


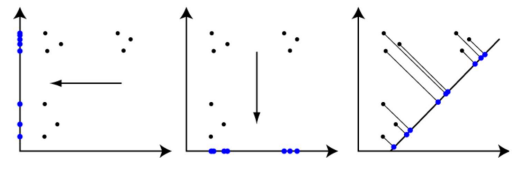
Class period 17

บทที่ 7 การแสดงผลการเปรียบเทียบข้อมูล
Visualize_Data_Distribution_(PCA)

1




Projection

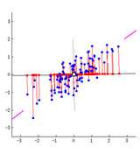


- การฉายแสงให้จุดข้อมูลให้เงาของจุดไปตกที่แกนที่กำหนด
- ลูกศรภายในกราฟคือเส้นทแยงของแสง จุดสีน้ำเงินคือข้อมูล

2




PCA (Principal component Analysis)



- PCA คือ การหาแกนใหม่ที่สามารถอธิบายการกระจายตัวของข้อมูลได้ดีที่สุด เมื่อมี ตัวแปร ที่จะนำมาแสดงการกระจายของข้อมูลมากกว่า 2 ตัวแปร สามารถใช้ PCA (Principle Component Analysis) เพื่อลดจำนวนตัวแปรลงมาได้โดยรักษาลักษณะการกระจายของข้อมูลเดิมมากที่สุด


3



sklearn -> scikit-learn

- <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
- เป็น package ที่รวบรวม function การทำ Data Science - Machine Learning - Data Mining เอาไว้ใช้งานแบบไม่ต้องเขียนเอง
- การใช้งาน import PCA จาก sklearn
- `from sklearn.decomposition import PCA`

4



การใช้ PCA มี 3 ขั้นตอน

- 1. Import
 - `from sklearn.decomposition import PCA`
- 2. Define
 - `pca = PCA()`
- 3. Fit - Transform คือ คำสั่งที่ใช้สำหรับหามาตรฐานแกนใหม่ที่สามารถอธิบายการกระจายตัวของข้อมูลได้ดีที่สุด
 - `new_axis = pca.fit_transform('ตัวแปรที่ใช้เก็บข้อมูลที่ต้องการทำPCA')`

5



เตรียมข้อมูลดอกไม้อิริส iris

- `import pandas as pd`
- `example_df = pd.read_csv('https://raw.githubusercontent.com/pandas-dev/pandas/master/pandas/tests/io/data/csv/iris.csv')`
- `thisdata = example_df.iloc[:, :-1]`
- `thisdata`

	SepalLength	SepalWidth	PetalLength	PetalWidth
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	2.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows x 4 columns

6

เริ่มทำ PCA

```
• from sklearn.decomposition import PCA
• pca = PCA()
• new_axis = pca.fit_transform(thisdata)
• new_axis.shape
• new_axis
```

6

7

ผลลัพธ์การทำ PCA

- จะได้ข้อมูลที่ถูกหมุนแกนแล้ว จำนวนข้อมูลเท่ากับข้อมูลที่ input ใช้ทำ PCA
- `new_axis.shape` จะได้ผลลัพธ์ (150, 4)
- `new_axis` จะได้ผลลัพธ์คือข้อมูลที่ถูกหมุนแกนแล้ว ในรูปแบบ numpy array

```
array([[ -2.68420713e+00,  3.26607315e-01, -2.15118379e-02,
         1.00613724e-03],
       [ -2.71533062e+00, -1.69556848e-01, -2.83521425e-01,
         9.96024240e-02],
       [ -2.88981954e+00, -1.37345610e-01,  2.47892410e-02,
         1.93845433e-02],
       [ -2.74643720e+00, -3.13124316e-01,  3.76719753e-02,
        -7.99552741e-02],
       [ -2.72895238e+00,  3.33924564e-01,  9.82296998e-02,
        -6.31287327e-02],
       [ -2.12789736e+00,  7.47782713e-01,  1.74325619e-01,
        -2.71468037e-02],
       [ -2.82089068e+00, -8.21845110e-02,  2.64251055e-01,
```

7

8

แปลงข้อมูล PCA ให้อยู่ในรูปแบบข้อมูลตาราง

- โดยใช้คำสั่ง `pd.DataFrame('ตัวแปรที่เก็บข้อมูล array PCA ', columns=['ชื่อคอลัมน์ที่ต้องการ 4 คอลัมน์'])` เช่น
- `PCAdf = pd.DataFrame(new_axis, columns = ['PCA1','PCA2','PCA3','PCA4'])`
- `PCAdf`

```
   PCA1    PCA2    PCA3    PCA4
0 -2.684207  0.266073 -0.021512  0.000613
1 -2.715331 -0.169568 -0.283521  0.099602
2 -2.889820 -0.137346  0.024789  0.019385
3 -2.746437  0.313124  0.037672 -0.079955
4 -2.728952  0.333925  0.098230 -0.063129
...
146 1.844017  0.179419  0.170030  0.402002
147 1.528844  0.379402  0.102046  0.202751
148 1.746046  0.070519  0.140704  0.106305
149 1.831829  0.110877  0.123074  0.040073
150 1.809460 -0.248387  0.262191 -0.100119
151 rows × 4 columns
```

8

9

pca.explained_variance_ratio_

- ใช้ดูประสิทธิภาพของการกระจายข้อมูล ตามจำนวนแกน เช่น
- `array([0.92461621, 0.05301557, 0.01718514, 0.00518309])`
- 0.92461621 คือ แกนแรก 1 แกนสามารถอธิบายการกระจายข้อมูลได้ 92.4%
- 0.05301557 คือ แกนแรก 2 แกนสามารถอธิบายการกระจายข้อมูลได้ $92.4 + 5.3 = 97.7\%$

9

10

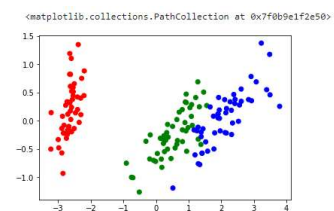
plot PCA data

- นำตารางข้อมูล PCA มาสร้างกราฟที่สามารถอธิบายการกระจายตัวของข้อมูลได้ดีที่สุด
- `from matplotlib import pyplot as plt`
- `example_df2 = example_df.replace({'Iris-setosa':'r', 'Iris-versicolor':'g', 'Iris-virginica':'b'})`
- `plt.scatter(PCAdf['PCA1'], PCAdf['PCA2'], c=example_df2['Name'])`
- `plt.scatter(example_df2['SepalWidth'], example_df2['PetalWidth'], c=example_df2['Name'])`

10

11

ผลลัพธ์จะได้ scatter plot ของข้อมูล PCA



11

12

