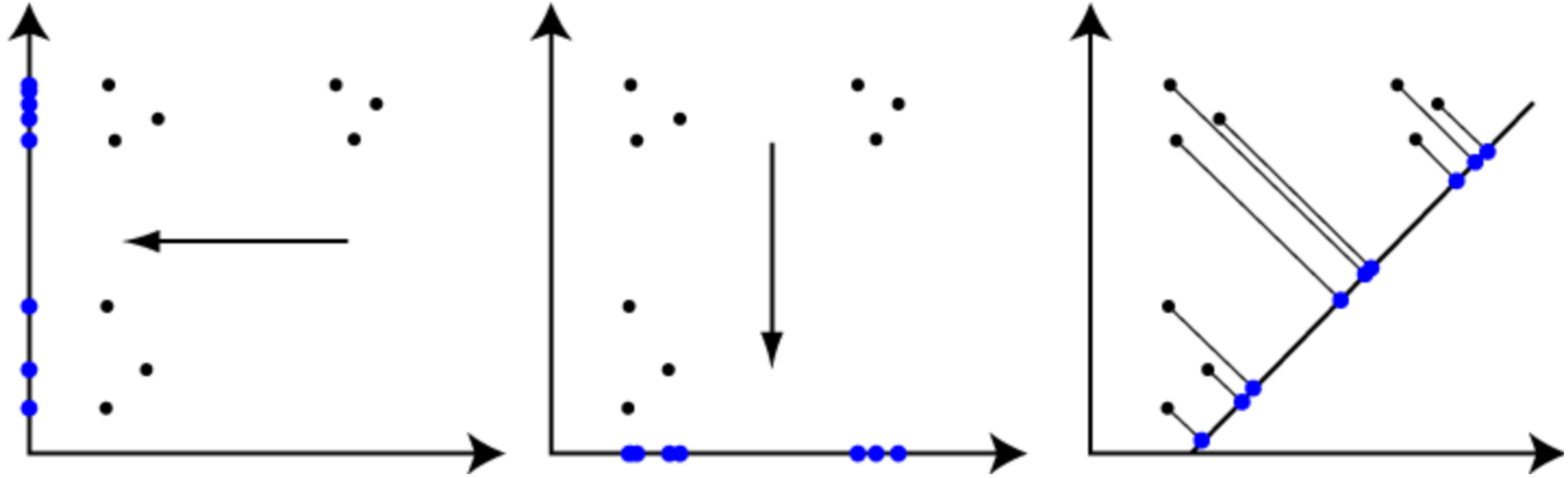
Class period 17

บทที่ 7 การแสดงผลการเปรียบเทียบข้อมูล

Visualize_Data_Distribution_(PCA)

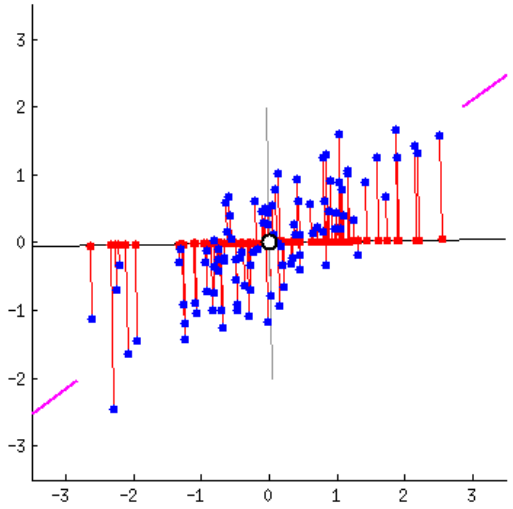
โดย ผศ. ดร.ธนพงศ์ อินทระ สาขาวิชาสถิติ มหาวิทยาลัยขอนแก่น

Projection



- การฉายแสงใส่จุดข้อมูลให้เงาของจุดไปตกที่แกนที่กำหนด
- ลูกศรภายในกราฟคือเส้นทางของแสง จุดสีน้ำเงินคือข้อมูล

PCA (Principal component Analysis)



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

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 ของ sklearn
- `from sklearn.decomposition import PCA`

การใช้ PCA มี 3 ขั้นตอน

- 1. Import
- `from sklearn.decomposition import PCA`

- 2. Define
- `pca = PCA()`

- 3. Fit – Transform คือ คำสั่งที่ใช้สำหรับหมุนแกนหาแกนใหม่ที่สามารถอธิบายการกระจายตัวของข้อมูลได้ดีที่สุด
- `new_axis = pca.fit_transform('ตัวแปรที่ใช้เก็บข้อมูลที่ต้องการทำPCA')`

เตรียมข้อมูลดอกไม้ iris

- `import pandas as pd`
- `example_df = pd.read_csv('https://raw.github.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	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

