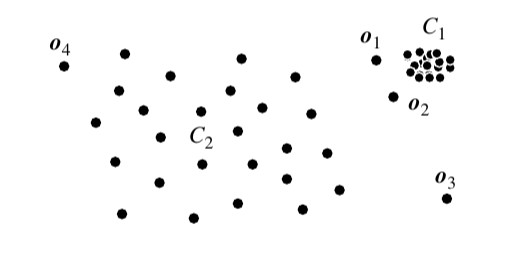
HW #3 Due: 4/11/2022

資工碩一 呂昀星 110598056

1. If we know the distributions of the samples are given below. Suppose that  and  are cluster centers with known respective covariance values (estimated from neighboring points on the plot). To detect outliers  and , of the Euclidean distance and the Mahalanobis distance, which one is better? Why?

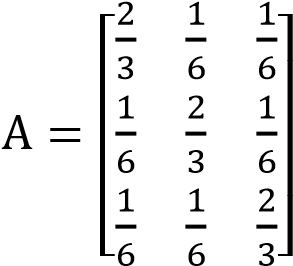
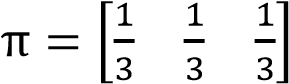


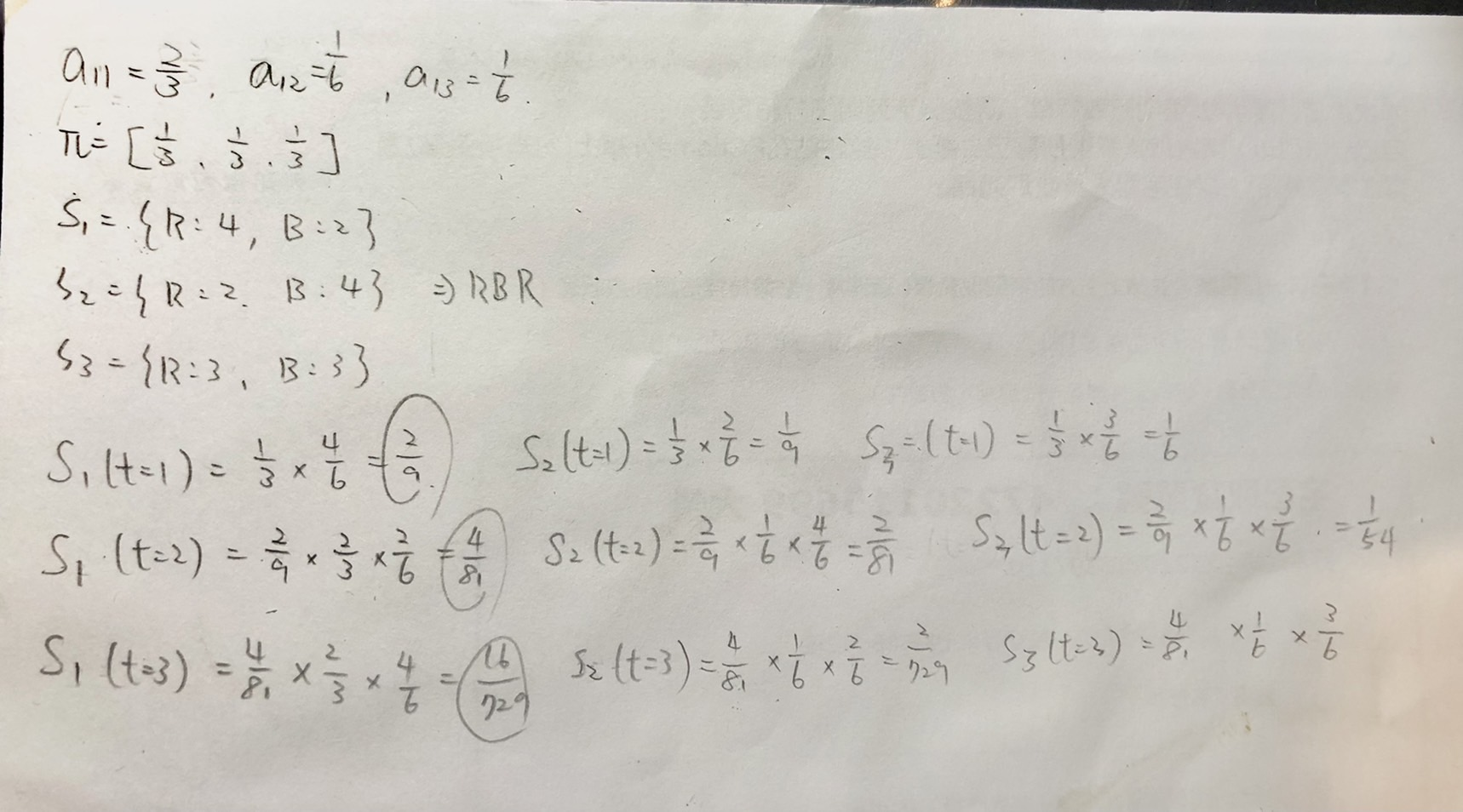
C1 與C2兩群的分散程度不一樣，Mahalanobis distance會考慮到特徵之間的

