HW2: Classification

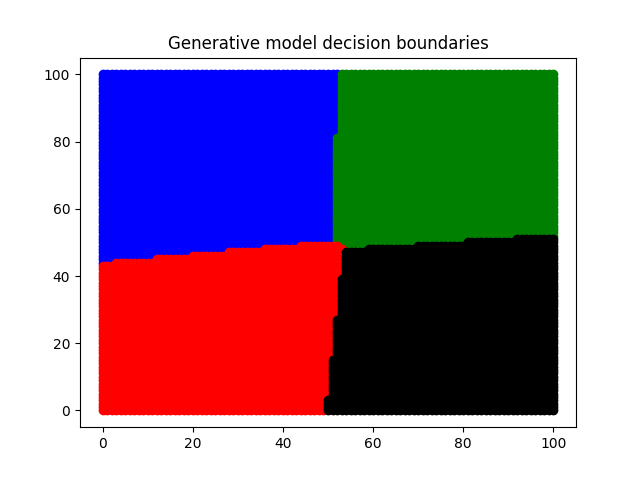
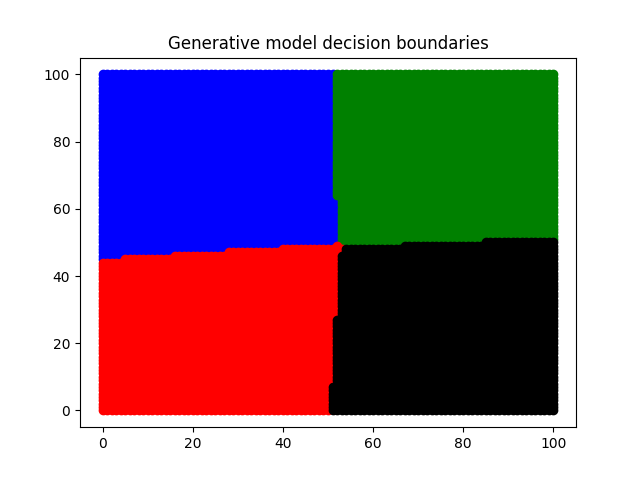
109511219 林錦樑

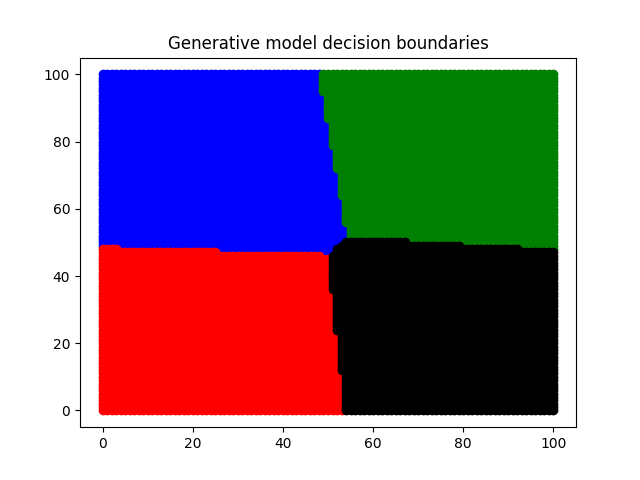
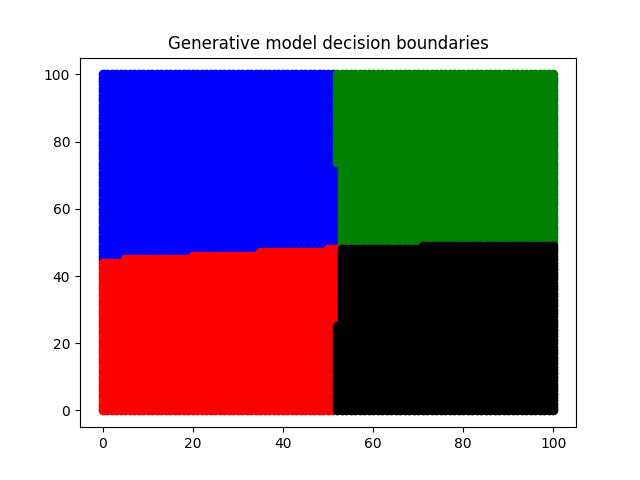
**Part I. 4 classes**

