目錄

Maximum Likelihood (ML)

$$W_{ML} = (\Phi^T\Phi)^{-1}\Phi^T \cdot t$$

$$\hat{y} = \Phi \cdot W_{ML}$$

Introduction

在這次的作業中，我用「R 語言」，並將 40000 筆 raw data 切成 4 等分，利用 30000 筆作為 Training data，另外 10000 筆作為 Validation data。

我將 1081 * 1081 的「**大平面**」切成數個「**正方形 subspaces**」，並在每個 subspace 都建立一個「二元高斯函數」作為我的 basic function（因為我認為地形圖就像是用高斯函數疊起來的，所以選高斯，公式如作業提示）

此外，我嘗試做了三個不同的 basic function 模型，分別為 Model A、Model B、Model C，其中的差異（如下表 1-1、圖 1-1 ～ 圖 1-3）。最後會在 comparison 中比較三種模型的 MSE（Mean Square Error）。

	Model A	Model B	Model C
Subspace 個數	47 * 47	53 * 53	59 * 59
Subspace 邊長	23	50	50
Subspace 之間 是否有 overlap?	否	是 （邊長重疊 30）	是 （邊長重疊 30）
大平面 range	1~1081 （1081 = 47*23）	1~1090	-59 ~ 1150
Subspace range	1~23 24~46 … 1059~1081 共 47 個	1~50 21~70 … 1041~1090 共 53 個	-59 ~ -10 -39 ~ 10 … 1101~1150 共 59 個

表 1-1、Model 差異比較

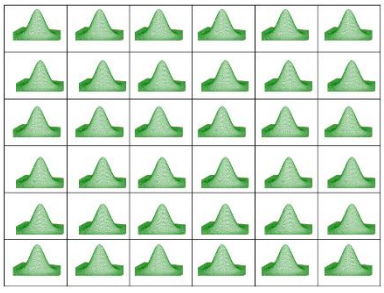


圖 1-1、Model A 示意圖

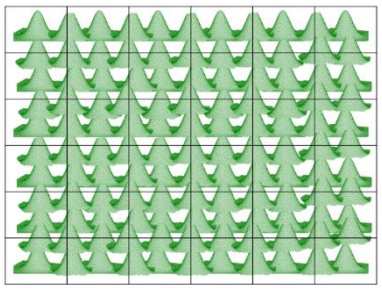


圖 1-2、Model B 示意圖

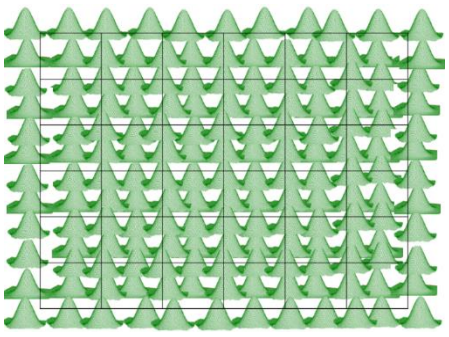
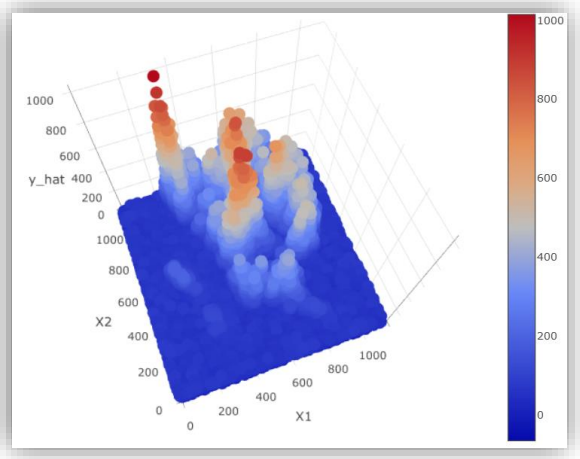


