | **D13CNPM5** | | | **Khóa:** | **2018-2023** | | |  |
|  |  |

***Hà Nội, tháng 10 năm 2021***

**PHIẾU CHẤM ĐIỂM**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và tên sinh viên** | **Nội dung thực hiện** | | **Điểm** | **Chữ ký** | |
| 1 | Đoàn Huy Đức  18810310421 |  | |  |  | |
| 2 | Trần Minh Hùng  18810310416 | |  |  | |  |
| 3 | Nguyễn Huy Kiên  18810310421 | |  |  | |  |

|  |  |  |
| --- | --- | --- |
| **Họ và tên giảng viên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

[LỜI CẢM ƠN 4](#_Toc75529527)

[LỜI MỞ ĐẦU 5](#_Toc75529528)

[CHƯƠNG 1: GIỚI THIỆU THUẬT TOÁN PHÂN CỤM K-MEANS 6](#_Toc75529529)

[**1. Giới thiệu** 6](#_Toc75529530)

[**2. Phân tích toán học** 8](#_Toc75529531)

[**2.1 Một số ký hiệu toán học** 8](#_Toc75529532)

[**2.2 Hàm mất mát và bài toán tối ưu** 9](#_Toc75529533)

[**2.3 Thuật toán tối ưu hàm mất mát** 10](#_Toc75529534)

[**2.3Tóm tắt thuật toán** 12](#_Toc75529535)

CHƯƠNG 2 : Phân cụm dữ liệu với thuật toán K-mean và ứng dụng trong phân đoạn ảnh………………………………………………………………………...13

## **2.1. Thuật toán K-means cho phân đoạn ảnh…………………………….**14

**2.2. Mô tả bài toán………………………………………………………….**14

## **2.3. Áp dụng thuật toán K-means cho phân đoạn ảnh…………………...**16

**2.4. Đánh giá và kết quả..…………………………………………………..**17

CHƯƠNG 3 : Kết luận……………………………………………………………18

# LỜI CẢM ƠN

Nhóm chúng em xin chân thành cảm ơn các thầy, cô giáo trong Khoa Công nghệ thông tin, trường Đại học Điện Lực, đã tạo điều kiện cho em thực hiện đề tài này.

Để có thể hoàn thành báo cáo đề tài “Nhận diện khuôn mặt dự đoán năm sinh và xu hướng tính cách”, nhóm em xin gửi lời cảm ơn chân thành nhất tới thầy*Ngô Hoàng Huy*, đã truyền đạt, giảng dạy cho chúng em những kiến thức, những kinh nghiệm qúy báu trong thời gian học tập và rèn luyện, tận tình hướng dẫn chúng em trong quá trình làm báo cáo này.

Nhóm em cũng gửi lời cảm ơn tới bạn bè đã đóng góp những ý kiến quý bàu để nhóm em có thể hoàn thành báo cáo tốt hơn.

Tuy nhiên, do thời gian và trình độ có hạn nên báo cáo này chắc chắn không tránh khỏi những thiếu sót, nhóm em rất mong được sự đóng góp ý kiến của các thầy và toàn thể các bạn.

Một lần nữa, em xin chân thành cảm ơn và luôn mong nhận được sự đóng góp quý báu của tất cả mọi người.

**Sinh viên thực hiện**

Đoàn Huy Đức

Trần Minh Hùng

Nguyễn Huy Kiên

# LỜI MỞ ĐẦU

Thuật toán phân cụm k-means là một phương pháp được sử dụng trong phân tích tính chất cụm của dữ liệu. Nó đặc biệt được sử dụng nhiều trong khai phá dữ liệu và thống kê. Nó phân vùng dữ liệu thành k cụm khác nhau. Giải thuật này giúp chúng ta xác định được dữ liệu của chúng ta nó thực sử thuộc về nhóm nào.

