TRƯỜNG ĐH CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập – Tự do – Hạnh phúc

THỰC HÀNH 1 CLUSTERING

Môn học

Máy học trong thị giác máy tính

Lớp : CS332.I11.KHTN

GVLT: TS Lê Đình Duy

SVTH: 1) Triệu Tráng Vinh – 14521097

TP.Hồ Chí Minh, tháng 10, năm 2017

Mục lục

Mục lục hình ảnh

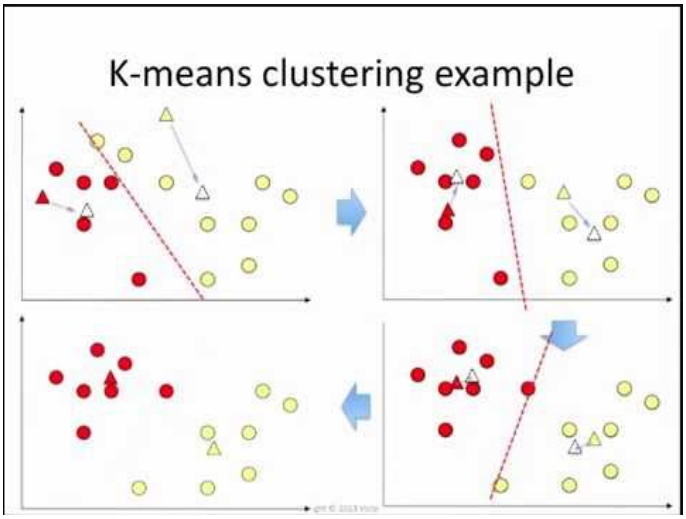
1. Giới thiệu bài toán

- Bài toán Clustering (gom nhóm) gom các dữ liệu thành các nhom liên quan gần nhau. Một số thuật toán như Kmean, DBSCAN…
- **Input**: Tập dữ liệu.
- **Output**: Tập dữ liệu đã được gom thành các nhóm với nhau.
- **Source code**: https://github.com/mrwen00/MachineLearning-TH1-Clustering

2. Ý tưởng các thuật toán Clustering:

a. Kmean:

- **Thuật toán K mean** là thuật toán phân dữ liệu thành K cụm, với K là số cụm cần gom, việc lựa chọn cụm dựa trên việc tính khoảng cách các điểm đến tâm cụm (centroid). Thuật toán được trình bày như sau:
    i. Chọn ngẫu nhiên K điểm từ tập hợp dữ liệu làm centroid. Mỗi cụm được đại diện bằng tâm của nó.
    ii. Tính khoảng cách các điểm đến mỗi centroid
    iii. Nhóm các điểm có khoảng cách gần centroid nhất.
    iv. Xác định centroid mới cho các điểm
    v. Lặp lại đến khi kết quả không đổi.



Hình 1 Kmean clustering example

b. Spectral:

- Đưa các đối tượng về dạng đồ thị tương đồng, dùng k dimension để phân chia đồ thị

c. DBSCAN

- Gom các nhóm điểm chứa lẫn nhau và chứa nhiều hơn một ngưỡng, nếu thấp hơn sẽ xem là nhiễu.

d. Agglomerative

- Đi từ bottom up gom các nhóm gần nhau nhất cho đến khi chỉ còn 1 cluster.

