Machine Learning 2020

1. Gaussian Process for Regression

- 根據題目定義，透過 exponential-quadratic kernel function（以下公式）以及題目給定的 theta

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp\left\{ -\frac{\theta_1}{2}||\mathbf{x}_n - \mathbf{x}_m||^2 \right\} + \theta_2 + \theta_3 \mathbf{x}_n^\top \mathbf{x}_m$$

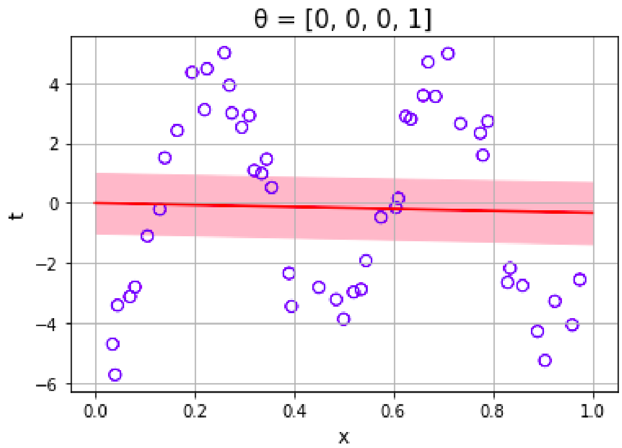
可以算出相對應的 covariance 和 inverse covariance，接著利用課本上的公式（以下公式）

$$\begin{aligned} m(\mathbf{x}_{N+1}) &= \mathbf{k}^{\mathrm{T}}\mathbf{C}_N^{-1}\mathbf{t} \\ \sigma^2(\mathbf{x}_{N+1}) &= c - \mathbf{k}^{\mathrm{T}}\mathbf{C}_N^{-1}\mathbf{k}. \end{aligned}$$

分別帶入 kernel 和 inverse covarianvce 可以求出 mean 跟 variance，有了 mean 與 variance 就可以畫出 predict result（這次使用 one standard deviation）以及算出 training data 和 testing data 的 RMS error。

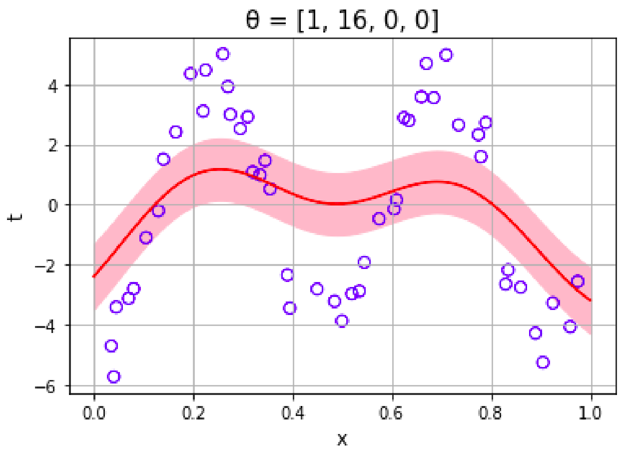
- Result Output :

1. theta = [0, 0, 0, 1]



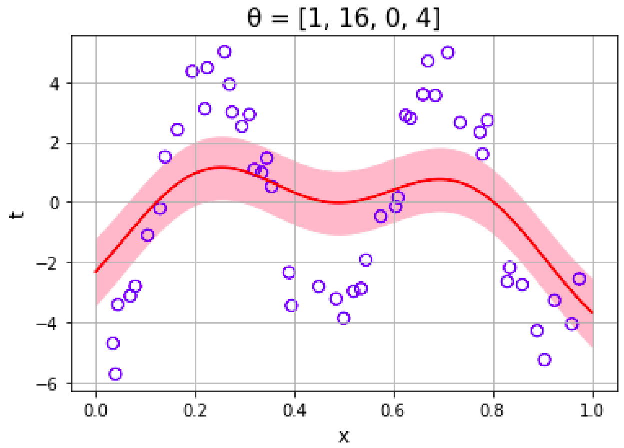

```
RMS error of training data : 3.1292014298222433
RMS error of testing  data : 3.3443986601861146
```

2. theta = [1, 16, 0, 0]