Để các bạn dễ hình dung ứng dụng của thuật toán. Chúng ta hãy quan sát một ví dụ thực tế như sau:

Trong các mô hình kinh doanh, doanh nghiệp sẽ chia nhỏ tệp khách hàng ra thành những nhóm đối tượng khác nhau để có thể áp dụng những chiến lược kinh doanh cụ thể cho từng nhóm đối tượng. Điều này giúp cho khách hàng được tiếp cận với các sản phẩm thật sự phù hợp với bản thân họ. Sự phù hợp đó sẽ kéo doanh số của chúng ta tăng lên. Vấn đề đặt ra là làm sao có thể chia nhỏ tệp khách hàng đó ra khi mà số lượng hóa đơn là rất lớn và chúng ta không thể ngồi để phân tích từng vị khách.

Và mục tiêu của các thuật toán phân cụm là từ tập dữ liệu khổng lồ đó. Làm sao chúng ta biết có những nhóm dữ liệu đặc trưng nào trong đó? Từng dữ liệu trong đó thuộc vào nhóm nào? Đó là cái mà thuật toán phân cụm của chúng ta cần đi tìm câu trả lời.

# CHƯƠNG 1: GIỚI THIỆU THUẬT TOÁN PHÂN CỤM K-MEANS

## **1. Giới thiệu**

Trong thuật toán K-means clustering, chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích là làm thể nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho **dữ liệu trong cùng một cụm có tính chất giống nhau**.

Ý tưởng đơn giản nhất về cluster (cụm) là tập hợp các điểm **ở gần nhau trong một không gian nào đó** (không gian này có thể có rất nhiều chiều trong trường hợp thông tin về một điểm dữ liệu là rất lớn). Hình bên dưới là một ví dụ về 3 cụm dữ liệu (từ giờ tôi sẽ viết gọn là **cluster**).

Chart, scatter chart

Description automatically generated

Bài toán với 3 clusters.

Giả sử mỗi cluster có một điểm đại diện (**center**) màu vàng. Và những điểm xung quanh mỗi center thuộc vào cùng nhóm với center đó. Một cách đơn giản nhất, xét một điểm bất kỳ, ta xét xem điểm đó gần với center nào nhất thì nó thuộc về cùng nhóm với center đó. Tới đây, chúng ta có một bài toán thú vị: **Trên một vùng biển hình vuông lớn có ba đảo hình vuông, tam giác, và tròn màu vàng như hình trên. Một điểm trên biển được gọi là thuộc lãnh hải của một đảo nếu nó nằm gần đảo này hơn so với hai đảo kia . Hãy xác định ranh giới lãnh hải của các đảo.**

Hình dưới đây là một hình minh họa cho việc phân chia lãnh hải nếu có 5 đảo khác nhau được biểu diễn bằng các hình tròn màu đen:

Chart

Description automatically generated

Phân vùng lãnh hải của mỗi đảo. Các vùng khác nhau có màu sắc khác nhau.

Chúng ta thấy rằng đường phân định giữa các lãnh hải là các đường thẳng (chính xác hơn thì chúng là các đường trung trực của các cặp điểm gần nhau). Vì vậy, lãnh hải của một đảo sẽ là một hình đa giác.

Cách phân chia này trong toán học được gọi là [Voronoi Diagram](https://en.wikipedia.org/wiki/Voronoi_diagram).

Trong không gian ba chiều, lấy ví dụ là các hành tinh, thì (tạm gọi là) **lãnh không** của mỗi hành tinh sẽ là một đa diện. Trong không gian nhiều chiều hơn, chúng ta sẽ có những thứ (mà tôi gọi là) **siêu đa diện** (hyperpolygon).

Quay lại với bài toán phân nhóm và cụ thể là thuật toán K-means clustering, chúng ta cần một chút phân tích toán học trước khi đi tới phần [tóm tắt thuật toán](https://machinelearningcoban.com/2017/01/01/kmeans/#tom-tat-thuat-toan) ở phần dưới. Nếu bạn không muốn đọc quá nhiều về toán, bạn có thể bỏ qua phần này. (**Tốt nhất là đừng bỏ qua, bạn sẽ tiếc đấy**).

## **2. Phân tích toán học**

Mục đích cuối cùng của thuật toán phân nhóm này là: từ dữ liệu đầu vào và số lượng nhóm chúng ta muốn tìm, hãy chỉ ra center của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng. Giả sử thêm rằng mỗi điểm dữ liệu chỉ thuộc vào đúng một nhóm.

