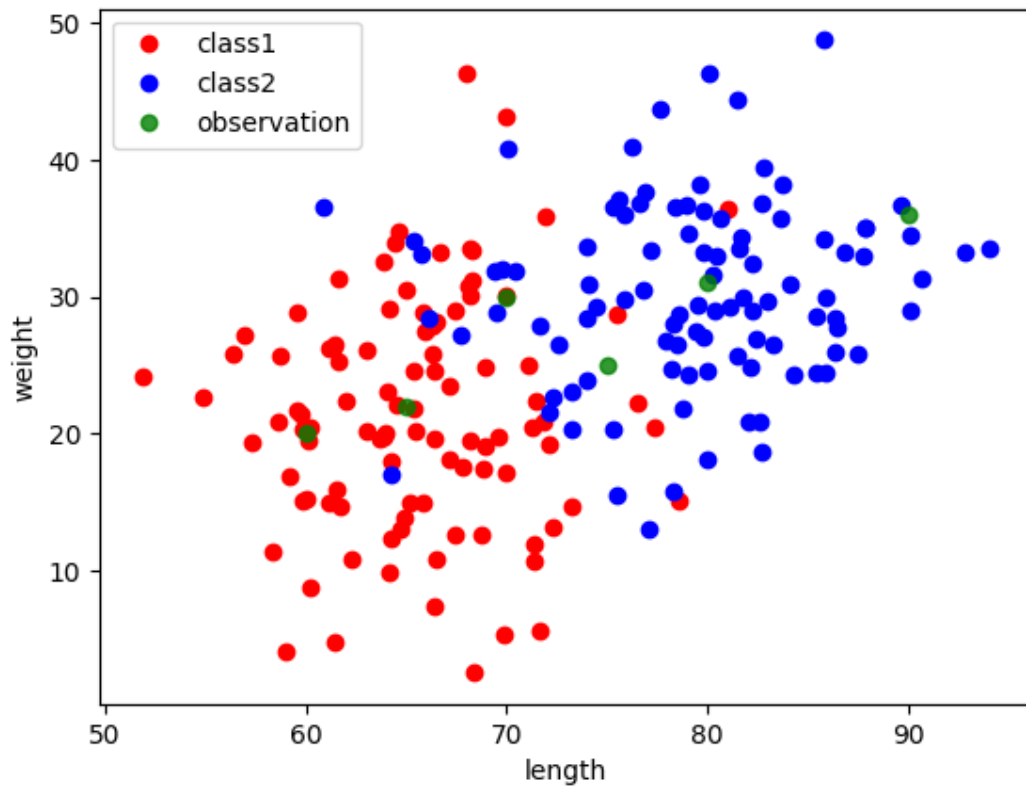


KECE407: Pattern Recognition

School of Electrical Engineering, KOREA UNIVERSITY(Homework #1)

2016170994 김다민

1. Data visualization



2. Data statistics

mean of class1

```
[65.55045436 21.39741652]
```

covariance of class1

```
[[27.02342512  3.56713765]
```

```
[ 3.56713765 70.77560176]]
```

mean of class2

```
[79.33297476 30.23232716]
```

covariance of class2

```
[[41.42775115  5.01652587]
```

```
[ 5.01652587 44.89646531]]
```

3. Data prediction

The likelihood calculations were performed Using multivariable normal distribution. It was assumed that priors of two class were the same.

a)

likelihood using two features

```
[[3.61921681e-03 1.93322815e-04]
 [6.72669372e-04 2.30652645e-03]
 [5.03300753e-05 3.67604637e-03]
 [2.28731204e-08 7.50553466e-04]
 [2.05844000e-03 2.02111574e-05]
 [1.60302040e-03 1.28679336e-03]]
```

b)

prediction using two features

```
[1 2 2 2 1 1]
```

c)

likelihood using only length

```
[[7.63141562e-02 5.19353358e-03]
 [1.47067582e-02 4.94145999e-02]
 [1.61179189e-03 6.16498870e-02]
 [1.20660513e-06 1.56981311e-02]
 [4.33995546e-02 6.81000774e-04]
 [5.32045777e-02 2.16620142e-02]]
```

prediction using only length

```
[1 2 2 2 1 1]
```

d)

likelihood using only weight

[[0.04729925 0.02799112]

