MACHINE LEARNING: BÀI THỰC HÀNH SỐ 4.

PHÂN TÍCH THÀNH PHẦN CHÍNH (PRINCIPAL COMPONENTS ANALYSIS - PCA)

Trong phần này chúng ta sẽ thực hành một số nội dung liên quan đến PCA và ứng dụng trong việc giảm số chiều dữ liệu. Dữ liệu sau khi được giảm số chiều sẽ được sử dụng cho một số mô hình phân loại đã học (ví dụ Logistic Regression, SoftMax).

Do phần PCA sẽ cần khá nhiều tính toán với ma trận, nên chúng ta sẽ sử dụng thư viện Sci-Kit Learn (sklearn) hoặc ít nhất là numpy để tận dụng việc tính toán kiểu vector hóa.

Ví dụ 1. Trong ví dụ này, chúng ta sử dụng PCA để phân tích và giảm số chiều của bộ dữ liệu hoa IRIS (xem lại bài thực hành phần Naïve Bayes). Dữ liệu gồm có 150 mẫu hoa phân đều trong 03 loại (class), mỗi mẫu có 04 chiều. Dữ liệu có thể lấy từ tệp Iris.csv đính kèm hoặc link: https://drive.google.com/file/d/1rb2ZQCllKCIP7Z1SZe9_T3XMKlUUfeB3/view . Thông tin thêm về dữ liệu có thể tham khảo lại trong bài hướng dẫn thực hành các phần Naïve Bayes hoặc SoftMax, hoặc trong link https://archive.ics.uci.edu/ml/datasets/iris. Do kích thước dữ liệu khá nhỏ nên trong ví dụ này chúng ta sẽ không sử dụng công cụ PCA của sklearn mà xây dựng từ các công cụ đại số tuyến tính của numpy. Ngoài ra chúng ta sẽ sử dụng công cụ PANDAS để đọc tệp CSV và một số công cụ vẽ để hiển thị kết quả (matplotlib.pyplot và seaborn).

Code minh họa trong Python:

Đoạn chương trình đọc tệp dữ liệu, truy xuất một số thông tin thống kê trên dữ liệu. Khai báo thư viện và đường dẫn đến tệp dữ liệu. Cần sửa đường dẫn này đến vị trí đặt dữ liệu cụ thể.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#Read csv data file, change to your location
df = pd.read_csv("D:\\Teach_n_Train\\Machine Learning\\code\\data\\Iris.csv")
```

Đoạn chương trình dưới đây chỉ thực hiện in thông tin hoặc hiển thị dữ liệu để chúng ta có cái nhìn tổng quan, trực giác

```
# show data information
df

df.describe()

sns.pairplot(df, hue = 'variaty')
```

Chuẩn hóa và tính ma trận hiệp phương sai của dữ liệu:

```
# We're seperating the species column
species = df["variety"].tolist()
X = df.drop("variety", 1)

# Standardize the data
X = (X - X.mean()) / X.std(ddof=0)

# Calculating the correlation matrix of the data
X_corr = (1 / 150) * X.T.dot(X)

# Plotting the correlation matrix
plt.figure(figsize=(10,10))
sns.heatmap(X_corr, vmax=1, square=True,annot=True)
plt.title('Correlation matrix')
```

Trong đoạn chương trình dưới đây, chúng ta tính hệ riêng của ma trận hiệp phương sai (X_corr) bằng cả 2 công cụ của thư viện Numpy.Linalg là SVD (khai triển kỳ dị) và Eig (Khai triển hệ giá trị-vector riêng), sau đó in ra kết quả để so sánh.