### **2.1 Một số ký hiệu toán học**

Giả sử có NN điểm dữ liệu là:

X=[x1,x2,…,xN]∈Rd×NX=[x1,x2,…,xN]∈Rd×N và K<NK<N

Là số cluster chúng ta muốn phân chia. Chúng ta cần tìm các center  m1,m2,…,mK∈Rd×1m1,m2,…,mK∈Rd×1 và label của mỗi điểm dữ liệu.

Lưu ý về ký hiệu toán học: **trong các bài viết của tôi, các số vô hướng được biểu diễn bởi các chữ cái viết ở dạng không in đậm, có thể viết hoa, ví dụ**x1,N,y,kx1,N,y,k**. Các vector được biểu diễn bằng các chữ cái thường in đậm, ví dụ**m,x1m,x1**. Các ma trận được biểu diễn bởi các chữ viết hoa in đậm, ví dụ**X,M,YX,M,Y**. Lưu ý này đã được nêu ở bài**[*Linear Regression*](https://machinelearningcoban.com/2016/12/28/linearregression/)**. Tôi xin được không nhắc lại trong các bài tiếp theo.**

Với mỗi điểm dữ liệu xixi đặt yi=[yi1,yi2,…,yiK]yi=[yi1,yi2,…,yiK] là label vector

của nó, trong đó nếu xi được phân vào cluster kk thì yik=1yik=1 và yij=0,∀j≠kyij=0,∀j≠k. Điều này có nghĩa là có đúng một phần tử của vector yiyi là bằng 1 (tương ứng với cluster của xixi), các phần tử còn lại bằng 0. Ví dụ: nếu một điểm dữ liệu có label vector là [1,0,0,…,0][1,0,0,…,0] thì nó thuộc vào cluster 1, là [0,1,0,…,0][0,1,0,…,0] thì nó thuộc vào cluster 2, ……. Cách mã hóa label của dữ liệu như thế này được goi là biểu diễn [**one-hot**](https://en.wikipedia.org/wiki/One-hot). Chúng ta sẽ thấy cách biểu diễn one-hot này rất phổ biến trong Machine Learning ở các bài tiếp theo.

Ràng buộc của yiyi có thể viết dưới dạng toán học như sau:yik∈{0,1},   K∑k=1yik=1   (1)yik∈{0,1},   ∑k=1Kyik=1   (1)

### **2.2 Hàm mất mát và bài toán tối ưu**

Nếu ta coi center mkmk là center (hoặc representative) của mỗi cluster và **ước lượng** tất cả các điểm được phân vào cluster này bởi mkmk, thì một điểm dữ liệu xixi được phân vào cluster kk sẽ bị sai số là (xi−mk)(xi−mk). Chúng ta mong muốn sai số này có trị tuyệt đối nhỏ nhất nên ([giống như trong bài Linear Regression](https://machinelearningcoban.com/2016/12/28/linearregression/#sai-so-du-doan)) ta sẽ tìm cách để đại lượng sau đây đạt giá trị nhỏ nhất:∥xi−mk∥22‖xi−mk‖22

Hơn nữa, vì xixi được phân vào cluster kk nên yik=1,yij=0, ∀j≠kyik=1,yij=0, ∀j≠k. Khi đó, biểu thức bên trên sẽ được viết lại là:yik∥xi−mk∥22=K∑j=1yij∥xi−mj∥22yik‖xi−mk‖22=∑j=1Kyij‖xi−mj‖22

(**Hy vọng chỗ này không quá khó hiểu**)

Sai số cho toàn bộ dữ liệu sẽ là:L(Y,M)=N∑i=1K∑j=1yij∥xi−mj∥22L(Y,M)=∑i=1N∑j=1Kyij‖xi−mj‖22

Trong đó Y=[y1;y2;…;yN]Y=[y1;y2;…;yN], M=[m1,m2,…mK]M=[m1,m2,…mK] lần lượt là các ma trận được tạo bởi label vector của mỗi điểm dữ liệu và center của mỗi cluster. Hàm số mất mát trong bài toán K-means clustering của chúng ta là hàm L(Y,M)L(Y,M) với ràng buộc như được nêu trong phương trình (1)(1).