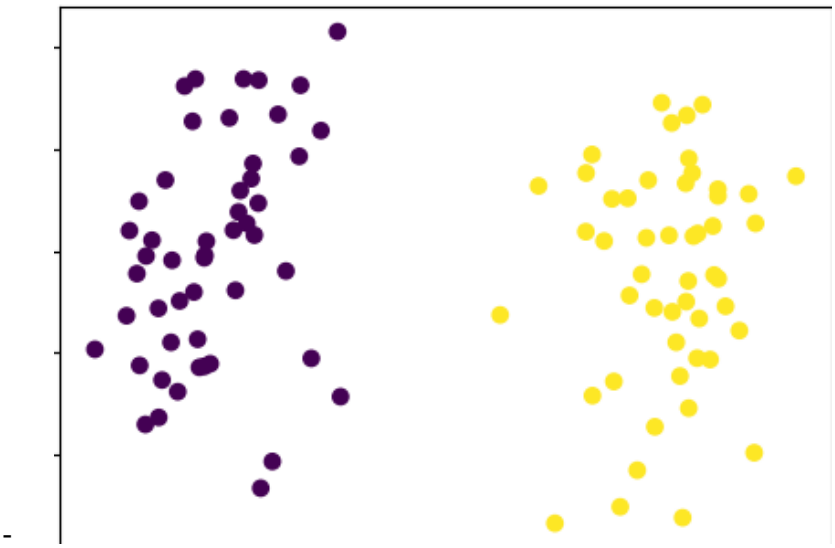
3. Yêu cầu 1 Thực hiện thuật toán Kmean

- Bài toán Kmean giúp gom nhóm (clustering) các điểm thành các nhóm với nhau.
- Dưới đây là minh họa code thực hiện gom nhóm bằng phương phap Kmean.

```
import numpy as np
import matplotlib.pyplot as plt
from random import randint

from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

plt.figure(figsize=(12, 12))

n_samples = 100
random_state = randint(1, 1000)
X, y = make_blobs(n_samples=n_samples,centers = 2, random_state=random_state)

y_pred = KMeans(n_clusters=2, random_state=random_state).fit_predict(X)

plt.subplot(221)
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
plt.title("Kmean")

print X

plt.show()
```

Hình 2 Thuật toán Kmean

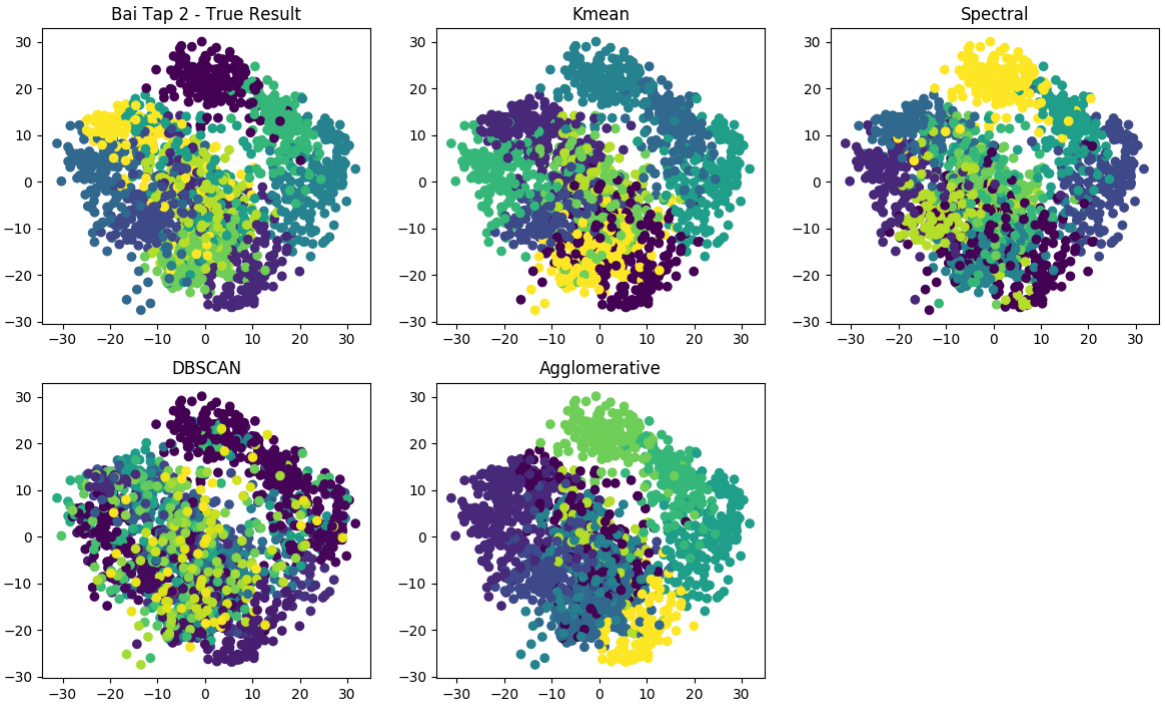
- Đầu tiên ta sẽ phát sinh dữ liệu mẫu, với n_samples = 100 số điểm. Với mỗi điểm gồm cặp tọa độ x, y. Tại dòng 18, ta sử dụng thư viện Kmean để gom nhóm dữ liệu với cluster = 2, sau đó hiển thị các điểm trên đồ thị. Với những điểm cùng màu thì sẽ nằm trong 1 nhóm.