```
# method1
u,s,v = np.linalg.svd(X_corr)
eig_values, eig_vectors = s, u
eig_values, eig_vectors

# method2
np.linalg.eig(X_corr)
```

Đoạn chương trình dưới đây hiển thị mức độ "quan trọng" của các trường dữ liệu (gồm 04 trường theo thứ tự như trong mô tả dữ liệu)

```
# plotting the variance explained by each PC
explained_variance=(eig_values / np.sum(eig_values))*100
plt.figure(figsize=(8,4))
plt.bar(range(4), explained_variance, alpha=0.6)
plt.ylabel('Percentage of explained variance')
plt.xlabel('Dimensions')
```

Đoạn chương trình chứa thủ tục chiếu dữ liệu (X) xuống không gian con 02 chiều ứng với 2 giá trị riêng lớn nhất (tức là trị riêng [0] và trị riêng [1])

```
# calculating our new axis
pc1 = X.dot(eig_vectors[:,0])
pc2 = X.dot(eig_vectors[:,1])
```

Vẽ dữ liệu trong không gian 2 chiều mới thành lập để quan sát:

```
# plotting in 2D
def plot_scatter(pc1, pc2):
    fig, ax = plt.subplots(figsize=(15, 8))

    species_unique = list(set(species))
    species_colors = ["r","b","g"]

    for i, spec in enumerate(species):
        plt.scatter(pc1[i], pc2[i], label = spec, s = 20,
c=species_colors[species_unique.index(spec)])
        ax.annotate(str(i+1), (pc1[i],pc2[i]))

    from collections import OrderedDict
    handles, labels = plt.gca().get_legend_handles_labels()
    by_label = OrderedDict(zip(labels, handles))
    plt.legend(by_label.values(), by_label.keys(), prop={'size': 15}, loc=4)

    ax.set_xlabel('Principal Component 1', fontsize = 15)
    ax.set_ylabel('Principal Component 2', fontsize = 15)
    ax.axhline(y=0, color="grey", linestyle="--")
    ax.axvline(x=0, color="grey", linestyle="--")

    plt.grid()
    plt.axis([-4, 4, -3, 3])
    plt.show()

plot_scatter(pc1, pc2)
```

Chúng ta có thể sử dụng thư viện sklearn để thực hiện yêu cầu trên. Toàn bộ đoạn lệnh như sau:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#Read csv data file, change to your location
df = pd.read_csv("D:\\Teach_n_Train\\Machine Learning\\code\\data\\Iris.csv")

from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

X = df.drop("variety", 1)
X = StandardScaler().fit_transform(X)
pca = PCA()
result = pca.fit_transform(X)
# Remember what we said about the sign of eigen vectors that might change ?
pc1 = - result[:,0]
pc2 = - result[:,1]
plot_scatter(pc1, pc2)
```

Yêu cầu thực hành

1) Sử dụng đoạn code chọn số chiều chính trong ví dụ 2, đưa tập dữ liệu đã đọc về còn 2 chiều, sau đó hiển thị lên màn hình để xem quan hệ giữa các lớp dữ liệu.

2) Với đoạn chương trình đọc dữ liệu đã có, hãy chạy lại ví dụ này với các thư viện của gói linear_model, lớp LogisticRegression và so sánh kết quả, chia Train:Test = 4:1 (theo từng loại hoa để tránh phân bố các loại hoa trong tập train và tập test mất cân bằng), tương ứng với 02 trường hợp:

    a. Chạy với dữ liệu nguyên bản, lưu lại độ chính xác, ma trận nhầm lẫn trong trường hợp này;

    b. Chạy với dữ liệu giảm chiều, hãy xử lý theo 02 quy trình dưới đây:

        i. Chia dữ liệu thành Train – Test, sau đó thực hiện giảm chiều một cách phù hợp và thực hiện bài toán phân loại. Lưu lại các kết quả.

        ii. Thực hiện giảm chiều trên toàn bộ dữ liệu, sau đó chia thành dữ liệu train:test và thực hiện bài toán phân loại.

    Hãy so sánh kết quả trong 2 trường hợp i) và ii) để tìm ra quy trình phù hợp khi xử lý giảm chiều dữ liệu để phục vụ bài toán phân loại (tương tự với hồi quy). Hãy giải thích. Sau đó so sánh kết quả trong trường hợp dữ liệu nguyên bản và dữ liệu giảm chiều.

Ví dụ 2 (tự thực hành). Chúng ta sử dụng thư viện PCA để giảm số chiều của tập dữ liệu bệnh nhân Parkinson, với 754 trường (thuộc tính) và 756 records. Trường 'class' chứa hai giá trị 0 và 1 xác định bệnh nhân có mắc Parkinson hay không (1 ~ có mắc). Dữ liệu có trong tệp đính kèm (dạng CSV) hoặc tại link: https://www.kaggle.com/datasets/dipayanbiswas/parkinsons-disease-speech-signal-features .

