

0860908 李少琪
Machine Learning HW3

1. Gaussian Process for Regression

a. 過程

(1) 利用 kernel

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

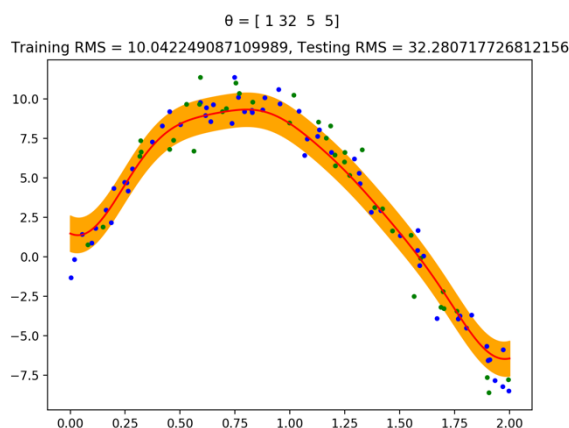
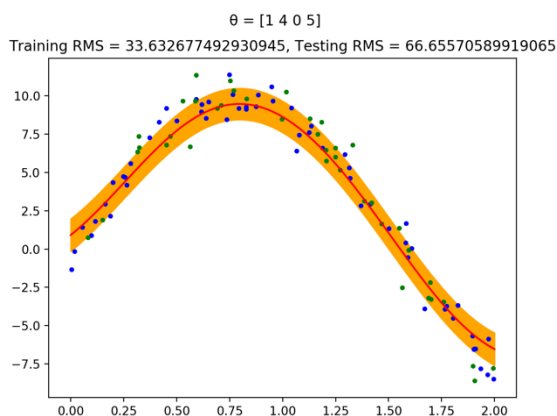
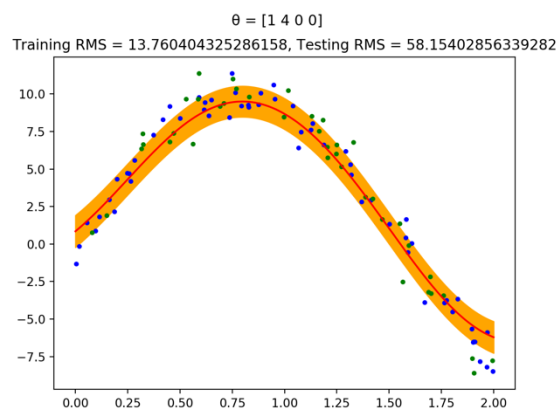
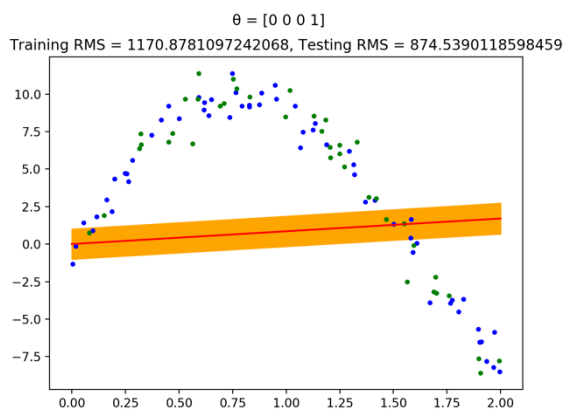
實作迴歸預測的高斯過程，並計算 root-mean-square error。

(2) 調整超參數 θ

我先以 $\theta = [1, 4, 0, 5], [5, 4, 0, 5], [8, 4, 0, 5], [15, 4, 0, 5]$ ，找出 $\theta = [8, 4, 0, 5]$ 的 root-mean-square error 最低，再以 $\theta = [8, 4, i, 5]$ ， $i=1, \dots, 100$ ，找出 root-mean-square error 的組合