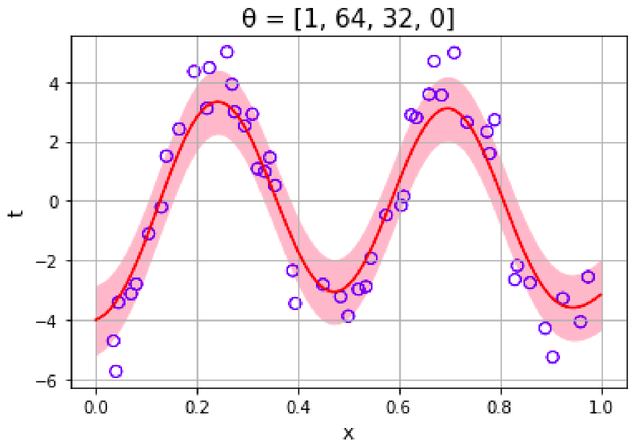

```
RMS error of training data : 2.4239279278312194
RMS error of testing  data : 2.6680517502524466
```

3. theta = [1, 16, 0, 4]



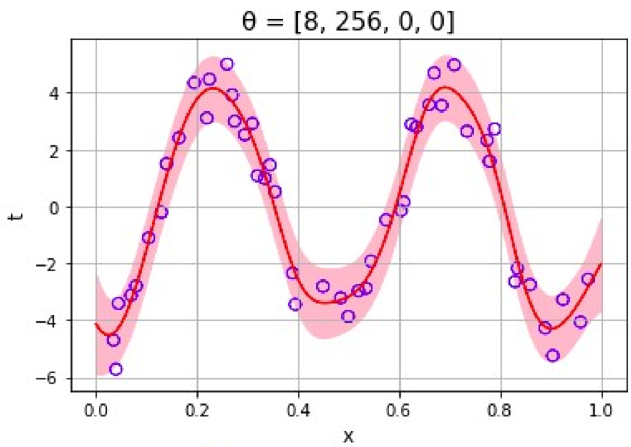

```
RMS error of training data : 2.410576487125207
RMS error of testing  data : 2.6569980001669165
```

4. theta = [1, 64, 32, 0]




```
RMS error of training data : 1.0428861621832217
RMS error of testing  data : 1.1627590936118453
```

5. theta = [8, 256, 0, 0] （trial and error 的結果）



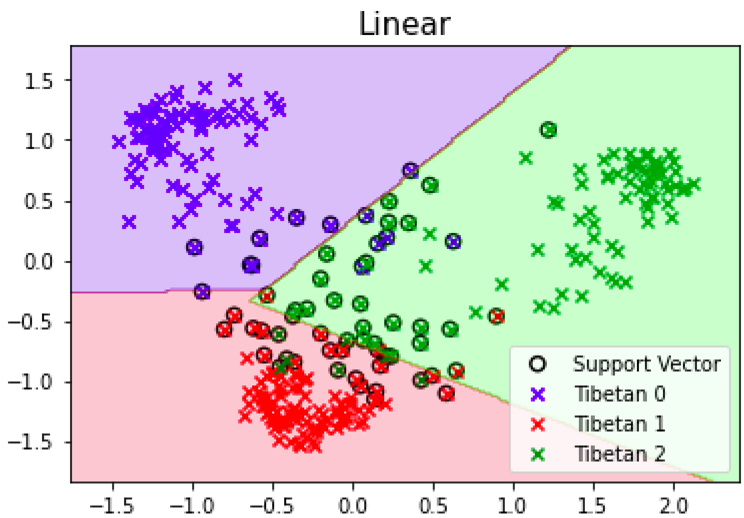

```
RMS error of training data : 0.6727647429593011
RMS error of testing  data : 1.0411559204230687
```

- Explain your findings and do some discussion :
  由結果可以看出第一組 theta 的 RMS error 不論是在 training data 還是 testing data 都相對其他組來得高，原因是它只用到了 $\mathbf{x}_n^\top \mathbf{x}_m$ 的部份，因此複雜度較低，無法良好的 fit data。由第二組和