Đoạn chương trình dưới đây đọc dữ liệu. Hãy tham khảo code ở phần trước để giảm số chiều xuống còn 02 chiều, sau đó vẽ ra màn hình để có thể quan sát:

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

```
import warnings
warnings.filterwarnings('ignore')

df= pd.read_csv("D:\\Teach_n_Train\\Machine
Learning\\code\\data\\pd_speech_features.csv")

df.head()

df['class'].value_counts()
```

Bài tập thực hành 1: Hãy sử dụng các bước và công cụ phù hợp để thực hiện được các công việc sau:

1) Sử dụng đoạn code chọn số chiều chính trong ví dụ 2, đưa tập dữ liệu đã đọc về còn 2 chiều, sau đó hiển thị lên màn hình để xem quan hệ giữa các lớp dữ liệu.
2) Giảm số chiều xuống còn 200 ứng với thành phần chính của dữ liệu bằng PCA nhằm phục vụ cho việc phân loại dữ liệu (theo class), với phần dữ liệu Train gồm 500 bản ghi và dữ liệu kiểm tra là phần còn lại. Dùng các công cụ đo độ chính xác để kiểm tra đánh giá mô hình.
3) Lấy dữ liệu gốc và chia thành tập train:test như tỷ lệ trong ý 2), sau đó giảm chiều về 200 chiều cho cả hai tập bằng PCA và thực hiện lại bài toán phân lớp dữ liệu. So sánh độ chính xác với ý 2).
4) Thực hiện các yêu cầu sau
   a. Hãy sử dụng phương phap PCA để đưa dữ liệu về số chiều nhỏ nhất sao cho lượng thông tin được giữ lại ít nhất đạt 80%.

      i. Cho biết dữ liệu mới còn bao nhiêu chiều.

      ii. Thực hiện lại bài toán phân loại với tỷ lệ như trên, ứng với dữ liệu mới và so sánh độ chính xác.
5) Sử dụng dữ liệu ban đầu, chia thành các tập Train- Test với tỷ lệ 4:2, sau đó áp dụng phương pháp Naïve Bayes phù hợp và phương pháp Hồi quy Logistic để thực hiện bài toán phân loại (theo class). Tiếp theo lại thực nghiệm các mô hình nói trên với dữ liệu đã giảm chiều ở ý b), cùng tỷ lệ chia như trên. Hãy đánh giá xem 2 mô hình này, mô hình nào có độ chính xác thay đổi nhiều hơn.

Ví dụ 3 (Bài tập thực hành 2 – Nộp trong buổi thực hành): Hãy sử dụng đoạn mã lệnh giảm số chiều bằng phương pháp phân tích thành phần chính đã học và tham khảo phần đọc, hiển thị dữ liệu hình ảnh chữ số viết tay đã có trong bài thực hành phần Multinomial Logistic Regression, để áp dụng vào dữ liệu ảnh chữ số viết tay đã được cung cấp trong bài thực hành tuần trước:

1) Đọc dữ liệu ảnh, lấy tập dữ liệu 5000 ảnh bất kỳ, giảm số chiều xuống còn 2 chiều để biểu diễn chúng trên mặt phẳng dưới dạng chấm điểm.
2) Áp dụng phương pháp phân loại nhiều lớp Multinomial Logistic Regression (tham khảo bài trước) để phân loại để phân loại tập dữ liệu đã chọn trong ý 1), tỷ lệ train:validation là 0.7:0.3. Lưu lại các độ đo tính chính xác và thời gian chạy của mô hình trong thực nghiệm này.
3) Tiến hành thực nghiệm lại với cùng tập dữ liệu nói trên và so sánh kết quả (độ chính xác và thời gian) chạy mô hình phân loại trong hai trường hợp:
   a) Sử dụng toàn dữ liệu nguyên bản (giữ nguyên 28x28 = 784 chiều), sau đó giảm số chiều xuống còn 100 chiều, chia thành 2 tập train:validation theo tỷ lệ là 0.7:0.3. Sau đó thực hiện mô hình Multinomial Logistic Regression để phân loại.

b) Chia tập 5000 ảnh nói trên thành 2 tập train:validation theo tỷ lệ là 0.7:0.3. Thực hiện giảm số chiều của mỗi tập về còn 100 chiều một cách độc lập. Sau đó thực hiện lại bước phân loại với mô hình Multinomial Logistic Regression.

