

請實做以下兩種不同 feature 的模型，回答第 (1) ~ (3) 題：

- (1) 抽全部 9 小時內的污染源 feature 的一次項(加 bias)
- (2) 抽全部 9 小時內 pm2.5 的一次項當作 feature(加 bias)

備註：

- a. NR 請皆設為 0，其他的數值不要做任何更動
- b. 所有 advanced 的 gradient descent 技術(如: adam, adagrad 等) 都是可以用的

1. (2%)記錄誤差值 (RMSE)(根據 kaggle public+private 分數)，討論兩種 feature 的影響

* Gradient Descent with $10^{(-8)}$ learning rate

* 9 hours features

* 污染源 features: 100000 iterations

pm2.5 features: 30000 iterations

* 我將污染源 features 定義為 SO₂, CO, PM₁₀, PM_{2.5}, NO_x, NO, NO₂, THC, NMHC, CH₄

* 我的 Training Data 和 Validation Data 是從原始的 Training Data 去 random shuffle 然後切除 2/3 為 Training, 1/3 為 Validation

Features \ Error Score	Training	Validation	Public	Private
All Pollution features	5.8487916838	5.98204409851	7.65911	5.57930
Pm2.5	6.11281674618	6.15122550959	7.43985	5.62742

比較可以發現 validation 的 error 都會略高於 Training data 的 error。

Pollution 的含有較多 features 且包含了 Pm2.5，所以在 Training 的表現更好。

無論是 pm2.5 還是 pollution 在 public 和 private 的表現差距都有一點大，private 表現較好，我認為這是由於兩者的 testing 資料在切的時候不夠平均或不夠多，使得 Pollution 在 private 表現好，而 pm2.5 在 public 表現好。

這兩次的 submit 我都沒有做什麼 tuning，因此 private 表現較好而 public 很糟糕，我不認為是 overfitting 所導致。

2. (1%)將 feature 從抽前 9 小時改成抽前 5 小時，討論其變化

* 除了 hours 改成 5 小時，其他參數及 validation 都和第 1 題相同。

Features \ Error Score	Training	Validation	Public	Private
All Pollution features	6.53883596039	6.31968703215	8.06458	5.68668
Pm2.5	6.21701842044	6.22728179451	7.57762	5.79381

和第一題的結果比較，5 小時的 error 都比 9 小時大一些，表示 9 小時的 feature 對於 accuracy 比較有幫助。而 pollution 的 feature 在 training 的 error 比 validation 大，我認為應該是由於該 model 剛好有 fit 到所切的 validation data。

3. (1%)Regularization on all the weight with $\lambda = 0.1$ 、 0.01 、 0.001 、 0.0001 ，並作圖

* 參數和模型都和第一題相同設定，加上 regularization

$\lambda = 0.0001$

Features \ Error Score	Training	Validation
All Pollution features	5.85143877732	5.98225594085
Pm2.5	6.11281674618	6.15122551004

$\lambda = 0.001$

Features \ Error Score	Training	Validation
All Pollution features	5.8486805535	5.98198378618
Pm2.5	6.11281674619	6.15122550787

$\lambda = 0.01$

Features \ Error Score	Training	Validation
All Pollution features	5.83136179518	5.97565268913
Pm2.5	6.11281674621	6.15122548614

$\lambda = 0.1$

Features \ Error Score	Training	Validation
All Pollution features	5.83182971205	5.97801002921
Pm2.5	6.11281674646	6.15122526882

$\lambda = 1$

Features \ Error Score	Training	Validation
All Pollution features	5.83162216839	5.97756707025
Pm2.5	6.1128167497	6.15122309644