1. Generative model

在Generative model，我們假設不同class的data都來自高斯分布，使用不同的mean vector，covariance matrix則假設為相同的。一開始我使用的covariance matrix來自class 1，結果如下面左上方圖片，可大致分出四種類別。

右上方covariance matrix來自class 2，左下方covariance matrix來自class 3，右下方covariance matrix來自class 4。四種不同的covariance matrix在Generative model下皆能分出四種類別，僅邊界部分有些微差異。前三張圖class 3的區域看起來較大，較符合給定的data分布。class 4的covariance matrix則會讓class 1區域較大。



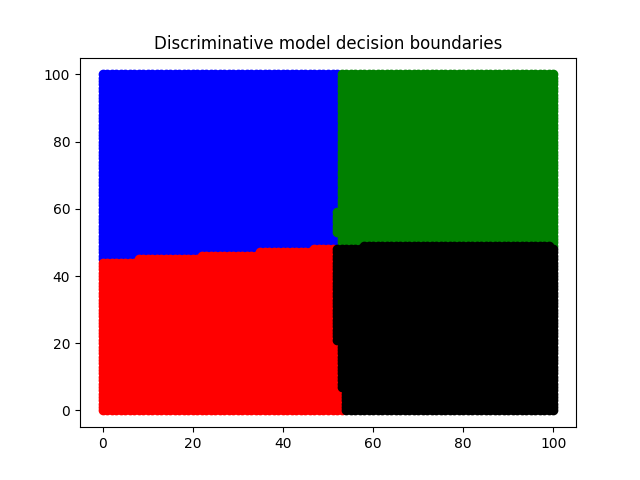
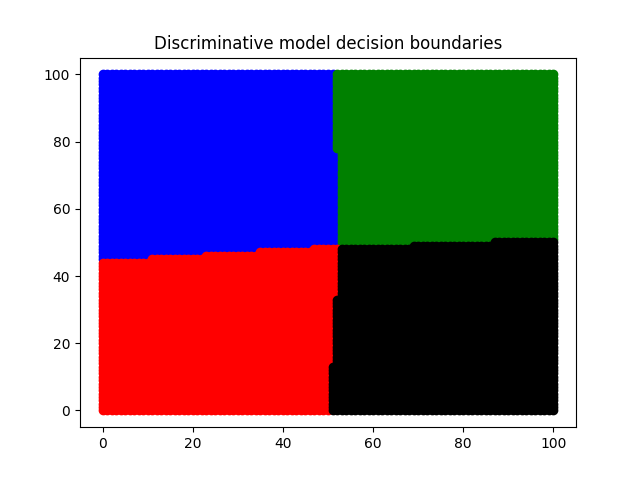


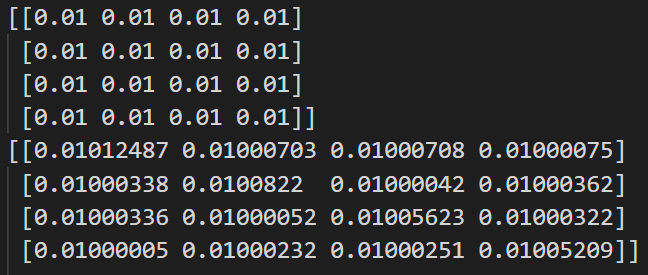
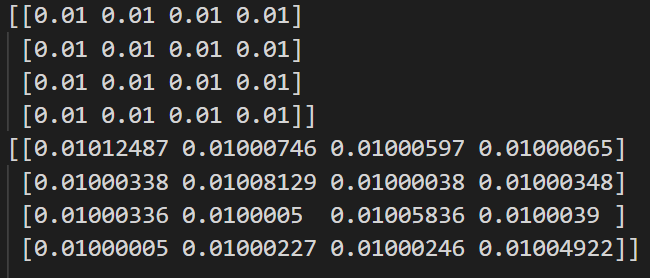
1. Discriminative model

