

HW2: Classification

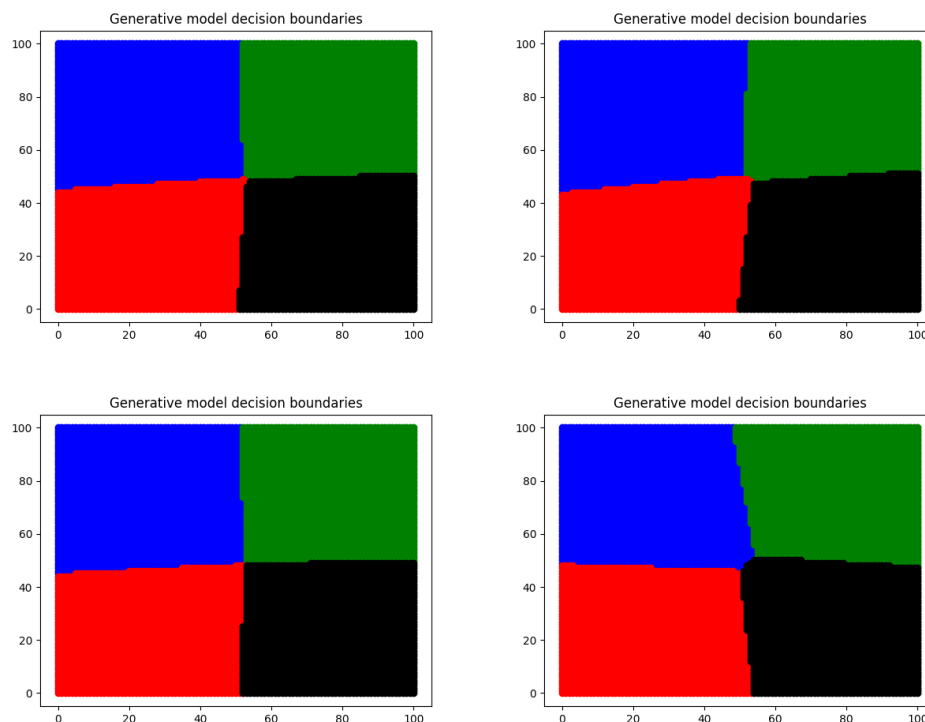
109511219 林錦樑

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。



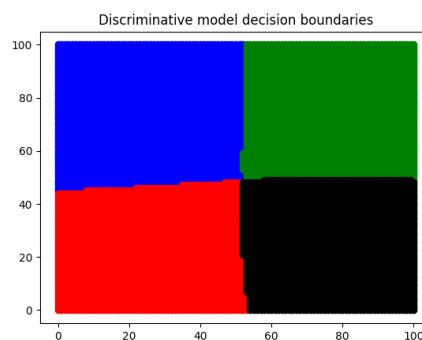
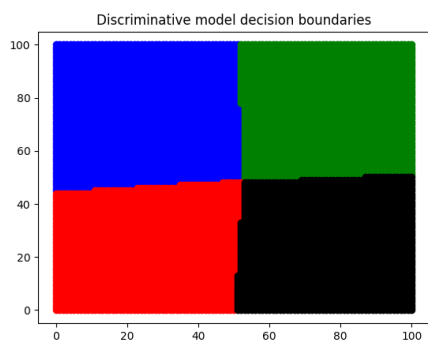
2. Discriminative model

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。

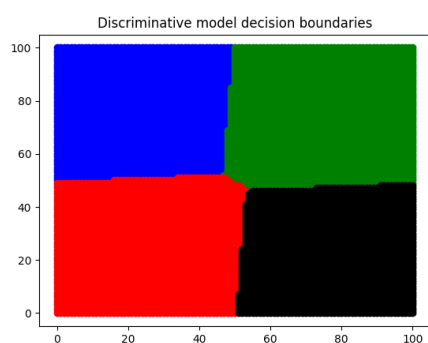
Learning rate 設定成 0.001，經過一次 gradient decent 即可分出四個 class。Weights 為 4x4 的 matrix，一開始 weight 皆設為 0.01