$\lambda = 10$

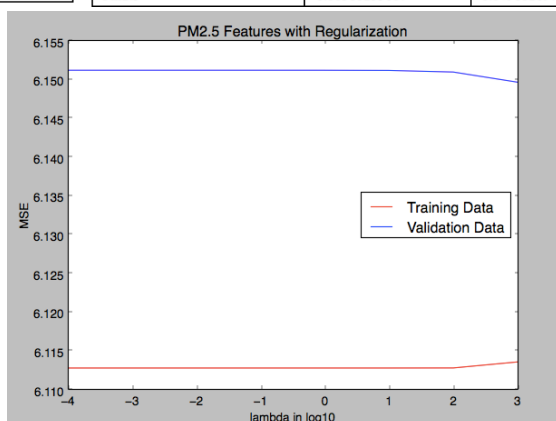
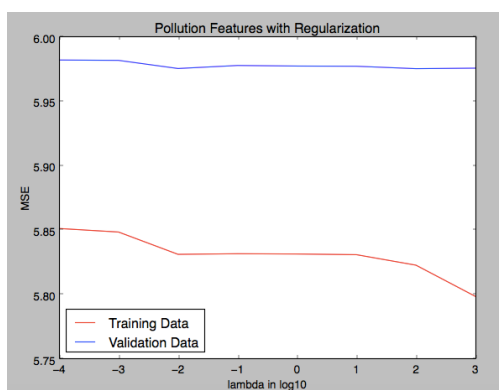
Features \ Error Score	Training	Validation
All Pollution features	5.83109258332	5.97741279883
Pm2.5	6.11281685617	6.15120145481

$\lambda = 100$

Features \ Error Score	Training	Validation
All Pollution features	5.82294653669	5.97552785426
Pm2.5	6.11282528073	6.1509932039

$\lambda = 1000$

Features \ Error Score	Training	Validation
All Pollution features	5.79867939479	5.97593640325
Pm2.5	6.11360235617	6.1496826974



可以發現，在 λ 0.0001~1 之間的 Regularization，performance 都沒有明顯的改變，我認為這是因為 learning rate 很小，乘上去之後對原本的 model 沒有貢獻太多的變化。因此我將 λ 放大到 1000，發現 training data 的 error 有所改變，而 validation 的 error 還是有輕微地下降，表示 regularization 是有一定幫助的。

在 PM2.5 中，regularization 的效果比較明顯，而在 pollution 中比較不明顯，我認為這是因為 pollution 的參數比較多，而 100000 的 iteration 之後仍然沒有到完全的收斂，因此 regularization 過程中反而增加了 validation 一些誤差。

4. (1%) 在線性回歸問題中，假設有 N 筆訓練資料，每筆訓練資料的特徵 (feature) 為一向量 x^n ，其標註 (label) 為一存量 y^n ，模型參數為一向量 w (此處忽略偏權值 b)，則線性回歸的損失函數 (loss function) 為 $\sum_{n=1}^N (y^n - x^n \cdot w)^2$ 。若將所有訓練資料的特徵值以矩陣 $X = [x^1 x^2 \dots x^N]^T$ 表示，所有訓練資料的標註以向量 $y = [y^1 y^2 \dots y^N]^T$ 表示，請問如何以 X 和 y 表示可以最小化損失函數的向量 w ？請寫下算式並選出正確答案。(其中 $X^T X$ 為 invertible)

- (a) $(X^T X)^T y$
- (b) $(X^T X)^T X^T y$
- (c) $(X^T X)^T X^T y$
- (d) $(X^T X)^T X^T y$

$$Loss = \sum_{n=1}^N (y^n - x^n \cdot w)^2$$

$$\text{對 } w_0 \text{ 求導: } \frac{\partial Loss}{\partial w_0} = 2 \sum_{n=1}^N [(y^n - x^n \cdot w) \cdot x^n]$$

$$\text{Let } \frac{\partial Loss}{\partial w_0} = 0 \Rightarrow 2 \sum_{n=1}^N [(y^n - x^n \cdot w) \cdot x^n] = 0$$

$$\Rightarrow \sum_{n=1}^N [y^n \cdot x^n] - \sum_{n=1}^N [x^n \cdot w_0 \cdot x^n] = 0$$

$$\Rightarrow X^T \cdot Y - X^T \cdot X \cdot w_0 = 0 \Rightarrow w_0 = (X^T X)^{-1} X^T Y$$

$$\text{So } \frac{\partial Loss}{\partial w} = 0 \Rightarrow 2 \cdot X^T (Y - X \cdot w) = 0$$

$$\Rightarrow 2 \cdot X^T X \cdot w = 2 X^T Y$$

$$\text{如果 } X^T X \text{ 可逆} \Rightarrow w = (X^T X)^{-1} X^T Y$$

選 (c)