b. 結果

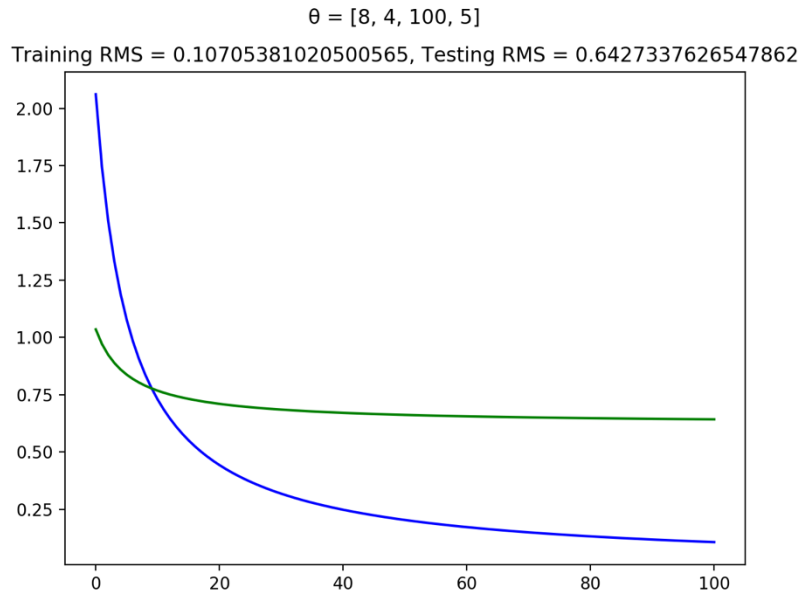
(1) $\theta = [0, 0, 0, 1], [1, 4, 0, 0], [1, 4, 0, 5], [1, 32, 5, 5]$



(2) 調整超參數 $\theta = [8, 4, i, 5], i=1, \dots, 100$

X 軸： i

Y 軸：root-mean-square error (藍:train, 綠:test)



c. 分析

從上述(1)可以發現，當 $\theta = [0, 0, 0, 1]$ 時， $k(x_n, x_m) = x_n^T x_m$ 是 linear kernel，完全無法擬和這次作業非線性的資料。比較 $\theta = [1, 4, 0, 0]$ 與 $\theta = [1, 4, 0, 5]$ ，可以發現當不考慮 $x_n^T x_m$ 時，training data 的 rms error 可以降的比較低，但 testing data 無法，而考慮 $x_n^T x_m$ 比較不會有太明顯 overfitting 的現象。

(3)我先比較了 $\theta = [1, 4, 0, 5], [5, 4, 0, 5], [8, 4, 0, 5], [15, 4, 0, 5]$ ，找出 $[8, 4, 0, 5]$ 的 rms 最低，在 $\theta = [8, 4, 10, 5]$ 時，training data 的 rms 是 0.7295，testing data 的 rms 是 0.7678，training data 的 rms 和 testing data 的 rms 並不會相差太遠，我認為是不錯的組合。

2. Support Vector Machine

a. 過程

Step 1. 讀取 x_train.csv 和 t_train.csv

Step 2. 利用 Principal Component Analysis 從 x_train 選出 2 個 Features

one-versus-the-rest

Step 3. 當要預測 class 1 時，將 t_train 的 class 1 改為 1，其餘改零；預測 class 2 時，將 t_train 的 class 2 改為 1，其餘改零；預測 class 3 時，將 t_train 的 class 3 改為 1，其餘改零。

one-versus-one

Step 3. 分別 class 1 V.S. class 2、class 1 V.S. class 3、class 2 V.S. class 3

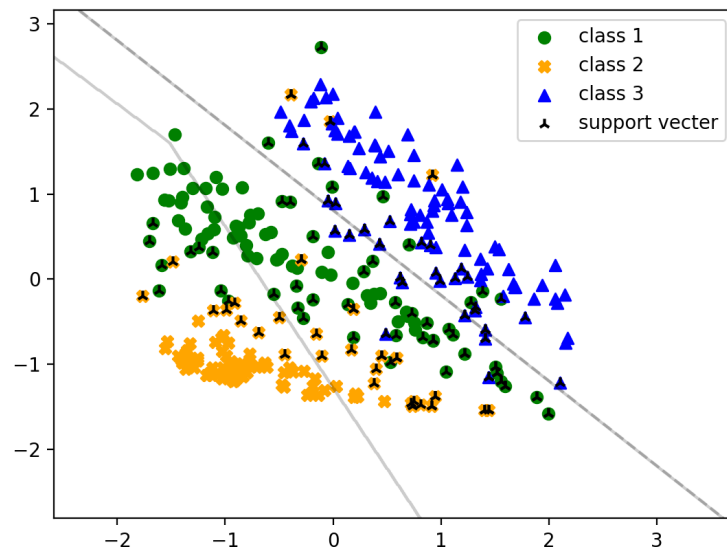
Step 4. Fit support vector machine (linear kernel & polynomial kernel)