Tóm lại chúng ta cần tối ưu bài toán sau:

Y,M=argminY,MN∑i=1K∑j=1yij∥xi−mj∥22     (2)Y,M=arg⁡minY,M∑i=1N∑j=1Kyij‖xi−mj‖22     (2)

subject to:  yij∈{0,1}  ∀i,j;   K∑j=1yij=1  ∀i

(**subject to** nghĩa là **thỏa mãn điều kiện**).

Nhắc lại khái niệm argminarg⁡min: Chúng ta biết ký hiệu minmin là **giá trị nhỏ nhất của hàm số**, argminarg⁡min chính là **giá trị của biến số để hàm số đó đạt giá trị nhỏ nhất đó**. Nếu f(x)=x2−2x+1=(x−1)2f(x)=x2−2x+1=(x−1)2 thì giá trị nhỏ nhất của hàm số này bằng 0, đạt được khi x=1x=1. Trong ví dụ này minxf(x)=0minxf(x)=0 và argminxf(x)=1arg⁡minxf(x)=1. Thêm ví dụ khác, nếu x1=0,x2=10,x3=5x1=0,x2=10,x3=5 thì ta nói argminixi=1arg⁡minixi=1 vì 11 là chỉ số để xixi đạt giá trị nhỏ nhất (bằng 00). Biến số viết bên dưới minmin là biến số cúng ta cần tối ưu. Trong các bài toán tối ưu, ta thường quan tâm tới argminarg⁡min hơn là minmin.

### **2.3 Thuật toán tối ưu hàm mất mát**

Bài toán (2)(2) là một bài toán khó tìm **điểm tối ưu** vì nó có thêm các điều kiện ràng buộc. **Bài toán này thuộc loại mix-integer programming (điều kiện biến là số nguyên) - là loại rất khó tìm nghiệm tối ưu toàn cục (global optimal point, tức nghiệm làm cho hàm mất mát đạt giá trị nhỏ nhất có thể).** Tuy nhiên, trong một số trường hợp chúng ta vẫn có thể tìm được phương pháp để tìm được nghiệm gần đúng hoặc điểm cực tiểu. (**Nếu chúng ta vẫn nhớ chương trình toán ôn thi đại học thì điểm cực tiểu chưa chắc đã phải là điểm làm cho hàm số đạt giá trị nhỏ nhất**).

Một cách đơn giản để giải bài toán (2)(2) là xen kẽ giải YY và MM khi biến còn lại được cố định. Đây là một thuật toán lặp, cũng là kỹ thuật phổ biến khi giải bài toán tối ưu. Chúng ta sẽ lần lượt giải quyết hai bài toán sau đây:

#### **2.3.1 Cố định MM, tìm YY**

Giả sử đã tìm được các centers, hãy tìm các label vector để hàm mất mát đạt giá trị nhỏ nhất. Điều này tương đương với việc tìm cluster cho mỗi điểm dữ liệu.

Khi các centers là cố định, bài toán tìm label vector cho toàn bộ dữ liệu có thể được chia nhỏ thành bài toán tìm label vector cho từng điểm dữ liệu xixi như sau:

yi=argminyiK∑j=1yij∥xi−mj∥22   (3)yi=arg⁡minyi∑j=1Kyij‖xi−mj‖22   (3)subject to:  yij∈{0,1}  ∀j;   K∑j=1yij=1subject to:  yij∈{0,1}  ∀j;   ∑j=1Kyij=1

Vì chỉ có một phần tử của label vector yiyi bằng 11 nên bài toán (3)(3) có thể tiếp tục được viết dưới dạng đơn giản hơn:j=argminj∥xi−mj∥22j=arg⁡minj‖xi−mj‖22

Vì ∥xi−mj∥22‖xi−mj‖22 chính là bình phương khoảng cách tính từ điểm xixi tới center mjmj, ta có thể kết luận rằng mỗi điểm xixi thuộc vào cluster có center gần nó nhất! Từ đó ta có thể dễ dàng suy ra label vector của từng điểm dữ liệu.