在Discriminative model，我們沒有假設資料是何種分布，會以basis function將data轉換成另一種分布，再以gradient decent找出適當的weights。

我的basis function如同generative的方法一樣，是四個Gaussian distribution。一開始我先利用四個class各自的mean vector與covariance matrix生出四個basis function，其結果與weights如下方左圖。之後我將全部的covariance matrix改為class 1的covariance matrix，其結果與weights如下方右圖。

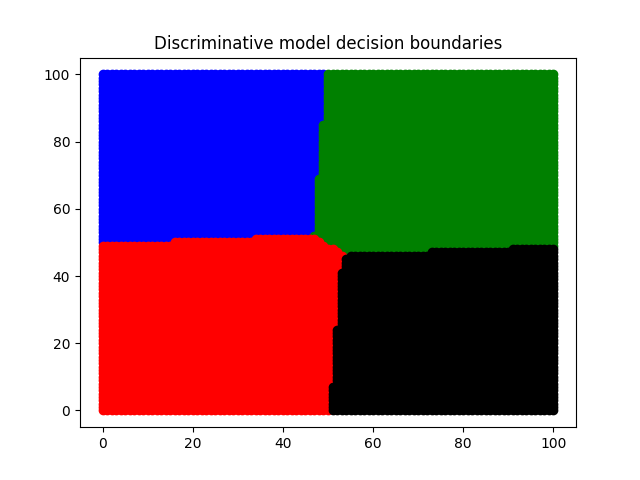
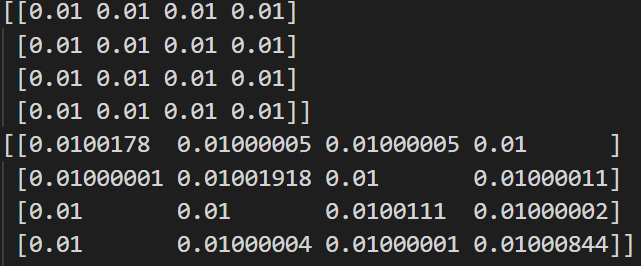
Learning rate設定成0.001，經過一次gradient decent即可分出四個class。Weights為4x4的matrix，一開始weight皆設為0.01



