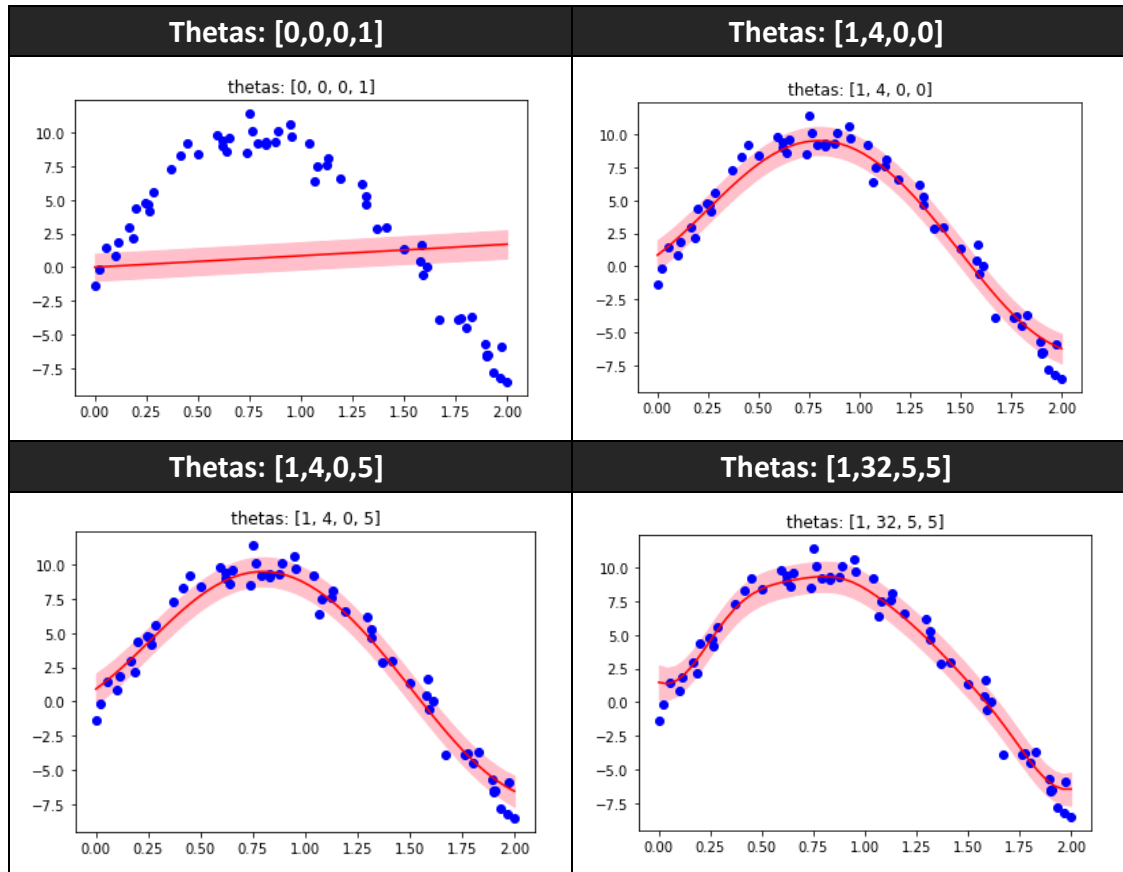


Machine Learning HW3

0853412 資管碩一 吳宛儒

1-2

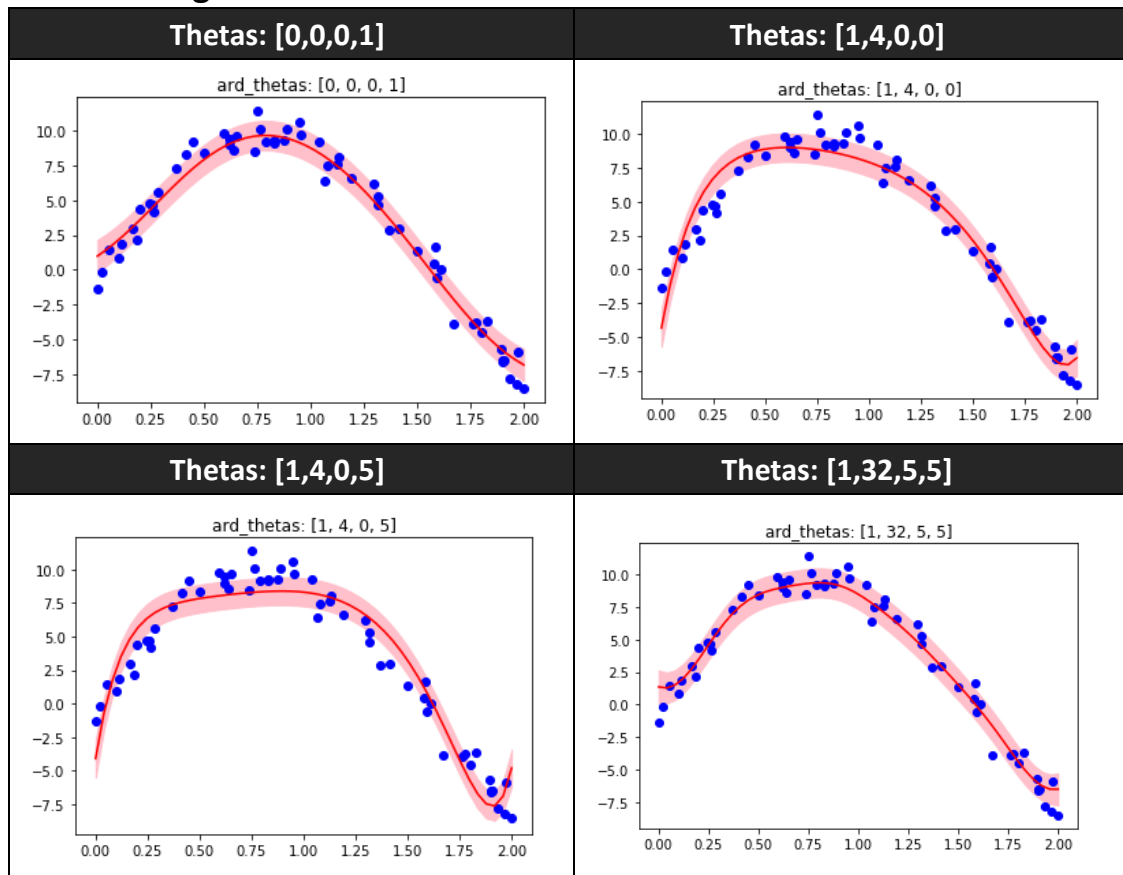


1-3

Thetas: [0,0,0,1]		Thetas: [1,4,0,0]	
Train	Test	Train	Test
[[6.65758954]]	[[6.74853909]]	[[1.05224307]]	[[1.29879576]]
Thetas: [1,4,0,5]		Thetas: [1,32,5,5]	
Train	Test	Train	Test
[[1.0288404]]	[[1.28609023]]	[[0.96404492]]	[[1.25846852]]