圖 1-3、Model C 示意圖

Models

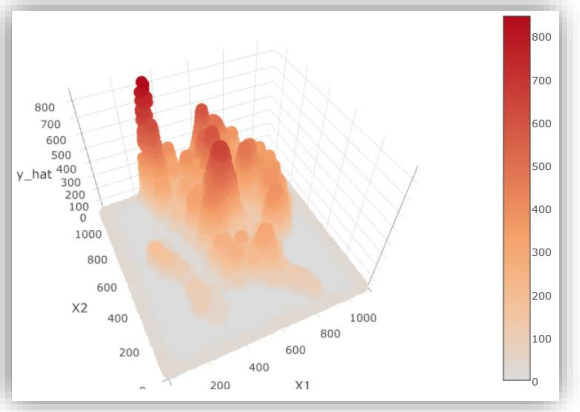
Model A



結果如右圖，雖然看起來與 ground truth data 有點像，但可以發現整個誤差非常大，且在低海拔更為嚴重，甚至還會出現負值。

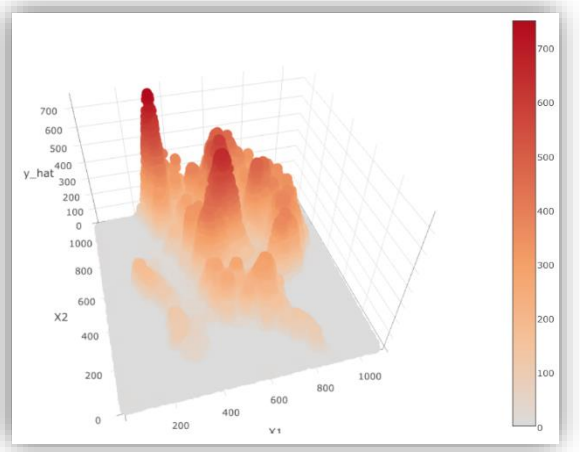
$E(w) = \frac{MSE}{2} = 1088.34$，為一個不理想的 Model。後來覺得可以不用把平面完全分割成獨立的子區域，因此在 Model B 中，將嘗試以 overlap 的方式去建立 basic function $\phi(x)$，此外，還會將算出來的高度做一次檢查，若小於零，則令為零，以降低 MSE。

Model B



結果如右圖，透過 overlap 以及檢測高度是否小於零，這兩個步驟改進了整個模型的準確度，$E(w) = 140.19$，然而，邊界還是有錯誤的高度出現(原本應該為 0)，推測如果再超過邊界的地方加入 basic function，應該可以改進這個問題。

Model C



由右圖可知，邊邊的點幾乎都消失了，$E(w)$ 降至 60.35。反推回去，誤差大概介於 ± 10 之間，應該是個不錯的適配程度。

(MAP、bayesian 出來的 3D 圖都跟他長得差不多，為了節省版面就不放了)

Comparison

	Model A	Model B	Model C
E(w) (Training data)	1088.34	140.19	60.35
E(w) (Validation data)	1638.44	170.48	81.73

* $E(w) = \frac{MSE}{2}$

Model A 將整個平面直接分割，故每個 subspace 並沒有重疊，因此每塊區域只會有一個小山丘，但與現實生活不同（地形不像金字塔一樣，一座一座分得這麼開，通常都會有交疊），且因為沒有做任何限制，所以高度可能會出現負值，E(w)超過 1000，不慎理想。

Model B 改進了以上缺點，將 subspace 加入重疊的部分，並在最後算出 $\hat{y}$ 後，若高度有負值，則直接令為 0，如此一來E(w)可降至 150 左右。然而，在 3D-plot 中可以發現，大平面的邊邊出現高度，這是不合理的（因為 true value = 0），猜想是因為 range 設置在 1~1090，故超過 range 的話，並沒有其他 basic function 可以來平衡高度。

Model C 將大平面 range 增廣為 -59 ~ 1150，如此一來邊邊的高度消失了，E(w)也降至 100 以下，對我來說已經很滿意了，所以就做到這邊，**後面的 MAP、Bayesian 也會採用此 model**。

Maximum a posteriori approach (MAP)

$$W_{MAP} = m_N = \beta \cdot S_N \cdot \Phi^T \cdot t$$

$$\hat{y} = \Phi \cdot W_{MAP}$$

where

$$S_N = (\alpha I + \beta \Phi^T \Phi)^{-1}$$

在這邊沿用 ML 的 Model C，透過改變 (α,β) 來控制 regularization coefficient λ ($\lambda = \frac{\alpha}{\beta}$)，以下比較$(\alpha,\beta)$不同組合下的 E(w)。

由下表可知，隨著 λ 的縮小，training data 的 E(w) 會越來越接近 ML 的，但即使把 λ 設定得很小，進步幅度也有限，最後選擇用$(\alpha,\beta) = (1,1000)$來計算 W_{MAP}（這邊 validation data 的 E(w) 已經比 ML 的小了）。