Hình 3 Visualize kết quả Kmean

4. Yêu cầu 2 Hand written digit.

- Dùng tập dữ liệu Hand written digit, Viết chương trình để clustering tập dữ liệu trên sử dụng các phương pháp Kmeans, Spectral Clustering, DBSCAN, Agglomerative Clustering. Visualize, so sánh và đánh giá kết quả.
- Dữ liệu: Lấy từ tập dữ liệu hand written digit của sklearn. Gồm 10 chữ số 0 →9.
- Dưới đây là kết quả sau quá trình clustering dựa trên các thuật toán.



Hình 4 So sánh các thuật toán Hand-written digits

- Việc đánh giá performance của thuật toán ta dựa trên hàm scikit metrics *metrics.adjusted_mutual_info_score*, đưa ra tỉ lệ phần trăm đúng giữa label và kết quả sau khi clustering.
- So sánh performance của các thuật toán dựa trên label có sẵn của dataset, ta có như sau.

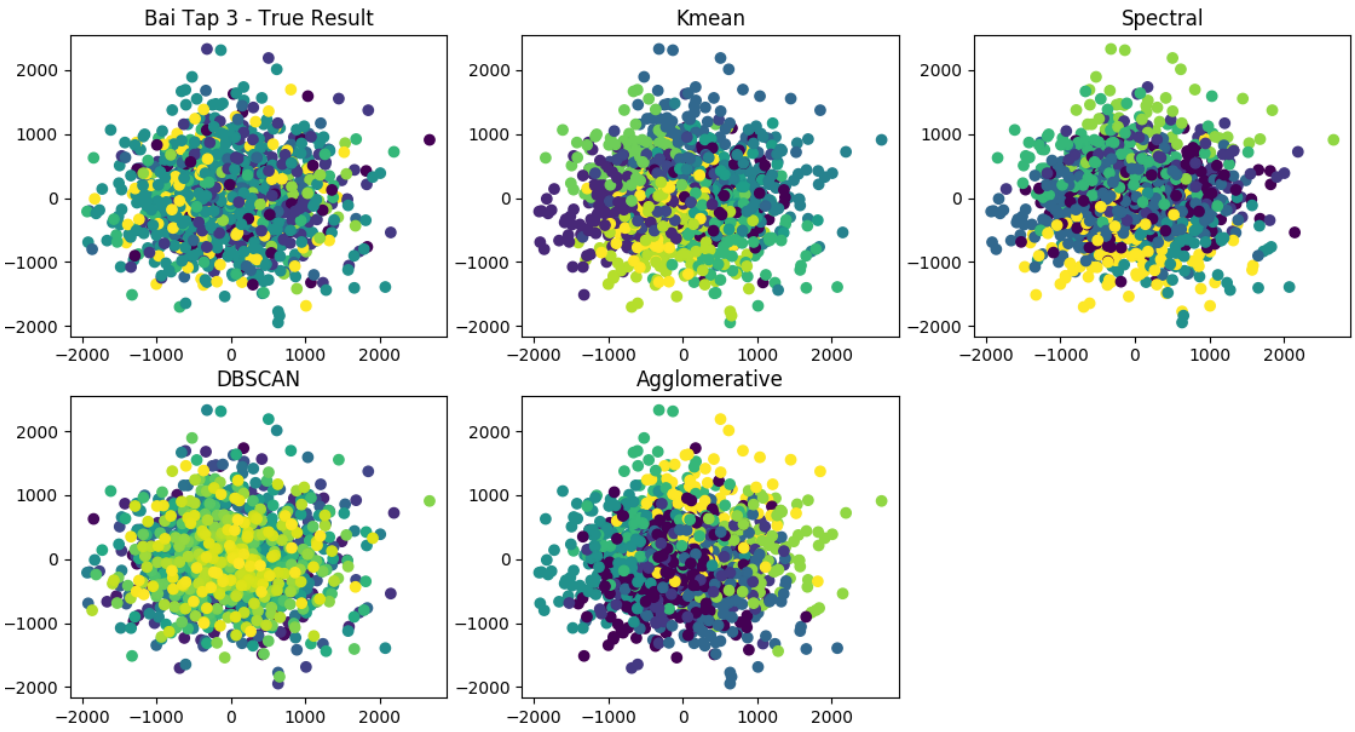
	Kmean	Spectral	DBSCAN	Agglomerative
Percentage	73.54%	70.85%	29.53%	85.60%

