Machine Learning HW1

B09902005 資工三盧冠綸

October 2, 2022

programming part

For this homework, I discussed with B09901133 and B09502019, and they also gave me some hints for the programming parts.

problem 1: 解釋什麼樣的 data preprocessing 可以 improve 你的 training/ testing accuracy。請提供數據 (例如 kaggle public score RMSE) 以佐證你的想法。(1pts)

To prove that my preprocessing is really helpful to improve the prediction accuracy, here I use two variables, E_{test} and E_{out}, where E_{test} is the average RMSE when I do 10-fold cross validation to test it, and E_{out} is its kaggle public score.

I use third steps to preprocess my data. After the four steps, my E_{test} is 3.62216, and my E_{out} is 1.87669.

1. First, I modify all the data value (in `standardize` function). I calculate mean and variance for all features and then transform all values into their Z-score and then plus 2. In fact, this won't affect the result of 1st-order polynomial, but will affect the result of 2nd-order polynomial. With doing this step, My E_{test} is 3.43258, and E_{out} is 3.70323 for 2nd-order polynomial. But without this step, my E_{test} is 11.97816, E_{out} is 13.50383, which is much larger.

2. Second, I use correlation coefficient to choose proper features (in `important_feat` function). I choose a feature only if its correlation coefficient with PM2.5 is larger than 0.8 or smaller than -0.8. If I choose all the features instead, my E_{test} will be 3.24463, and my E_{out} will be 2.60535, which is also larger because of over-fitting.

3. Third, I eliminate some extreme data (in `get_extreme` and `valid` function). If there is an feature whose Z-score is larger than 10 or less than -10, then I will not put any data that contains it into my training data. Without this step, my E_{test} is 3.69597, and my E_{out} is 1.98415, which is larger than when

using this step. And, if I use a tighter bound, that is, eliminate data whose Z-score is larger than 2 or smaller than -2. This way, E_{test} is 3.13312, and my E_{out} is 2.11100, which also has a larger E_{out} than the original one.

problem 2: 請實作 2nd-order polynomial regression model (不用考慮交互項) (1pts)

(a) **貼上 polynomial regression 版本的 Gradient descent code 內容**

Here, instead of changing the code in the gradient descent part, I modify the features of `train_x` instead. If a data $\texttt{train\_x}_i$ is a vector $\begin{bmatrix} x_{i,1} & x_{i,2} & ... & x_{i,n} \end{bmatrix}^T$, then I transform it into $\begin{bmatrix} x_{i,1} & x_{i,2} & ... & x_{i,n} & x_{i,1}^2 & x_{i,2}^2 & ... & x_{i,n}^2 \end{bmatrix}^T$ and then do gradient descent on it. Below is my code to modify the features.

```
def add_squares(x):
  xx = x.copy()
  for i in range (len(x)):
    for j in range (len(x[i])):
      xx[i][j] *= xx[i][j]
  # print(x, x.shape)
  # print(xx, xx.shape)
  xxx = np.concatenate((x,xx), axis=1)
  # print(xxx, xxx.shape)
  return xxx

train_x = add_squares(train_x)
```

And, my code in gradient descent part is same with what TAs provides in the sample code.

(b) **在只使用 NO 數值作爲 feature 的情況下，紀錄該 model 所訓練出的 parameter 數值 (w2, w1, b) 以及 kaggle public score.**

Let mean of NO in our data be m, and let standard deviation of NO be std, then given a data of NO with value $x_1 = \begin{bmatrix} x_8 & x_7 & ...x_1 \end{bmatrix}^T$, and $x_2 = \begin{bmatrix} x_8^2 & x_7^2 & ...x_1^2 \end{bmatrix}^T$, where x_i is the NO value of the i-th day before the day we want to predict.

then due to my preprocessing of data, then we define $x_1' = \begin{bmatrix} z(x_8)+2 & z(x_7)+2 & ...z(x_1)+2 \end{bmatrix}^T$, and $x_2' = \begin{bmatrix} (z(x_8)+2)^2 & (z(x_7)+2)^2 & ...(z(x_1)+2)^2 \end{bmatrix}^T$, where $z(x_i)$ is the Z-score of x_i in the NO data, which is $(x_i - m)/std$. (Since I eliminate all data whose Z-score is larger than 2 or smaller than -2 when preprocessing, so the value of $z(x_i) + 2$ must be larger or equal to 0 when training.)

Then, the prediction of my code is $w_1 x_1' + w_2 x_2' + b$, where $b = 0.6378$, and

$w_2 = \begin{bmatrix} -0.243 & 0.057 & -0.564 & -0.175 & -0.014 & -0.082 & -0.197 & -0.216 \end{bmatrix}^T$

$w_1 = \begin{bmatrix} 0.597 & 0.345 & 0.978 & 1.208 & 0.798 & 0.677 & 1.320 & 1.355 \end{bmatrix}^T$

Using these, the public score on kaggle will be 4.94351.

mathematics part

problem 1: Mathematic background (0.8pts)

(a) According to the definition of positive semi-definite matrix, we want to prove that $x^T(AA^T)x \geq 0\ \forall x \in \mathbb{R}^n$ given any $A \in \mathbb{R}^{n*n}$.

