

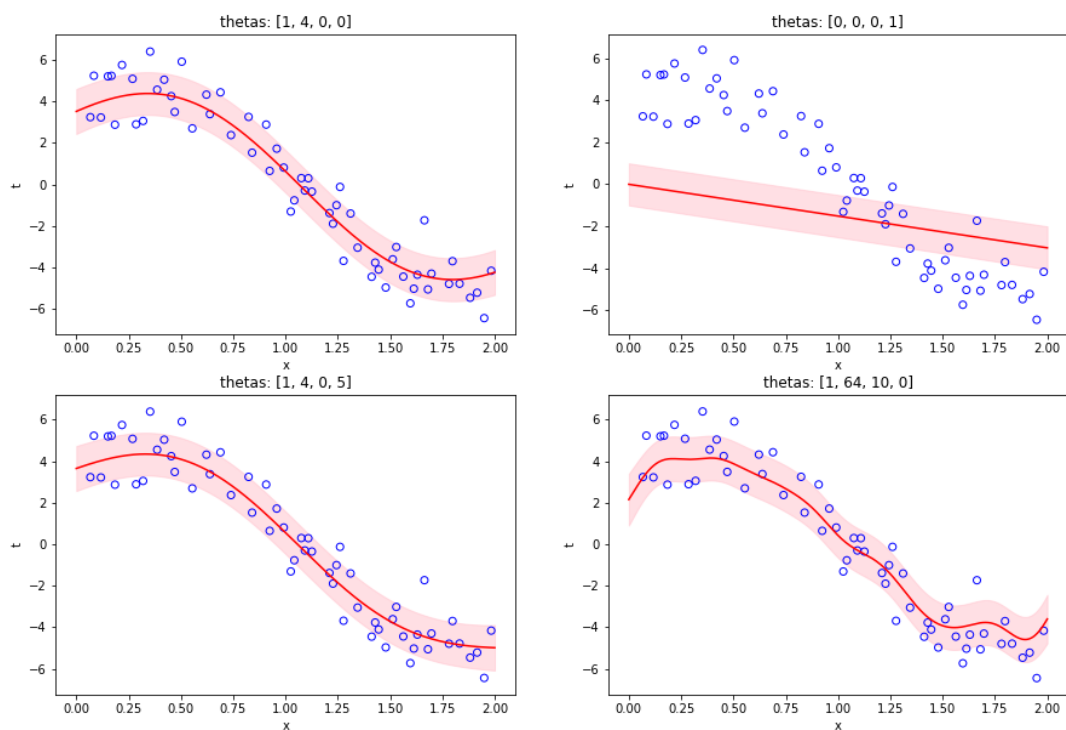
ML HW3 Report

姓名：陳品媛

學號：A063021

Problem 1 Gaussian Process

Prediction result



Root-mean-square errors

Thetas	train_RMSE	test_RMSE
[1, 4, 0, 0]	1.03657014710737	1.1042120692453026
[0, 0, 0, 1]	3.3761505372595733	3.843846395272742
[1, 4, 0, 5]	1.0234011882136607	1.093381396828782
[1, 64, 10, 0]	1.048217914229123	1.294071842323675