- Kết luận: Agglomerative cho kết quả khá cao đến 85%, DBSCAN kết quả thấp nhất 29.53%.

5. Yêu cầu 3 Dùng dữ liệu Face.

- Dùng bộ dữ liệu Face, áp dụng tương tự cho 4 thuật toán. Với Feature là LBP

- Các bước thực hiện:
    - Sau khi load tập dataset từ thư viện ảnh, dùng hàm LBP để rút trích đặc trưng của mỗi ảnh. Với tập dữ liệu lfw_people, số cluster là 7
    - Sau đó dùng thuật toán Kmean, DBSCAN … để gom nhóm thành các cluster rồi hiển thị.



Hình 5 So sánh các thuật toán Face dataset

- So sánh độ performance của các thuật toán.

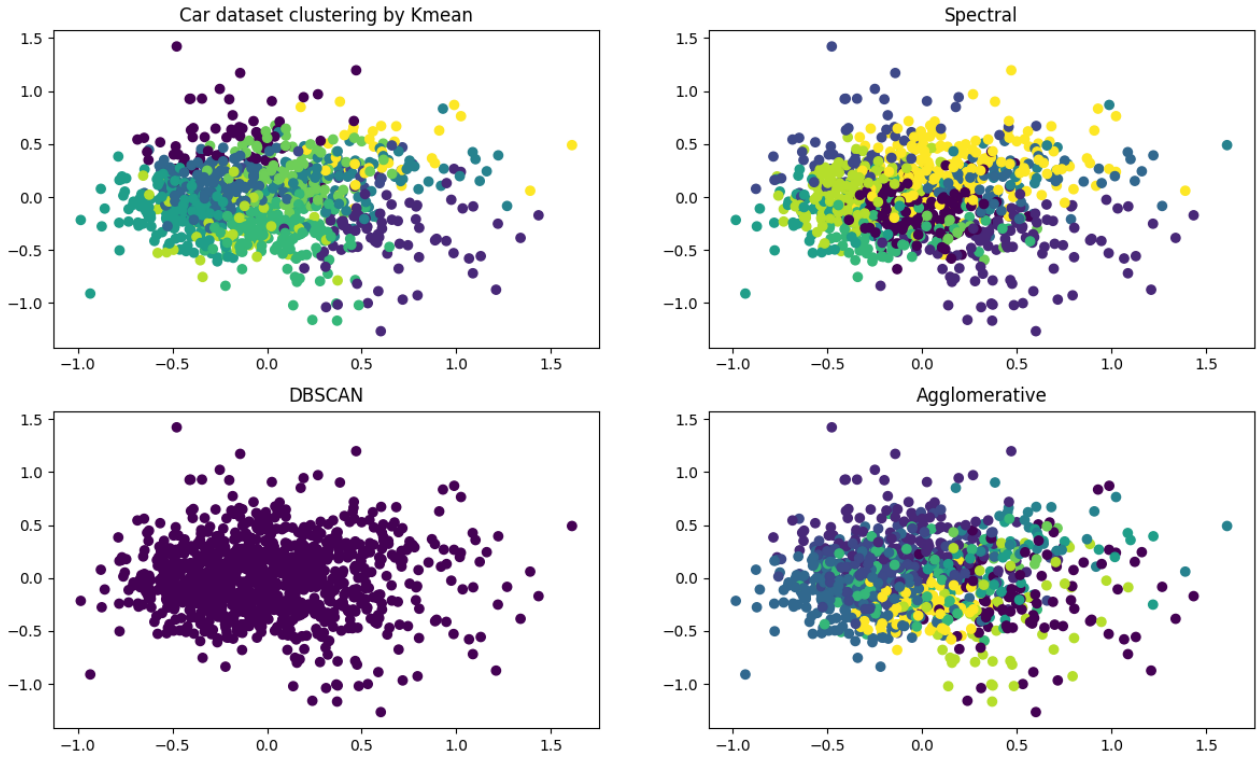
	Kmean	Spectral	DBSCAN	Agglomerative
Percentage	0.19%	0.18%	0.0%	0.17%