Obviously,
$x^T(AA^T)x = x^T AA^T x = (x^T A)(A^T x) = (A^T x)^T(A^T x) = (A^T x) \cdot (A^T x) \geq 0.$

(b) Since $f(x_1, x_2) = (x_1 \sin x_2)(e^{-x_1 x_2})$, so
$\frac{\partial f}{\partial x_1} = (\sin x_2)(e^{-x_1 x_2}) + (x_1 \sin x_2)(-x_2 e^{-x_1 x_2}) = (1 - x_1 x_2)(\sin x_2)(e^{-x_1 x_2})$
$\frac{\partial f}{\partial x_2} = (x_1 \cos x_2)(e^{-x_1 x_2}) + (x_1 \sin x_2)(-x_1 e^{-x_1 x_2}) =$
$(x_1 e^{-x_1 x_2})(\cos x_2 - x_1 \sin x_2)$

So, $\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} (1 - x_1 x_2)(\sin x_2)(e^{-x_1 x_2}) \\ (x_1 e^{-x_1 x_2})(\cos x_2 - x_1 \sin x_2) \end{bmatrix}$

(c) Note that $P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = p^{\sum x_i}(1-p)^{n - \sum x_i}$, and we want to find p to maximize it, and we do this by finding p to maximize $\log(p^{\sum x_i}(1-p)^{n-\sum x_i}) = (\sum x_i) \log p + (n - \sum x_i) \log(1-p)$.

That is, we want to find p so that $\frac{(\sum x_i)}{p} - \frac{(n - \sum x_i)}{1-p} = 0$. This means $(1-p)(\sum x_i) - p(n - \sum x_i) = (\sum x_i) - np = 0$.

So, $p = \frac{\sum x_i}{n}$.

problem 2: Closed-Form Linear Regression Solution (0.8pts)

(a) Let $L(\theta) = \sum_i \omega_i (y_i - X_i \theta)^2$, then
$L(\theta) = (y - X\theta)^T \Omega (y - X\theta) = y^T \Omega y - 2y^T \Omega X \theta + \theta^T X^T \Omega X \theta$

We want to let $\nabla_\theta L(\theta) = -2y^T \Omega X + 2X^T \Omega X \theta = 0$.

So, $\theta = (X^T \Omega X)^{-1} y^T \Omega X = (X^T \Omega X)^{-1} X^T \Omega y$.

(b) Let $L(\theta) = \sum_i (y_i - X_i\theta)^2 + \lambda \sum_j \omega_j^2$, then $L(\theta) = (y - X\theta)^T(y - X\theta) + \lambda\theta^T\theta$

We want to let $\nabla_\theta L(\theta) = -2y^T X + 2X^T X\theta + 2\lambda\theta = 0$.

So, $\theta = (X^T X - \lambda I)^{-1} y^T X = (X^T X - \lambda I)^{-1} X^T y$.

problem 3: Logistic Sigmoid Function and Hyperbolic Tangent Function (0.8pts)

(a) $2\sigma(2a) - 1 = \frac{2}{1+e^{-2a}} - 1 = \frac{1-e^{-2a}}{1+e^{-2a}} = \frac{e^a - e^{-a}}{e^a + e^{-a}} = \tanh a$

(b) .

$$\begin{aligned} y(x, \mathbf{u}) &= u_0 + \Sigma_{j=1}^M u_j \tanh(\tfrac{x-\mu_j}{2s}) \\ &= u_0 + \Sigma_{j=1}^M u_j (2\sigma(\tfrac{x-\mu_j}{s}) - 1) \\ &= u_0 + (\Sigma_{j=1}^M 2u_j \sigma(\tfrac{x-\mu_j}{s})) - (\Sigma_{j=1}^M u_j) \end{aligned}$$

So, a linear combination of logistic sigmoid functions of the form $y(x, \mathbf{w})$ is equivalent to a linear combination of tanh functions of the form $y(x, \mathbf{u})$, where $w_0 = u_0 - \Sigma_{j=1}^M u_j$, and $w_i = 2u_i \ \forall 1 \le i \le M$.