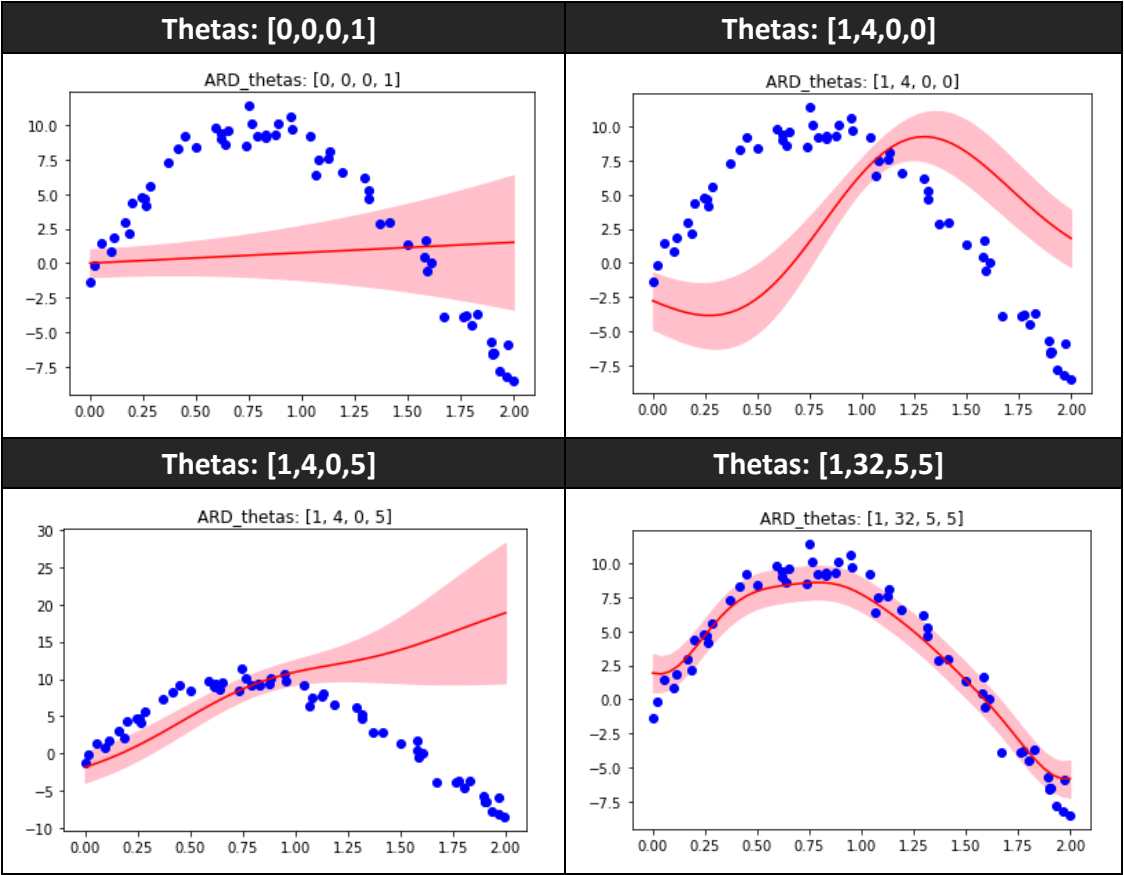
1-4

ARD learning rate= 0.0001



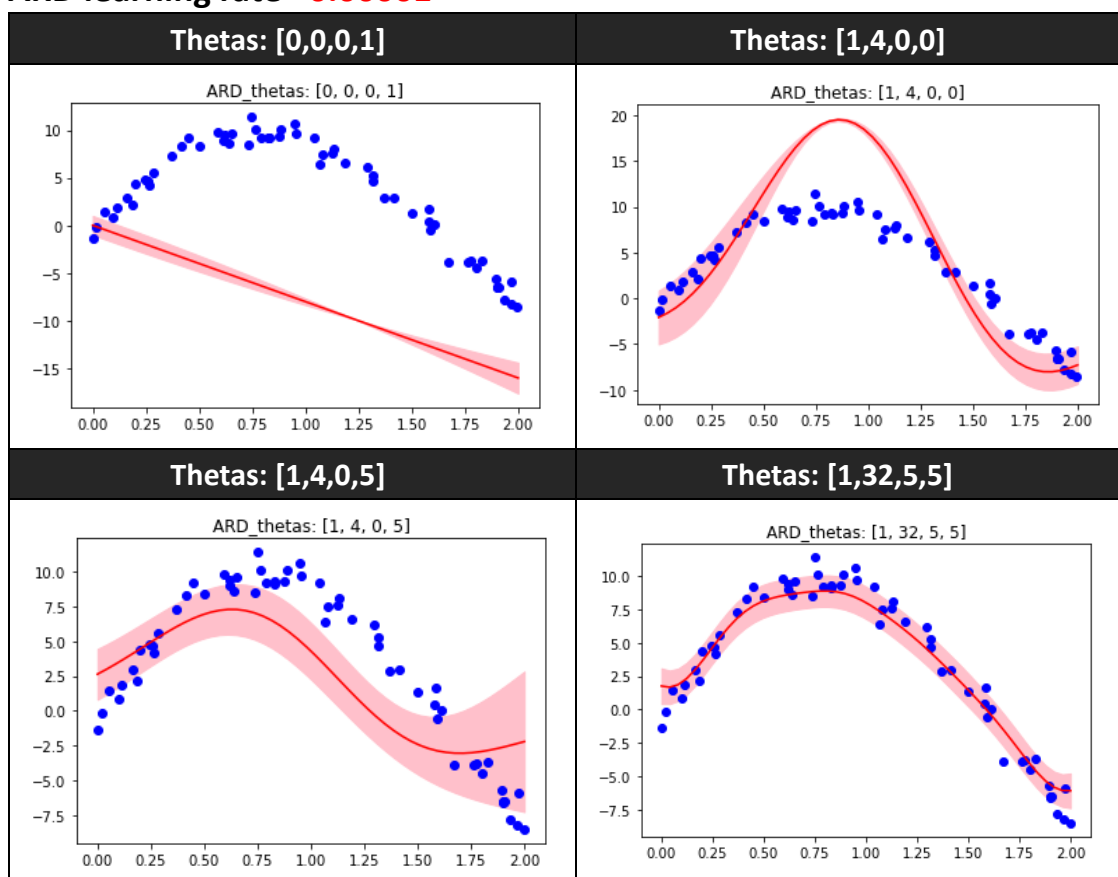
Thetas: [0,0,0,1]		Thetas: [1,4,0,0]	
Train	Test	Train	Test
[[0.97162025]]	[[1.21648941]]	[[1.32446569]]	[[1.33961707]]
Thetas: [1,4,0,5]		Thetas: [1,32,5,5]	
Train	Test	Train	Test
[[1.58240701]]	[[1.57015261]]	[[0.94423603]]	[[1.24366664]]