เริ่มทำ PCA

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

ผลลัพธ์การทำ PCA

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

```
array([[-2.68420713e+00,  3.26607315e-01, -2.15118370e-02,
         1.00615724e-03],
       [-2.71539062e+00, -1.69556848e-01, -2.03521425e-01,
         9.96024240e-02],
       [-2.88981954e+00, -1.37345610e-01,  2.47092410e-02,
         1.93045428e-02],
       [-2.74643720e+00, -3.11124316e-01,  3.76719753e-02,
        -7.59552741e-02],
       [-2.72859298e+00,  3.33924564e-01,  9.62296998e-02,
        -6.31287327e-02],
       [-2.27989736e+00,  7.47782713e-01,  1.74325619e-01,
        -2.71468037e-02],
       [-2.82089068e+00, -8.21045110e-02,  2.64251085e-01,
```

แปลงข้อมูล PCA ให้อยู่ในรูปแบบข้อมูลตาราง

- โดยจะใช้คำสั่ง pa.DataFrame(‘ตัวแปรที่ใช้เก็บข้อมูล 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	0.326607	-0.021512	0.001006
1	-2.715391	-0.169557	-0.203521	0.099602
2	-2.889820	-0.137346	0.024709	0.019305
3	-2.746437	-0.311124	0.037672	-0.075955
4	-2.728593	0.333925	0.096230	-0.063129
...	...	...	...	...
145	1.944017	0.187415	0.179303	0.425082
146	1.525664	-0.375021	-0.120636	0.255723
147	1.764046	0.078519	0.130784	0.136295
148	1.901629	0.115877	0.722874	0.040873
149	1.389666	-0.282887	0.362318	-0.156310

150 rows × 4 columns

pca.explained_variance_ratio_

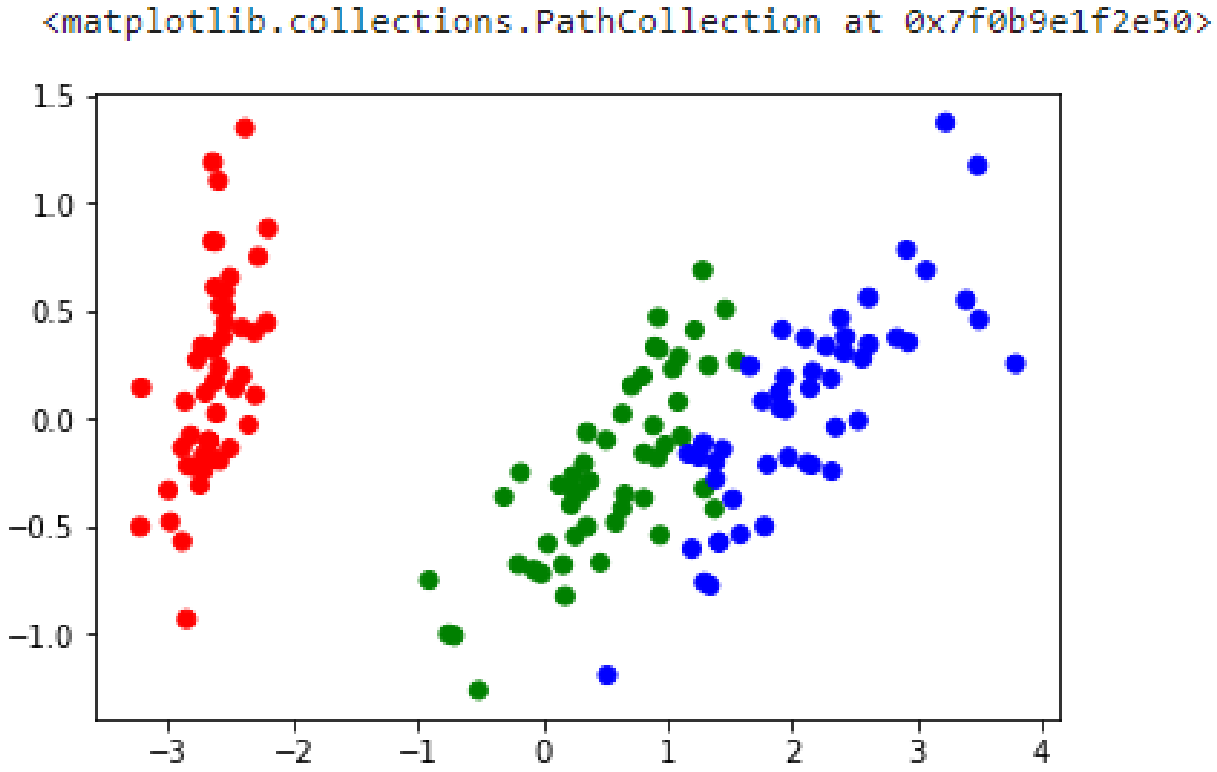
- ใช้ดูประสิทธิภาพของการกระจายข้อมูล ตามจำนวนแกน เช่น
- `array([0.92461621, 0.05301557, 0.01718514, 0.00518309])`