- Kết luận: Với tập dữ liệu Face dataset cho kết quả khá thấp chưa đến 1%.

6. Tự chọn tập dữ liệu, dùng rút trích đặc trưng tiên tiến.

- Sử dụng tập dữ liệu Car Dataset cho 1000 ảnh.
- Dùng phương pháp rút trích đặc trưng HoG (đã rút trích, được lưu thành data.npy).

- Ta sẽ load từng ảnh, rút trích đặc trưng HoG dựa trên thư viện Scikit.
- Ứng với mỗi đặc trưng ảnh xem như 1 vector. Tập hợp nhiều vector của các ảnh.
- Dùng các thuật toán clustering để gom nhóm tập các đặc trưng.
- Visualize kết quả.



Hình 6 So sánh thuật toán dựa trên Car Dataset

- So sánh độ performance các thuật toán:

	Kmean	Spectral	DBSCAN	Agglomerative
Percentage	11.19%	12.18%	0.0%	5.17%

- Kết luận đánh giá:Thông qua độ chính xác của các thuật toán trên cho thấy kết quả mang lại khá thấp, nguyên nhân có thể do quá trình điều chỉnh thông số, thuật toán chưa được tối ưu, hoặc tập dữ liệu khá phức tạp.

7. Tổng hợp các điểm mạnh yếu của các thuận toán Clustering

	Use case	Độ đo
Kmean	Áp dụng tổng quát cho các loại dữ liệu thông thường, cho flat-geometry. Số cluster ít, không áp dụng được cho dữ liệu lớn.	Tính khoảng cách các điểm
Spectral	Áp dụng cho non flat geometry. Một vài dạng cluster.	Dựa trên tính khoảng cách các điểm trên đồ thị.
DBSCAN	Áp dụng cho non flat geometry.	Khoảng cách điểm gần nhất
Agglomerative	Áp dùng nhiều cluster, không dùng khoảng cách Euclidean	Tính khoảng các giữa các cặp điểm.

8. Tham khảo

- http://scikit-learn.org/stable/modules/clustering.html.
- http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html
- http://ai.stanford.edu/~jkrause/cars/car_dataset.html
- https://gurus.pyimagesearch.com/lesson-sample-histogram-of-oriented-gradients-and-car-logo-recognition/#
- https://en.wikipedia.org/wiki/Local_binary_patterns