Step 5. 以 $\text{decision function} > 0$ ，為預測到該類別， $\text{decision function} < 0$ ，預測到非該類別

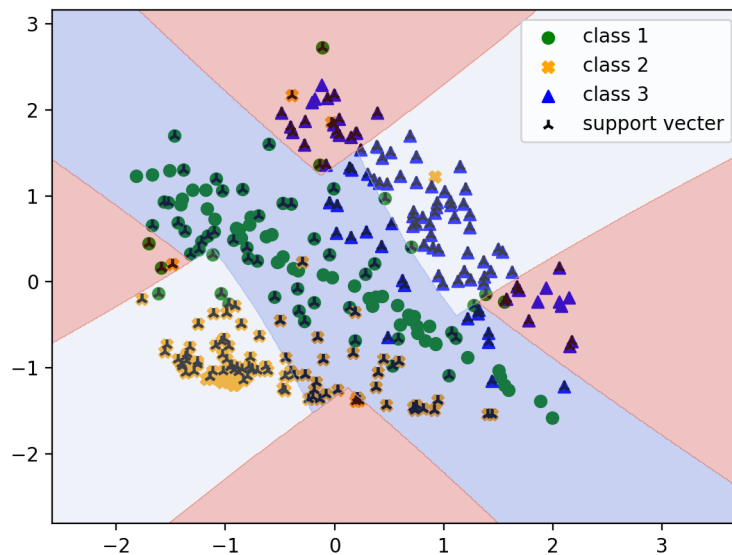
Step 6. 以投票方式選出最終預測類別

b. 結果

linear kernel , one-versus-one



Polynomial kernel, one-versus-one



c. 分析

在實作 linear kernel 的 one-versus-the-rest 時，預測出來的 decision

function 都是負的，無法作出預測，直接使用 scikit-learn SVM 的 predict 也是一樣，所以直接實作 one-versus-one。

從上述結果可以發現 linear kernel 只能將資料以一線區隔不同類別，所以只能將 linear kernel 實作於有這種現象的資料上。而 polynomial kernel 可以的於此資料分類的較好。

3. Gaussian Mixture Model

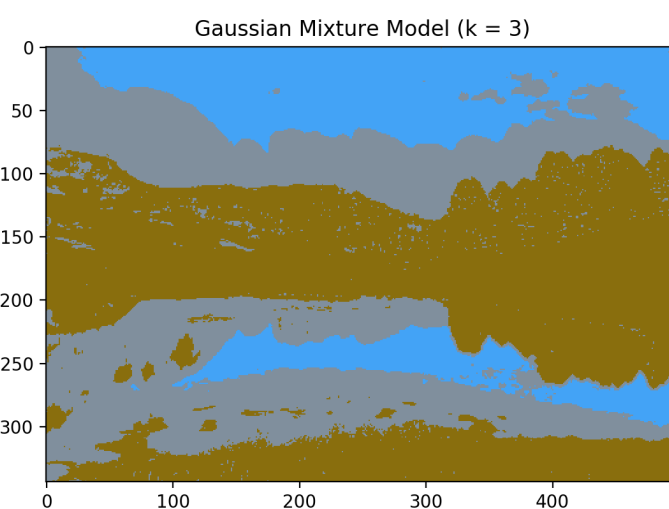
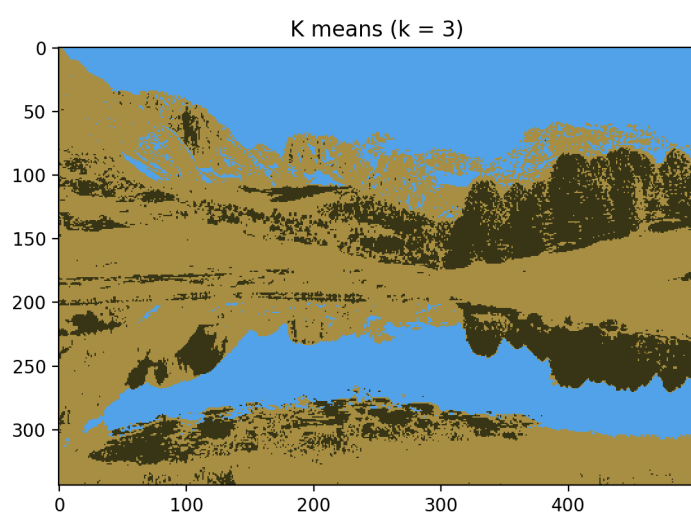
a. 過程