[0.04326617 0.04389253]

[0.02472061 0.05914985]

[0.01051329 0.04110633]

[0.04677103 0.0185527]

[0.02811348 0.05950356]]

prediction using only weight

[1 2 2 2 1 2]

e)

두개의 피쳐중 더 나은 것은 length 라고 생각한다. 왜냐하면 위에 시각화 그림과 통계치에서 볼 수 있듯이, 두클래스의 length 평균 차이가 14.8 weight 의 평균 차이 9.7 보다 크며, 분산 또한 Length 27.02, 41.43 가 weight 가 70.78 44.90 으로 더 크다, 이는 하나의 변수를 갖는 normal distribution 으로 바꾸게 되면 length 의 경우 두 그래프의 중심간 거리가 더 멀고 분산이 적어 확률분포가 중심에 밀집해 있다. 반면 weight 의 경우 두확률분포 간 거리가 더 가깝고 분산 또한 커서 확률분포가 퍼져있으며 이는 확률이 비슷한 공통영역이 넓어지는 것을 의미하고, 해당 영역에서 분류에 어려움이 생긴다.

추가적으로 이번 문제에 의해 위에서 나온 분류 결과를 보면,

두개 모두사용:[1 2 2 2 1 1]

Length 만 사용:[1 2 2 2 1 1]

Weight 만 사용:[1 2 2 2 1 2]

두개를 사용했을 때와 length 만 이용했을 때의 결과가 더 비슷하다는 것을 알 수 있다. 따라서 둘 중 하나의 피쳐만 사용해야한다면 length 를 이용하는 것이 적절해 보인다.

Source code

```
from scipy import io
from scipy.stats import multivariate_normal
import matplotlib.pyplot as plt
import numpy as np

# problem1 Data visualization
class1=io.loadmat('data/class1.mat')['R']
class2=io.loadmat('data/class2.mat')['R']
observation=io.loadmat('data/observation.mat')['obs']

plt.plot(class1[:,0],class1[:,1],"o",color="red",label="class1")
plt.plot(class2[:,0],class2[:,1],"o",color="blue", label="class2")
plt.plot(observation[:,0],observation[:,1],"o",color="green", label="observation",
alpha=0.8)
plt.xlabel("length")
plt.ylabel("weight")
plt.legend()
plt.show()

# problem2 Data statistics
mean1=np.mean(class1,axis=0)
cov1=np.cov(class1.T)
mean2=np.mean(class2, axis=0)
cov2=np.cov(class2.T)
print('mean of class1\n',mean1)
print('covariance of class1\n',cov1,'\n')
print('mean of class2\n',mean2)
print('covariance of class2\n',cov2)

# problem3 Data prediction
def get_prediction(p1, p2, obs_data):
    predict=[]
    likelihood=[]
    for fish in obs_data:
        l1,l2=p1.pdf(fish), p2.pdf(fish)
        if l1 > l2:
            predict.append(1)
        else:
            predict.append(2)
        likelihood.append([l1,l2])
    return np.array(predict),np.array(likelihood)

p1=multivariate_normal(mean=mean1,cov=cov1)
p2=multivariate_normal(mean=mean2,cov=cov2)
multi_predict, multi_likelihood=get_prediction(p1,p2,observation)
print('\n\nlikelihood using two features\n',multi_likelihood)
print('prediction using two features\n',multi_predict)

p1=multivariate_normal(mean=mean1[0],cov=cov1[0][0])
p2=multivariate_normal(mean=mean2[0],cov=cov2[0][0])
length_predict, length_likelihood=get_prediction(p1,p2,observation[:,0])
```

```
print('\n\nlikelihood using only length\n',length_likelihood)
print('prediction using only length\n',length_predict)

p1=multivariate_normal(mean=mean1[1],cov=cov1[1][1])
p2=multivariate_normal(mean=mean2[1],cov=cov2[1][1])
weight_predict, weight_likelihood=get_prediction(p1,p2,observation[:,1])
print('\n\nlikelihood using only weight\n',weight_likelihood)
print('prediction using only weight\n',weight_predict)
```