Bài tập về nhà 9-10

Nguyễn Khắc Sơn - DHIOT17A - MSSV: 21085691

Bài 1:

Dựa trên những kiến thức và tìm hiểu của em về bài toán phân cụm (clustering), em hãy

- Lấy một ví dụ liên quan tới một vấn đề trong lĩnh vực IoT mà em có thể giải quyết thông qua việc sử dụng bài toán phân cụm.
- Giải thích về lý do em lựa chọn
- Nêu những ưu điểm của phân cụm khi em áp dụng để giải quyết vấn đề này?

Giải

- Sử dụng phân cụm trong lĩnh vực xử lý ảnh (Computer Vision) là phân đoạn ảnh (image segmentation) trong **hệ thống giám sát giao thông thông minh**
- Lý do em lựa chọn:
    - **Xử lý tự động và nhanh chóng:** Phân cụm giúp phân đoạn ảnh nhanh chóng mà không cần yêu cầu can thiệp thủ công, tiết kiệm thời gian và chi phí trong các hệ thống giám sát thời gian thực.
    - **Tăng cường khả năng phát hiện đối tượng:** Phân cụm pixel dựa trên đặc điểm tương đồng giúp cải thiện độ chính xác trong việc phân biệt các đối tượng khác nhau trong ảnh.
    - **Tối ưu hóa trong các tình huống phức tạp:** Trong các cảnh giao thông đông đúc, phân cụm giúp nhận diện các đối tượng một cách có tổ chức, đặc biệt khi có nhiều phương tiện và người đi bộ.

- Ưu điểm của phân cụm:
    - **Hiệu quả trong phân đoạn ảnh:** Phân cụm giúp chia hình ảnh thành các vùng khác nhau dựa trên đặc điểm như màu sắc, độ sáng, và kết cấu, giúp hệ thống dễ dàng nhận diện và phân tích các đối tượng trong cảnh giao thông.
    - **Cải thiện độ chính xác trong nhận diện đối tượng:** Việc phân nhóm các pixel tương đồng giúp hệ thống nhận diện các phương tiện và người đi bộ chính xác hơn, giảm thiểu các lỗi phát hiện sai.
    - **Tăng hiệu suất xử lý:** Phân cụm cho phép hệ thống giám sát xử lý dữ liệu ảnh một cách hiệu quả, đảm bảo phản hồi nhanh và chính xác trong thời gian thực, đặc biệt là trong các hệ thống giám sát giao thông liên tục.
    - **Dễ dàng mở rộng**

- Ưu điểm?
  Hệ thống giám sát giao thông thông minh có thể sử dụng phân cụm để phân tích các cảnh quay thời gian thực từ camera, giúp phân loại và theo dõi đối tượng một cách hiệu quả. Điều này có thể hỗ trợ việc **phát hiện tắc đường, xác định hành vi lái xe nguy hiểm, hoặc tự động phát hiện vi phạm giao thông trong** một khu vực.

Bài 2:

Phân cụm (clustering) có nhiều các ứng dụng khác nhau trong nhiều lĩnh vực. Em hãy lựa chọn và trình bày về một ứng dụng của phân cụm mà em quan tâm nhất.

Giải

- Một ứng dụng em quan tâm nhất là **phân tích dữ liệu khách hàng trong lĩnh vực game và thể thao điện tử (eSports)**
- Phân cụm giúp giải quyết 1 số vấn đề như
  - *Phân loại người chơi theo kỹ năng và hành vi*
  - *Cá nhận hóa khuyến mãi và nội dung trong game*
  - *Phân tích và phát triển chiến lược eSports*
- Em nghĩ ứng dụng này thú vị vì chúng giúp **cá nhân hóa trải nghiệm của người chơi**. Yếu tố này là quan trọng nhất trong việc giữ chân người chơi, giúp xác định chinh xác nhưng gì người chơi muốn, từ đó tạo ra trải nghiệm thu hút, hấp dẫn hơn.
  Tiếp theo, phân cụm giúp cho các nhà phát triển **điều chỉnh trò chơi, phát triển, cân bằng game** 1 cách phù hợp, đồng thời xây dựng cộng đồng eSports chuyên nghiệp và lành mạnh hơn. Ngoài ra, các vật phẩm hay dịch vụ cũng được tạo ra để **tăng doanh thu** từ các giao dịch trong trò chơi

- Mặc dù phân cụm mang lại nhiều lợi ích cho việc phân tích dữ liệu khách hàng nhưng một số khó khăn vẫn còn tồn đọng:

Khó khăn	Giải thích	Giải pháp
Phân cụm tĩnh, không theo thời gian thực	Các thuật toán phân cụm truyền thống như **k-means** hoặc **hierarchical clustering** (gọi là Phân cụm phân cấp) thường tạo ra các cụm cố định và không cập nhật theo thời gian thực. Tuy nhiên, trong lĩnh vực game, hành vi của người chơi thay đổi liên tục theo thời gian.	Các thuật toán **học máy trực tuyến** (online learning) hoặc **phân cụm động** (dynamic clustering) có thể sử dụng để giải quyết
Khó xử lý dữ liệu không đồng nhất	Dữ liệu của người chơi game có thể rất đa dạng và phức tạp, bao gồm cả dữ liệu số, dữ liệu văn bản, dữ liệu về tương tác trong game,... Các thuật toán phân cụm truyền thống khó xử lý hiệu quả các loại dữ liệu không đồng nhất này.	Thuật toán **học sâu** (deep learning) có thể được sử dụng để trích xuất các đặc trưng từ dữ liệu không đồng nhất và sau đó áp dụng phân cụm trên các đặc trưng đã được mã hóa.
Không tận dụng được dữ liệu nhãn có sẵn	Trong nhiều trường hợp, đã có dữ liệu nhãn từ các nhóm người chơi trước (ví dụ: người chơi mới, người chơi chuyên nghiệp, người tiêu dùng nhiều). Không sử dụng thông tin nhãn này có thể dẫn đến kết quả phân cụm kém hiệu quả.	Sử dụng các thuật toán **bán giám sát.** Ngoài ra, các thuật toán như **K-Means Semi-Supervised** có thể tận dụng các nhãn đã biết để cải thiện độ chính xác phân cụm.

Bài 3:

Để xây dựng một tòa nhà văn phòng thông minh, một công ty đã tiến hành thu thập được một tập dữ liệu từ những nhân viên làm việc trong văn phòng thông qua khảo sát (xem Bảng 1). Mặc dù vậy có nhiều nhân viên đã không điền đầy đủ thông tin trong mục Mức độ hài lòng.

Em hãy sử dụng thuật toán **k-means** để tự động điền các thông tin còn thiếu vào cột Mức độ hài lòng trong Bảng 1. Giả sử rằng

Mẫu	Nhiệt độ (°C)	Độ ẩm (%)	Mức độ hài lòng
S1	30	60	
S2	25	70	Dễ chịu
S3	35	80	
S4	28	75	
S5	33	85	Khó chịu
S6	26	65	
S7	36	90	
S8	27	68	Dễ chịu
S9	32	55	Tuyệt vời

mức độ hài lòng của các nhân viên chỉ gồm có 3 mức là: Khó chịu, Dễ chịu, và Tuyệt vời.

Gợi ý: Mỗi mẫu dữ liệu gồm 2 đặc trưng (Nhiệt độ và Độ ẩm) và Mức độ hài lòng có thể được coi là nhãn (để sử dụng trong các bài toán học có giám sát khác).

Giải

BT - Học Máy Ngày: 21-10

Bài 3:

Mẫu	t°C	Độ ẩm	Mức độ hài lòng
s1	30	60	Tuyệt vời
s2	25	70	Dễ chịu (1)
s3	35	75	Khó chịu
s4	28	80	Dễ chịu
s5	33	85	Khó chịu (2)
s6	26	65	Dễ chịu
s7	36	90	Khó chịu
s8	27	68	Dễ chịu
s9	32	55	Tuyệt vời (3)

B1> Lựa chọn tâm cụm

c_1 (25, 70) "Dễ chịu"

c_2 (33, 85) "Khó chịu"

c_3 (32, 55) "Tuyệt vời"

B2> Tính khoảng cách từ các mẫu đến các tâm cụm

- $d(s_1, c_1) = \sqrt{(30-25)^2 + (60-70)^2}$ * $= 11.18$
- $d(s_1, c_2) =$ " $= 25.18$
- $d(s_1, c_3) =$ " $= 5.39$ → gần c_3 nhất

=> $s_1 \in c_3$

* công thức Euclidean: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Ngày:

$S_2(35,06)$

$d(S_2, c_1) = 14,14$

$d(S_2, c_2) = 5,39$

$d(S_2, c_3) = 25,2$

$\Rightarrow S_2 \in c_2$

$S_4(28,75)$

$d(S_4, c_1) = 5,83$

$d(S_4, c_2) = 11,18$

$d(S_4, c_3) = 20,39$

$\Rightarrow S_4 \in c_1$

$S_6(26,65)$

$d(S_6, c_1) = 5,1$

$d(S_6, c_2) = 21,2$

$d(S_6, c_3) = 11,66$

$\Rightarrow S_6 \in c_1$

$d(S_7, c_1) = 22,32$

$d(S_7, c_2) = 5,83$

$d(S_7, c_3) = 35,23$

$\Rightarrow S_7 \in c_2$

Bài 4: (Thực hành với Python)

Phân đoạn hình ảnh (Image Segmentation) đề cập đến việc phân chia một hình ảnh thành nhiều phân đoạn (vùng) dựa trên sự giống nhau của các giá trị pixel. Phân đoạn hình ảnh thường được sử dụng để phát hiện đối tượng hoặc phân biệt các phần khác nhau của hình ảnh. Một bạn sinh viên đã áp dụng thuật toán phân cụm k-means với k = 3 để phân cụm bức ảnh lenna.png (Hình 1).

Dựa trên ví dụ của bạn sinh viên đã thực hiện (file Image_Segmentation.ipynb), em hãy thực hiện các tác vụ sau:

a) **Em hãy sửa đổi giá trị của số lượng cụm để xem số lượng cụm khác nhau ảnh hưởng đến việc phân đoạn hình ảnh như thế nào. (Hãy thử với số lượng cụm khác nhau, ví dụ 2, 4, 7, ...). Việc tăng số lượng cụm ảnh hưởng đến mức độ chi tiết trong hình ảnh được phân đoạn như thế nào?**

Quá trình phân đoạn ảnh dựa trên thuật toán K-means clustering đã được thực hiện với các giá trị K khác nhau (K là số lượng cụm). Dưới đây là sự ảnh hưởng của việc tăng số lượng cụm (K) đến mức độ chi tiết của hình ảnh phân đoạn:

- **K=1**: Toàn bộ hình ảnh được gán vào một cụm, tạo ra một màu đồng nhất và mất hoàn toàn chi tiết.
- **K=2**: Hình ảnh được chia thành 2 màu chính, xuất hiện một số đặc trưng cơ bản nhưng vẫn thiếu chi tiết.
- **K=5**: Chi tiết hình ảnh cải thiện rõ rệt, các vùng khác nhau bắt đầu phân biệt tốt hơn, bao gồm da, mũ, và nền.
- **K=10**: Chi tiết trở nên rõ ràng hơn, màu sắc phân chia mịn màng, và các chi tiết nhỏ như mũ và tóc rõ nét hơn.

Như vậy, khi số cụm K tăng, mức độ chi tiết trong hình ảnh phân đoạn cũng tăng theo. Tuy nhiên, nếu tăng K quá cao, việc phân đoạn có thể trở nên quá phức tạp và làm mất tính tổng quát của ảnh phân đoạn.

b) Theo mặc định, k-means sử dụng phương pháp khởi tạo k-means++. Thay đổi phương pháp khởi tạo thành ngẫu nhiên và quan sát sự khác biệt. Phương pháp khởi tạo ảnh hưởng như thế nào đến kết quả phân đoạn và tốc độ hội tụ?

k-means++	**"random"**
Thay vì chọn tâm cụm ban đầu một cách ngẫu nhiên, k-means++ sẽ chọn các tâm ban đầu sao cho chúng cách xa nhau nhất có thể. Điều này giúp tránh tình trạng các tâm cụm ban đầu gần nhau, dẫn đến kết quả phân cụm không hiệu quả.	Khi sử dụng phương pháp này, các tâm cụm ban đầu được chọn một cách tùy ý từ các điểm trong tập dữ liệu. Điều này có thể dẫn đến việc chọn các tâm cụm không tối ưu, gây ra kết quả phân đoạn kém chính xác hoặc không hợp lý, đặc biệt nếu các tâm cụm ban đầu rơi vào các vùng dữ liệu gần nhau.