- `0.92461621` คือ ใช้แกน 1 แกนสามารถอธิบายการกระจายข้อมูลได้ `92.4%`
- `0.05301557` คือ ใช้แกน 2 แกนสามารถอธิบายการกระจายข้อมูลได้ `92.4+5.3 = 97.7%`

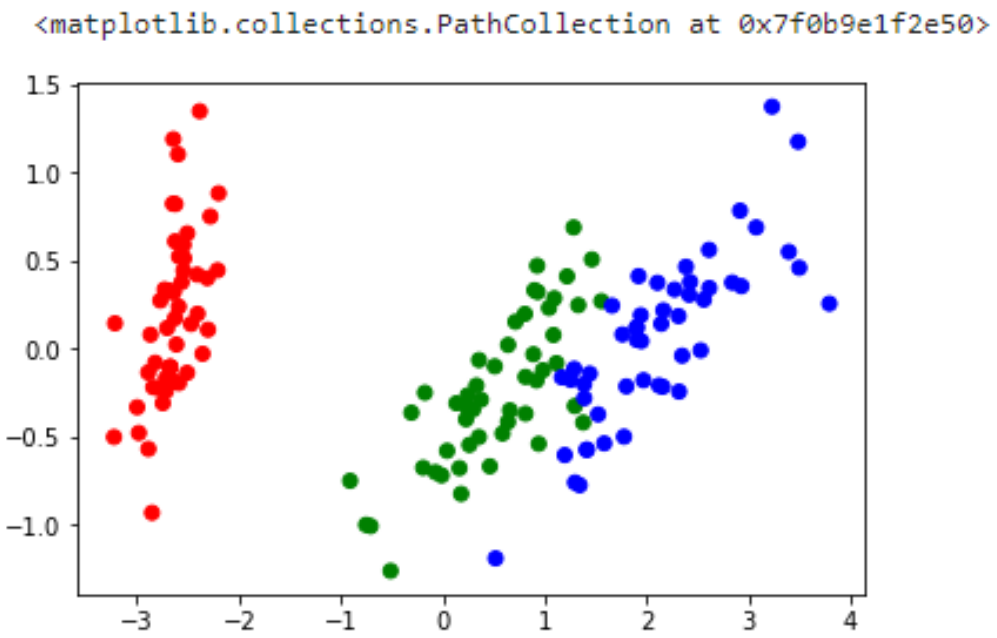
plot PCA data

- 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'])`

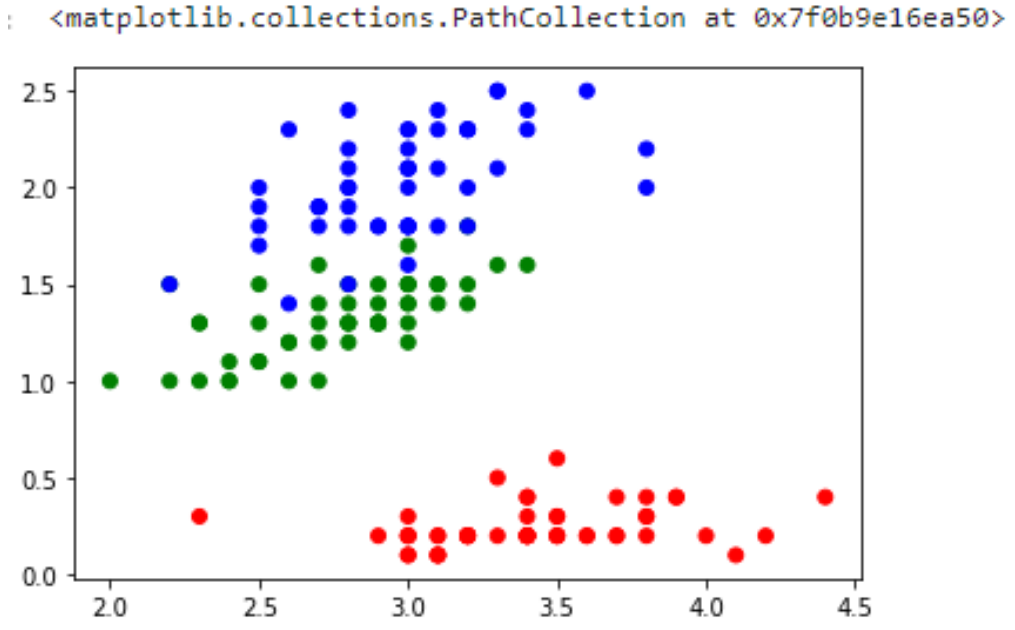
ผลลัพธ์จะได้ scatter plot ของข้อมูล PCA



เปรียบเทียบข้อมูลที่ทำ PCA แล้วกับข้อมูลก่อนทำ



กราฟของข้อมูลที่ทำ PCA แล้ว



กราฟของข้อมูลก่อนทำ