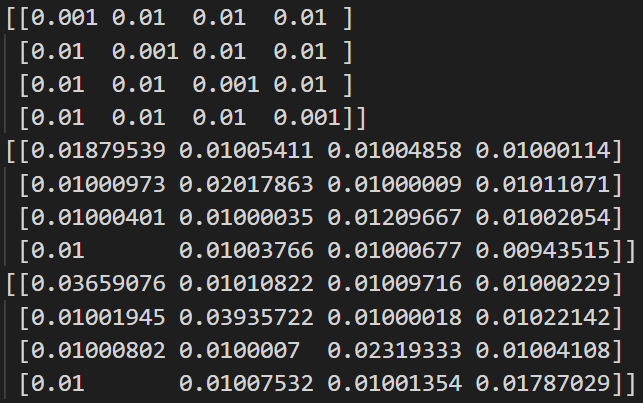
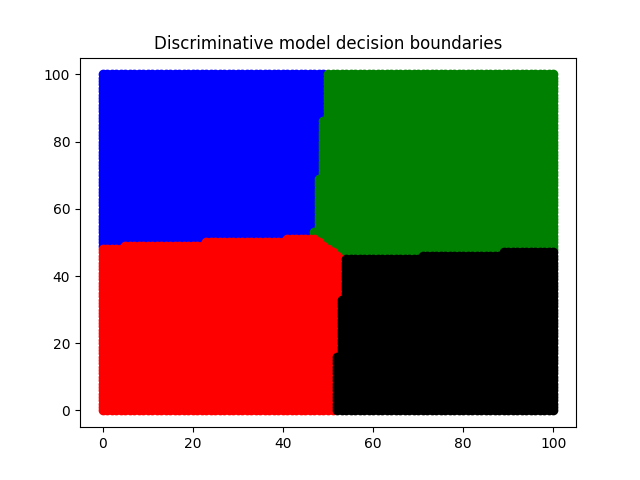
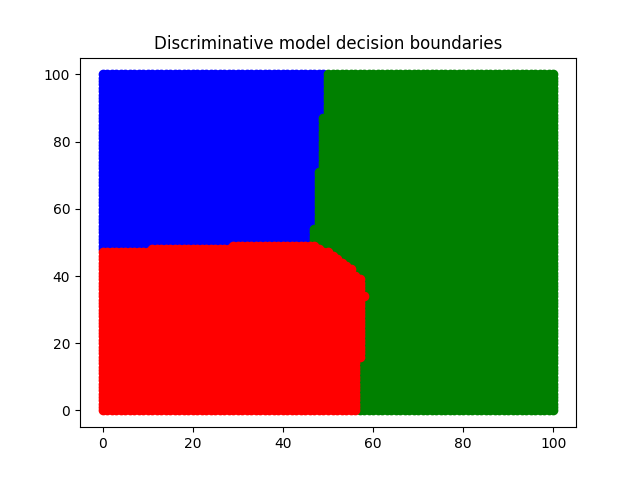
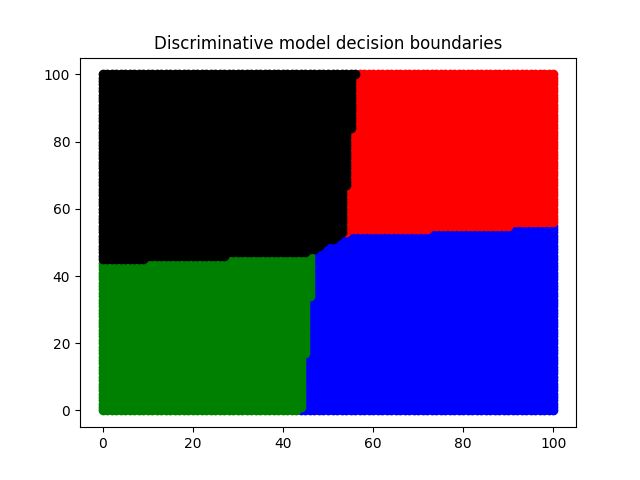


在KxM的Weights matrix中(K=classes=4, M=basis=4)，不同column對應到不同basis，不同row則對應到不同class。在計算class k的a時，如果該筆data屬於class k，則要讓第k個basis function的值越大越好，因此class k的第k個weight要增加，才能使a變大。 從上方的weight matrix可以看出對角線上的element，值都較大，符合前面提到的推論。

我再將四個是四個Gaussian distribution的basis function修改，將mean vector改在四個角落，covariance matrix一樣來自class 1。其結果與weights如下圖，四個class的區域大小不平均，其中class1的區域偏小。

我再修改最初Weights matrix的部分，將對角線的element的值將低到0.001，其他則保持0.01，learning rate則調升到1，其結果與weights如下面四張圖。



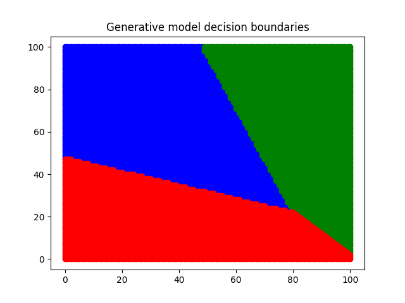
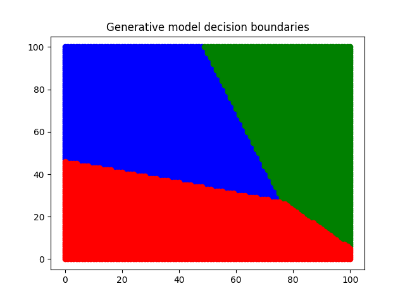
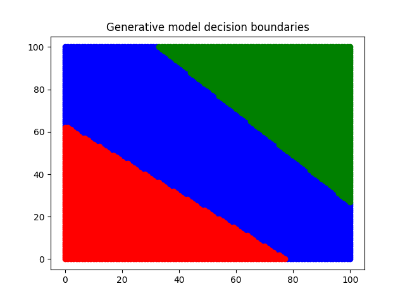
左上圖為使用最初Weights matrix得到的decision boundaries，可以看出class 1與class 4分類完全顛倒。右上圖為經過第一次gradient decent後的decision boundaries，可以看出class 1有出現在合理的位置，但class 4的區域被class 2合併了。左下圖為經過第二次gradient decent後的decision boundaries，可以合理地分出四種類別。