分別利用 k means 和 gaussian mixture model 實作 image segmentation

b. 結果

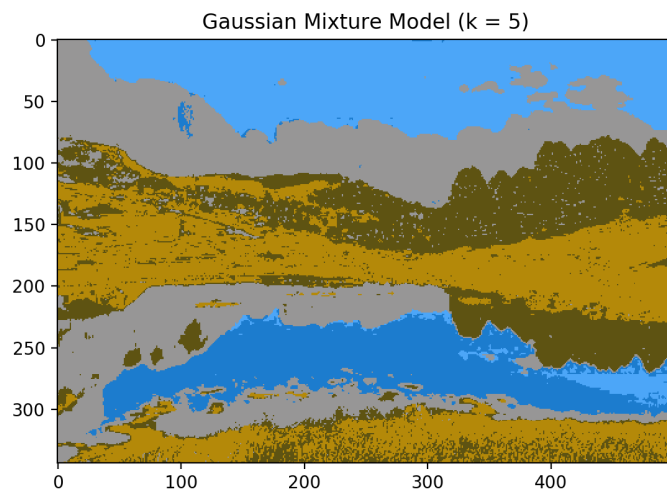
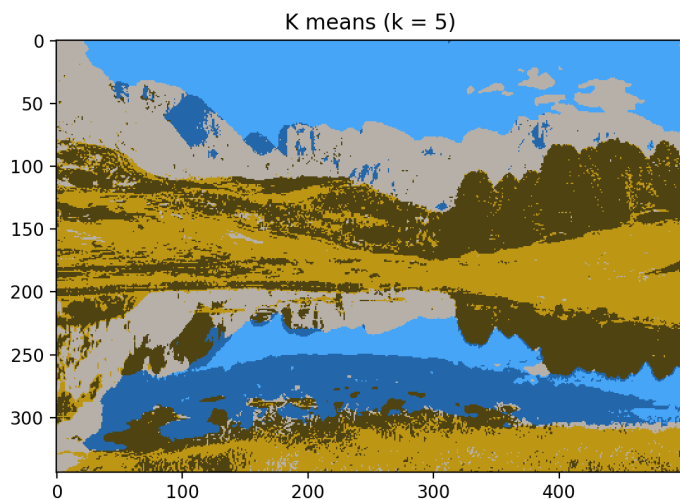
K = 3

K means			Gaussian Mixture Model		
R	G	B	R	G	B
55	53	21	137	110	13
82	162	233	67	163	246
167	142	66	128	143	157