My discussion

Exponential-quadratic kernel function

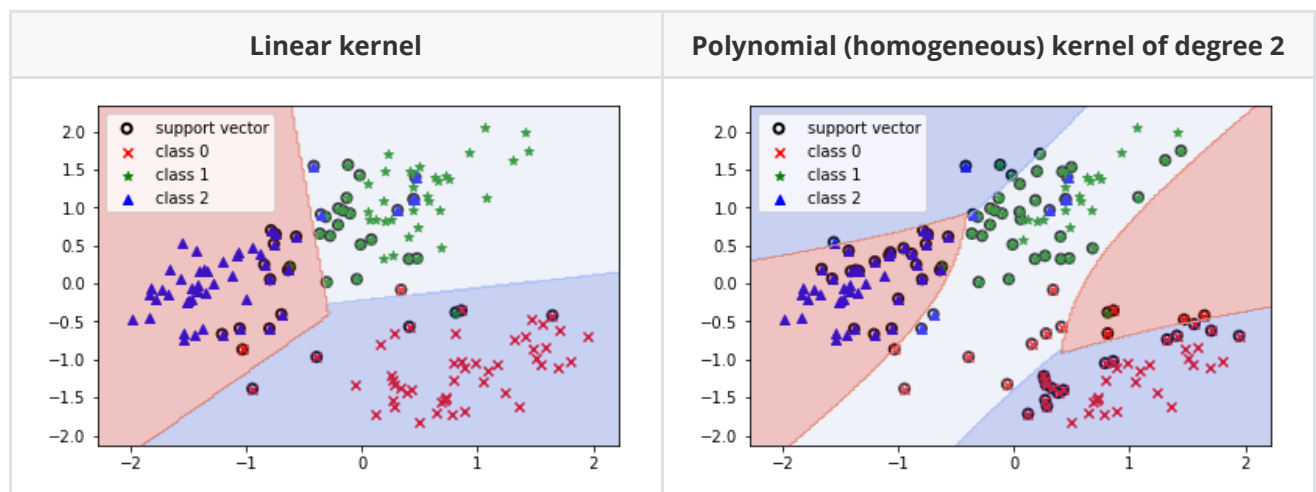
$$k(x_n, x_m) = \theta_0 \exp\left\{-\frac{\theta_1}{2} \|x_n - x_m\|^2\right\} + \theta_2 + \theta_3 x_n^T x_m$$

觀察實驗結果，明顯注意到當 $\theta = [0, 0, 0, 1]$ 的錯誤相對其他設定偏高，因為他只用了linear kernel： $x_n^T x_m$ ，所以相對變化較少，因此在training data與testing data都表現不好，underfitting。

另外， $\theta = [1, 64, 10, 0]$ 我認為是overfitting的結果（train_RSME與其中兩種設定差不多，但test_RMSE卻比其他的大了許多），他在 $\|x_n - x_m\|^2$ 的部份配的權重偏多，然而這裡能表示的也較複雜許多，對training data可以fit的不錯，但也因此對testing data的fitting狀況沒有像其他兩者設定來的好。

最後，第一與第三組的設定很相似，僅差在第三組添加 $x_n^T x_m$ ，相較起來，增加一些多樣性，但是可用第二組的設定看出其實對整體model的成效並不是很多，因此結果上第一與第三組的結果並不會差太多。