<pre>[[0.01 0.01 0.01 0.01] [0.01 0.01 0.01 0.01] [0.01 0.01 0.01 0.01] [0.01 0.01 0.01 0.01]] [[0.01012487 0.01000746 0.01000597 0.01000065] [0.01000338 0.01008129 0.01000038 0.01000348] [0.01000336 0.0100005 0.01005836 0.0100039] [0.01000005 0.01000227 0.01000246 0.01004922]]</pre>	<pre>[[0.01 0.01 0.01 0.01] [0.01 0.01 0.01 0.01] [0.01 0.01 0.01 0.01] [0.01 0.01 0.01 0.01]] [[0.01012487 0.01000703 0.01000708 0.01000075] [0.01000338 0.0100822 0.01000042 0.01000362] [0.01000336 0.01000052 0.01005623 0.01000322] [0.01000005 0.01000232 0.01000251 0.01005209]]</pre>
---	---

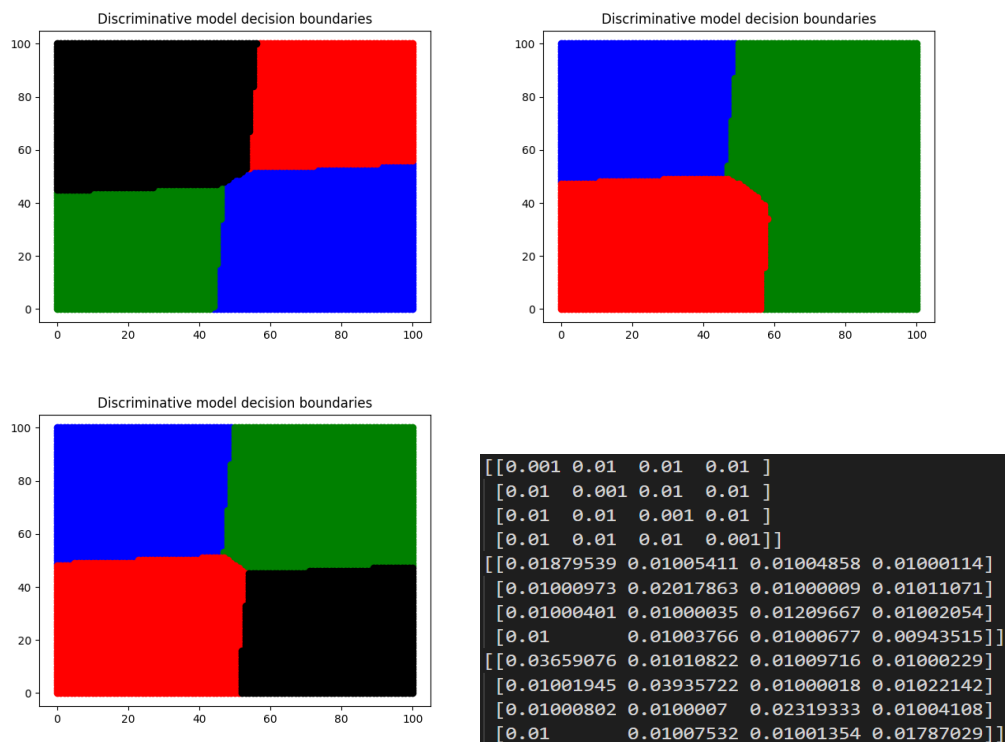
在 $K \times M$ 的 Weights matrix 中($K=\text{classes}=4$, $M=\text{basis}=4$)，不同 column 對應到不同 basis，不同 row 則對應到不同 class。在計算 class k 的 a 時，如果該筆 data 屬於 class k ，則要讓第 k 個 basis function 的值越大越好，因此 class k 的第 k 個 weight 要增加，才能使 a 變大。從上方的 weight matrix 可以看出對角線上的 element，值都較大，符合前面提到的推論。