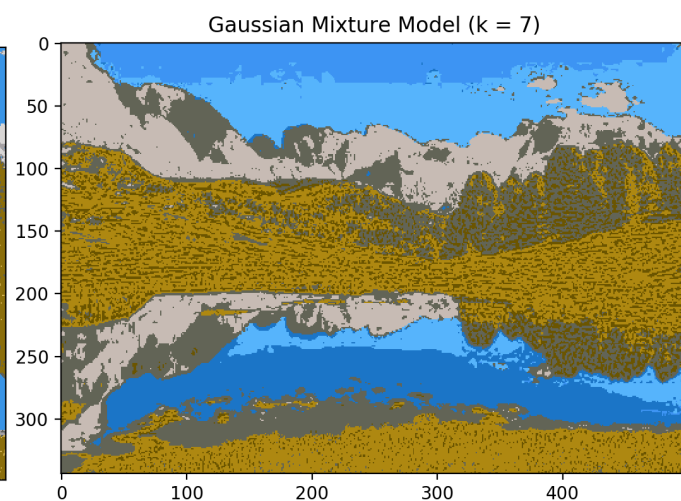
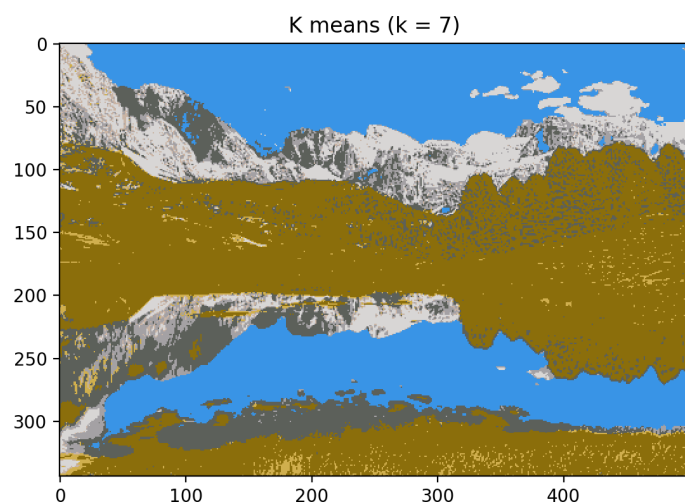
K = 5

K means			Gaussian Mixture Model		
R	G	B	R	G	B
71	165	247	76	166	248
79	67	17	94	83	18
35	103	170	28	124	206
189	150	19	180	137	9
183	176	169	151	150	150



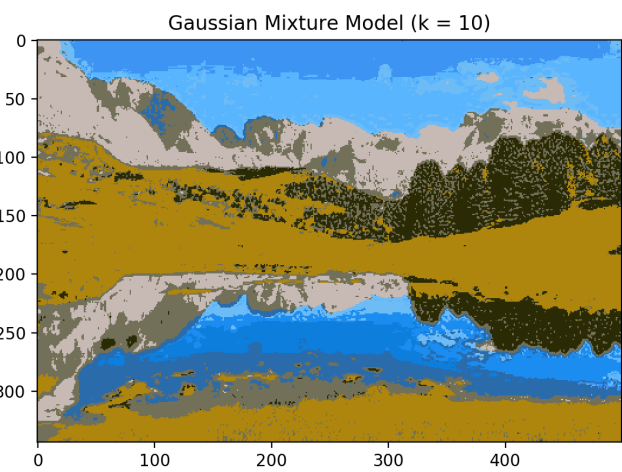
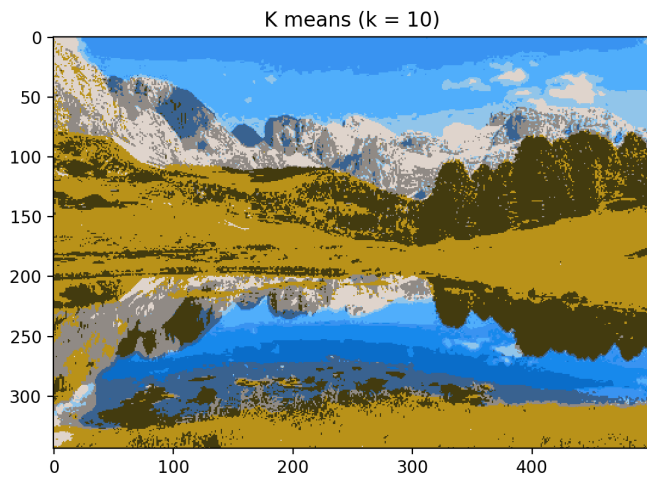
K = 7

K means			Gaussian Mixture Model		
R	G	B	R	G	B
164	161	164	98	100	86
57	148	230	61	148	243
208	175	78	174	136	17
216	214	213	86	180	252
92	96	90	27	117	199
139	110	11	107	85	0
201	180	164	199	187	180