#### **2.3.2 Cố định YY, tìm MM**

Giả sử đã tìm được cluster cho từng điểm, hãy tìm center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất.

Một khi chúng ta đã xác định được label vector cho từng điểm dữ liệu, bài toán tìm center cho mỗi cluster được rút gọn thành:

mj=argminmjN∑i=1yij∥xi−mj∥22.mj=arg⁡minmj∑i=1Nyij‖xi−mj‖22.Tới đây, ta có thể tìm nghiệm bằng phương pháp giải đạo hàm bằng 0, vì hàm cần tối ưu là một hàm liên tục và có đạo hàm xác định tại mọi điểm. **Và quan trọng hơn, hàm này là hàm convex (lồi) theo**mjmj**nên chúng ta sẽ tìm được giá trị nhỏ nhất và điểm tối ưu tương ứng. Sau này nếu có dịp, tôi sẽ nói thêm về tối ưu lồi (convex optimization) - một mảng cực kỳ quan trọng trong toán tối ưu**.

Đặt l(mj)l(mj) là hàm bên trong dấu argminarg⁡min, ta có đạo hàm:∂l(mj)∂mj=2N∑i=1yij(mj−xi)∂l(mj)∂mj=2∑i=1Nyij(mj−xi)

Giải phương trình đạo hàm bằng 0 ta có:mjN∑i=1yij=N∑i=1yijximj∑i=1Nyij=∑i=1Nyijxi⇒mj=∑Ni=1yijxi∑Ni=1yij⇒mj=∑i=1Nyijxi∑i=1Nyij

Nếu để ý một chút, chúng ta sẽ thấy rằng mẫu số chính là phép đếm **số lượng các điểm dữ liệu** trong cluster jj (**Bạn có nhận ra không?**). Còn tử số chính là **tổng các điểm dữ liệu** trong cluster jj. (**Nếu bạn đọc vẫn nhớ điều kiện ràng buộc của các** yijyij **thì sẽ có thể nhanh chóng nhìn ra điều này**).

Hay nói một cách đơn giản hơn nhiều: mjmj là trung bình cộng của các điểm trong cluster jj.

Tên gọi **K-means clustering** cũng xuất phát từ đây.

### **2.4 Tóm tắt thuật toán**

Tới đây tôi xin được tóm tắt lại thuật toán (**đặc biệt quan trọng với các bạn bỏ qua phần toán học bên trên**) như sau:

Đầu vào: Dữ liệu XX và số lượng cluster cần tìm KK.

Đầu ra: Các center MM và label vector cho từng điểm dữ liệu YY.

1. Chọn KK điểm bất kỳ làm các center ban đầu.
2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.
3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.
4. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất các các điểm dữ liệu đã được gán vào cluster đó sau bước 2.
5. Quay lại bước 2.

Chúng ta có thể đảm bảo rằng thuật toán sẽ dừng lại sau một số hữu hạn vòng lặp. Thật vậy, vì hàm mất mát là một số dương và sau mỗi bước 2 hoặc 3, giá trị của hàm mất mát bị giảm đi. Theo kiến thức về dãy số trong chương trình cấp 3: **nếu một dãy số giảm và bị chặn dưới thì nó hội tụ!** Hơn nữa, số lượng cách phân nhóm cho toàn bộ dữ liệu là hữu hạn nên đến một lúc nào đó, hàm mất mát sẽ không thể thay đổi, và chúng ta có thể dừng thuật toán tại đây.

Chúng ta sẽ có một vài [thảo luận](https://machinelearningcoban.com/2017/01/01/kmeans/#-thao-luan) về thuật toán này, về những hạn chế và một số phương pháp khắc phục. Nhưng trước hết, hãy xem nó thể hiện như thế nào trong một ví dụ cụ thể dưới đây.

