

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
วิชา Machine Learning Laboratory

**การทดลองที่ 1 : พื้นฐานการอ่านไฟล์ข้อมูล การแก้ปัญหาค่าข้อมูลหาย การปรับช่วงค่าของข้อมูล การ
ปรับลดมิติข้อมูล และ แสดงผลข้อมูลในเชิงกราฟ**

วัตถุประสงค์

1. เพื่อศึกษาและทดลองการอ่านไฟล์ข้อมูล (.csv)
2. เพื่อศึกษาและทดลองการแก้ปัญหาค่าข้อมูลหาย
3. เพื่อศึกษาและทดลองการปรับช่วงค่าของข้อมูล
4. เพื่อศึกษาและทดลองการปรับลดมิติข้อมูล
5. เพื่อศึกษาและทดลองการแสดงผลข้อมูลเชิงกราฟ

อุปกรณ์ และเครื่องมือที่ใช้ในการทดลอง

1. โปรแกรม python

ข้อกำหนดในการตรวจการทดลอง

1. แสดงโค้ดและภาพผลการทดลองที่ทำพร้อมอธิบาย
2. นศ.ที่ได้รับการตรวจจากอาจารย์เรียบร้อยแล้ว อาจารย์จะเช็คส่งงานในระบบ
3. ให้นศ. นำ source code และ ภาพ figure ที่ให้แสดงทุกภาพ โฟสลงใน google classroom ส่งภายในวันที่ 6 กพ. 2562 เวลา 17.00 น.

ตอนที่ 1: การทดลองอ่านไฟล์ข้อมูล (.csv) การแก้ปัญห ข้อมูลหาย และการปรับช่วงค่าของข้อมูล และแสดงข้อมูลเชิงกราฟ

1.1 Import Lib (numpy, panda, matplotlib, seaborn)

1.2 ฟังก์ชันใช้งาน

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns; sns.set()
```

1.3 อ่านไฟล์ “human_sensor_data.csv” เข้ามาเป็น dataframe ด้วย pd.read_csv()

1.4 ปรับรูปแบบของข้อมูล

1.4.1 ลบ column เวลา ‘uts’ ด้วย pd.drop()

1.4.2 แทนค่า Missing (NA) และ 0 ด้วยค่า mean ด้วย pd.fillna() และ pd.replace()

1.4.3 ลบข้อมูลชุดที่มีความซ้ำซ้อน ด้วย pd.drop_duplicates()

1.5 ปรับช่วงค่าข้อมูลโดยการใช้ข้อมูลสถิติ (zero mean / unit variance) (แสดงการคำนวณในโปรแกรม ไม่ใช่ฟังก์ชันสำเร็จ)

```
Xmean = pd.mean()
Xstd = pd.std()
Xnorm = (X - Xmean)/Xstd
```

1.6 Figure 1.1 แบ่ง subplot() โดยใช้ layout=(2, 4) พร้อมแสดงกราฟเส้นตรงของค่า features (variables) ของ Xnorm ที่ได้จากข้อ 1.5 ด้วยฟังก์ชัน plot()

1.7 Figure 1.2 แบ่ง subplot() โดยใช้ layout=(2, 4) พร้อมแสดงกราฟเส้นตรงของค่า features (variables) ของ Xnorm ที่ได้จากข้อ 1.5 ด้วยฟังก์ชัน hist()

1.8 Figure 1.3 แสดงกราฟ **Scatter Pair Plot** เพื่อดูความสัมพันธ์ของคู่ข้อมูลในชุดข้อมูลทั้งหมดของ Xnorm ได้จากข้อ 1.5 โดยใช้ฟังก์ชัน sns.pairplot()

1.9 Figure 1.4 แบ่ง subplot() (1 row, 2 columns) แสดงกราฟ **3D Scatter Plot** ของ Xnorm ได้จากข้อ 1.5

ตำแหน่ง (1,2,1) แสดง Xnorm เฉพาะ ['accelerate.x', 'accelerate.y', 'accelerate.z']

```
ตย.    x = Xnorm['accelerate.x']
        y = Xnorm['accelerate.y']
        z = Xnorm['accelerate.z']
```

ตำแหน่ง (1,2,2) แสดง Xnorm เฉพาะ ['gyro.x', 'gyro.y', 'gyro.z']

ปรับ (ความกว้าง x ความสูง: w,h) ของเฟรมภาพ

ปรับสีของจุดที่ c / ปรับขนาดของจุดที่ s / ปรับสีของเส้นขอบจุดที่ edgecolor
ปรับ 3d view point ของกราฟ ที่ view_init(elev, azim)

‘elev’ stores the elevation angle in the z plane.

‘azim’ stores the azimuth angle in the x,y plane.