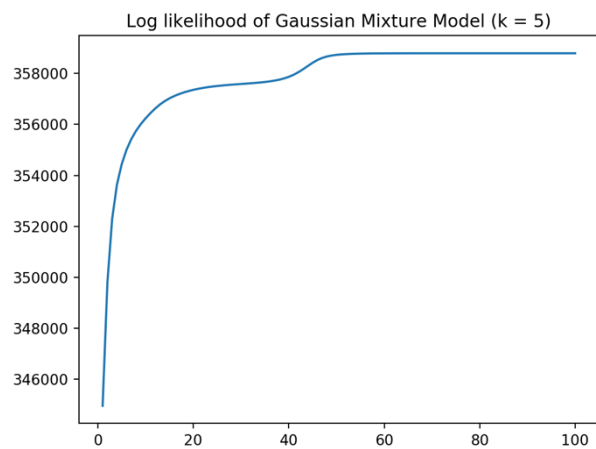
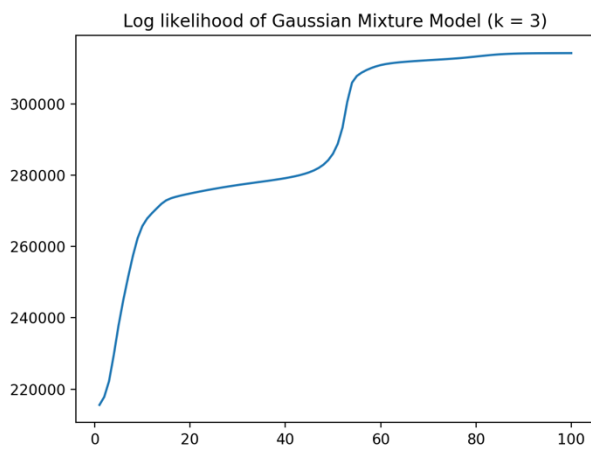


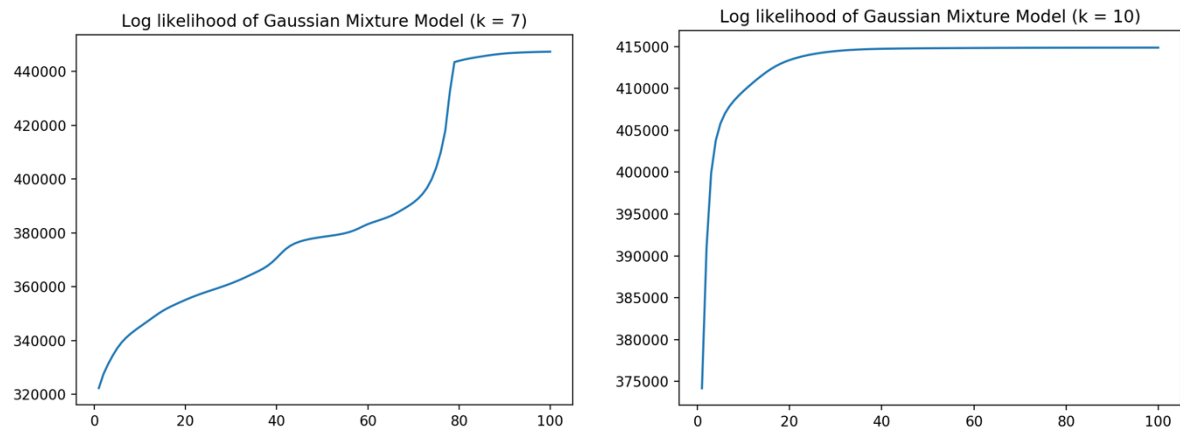
K = 10

K means			Gaussian Mixture model		
R	G	B	R	G	B
80	174	251	89	181	253
10	109	201	13	126	220
145	197	234	110	188	246
57	98	144	41	107	171
67	59	15	42	43	5
23	137	236	27	140	240
144	138	133	114	112	89
185	146	25	174	134	13
223	212	204	199	186	180
57	147	241	61	148	243



Log likelihood of Gaussian Mixture Model





c. 分析

從上述的結果表現，就影像分割我認為 K means 的表現比較好，比較有把重要的訊息留下來，且 Gaussian Mixture Model 容易將邊緣的兩側都找出來，讓結果感覺出有很多雜訊。