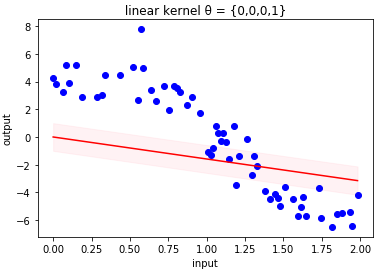
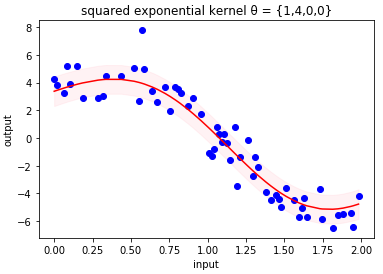
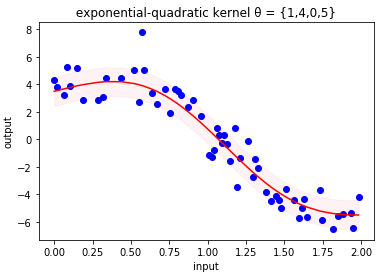
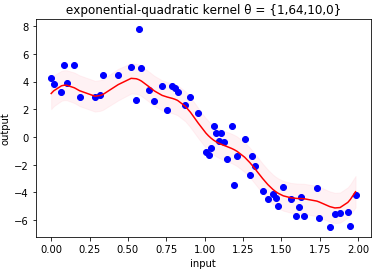
ML lab3

1 Gaussian Process (60%)

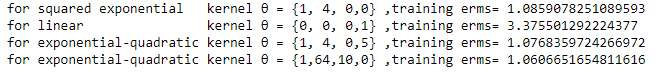
(1) plot the prediction result like Figure 6.8 of textbook for training set but one standard deviation



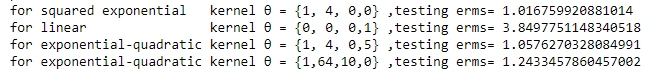
 

(2) the corresponding root-mean-square errors for both training and test sets with respect to four kernels.

For training data



For testing data



心得:

以結果來看，exponential-quadratic的training erms比squared exponential和linear低

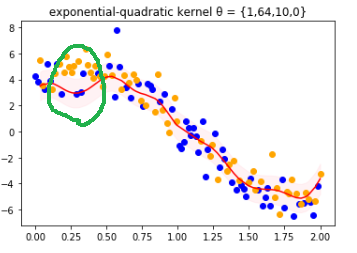
這件事是可以預期的，因為愈複雜的參數的model一定能比較fit training data

但是exponential-quadratic的testing erms比squared exponential高

這是因為training data中參雜著noise，因此太fit training data反而會在predict新資料時比較差，這稱為over-fitting。像是這次lab的，函數圖形已經各種歪七扭八。

下圖為的訓練結果，predict的函數為紅線，training data為藍點，測資為橘點

左邊綠色圈圈是很明顯overfitting的地方



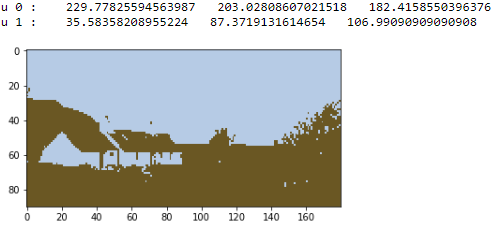
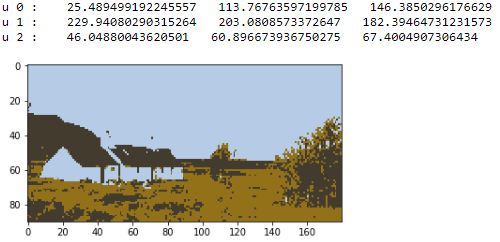
2 Support Vector Machine (40%) 沒做