```
fig = plt.figure(figsize=(w,h))
#=====
# First subplot
#=====
# set up the axes for the first plot
ax = fig.add_subplot(1, 2, 1, projection='3d')

ax.scatter(x,y,z, c='cyan',s=20,edgecolor='k')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')

# optional view configurations
ax.view_init(30, -60)

#=====
# Second subplot
#=====
# set up the axes for the second plot
ax = fig.add_subplot(1, 2, 2, projection='3d')

ax.scatter((x,y,z, c='violet',s=20,edgecolor='k')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')

# optional view configurations
ax.view_init(30, -10)

plt.show()
```

1.10 Figure 1.5 แสดง Heatmap การกระจายของคู่ข้อมูล (joint distribution) ['accelerate.x', 'accelerate.y', 'accelerate.z'] ที่ละคู่ โดยใช้คำสั่ง

```
sns.jointplot(x="x", y="y", data=dataframe, kind="kde", color="g");
```

(อาจารย์ตรวจผลการทดลอง)

ตอนที่ 2: การทดลองการลดมิติของข้อมูลด้วยค่า Principle Component Analysis

2.1 คำนวณค่า covariance matrix ของชุดข้อมูลจาก

```
covM = x.T.dot(x) / (x.shape[0]-1)
```

2.2 คำนวณค่า eigenvalue / eigenvector จาก covariance matrix ที่คำนวณได้จากข้อ 2.1

```
eig_vals, eig_vecs = np.linalg.eig(covM)
```

```
print('\nEigenvalues \n%s' %eig_vals)
```

```
print('Eigenvectors \n%s' %eig_vecs)
```

2.2 Figure 2.1 แสดง bar plot ของค่า eigenvalue ที่คำนวณได้จากข้อ 2.2

```
y_pos = np.arange(len(eig_vals))
```

```
plt.bar(y_pos, eig_vals, color=['cyan','darkviolet','yellow'])
```

2.3 Figure 2.2 3D point plot และ eigenvector

```
# Split each eigenvector and scale with its sqrt(eigenvalue)
ev1 = eig_vecs[:,0]*np.sqrt(eig_vals[0])
ev2 = eig_vecs[:,1]*np.sqrt(eig_vals[1])
ev3 = eig_vecs[:,2]*np.sqrt(eig_vals[2])

#####
#plotting eigenvectors
#####

fig = plt.figure(figsize=(20,8))

# 3D eigenvector view
ax = fig.add_subplot(141, projection='3d')

ax.plot(Xnorm['accelerate.x'], Xnorm['accelerate.y'], Xnorm['accelerate.z'], 'o', markersize=10, color='green', alpha=0.2)
ax.plot([mean_x], [mean_y], [mean_z], 'o', markersize=10, color='red', alpha=0.5)

ax.plot([0, ev1[0]], [0, ev1[1]], [0, ev1[2]], color='red', alpha=0.8, lw=2)
ax.plot([0, ev2[0]], [0, ev2[1]], [0, ev2[2]], color='violet', alpha=0.8, lw=2)
ax.plot([0, ev3[0]], [0, ev3[1]], [0, ev3[2]], color='cyan', alpha=0.8, lw=2)

ax.set_xlabel('x_values')
ax.set_ylabel('y_values')
ax.set_zlabel('z_values')

plt.title('Eigenvectors')

# optional view configurations (x,y,z)=(10,60); (x,z)=(0,90); (x,y)=(90,90); (y,z)=(180,180)
ax.view_init(10,60)

plt.show()
```

ให้แบ่ง subplot layout (1,4) โดย

subplot(141) แสดง

1) 3d point plot ของค่า ['accelerate.x', 'accelerate.y', 'accelerate.z']

2) 3d vector plot ของ eigenvector ตัวแรก หรือ principle component แรก

3) กำหนด view initial ไว้ที่ view_init(elev, azim) -> (10, 60) เพื่อดู 3D eigenvector

Subplot(142) แสดง เช่นเดียวกับ (141) แต่ปรับ

view_init(elev, azim) -> (0, 90) เพื่อ 3D onto 2D x-z axis

Subplot(143) แสดง เช่นเดียวกับ (141) แต่ปรับ

view_init(elev, azim) -> (90, 90) เพื่อ 3D onto 2D x-y axis

Subplot(144) แสดง เช่นเดียวกับ (141) แต่ปรับ

view_init(elev, azim) -> (180, 180) เพื่อ 3D onto 2D y-z axis

2.3 ลดมิติของข้อมูลจาก 3D ลงเหลือ 2D โดย เลือก eigenvector 2 vector แรก ที่สัมพันธ์กับ eigenvalue ที่มีค่าสูงสุด 2 อันดับแรก

project Xnorm onto new principle components (eigenvectors)

$X_new_pca1 = Xnorm.T.dot(eig_vecs[:,0])$

$X_new_pca2 = Xnorm.T.dot(eig_vecs[:,1])$

2.4 Scatter plot (X_new_pca1, X_new_pca2)

(อาจารย์ตรวจผลการทดลอง)

Tutorial

[Panda \(Reading .csv / Drop Duplicate\)](#)

[1] <https://www.ritchieng.com/pandas-removing-duplicate-rows/>

[2] <http://www.datasciencemadesimple.com/delete-drop-duplicate-row-dataframe-python-pandas/>

[Panda \(Fill missing value\)](#)

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html>

[Panda \(Replace with specific value\)](#)

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.replace.html>

[Panda \(drop row or column\)](#)

https://chrisalbon.com/python/data_wrangling/pandas_dropping_column_and_rows/

[PCA](#)

[Inside calculation example](#)

<https://plot.ly/ipython-notebooks/principal-component-analysis/>

[Using built in function](#)

<https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html>

Visualization in Graph Plot

[1] https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

[2] https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html#visualization-hist

[3] 3D Subplot: <https://matplotlib.org/gallery/mplot3d/subplot3d.html>

[4] Jointdistribution: <https://seaborn.pydata.org/tutorial/distributions.html>

Color code

https://matplotlib.org/examples/color/named_colors.html

https://seaborn.pydata.org/tutorial/color_palettes.html#setting-the-default-color-palette