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。



```
[[0.01 0.01 0.01 0.01]
 [0.01 0.01 0.01 0.01]
 [0.01 0.01 0.01 0.01]
 [0.01 0.01 0.01 0.01]]
[[0.0100178 0.01000005 0.01000005 0.01
 [0.01000001 0.01001918 0.01 0.01000011]
 [0.01 0.01 0.0100111 0.01000002]
 [0.01 0.01000004 0.01000001 0.01000844]]
```

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。



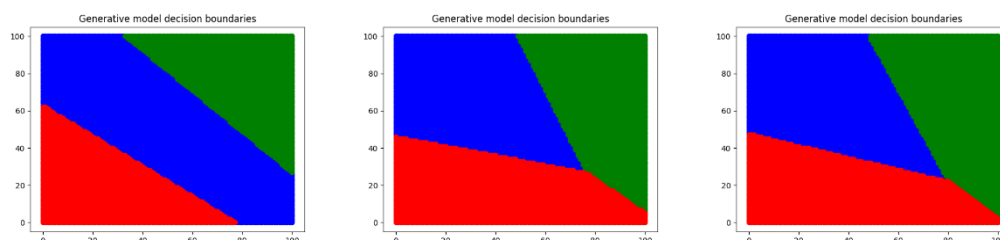
左上圖為使用最初 **Weights matrix** 得到的 **decision boundaries**，可以看出 **class 1** 與 **class 4** 分類完全顛倒。右上圖為經過第一次 **gradient decent** 後的 **decision boundaries**，可以看出 **class 1** 有出現在合理的位置，但 **class 4** 的區域被 **class 2** 合併了。左下圖為經過第二次 **gradient decent** 後的 **decision boundaries**，可以合理地分出四種類別。

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

3. 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**。

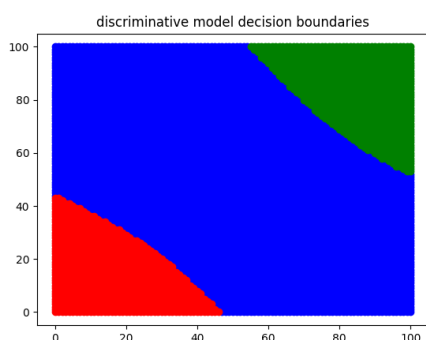


可以看出 class 1 在合併後，data 變得更加分散，variance 也隨之變大，covariance matrix 與其他兩個 class 相比較為不同，分類的結果也有所不同，用 class 1 的 covariance matrix 得到的 model 較能將原本在 class 4 的 data 分類至 class 1，但也會有部分 class2 和 3 被誤分到 class 1。

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。

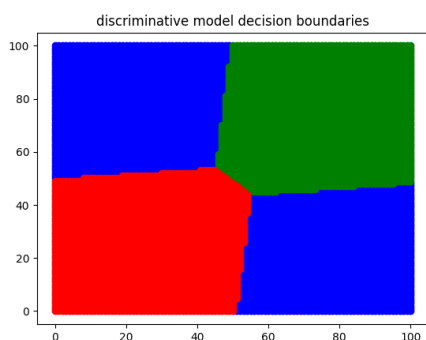
4. Discriminative model

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。



```
[[0.01 0.01 0.01 0.01]
 [0.01 0.01 0.01 0.01]
 [0.01 0.01 0.01 0.01]]
[[0.01003562 0.01000007 0.01000006 0.01001563]
 [0.01000114 0.01000774 0.01      0.01000102]
 [0.01000085 0.01      0.01000473 0.0100012 ]]
```

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。



```
[[0.01 0.01 0.01 0.01]
 [0.01 0.01 0.01 0.01]
 [0.01 0.01 0.01 0.01]]
[[0.01001503 0.01000008 0.01000005 0.01000718]
 [0.01000001 0.01001751 0.01      0.01000009]
 [0.01      0.01      0.01001007 0.01000001]]
```