**CHƯƠNG 2: Phân cụm dữ liệu với thuật toán K-mean và ứng dụng trong phân đoạn ảnh**

## **2.1. Thuật toán K-means cho phân đoạn ảnh**

Tầm quan trọng và những khó khăn của việc gom nhóm các đối tƣợng mang tính tri giác của con người từ lâu đã được nghiên cứu nhiều trong các lĩnh vực của thị giác máy tinh đặc biệt trong lĩnh vực của xử lí ảnh. Và phân đoạn ảnh đã có những ứng dụng mạnh mẽ và rộng rãi trong các bài toán phân tích và hiểu ảnh tự động, nhƣng nó cũng là một bài toán khó mà đến bây giờ các nhà khoa học vẫn chƣa giải quyết đƣợc một cách hoàn toàn thấu đáo. Làm thế nào để phân chia một ảnh thành các tập con. Những cách khả thi để có thể làm đƣợc điều đó. Đó là những câu hỏi mà người ta đã đặt ra từ lâu và mong muốn tìm được câu trả lời. Trong khoảng 30 năm trở lại đây đã có rất nhiều các thuật toán đƣợc đề xuất để giải quyết bài toán phân đoạn ảnh. Các thuật toán hầu hết đều dựa vào hai thuộc tính quan trọng của mỗi điểm ảnh so với các điểm lân cận của nó, đó là: sự khác (dissimilarity) và giống nhau (similarity). Các phương pháp dựa trên sự giống nhau của các điểm ảnh đƣợc gọi là phương pháp miền (region-based methods), còn các phương pháp dựa trên sự khác nhau của các điểm ảnh đƣợc gọi là các phương pháp biên (boundarybased methods). Trong bài báo cáo này em xin phép được trình bày thuật toán K-means để giải quyết bài toán phân đoạn ảnh.

## **2.2. Mô tả bài toán**

Input:

-Ảnh có kích thước m\*n

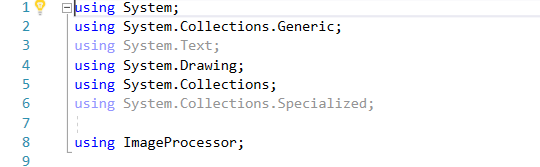
-Số cụm (k) muốn phân đoạn.

Output:

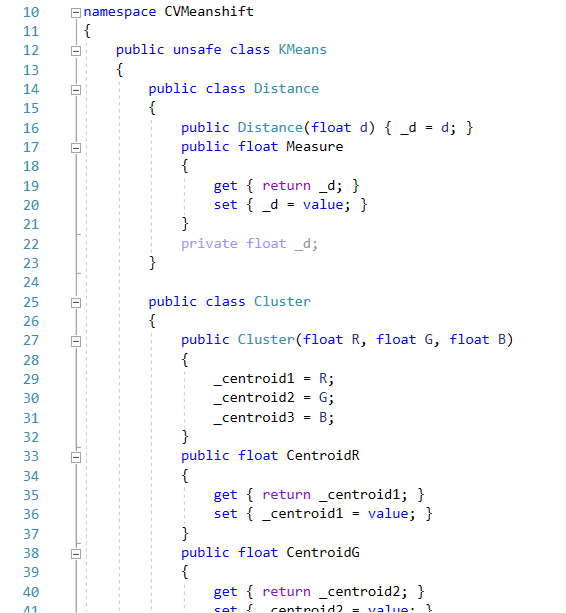
-Ảnh được phân thành k đoạn có màu sắc tương đồng nhau.