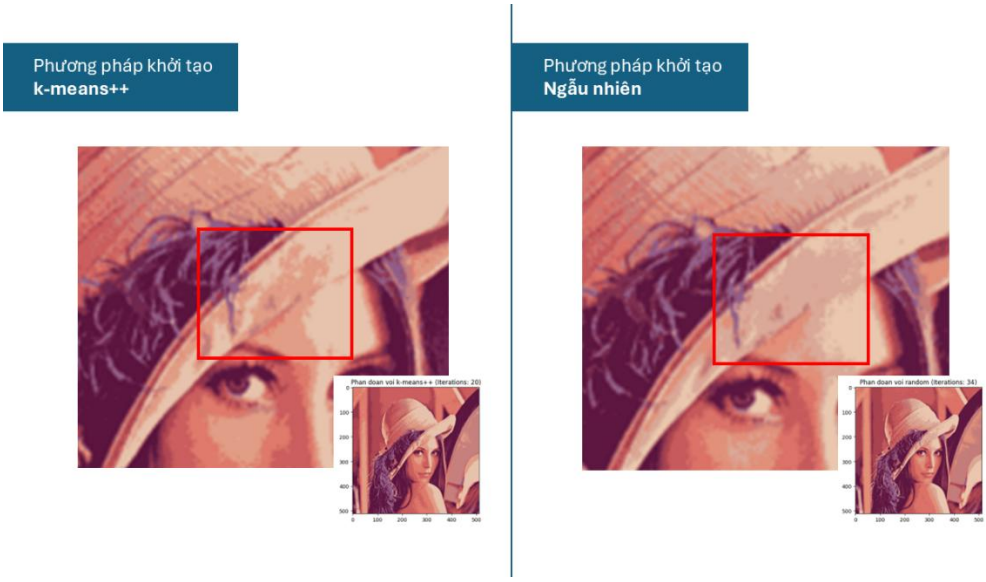
Đầu tiên, thay đổi phương pháp khởi tạo k-means++ thành ngẫu nhiên

```
# Chọn số lượng cụm (clusters)
K = 10
kmeans = KMeans(n_clusters=K, init='random', random_state=0)
```

Tiếp theo, chạy 2 đoạn code để kiểm tra kết quả phân đoạn cũng như tốc độ hội tụ:

Đối với phương pháp khởi tạo **k-means++**	Đối với phương pháp khởi tạo **ngẫu nhiên**
○ Tốc độ hội tụ nhanh hơn : **0.9s** ○ Số lần lặp với k-means++: **20**	○ Tốc độ hội tụ chậm hơn : **1.9s** ○ Số lần lặp với k-means++: **34**

Quan sát sự khác biệt:



Nhận xét:

- Đối với **khởi tạo thông minh** (K-means++), ảnh sau khi phân đoạn sẽ có các vùng màu đồng nhất và dễ phân biệt hơn. Các khu vực khác nhau trên ảnh được phân thành cụm một cách rõ ràng và hợp lý.
- Đối với **khởi tạo ngẫu nhiên**, ảnh có thể bị phân đoạn với các cụm màu sắc ít rõ ràng, các vùng màu có thể không chính xác và kết quả có thể thay đổi rất nhiều giữa các lần chạy, vì khởi tạo ngẫu nhiên có thể dẫn đến các kết quả không nhất quán.

c) Sau khi thực hiện phân cụm k-means, hãy trực quan hóa các màu chủ đạo (trung tâm cụm) biểu diễn các vùng được phân đoạn. Các màu chủ đạo (trung tâm cụm) được xác định bởi k-means trong hình ảnh là gì?

Lấy các trung tâm cụm từ kết quả của quá trình K-means.

Chuyển các giá trị trung tâm cụm (đang ở dạng float32) về kiểu dữ liệu uint8, vì màu sắc trong hình ảnh thông thường được biểu diễn với các giá trị từ 0 đến 255.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

```
# Load file hình ảnh (ví dụ: file ảnh 'lenna.png' chứa cùng chung trong thư
mục)
image = cv2.imread('lenna.png')
# Chuyển từ BGR (OpenCV) sang RGB
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Định dạng ảnh thành một mảng 2D chứa các pixel (width * height, 3)
pixel_values = image.reshape((-1, 3))
pixel_values = np.float32(pixel_values)

# Chọn số lượng cụm (clusters)
K = 10

# Thực hiện phân cụm k-means với khởi tạo 'k-means++'
kmeans = KMeans(n_clusters=K, init='k-means++', random_state=0)
kmeans.fit(pixel_values)

# Lấy các trung tâm cụm (màu chủ đạo)
centers = kmeans.cluster_centers_
# Chuyển đổi trung tâm cụm sang kiểu uint8 (0-255)
centers = np.uint8(centers)

# Hiển thị các màu chủ đạo (trung tâm cụm)
plt.figure(figsize=(8, 2))
for i in range(K):
    plt.subplot(1, K, i+1)
    plt.imshow([[centers[i]]])
    plt.axis('off')

plt.suptitle(f'Cac mau chu dao (trung tam cum) voi K={K}')
plt.show()
```