ARD learning rate= 0.001



Thetas: [0,0,0,1]		Thetas: [1,4,0,0]	
Train	Test	Train	Test
[[6.65876231]]	[[6.77036825]]	[[7.56589985]]	[[7.79003101]]
Thetas: [1,4,0,5]		Thetas: [1,32,5,5]	
Train	Test	Train	Test
[[12.14293537]]	[[11.12835704]]	[[1.28399296]]	[[1.57993645]]

ARD learning rate= 0.00001



Thetas: [0,0,0,1]		Thetas: [1,4,0,0]	
Train	Test	Train	Test
[[12.46505149]]	[[13.5411752]]	[[5.28553288]]	[[4.76680009]]
Thetas: [1,4,0,5]		Thetas: [1,32,5,5]	
Train	Test	Train	Test
[[3.5155231]]	[[4.03419488]]	[[1.14106667]]	[[1.44280739]]

1-5

第一次實驗在未使用 ARD 並且僅有 θ_3 非零的時候，其結果為線性且幾乎完全無法呈現真實 data 的分布情形，但是當 θ_0 和 θ_1 為非零且 θ_2 和 θ_3 為零的時候，效果便改善很多，到最後 thetas 為 $[1, 32, 5, 5]$ 的時候已經能有很不錯的呈現效果。在使用 ARD 以後，我嘗試了 learning rate=0.0001 和 0.001 兩種，發現兩種所得出的結果差異蠻大，尤其是在 thetas 為 $[0, 0, 0, 1]$ 和 $[1, 4, 0, 0]$ 和 $[1, 4, 0, 5]$ 的時候，較大的 learning rate 似乎無法真正學習到較好 fitting data 的方式，也有較大的 RMS error。因著這樣的結果我又嘗試了 learning rate 為 0.00001 的方式，發現其效果並沒有想像中來得好。由此推測出 learning rate 並非越大越好，而是在設定為 0.001 的時候為最適合此 data 的 learning rate。