Problem 2 Support Vector Machine



My discussion

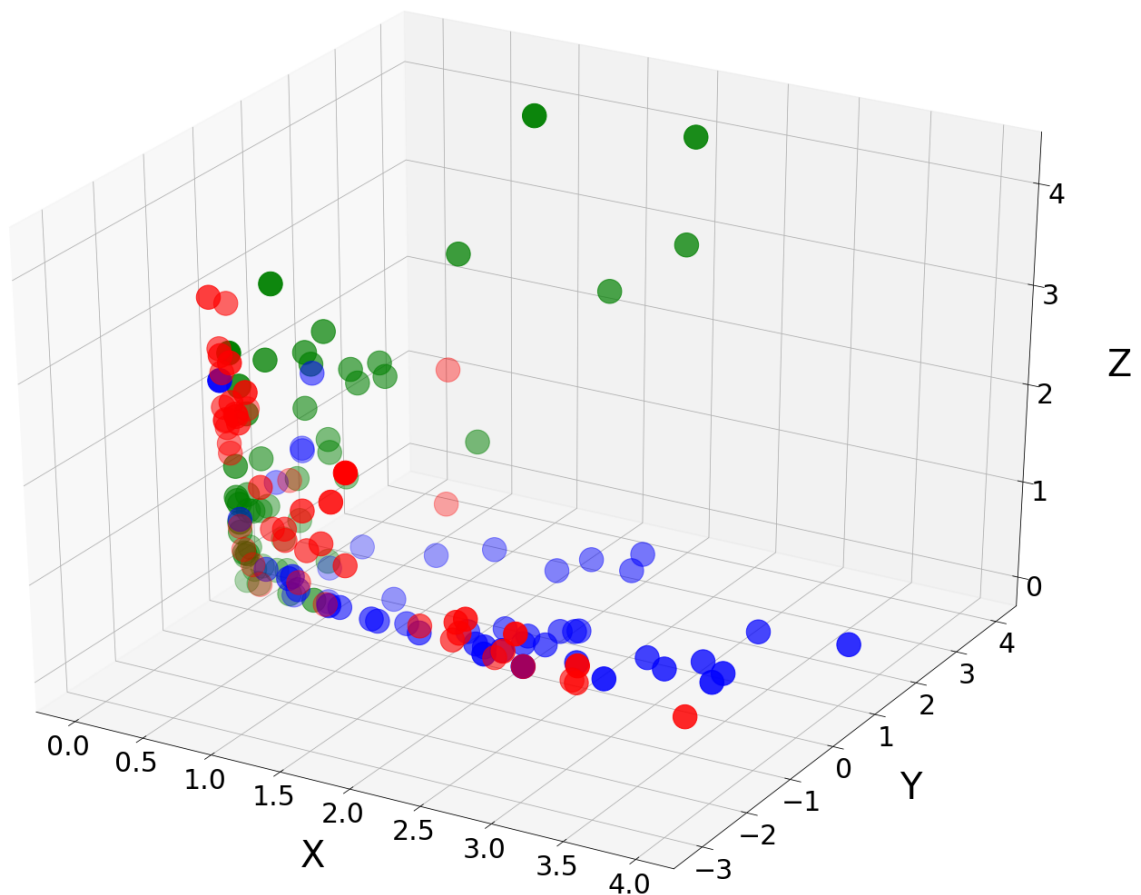
首先，我有對資料做標準化，實作採用One-versus-one解決multiclass，並利用sklearn裡的工具fit data後取得coefficient，分配multiplier給對應的分類器，算出每個分類器的weight與bias進行預測，最後採用投票作為分類結果。

利用下圖coefficient所位在的位置取絕對值（ $dual_coef[i] = labels[i] * alphas[i]$ ），並依據所屬類別分配給對應的分類器。

Consider a three class problem with class 0 having three support vectors v_0^0, v_0^1, v_0^2 and class 1 and 2 having two support vectors v_1^0, v_1^1 and v_2^0, v_2^1 respectively. For each support vector v_i^j , there are two dual coefficients. Let's call the coefficient of support vector v_i^j in the classifier between classes i and k $\alpha_{i,k}^j$. Then `dual_coef_` looks like this:

$\alpha_{0,1}^0$	$\alpha_{0,2}^0$	Coefficients for SVs of class 0
$\alpha_{0,1}^1$	$\alpha_{0,2}^1$	
$\alpha_{0,1}^2$	$\alpha_{0,2}^2$	
$\alpha_{1,0}^0$	$\alpha_{1,2}^0$	Coefficients for SVs of class 1
$\alpha_{1,0}^1$	$\alpha_{1,2}^1$	
$\alpha_{2,0}^0$	$\alpha_{2,1}^0$	Coefficients for SVs of class 2
$\alpha_{2,0}^1$	$\alpha_{2,1}^1$	

我認為單純觀察資料分佈，分佈上大致可以分為三群，overlap的狀況並不是說很嚴重。而從實驗的分類結果，我認為linear kernel就分類的還不錯，雖然overlap的點無法分類正確，但由於資料本身的重疊狀況就沒有很多，所以我覺得結果還不錯。反觀polynomial kernel，可以明顯發現在邊界的地方圓滑許多，可以順著資料的分佈，但是注意到中下半部的紅色點，都被分類到錯誤的類別，我猜測可能原因是polynomial kernel function將資料投影到feature space後的情況不並能明顯將不同類別的資料區隔出來，如下圖將資料投影後似乎也是無法線性分離開。