從Weights matrix來看，第一個matrix對角線的element皆為最小值，不太能正確分類。經過一次gradient decent後的matrix，只有class 4的第四個basis function的weight為最小值，因此只有class 4無法正確分類，會被誤分到basis function的weight有最大值的class 2。經過兩次gradient decent後的matrix，對角線的element皆為最大值，可分出四種類別。

**Part II. 3 classes**

1. Generative model

這邊將class 4歸類到class1，因此只剩下3個class。Generative model我依序使用了三個class的covariance matrix來生成model，下圖由左至右分別使用class 1、class 2、class 3的covariance matrix。

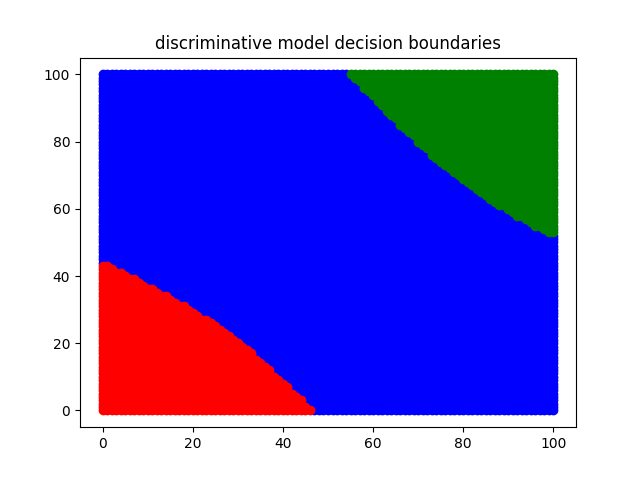
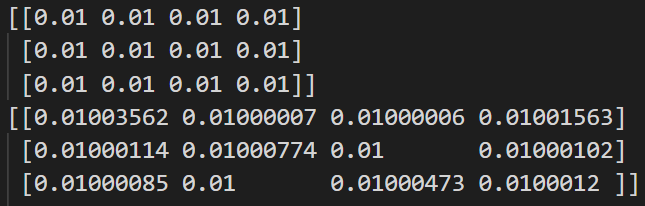


可以看出class 1在合併後，data變得更加分散，variance也隨之變大，covariance matrix與其他兩個class相比較為不同，分類的結果也有所不同，用class 1的covariance matrix得到的model較能將原本在class 4的data分類至class 1，但也會有部分class2和3被誤分到class 1。

Class 1從原本的data來看應該是兩個Gaussian distribution，但資料合併後，model生成的部分還是將整個class 1視為同一個Gaussian distribution，原本資料較少的地方反而成為pdf最高的地方，導致生成結果有誤差。如果將合併後的class 1視為mixture of two Gaussians，可能可以獲得較好的分類結果。

1. Discriminative model

Discriminative model的部分，一開始weight皆設為0.01，Learning rate設定成0.001，basis function一樣使用四個Gaussian distribution，mean vector在四個角落，covariance matrix來自class 1，其經過一次gradient decent的結果與weights如下圖。可以看到其分類結果與Generative model使用class 1 covariance matrix的結果相似，但class1的部分更寬，壓縮到class 2與3的部分，看不太出在原本的每個class附近都有一個basis function帶來的效果。

接著將covariance matrix改成來自class 2，其經過一次gradient decent的結果與weights如下圖。可以看到class 1分布在左上與右下，與原本的class 1和class 4分布相似。我推測是因為在原本的每個class附近都有一個basis function，加上較小的covariance matrix，可以讓basis function接近原本的資料分布，得出更好的分類結果。