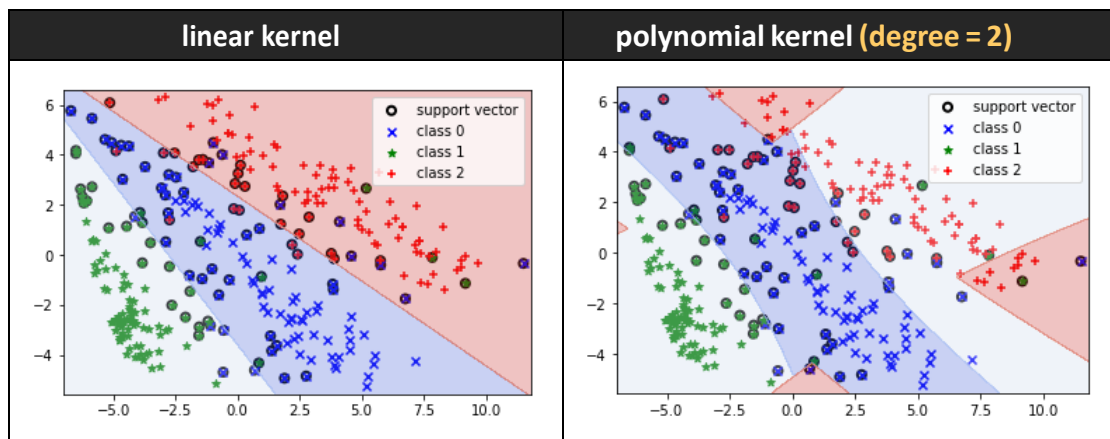
2-1

one-versus-the-rest: 在 training 的時候依據把某個類別的樣本歸為一類，期他剩餘的樣本歸為另一類(一次共兩類)，如果有 K 個類別的樣本，就會有 K 個 SVM。分類時將未知樣本分類為具有最大分類函數值的那類。假設有 4 個類別 (label)，當依上述方法建好 4 個 SVM 以後，在 testing 的時候，將 testing vector 分別利用這 4 個進行測試，得到 $f_1(x)$, $f_2(x)$, $f_3(x)$, $f_4(x)$ ，最終分的類別就是最大函數值的那個。

one-versus-one: 在任意兩個樣本之間設計一個 SVM，當有 k 個樣本的時候就需要設計 $k(k-1)/2$ 個 SVM。當面對一個未知的樣本欲將其作分類，得票最多的類別就會是它的類別。假設今有 4 個類別 A, B, C, D，便會有 A, B; A, C; A, D; B, C; B, D; C, D 六個 SVM，透過 training 得到六個 trained SVM，testing 的時候每個未知的樣本就會得到一組結果，並用投票的方式決定最終的分類結果。選擇後者，因為第一種方法可能會存有 bias 偏差(只用屬於某類/不屬於某類來分割)，第二種方法雖然在類別很多的情況下成本較高，但多數實驗皆說明這種方法會比第一種方法還要好。

2-3, 2-4

SVM with multi-class classification: the decision boundary and support vectors.