(α,β)	(1,5)	(1,10)	(1,1000)	(1,1000000)
E(w) (Training data)	69.02	65.78	61.55	60.42
E(w) (Validation data)	89.58	84.83	79.47	77.91

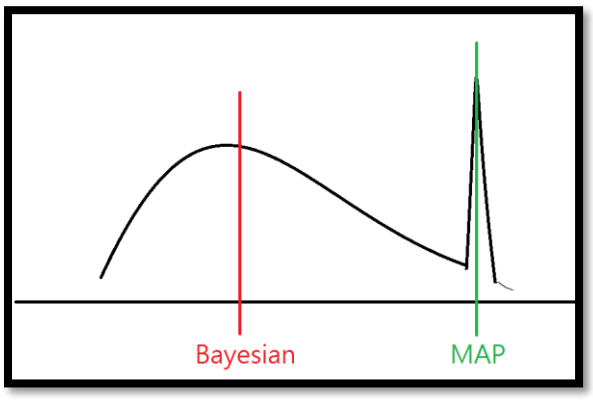
Bayesian approach

$$p(t|x,t,\alpha,\beta) = N(t|m_N^T \cdot \phi(x)\ , \sigma_N^2(x))$$

where

$$m_N = \beta \cdot S_N \cdot \Phi^T \cdot t \quad (\text{令 } m_0 = 0)$$

$$\sigma_N^2(x) = \frac{1}{\beta} + \phi(x)^T S_N \phi(x)$$



相較於點估計的 ML & MAP，Bayesian 考慮所有的 w，透過對 w 積分得到後驗分配的機率分布，並取其平均值。可避免如右上圖這種情況(MAP 會選擇後驗分配中，機率最大的那個，但有時候這個值並不是最佳解)。

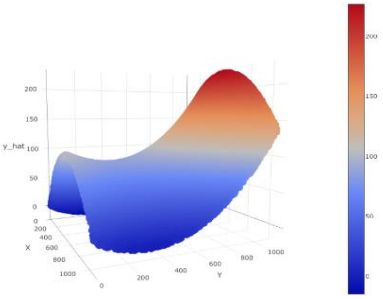
在這邊沿用 ML 的 Model C，以下比較(α,β)不同組合下的 E(w)，基本上與 MAP 的結果差不多，原因是這邊的平均值就是 MAP，只是再加上一些變異量σ_N^2，所以有可能表現會比 MAP 差一點。

(α,β)	(1,5)	(1,10)	(1,1000)	(1,1000000)
E(w) (Training data)	69.03	65.78	61.55	60.42
E(w) (Validation data)	89.60	84.83	79.49	77.91

Discussion

□ Underfitting：

ex：$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2$ 作為 model。得到如右圖的圖形，既不 Fit training data，也不 fit validation data，當然 testing data 也不會準。



□ Overfitting：

若將我在 ML 的 model C 繼續切更細，例如切成 300 * 300，這樣 model 會有 90001 個ϕ(x)，比 data 的量還多，可能會造成 overfitting 的現象（但我電腦效能有限，沒辦法切這麼細）。由於 ML 容易產生 overfitting，所以可以改用 MAP 去避免(因為有 regularization)，但要注意 $\lambda = \frac{\alpha}{\beta}$ 不可太小，否則也可能會發生 overfitting 的現象。

□ 三種方法的比較：

[ML] 找的是讓概似函數最大的那個參數，經過推導後可以得到 close-form 的解 $W_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \cdot t$。

[MAP] 除了概似函數之外，還把 prior 考慮進去，找出讓 posterior 最大的那個參數，在這邊假設 prior 服從$N(w|0, \alpha^{-1}I)$，將均值設成 0，方便計算，經過推導後，一樣可以得到 close-form 的解 $W_{MAP} = m_N = \beta \cdot S_N \cdot \Phi^T \cdot t$。在這題，隨著λ的減小，trianing data 的E(w)會自然地接近 ML 的(詳見公式)，但 validation data 的E(w)會表現得比 ML 來得好。

然而，不論是 ML 或是 MAP，都是屬於點估計的方式，找到最好的 W。

[Bayesian] 的方法則是透過對 w 積分，找出後驗機率的分布情況 $p(t|x, t, \alpha, \beta) = N(t|m_N^T \cdot \phi(x), \sigma_N^2(x))$，所以他的表現跟 MAP 應該會很接近（事實上，機率最高的那個就是 MAP），在這題 Bayesian 的表現略遜於 MAP