So sánh kết quả, thời gian chạy và giải thích lý do tại sao có kết quả như vậy. Từ đó nêu quy trình đúng để áp dụng giảm chiều dữ liệu khi phục vụ bài toán phân loại hoặc hồi quy.

4) Hãy thực hiện lại bài toán phân loại dữ liệu ở ý 2) và 3) nhưng sử dụng mô hình Naïve Bayes phù hợp. So sánh kết quả để chỉ ra độ chính xác của mô hình nào thay đổi nhiều hơn. Giải thích.

Ví dụ 4 (Bài tập tự thực hành)

Hãy sử dụng đoạn mã lệnh giảm số chiều bằng phương pháp phân tích thành phần chính đã học (tham khảo mã lệnh áp dụng cho dữ liệu ảnh đã được gửi):

Giải nén tệp face_data.zip đính kèm sẽ có 165 ảnh các khuôn mặt của 15 người, mỗi người chụp ở 11 trạng thái khác nhau, dưới định dạng tệp png. Có một số thư viện Python hỗ trợ việc đọc ảnh và trả về mảng số ứng với các điểm ảnh (xem lại khái niệm ảnh số trong phần đọc dữ liệu chữ số viết tay). Tên ảnh được ghép bởi prefix 'subject', chỉ số của người tương ứng ghi theo kiểu 01, 02 … đến 15 và dấu chấm "." . Tiếp theo là các trạng thái, gồm 11 trạng thái

['centerlight', 'glasses', 'happy', 'leftlight', 'noglasses', 'normal', 'rightlight', 'sad', 'sleepy', 'surprised', 'wink']

Cuối cùng là phần mở rộng trong tên tệp, .png.

Đoạn chương trình dưới đây cho phép sử dụng phương thức của thư viện OpenCV (cv2) để đọc tệp ảnh và lấy ra ma trận ứng với các điểm ảnh, sau đó duỗi

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

# path to the database - change it if needed
path = 'D:\\Teach_n_Train\\Machine Learning\\code\\yale\\yalefaces_data\\'

ids = range(1, 16) # 15 people
states = ['centerlight', 'glasses', 'happy', 'leftlight',
          'noglasses', 'normal', 'rightlight','sad',
          'sleepy', 'surprised', 'wink' ]
prefix = 'subject'
surfix = '.png' #file extension is png

# open one picture to get the image's size
fn = prefix + '01.' + states[0] + surfix
im = cv2.imread(path + fn, 0)

h = im.shape[0] # hight
w = im.shape[1] # width

D = h * w
N = len(states)*15
print(N, D, h, w)

X = np.zeros((D, N))

# collect all data
count = 0

# there are 15 people
for person_id in range(1, 16):
    for state in states:
```

```
        # get name of each image file
        fn = path + prefix + str(person_id).zfill(2) + '.' + state + surfix

        # open the file and read as grey image
        tmp = cv2.imread(fn, cv2.IMREAD_GRAYSCALE)

        # then add image to dataset X
        X[:, cnt] = tmp.reshape(D)
        count += 1
```

Áp dụng cho dữ liệu ảnh khuôn mặt (của 15 người, mỗi người có 11 trạng thái):

a. Giảm số chiều dữ liệu xuống còn 500.
b. Áp dụng các phương pháp phân loại nhiều lớp: Multinomial Logistic Regression và Naïve Bayes (đã có code) để phân loại, tỷ lệ train:test là 0.7:0.3
c. Coi toàn bộ 165 ảnh đầu vào là train, tìm 5 ảnh chân dung (tùy ý) sau đó đưa về cùng kích thước như trong ảnh dữ liệu (H = 320; W = 243), với phần chân dung lệch sang tay phải theo hướng người nhìn vào. Thử dụng mô hình đã huấn luyện, chạy test xem 5 ảnh dữ liệu mới sẽ thuộc nhóm nào.