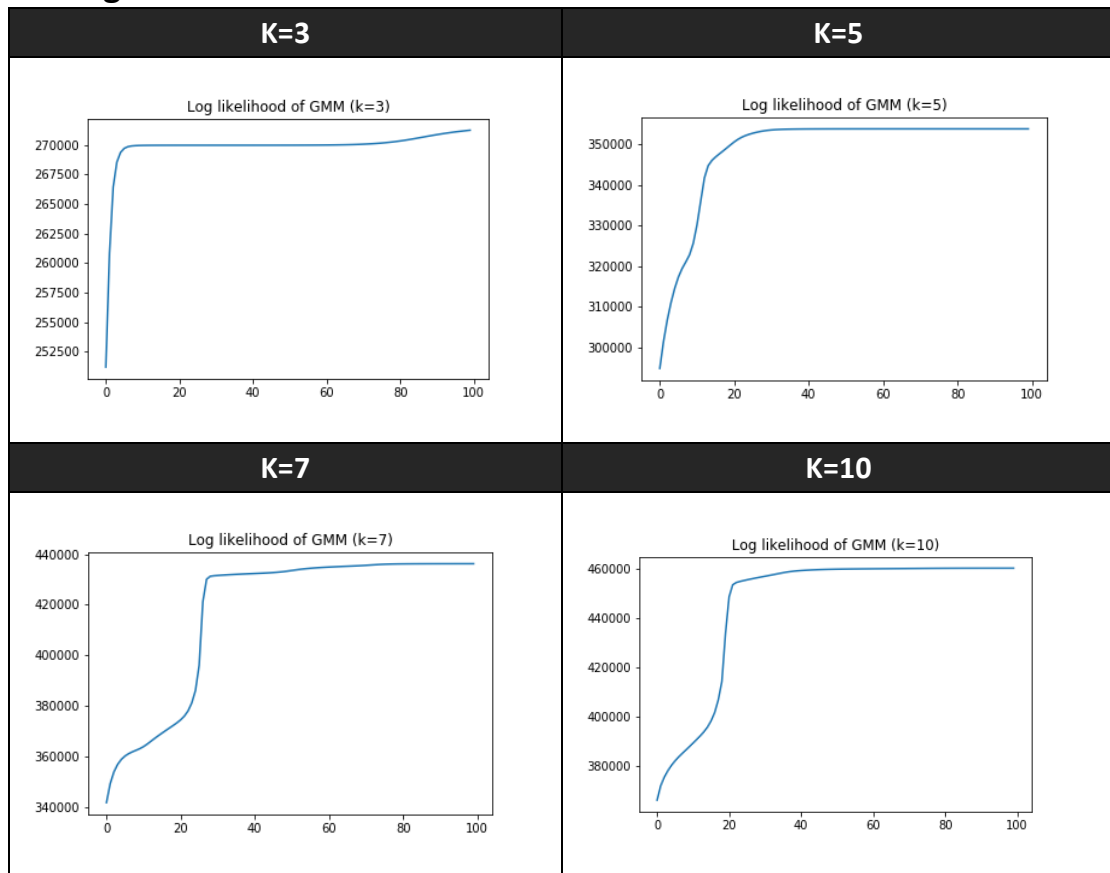


2-5

若實現在作業使用的資料上，線性 kernel 已可以大致將資料做正確分類，而 polynomial kernel 則可以將其用非線性的方法切割。原本預期 polynomial 可以有更少的 support vector 數量，但在印出來的圖上面並沒有辦法有太多的區別，預計應該用其他 degree 嘗試出不一樣的效果。

3-2

The log likelihood curve of GMM



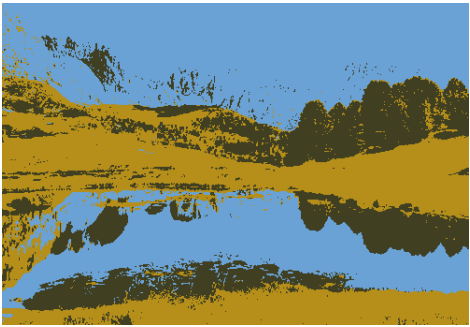

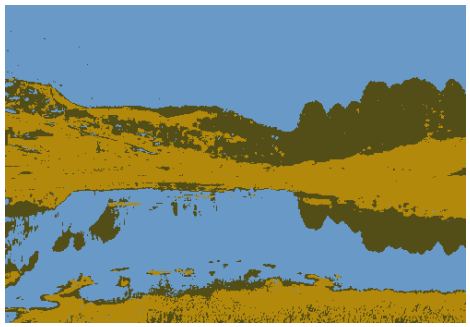
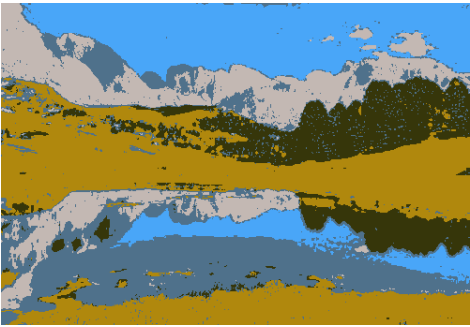


3-3

K = 3, 5, 7, and 10

K=3					K=5						
K-means		GMM			K-means		GMM				
K_means	R	G	B	GMM	R	G	B	GMM	R	G	B
0	181	143	26	0	178	137	11	0	79	113	139
1	106	162	213	1	105	153	199	1	176	136	13
2	64	63	34	2	83	77	24	2	195	184	179
								3	73	166	248
								4	54	54	10

K=7					K=10						
K-means		GMM			K-means		GMM				
K_means	R	G	B	GMM	R	G	B	GMM	R	G	B
0	215	175	19	0	181	139	15	0	112	115	117
1	98	82	22	1	53	54	12	1	48	50	10
2	31	33	9	2	108	85	0	2	202	190	184
3	161	123	24	3	126	112	46	3	169	132	8
4	64	159	243	4	76	166	248	4	21	133	229
5	62	102	146	5	29	124	205	5	75	165	248
6	195	188	184	6	159	158	162	6	76	145	204
								7	124	107	34
								8	108	85	0
								9	196	147	21

The resulting images

K=3	K=5
K-means	K-means
	
GMM	GMM
	
K=7	K=10
K-means	K-means
	
GMM	GMM