相關性，顧使用Mahalanobis distance會較合適。

1. Follow the numerical example in GMM and complete the computation of  in one step.

1. Use the three-urn example (on pp. 13 of the PPT file) to find the optimal state sequence (decoding problem) with the associated path probability if the observation sequence is “RBR.” Use the number of red and blue balls in each urn to compute the emission probability for each state. The transition probability

 and initial state distribution .



1. Repeat problem 5 of HW 2, but use GMM with 2 mixtures instead. The GMM tools are supported in sklearn. Remember to use one model per class. Of the k-NN, Naïve Bayes, and GMM classifiers, which one has the best accuracy?

GMM:0.355 ,Knn is better (0.97)

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

iris = datasets.load\_iris()

#print(iris.feature\_names)

#print(iris.target\_names)

data\_x=iris.data

data\_y=iris.target

x\_train,x\_test,y\_train,y\_test=train\_test\_split(data\_x,data\_y,test\_size=0.3)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x\_train = sc.fit\_transform(x\_train)

x\_test = sc.transform(x\_test)

from sklearn import mixture

gmm =mixture.GaussianMixture()

gmm.fit(x\_train, y\_train)

y\_pred = gmm.predict(x\_test)

from sklearn.metrics import accuracy\_score

print ("Accuracy : ", accuracy\_score(y\_test, y\_pred))

1. In this problem, you are asked to perform the wrapper-type feature selection using the Naïve Bayes classifier for cancer dataset (Breast Cancer Wisconsin (Original) Data Set, directly from the sklearn or downloading from [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original% 29)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29). To simplify the problem, we just want to keep 5 attributes out of 9 (hint: one attribute is useless. Which one is it?) To begin one experiment, randomly draw 60 % of the instances from each class for training, and 20% from each class for finding the best 5 attributes. Once the feature selection is complete, use the rest 20% for testing to obtain the accuracy. Repeat the selection 10 times to get the average accuracy. You need to deal with **missing attributes.** Compare the obtained accuracy with the same type of model trained with the full set of 9 features.

Accuracy : 0.9342105263157896

1. from sklearn.datasets import load\_breast\_cancer
2. from sklearn.model\_selection import train\_test\_split
3. from sklearn.feature\_selection import SelectKBest,f\_classif
4. from sklearn.metrics import accuracy\_score
5. data = load\_breast\_cancer()
6. data\_x=data.data
7. data\_y=data.target
8. train\_ratio = 0.6
9. validation\_ratio = 0.2
10. test\_ratio = 0.2
11. #x\_train x\_text x\_val
12. x\_train,x\_test,y\_train,y\_test=train\_test\_split(data\_x,data\_y,test\_size=1-train\_ratio)
13. x\_train,x\_val,y\_train,y\_val=train\_test\_split(data\_x,data\_y,test\_size=test\_ratio/(test\_ratio+validation\_ratio))
14. #feature selection
15. select=SelectKBest(f\_classif,k=5)
16. select.fit(x\_val,y\_val)
17. from sklearn.naive\_bayes import GaussianNB
18. classifier = GaussianNB()
19. classifier.fit(x\_val, y\_val)
20. dataframe=select.get\_support(True)
21. x\_train=x\_train[:,[dataframe[0],dataframe[1],dataframe[3],dataframe[4]]]
22. #avg accuracy for 10 times
23. count=0
24. for i in range(10):
25. y\_pred = classifier.predict(x\_test)
26. count+=accuracy\_score(y\_test, y\_pred)
28. count=count/10
29. print ("Accuracy : ", count)