另外，觀察support vector，linear kernel可以用三調線將資料分開，所以support vector單純分佈在分隔線的附近；polynomial kernel用了六條線去分隔，所以support vector的數量大幅增加，也同時增加記憶體用量。所以透過這個實驗，我認為應該要觀察資料實際分佈的特性，而去選用適合的kernel function去訓練。

Problem 3 Gaussian Mixture Model

K-means model

Estimated $\{\mu_k\}_{k=1}^K$

```
===== K = 2 (K_means) =====  
+-----+-----+-----+-----+
```

K_means	R	G	B
0	182	202	229
1	106	86	34

===== K = 3 (K_means) =====

K_means	R	G	B
0	66	60	45
1	182	202	229
2	146	113	25

===== K = 5 (K_means) =====

K_means	R	G	B
0	184	206	233
1	87	62	15
2	38	60	88
3	137	134	144
4	154	119	16

===== K = 20 (K_means) =====

K_means	R	G	B
0	171	197	227
1	137	127	136
2	160	183	214
3	84	66	61
4	182	186	191
5	85	60	10
6	173	151	69
7	245	248	249
8	57	75	106
9	207	223	245
10	49	33	6
11	148	155	178
12	109	98	104
13	117	85	6
14	193	213	242
15	182	207	238
16	120	95	41
17	180	140	15
18	149	113	10
19	7	46	81

Resulting images



GMM

Estimated $\{\mu_k\}_{k=1}^K$

===== K = 2 (GMM) =====

GMM	R	G	B
0	182	205	235
1	120	106	66

===== K = 3 (GMM) =====

GMM	R	G	B
0	104	84	25
1	182	205	235
2	147	141	133

===== K = 5 (K_means) =====

K_means	R	G	B
0	184	206	233
1	87	62	15
2	38	60	88
3	137	134	144

```

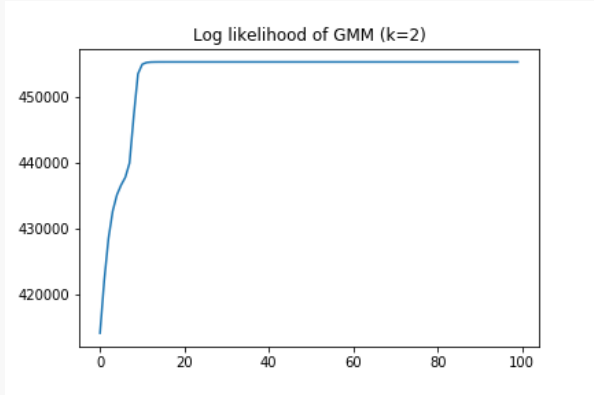
|    4    | 154 | 119 | 16 |
+-----+-----+-----+-----+

===== K = 20 (GMM) =====
+-----+-----+-----+-----+
| GMM |  R  |  G  |  B  |
+-----+-----+-----+-----+
|  0  | 173 | 201 | 231 |
|  1  | 147 | 135 | 149 |
|  2  | 167 | 193 | 224 |
|  3  |  88 |  69 |  64 |
|  4  | 189 | 202 | 213 |
|  5  |  94 |  6  |  11 |
|  6  | 138 | 120 |  75 |
|  7  | 248 | 250 | 251 |
|  8  |  60 |  77 | 110 |
|  9  | 202 | 219 | 245 |
| 10  | 103 |  74 |  0  |
| 11  | 165 | 184 | 213 |
| 12  |  52 |  74 |  91 |
| 13  | 122 |  89 |  4  |
| 14  | 193 | 213 | 241 |
| 15  | 183 | 209 | 240 |
| 16  |  96 |  72 |  20 |
| 17  | 163 | 137 |  26 |
| 18  | 157 | 119 |  13 |
| 19  |  4  |  46 |  83 |
+-----+-----+-----+-----+

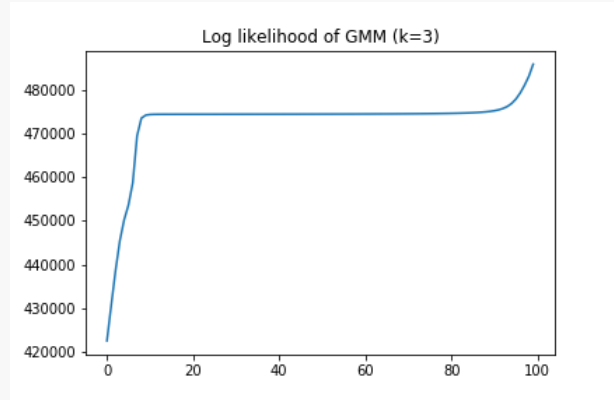
```