## **2.3. Áp dụng thuật toán K-means cho phân đoạn ảnh**

- Sử dụng các thư viện



-Áp dụng thuật toán K-means

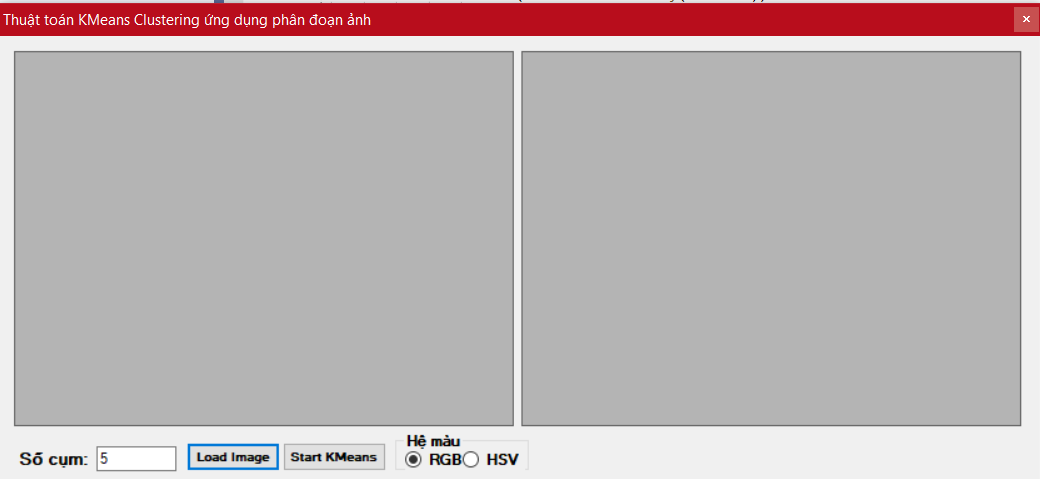
* 

**2.4. Đánh giá và kết quả**

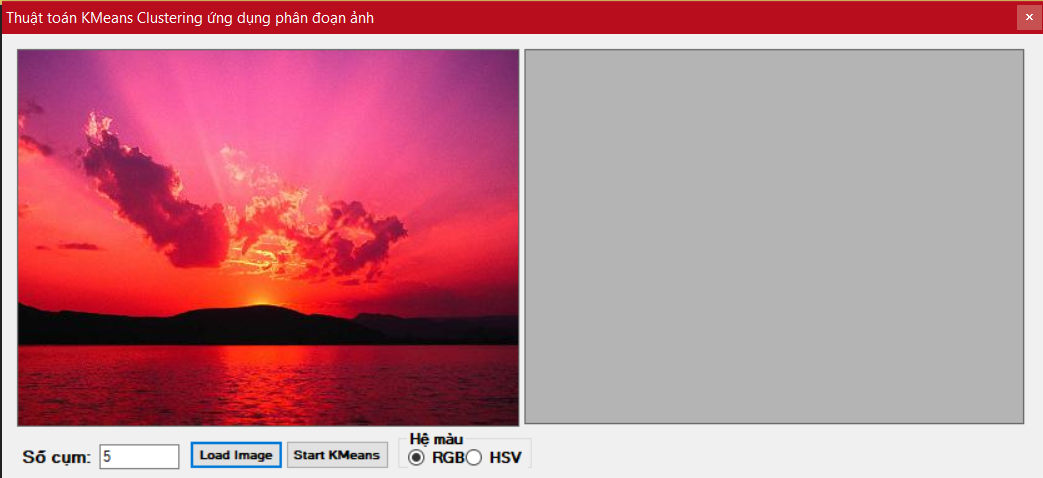
**2.4.1. Môi trường cài đặt**   
 Chương trình lập trình với ngôn ngữ C#, cài đặt và chạy thử nghiệm trên môi trường hệ điều hành Windows XP.