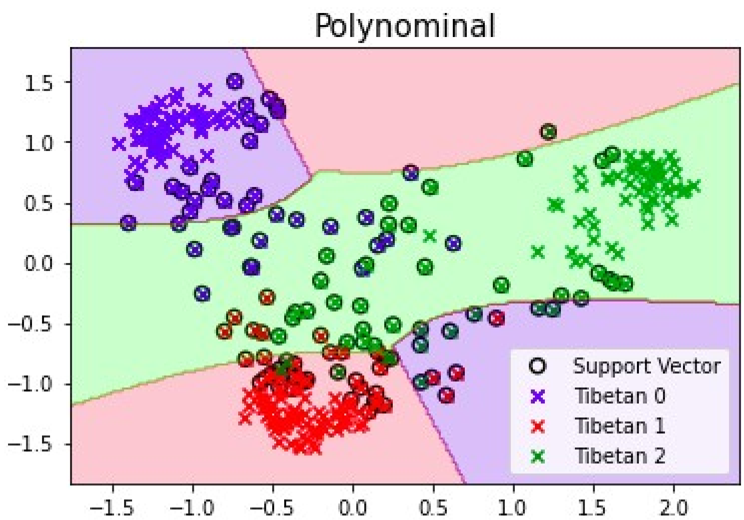
第三組 theta 可以看出兩組的 RMS error 都相似，因此調整 kernel function 中的 theta2 和 theta3 無法帶來顯著的 fitting 效果，所以我在最後一組 trial and error 的 theta 中主要調整的是 theta0 和 theta1，但是當 theta0 或 theta1 調太大時會發生 overfitting 的現象。

2. Support Vector Machine (SVM)

- Analyze the difference between two decision approaches (one-versus-the-rest and one-versus-one). Decide which one you want to use and explain why you choose this approach.
  one-versus-the-rest 和 one-versus-one 都是用來做 mulit-class 的分類，差別在於 one-versus-the-rest 會先將取其中一類當作+1 類，剩下的類別當作-1 類，然後得到一個分類器（decision hyperplane），接著一直做到所有類別都有當作+1 類，最後計算 decision value 哪個比較大，就把資料判給哪類。而 one-versus-one 則是取第 1 類和第 2 類得到一個分類器（decision hyperplane），然後第 1 類和第 3 類得到一個分類器，以此類推，直到所有類別都彼此分類學習過為止，最後判別類別的方式為看此樣本被判為哪一類的次數最多，這個樣本就判給那類。
  在作業中我使用 one-versus-one，原因是 one-versus-the-rest 會有兩種特殊情形需要做額外判定，例如兩類的 decision value 都是正的，或所有類的 decision value 都是負的，但是 one-versus-one 就相對容易且直觀，只須判斷樣本被判別為哪類最多次就好。
- Result Output :
  1. linear kernel



2. polynomial kernel



- linear kernel 分類的結果是用 3 條偏直的線去做切割，而 polynomial kernel 則是用 6 條較平滑的曲線做切割。我認為由 linear kernel 分出來的效果比較好，原因是分錯的類別看起來較少。

3. Gaussian Mixture Model

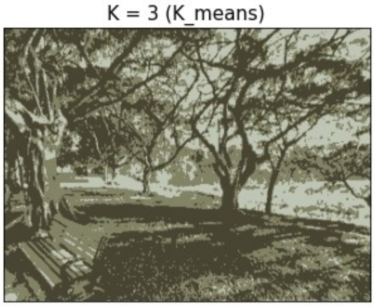
- K-means :

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

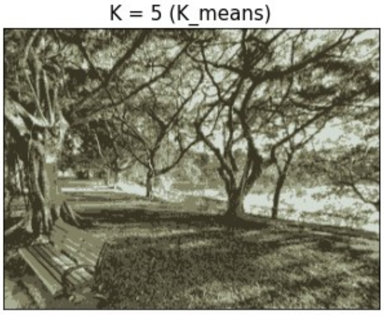
```
============== K = 3 (K_means) ==============
K_means :       R           G           B
0       :  133.042622  126.052325  104.298038
1       :  194.103966  195.771721  182.424996
2       :   73.954632   66.601523   52.842755

============== K = 5 (K_means) ==============
K_means :       R           G           B
0       :   93.636061   86.275496   69.826351
1       :  169.289581  167.959980  147.853204
2       :  133.202325  125.333674  103.057953
3       :   59.287432   52.028229   39.960196
4       :  213.851592  217.285669  210.257006

============== K = 7 (K_means) ==============
K_means :       R           G           B
0       :  159.693862  144.134152  103.724888
1       :  222.220654  225.917431  221.712303
2       :  133.040463  141.464008  141.422519
3       :  183.849133  183.897332  166.209271
4       :   82.934553   75.356402   61.276823
5       :   53.323517   46.382578   34.334622
6       :  115.238474  107.267481   87.405542

============== K = 10 (K_means) ==============
K_means :       R           G           B
0       :  188.578733  195.828142  192.700107
1       :   90.945066   84.098107   69.638626
2       :   44.594025   38.032862   25.750314
3       :   70.019759   62.222684   49.774196
4       :  199.192329  184.717343  130.075875
5       :  231.218728  233.680028  229.579129
6       :  151.906014  159.274768  157.467298
7       :  155.220886  139.248419  103.328910
8       :  119.313522  108.519658   82.901525
9       :  118.687049  124.488493  122.439274
```