Log likelihood curve of GMM

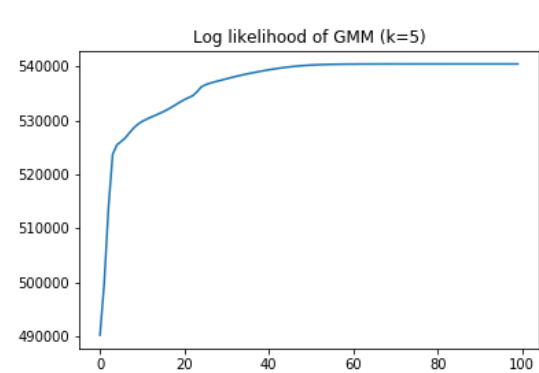
K = 2



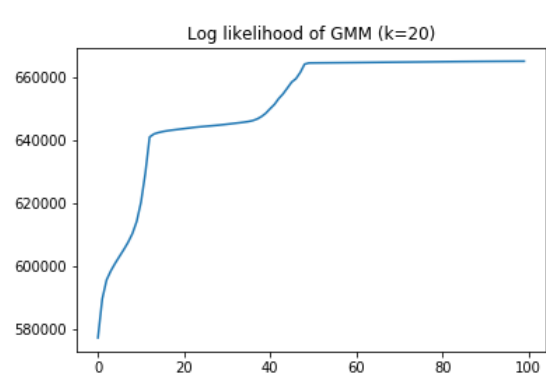
K = 3



K = 5

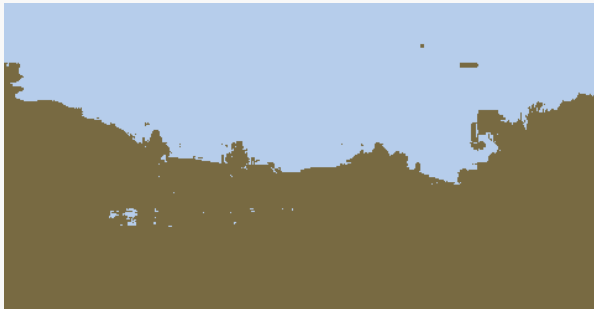


K = 20



Resulting images

K = 2



K = 3



K = 5



K = 20



My discussion

從實驗結果觀察到K-means相對於GMM更能用有限的顏色表示原先的圖片的對比感覺。另外，比較兩個model對於K=20的結果，於天空的部份，K-means相較之下圓滑許多，不像GMM出現許多的方格，我認為其原因為K-means會考慮顏色之間的距離，會把相近的顏色（實際在圖片上也是距離近的）拉在同一類，然而GMM是從機率的角度出發。

關於GMM的收斂，當K越大，會需要更多的iteration來達到收斂，我認為可能的原因是因為當可以使用的顏色越多，就可以產生更多的可能結果，所以需要更多時間去嘗試到對的方向。