3 Gaussian Mixture Model (30%)

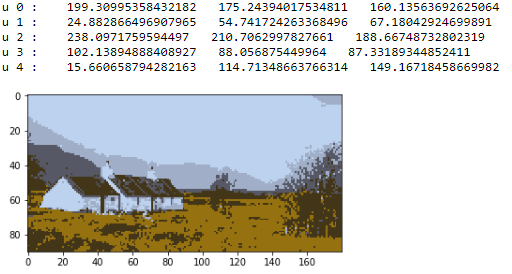
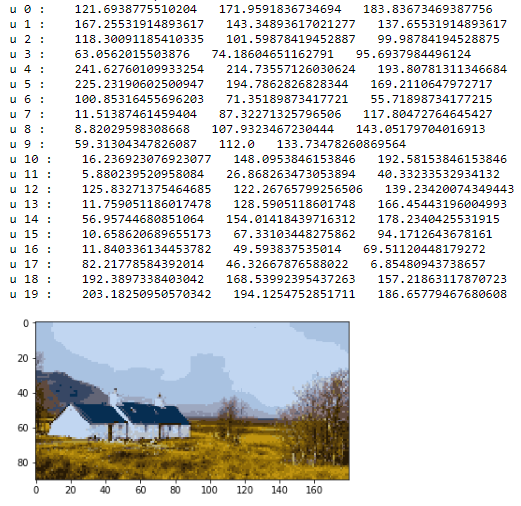
(因為原始圖片太大，助教說可以resize，所以我把圖片縮成90x180)

(1) K-means for K = 2, 3, 5, and 20 respectively

K=2 K=3

K=5 K=20

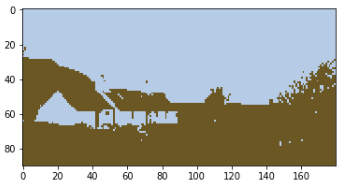
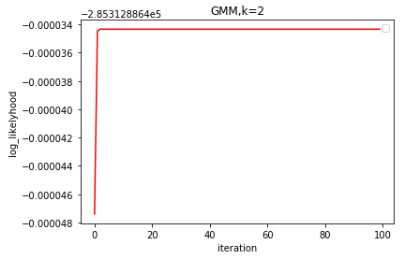
心得:

K-means有個嚴重的問題，因為E step是讓各個點選擇較近的cluster，若是cluster的µk初始的值選得不好會導致每一次都沒有點選擇這個cluster，以上面的k=20為例，往往20個cluster中只有11~13個cluster有分配到點，雖然是k=20，但是出來的segmentation效果只有k=11~13

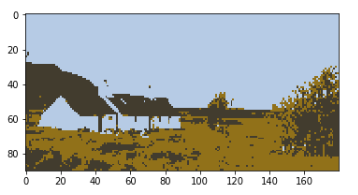
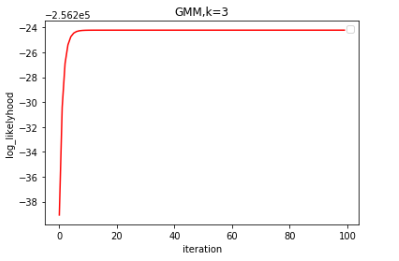
為了解決這個問題，我的作法是每次iteration把沒有點選的cluster的µk以亂數重新產生，實驗結果是跑完10個iteration後，每個cluster都有分配到點所以可以達到預期的segmentation效果

(2) GMM for K = 2, 3, 5, and 20 respectively

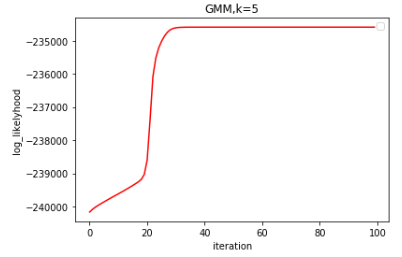
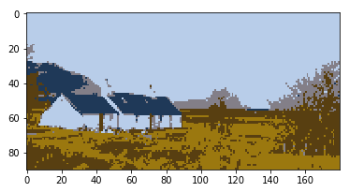
K=2



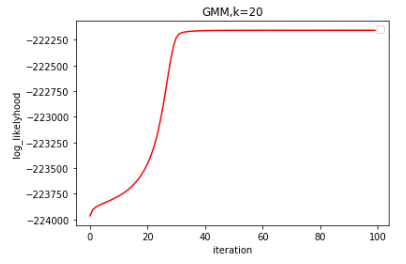
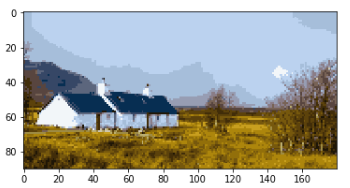
K=3



K=5

K=20

心得:

(1)GMM的運算量比K-means大很多，也因此我把圖片resize過才做

(2)K愈大，需要愈多iteration才能達到local optimal solution

(3)以本次結果來說，我覺得有沒有用GMM其實沒有什麼差，k means就已經有不錯的效果了

原圖 k means GMM