  2. Resulting images

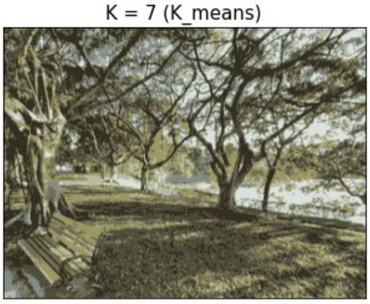
K = 3 (K_means)



K = 5 (K_means)



K = 7 (K_means)



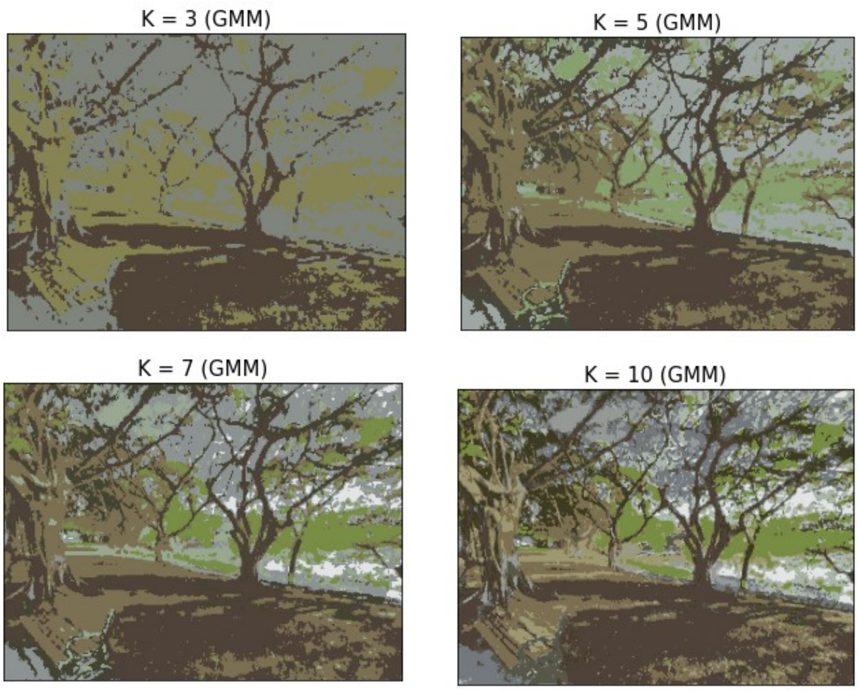
K = 10 (K_means)

- GMM:

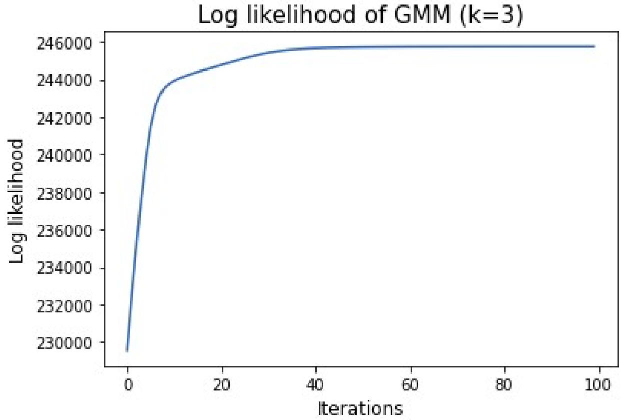
1. Estimated $\{\mu_k\}_{k=1}^{K}$

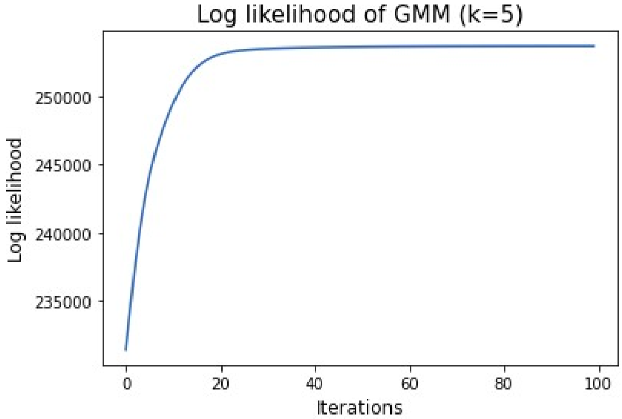
```
============== K = 3 (GMM) ==============
GMM      :       R          G          B
0        :  137.086419  125.350074   87.284775
1        :  128.226743  131.853131  124.523432
2        :   80.297593   67.365795   58.843944

============== K = 5 (GMM) ==============
GMM      :       R          G          B
0        :   74.807811   71.179823   60.243718
1        :  142.004612  151.857162  112.047559
2        :  126.748776  111.333607   85.991130
3        :   79.136895   65.641933   58.420525
4        :  160.663566  166.070501  165.977591

============== K = 7 (GMM) ==============
GMM      :       R          G          B
0        :  127.778992  131.460343   73.811755
1        :  227.822218  228.414917  228.398983
2        :  141.249307  147.030208  147.578251
3        :  156.264012  164.026257  146.415276
4        :   69.370623   65.303385   52.977128
5        :   78.897683   65.593494   58.405930
6        :  125.631374  108.746704   85.858646

============== K = 10 (GMM) ==============
GMM      :       R          G          B
0        :  173.617155  179.129187  170.004794
1        :   88.102079   85.335860   73.312074
2        :   62.868977   55.664040   35.696552
3        :   78.182141   65.313370   58.309823
4        :  126.834010  130.259682   71.632488
5        :  227.901905  228.078734  228.141621
6        :  150.124307  153.773636  151.718401
7        :  161.407784  147.491783  116.452990
8        :  126.377608  107.644005   84.047416
9        :  116.870977  122.468117  126.104567
```

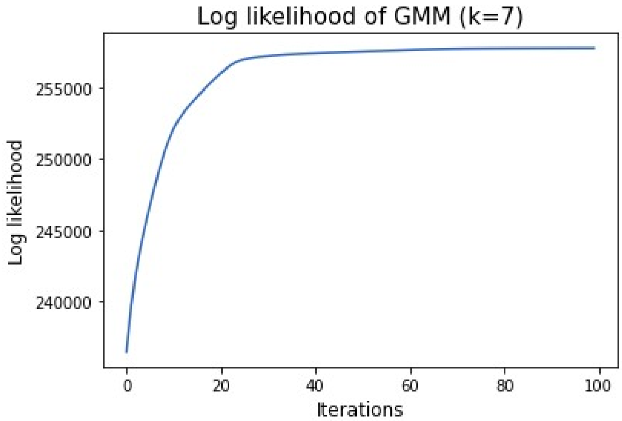
2. Resulting images

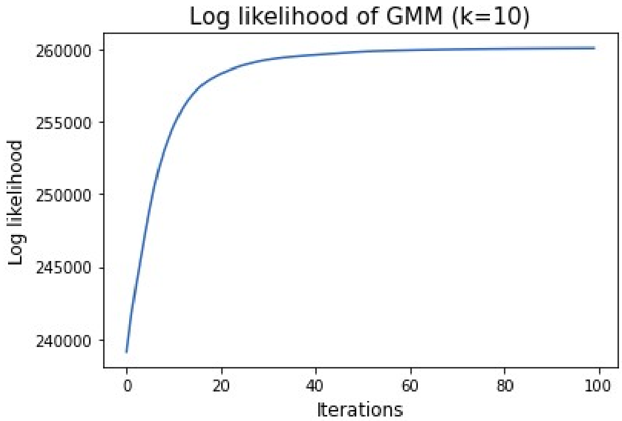


3. Graph of Log likelihood









- 由 K-means 和 GMM（K=10）的結果圖可以發現 K-means 的圖更接近原圖，對比跟銳度較高，會影響兩者圖片的差異，我認為是因為 K-means 會考慮距離，將相近的顏色拉在同一類別，而 GMM 則是從機率的角度出發。