**2.4.2 Một số giao diện**

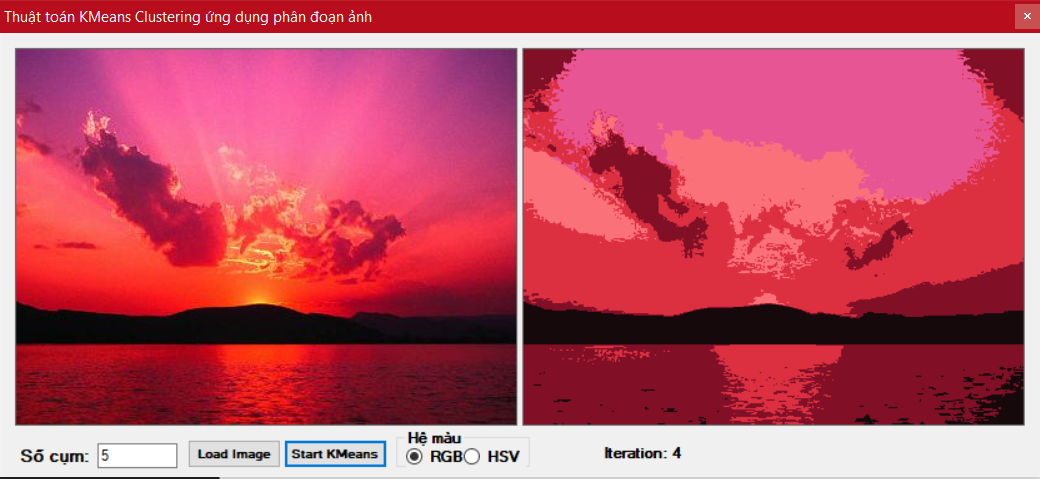
* Giao diện khởi động



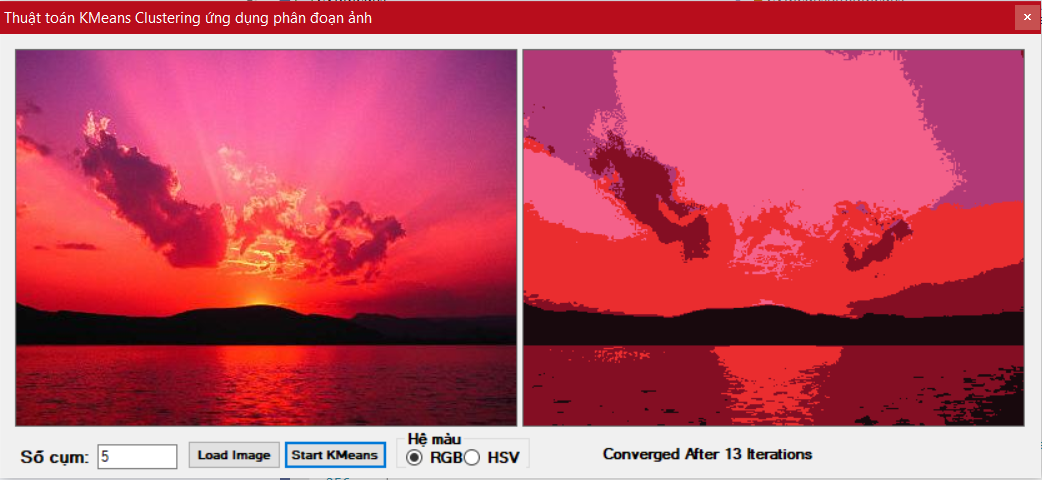
* Đưa dữ liệu vào



* Quá trình xử lý dữ liệu



* Kết quả phân cụm



**CHƯƠNG 3: KẾT LUẬN**  
 Trong quá trình nghiên cứu, tìm hiểu và hoàn thành đề tài đồ án tốt nghiệp “Tìm hiểu một số phuơng pháp phân cụm dữ liệu và ứng dụng”, em đã thu nhận đƣợc thêm những kiến thức và em cũng nhận thấy phân cụm dữ liệu trong khai phá dữ liệu là một lĩnh vực nghiên cứu rộng lớn, còn nhiều điều mà chúng ta cần khám phá. Trong đề tài em đã cố gắng tập trung tìm hiểu và nghiên cứu tổng quan khai phá dữ liệu, phân cụm dữ liệu và một số thuật toán của nó, tổng quan về phân vùng ảnh. Cài đặt thử nghiệm thuật toán k-means với ứng dụng là phân đoạn ảnh. Do thời gian thực hiện hạn chế nên em mới chỉ tìm hiểu đựơc một số kỹ thuật cơ bản trong phân cụm dữ liệu, cài đặt thử nghiệm với thuật toán Kmeans. Nhƣng còn một số các kỹ thuật em vẫn chƣa tìm hiểu, khai thác và ứng dụng cho các bài toán Trong thời gian tới em sẽ cố gắng tiếp tục nghiên cứu, tìm hiểu thêm một sô kỹ thuật phân cụm và nhất là có thể tìm hiểu và phát triển các kỹ thuật phân đoạn ảnh để có thể xử lý với ảnh động.