problem 4: Noise and Regulation (0.8pts)

$$\begin{aligned} \bar{L}_{ss}(\mathbf{w}, b) &= \mathbb{E}[\tfrac{1}{2N} \textstyle\sum_{i=1}^N (f_{\mathbf{w},b}(\mathbf{x}_i + \eta_i) - y_i)^2] \\ &= \mathbb{E}[\tfrac{1}{2N} \textstyle\sum_{i=1}^N (\mathbf{w}^T \mathbf{x}_i + \mathbf{w}^T \eta_i + b - y_i)^2] \\ &= \tfrac{1}{2N} \mathbb{E}[\textstyle\sum_{i=1}^N ((f_{\mathbf{w},b}(\mathbf{x}_i) - y_i) + (\mathbf{w}^T \eta_i))^2] \\ &= \tfrac{1}{2N} \mathbb{E}[\textstyle\sum_{i=1}^N (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2 + 2 * (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i) * (\mathbf{w}^T \eta_i) + (\mathbf{w}^T \eta_i)^2] \\ &= \tfrac{1}{2N} \mathbb{E}[\textstyle\sum_{i=1}^N (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2] + \tfrac{1}{N} \mathbb{E}[\textstyle\sum_{i=1}^N (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i) * (\mathbf{w}^T \eta_i)] + \tfrac{1}{2N} \mathbb{E}[\textstyle\sum_{i=1}^N (\mathbf{w}^T \eta_i)^2] \end{aligned}$$

(i) Note that $\frac{1}{2N} \mathbb{E}[\sum_{i=1}^N (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2] = \frac{1}{2N} \sum_{i=1}^N (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2$

(ii) And, since η_i is independent with $f_{\mathbf{w},b}(\mathbf{x}_i) - y_i$, and $\mathbb{E}[\eta_{i,j}] = 0 \ \forall i, j$, so

$$\begin{aligned} &\tfrac{1}{N} \mathbb{E}[\textstyle\sum_{i=1}^N (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i) * (\mathbf{w}^T \eta_i)] \\ &= \tfrac{1}{N} \textstyle\sum_{i=1}^N \mathbb{E}[(f_{\mathbf{w},b}(\mathbf{x}_i) - y_i) * (\mathbf{w}^T \eta_i)] \\ &= \tfrac{1}{N} \textstyle\sum_{i=1}^N (\mathbb{E}[f_{\mathbf{w},b}(\mathbf{x}_i) - y_i] * \mathbb{E}[\mathbf{w}^T \eta_i]) \\ &= \tfrac{1}{N} \textstyle\sum_{i=1}^N (\mathbb{E}[f_{\mathbf{w},b}(\mathbf{x}_i) - y_i] * (\textstyle\sum_{j=1}^k \mathbb{E}[\mathbf{w}_j \eta_{i,j}])) \\ &= 0 \end{aligned}$$

(iii) And, we also have $\frac{1}{2N}\mathbb{E}[\sum_{i=1}^{N}(\mathbf{w}^T\eta_i)^2] = \frac{1}{2N}\sum_{i=1}^{N}\mathbb{E}[\sum_{j=1}^{k}(\mathbf{w}_j\eta_{1,j})^2]$
$= \frac{1}{2N}\sum_{i=1}^{N}\mathbb{E}[\sum_{j=1}^{k}\sigma^2(\mathbf{w}_j)^2] = \frac{1}{2N}\sum_{i=1}^{N}\sigma^2\left\|\mathbf{w}\right\|^2 = \frac{\sigma^2}{2}\left\|\mathbf{w}\right\|^2$

By combining (i) and (ii) and (iii), we can finally derive that

$$\bar{L}_{ss}(\mathbf{w}, b) = \text{(i)} + \text{(ii)} + \text{(iii)}$$
$$= \frac{1}{2N}\sum_{i=1}^{N}(f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2 + \frac{\sigma^2}{2}\left\|\mathbf{w}\right\|^2$$

problem 5: Logistic Regression (0.8pts)

(a) $p(C_1|x) = \sigma(\mathbf{w}^T\mathbf{x}+b) = \sigma((-1)*7+2*0+(-1)*3+5*10+3) = \sigma(43) = \frac{1}{1+e^{-43}}$, which is very close to 1, so the prediction will be C_1.

(b) First, we. have $p(\mathbf{y}|\mathbf{x}) = \Pi_{i=1}^{N}p(y_i|\mathbf{x}_i) = \Pi_{i=1}^{N}(y_i f_{\mathbf{w},b}(\mathbf{x}_i) + (1-y_i)(1-f_{\mathbf{w},b}(\mathbf{x}_i)))$

Then, we can calculate $L(\mathbf{w}, b)$.

$$L(\mathbf{w}, b) = -\log p(\mathbf{y}|\mathbf{x})$$
$$= -\log\left(\Pi_{i=1}^{N}(y_i f_{\mathbf{w},b}(\mathbf{x}_i) + (1-y_i)(1-f_{\mathbf{w},b}(\mathbf{x}_i)))\right)$$
$$= \sum_{i=1}^{N}(-\log\left(y_i f_{\mathbf{w},b}(\mathbf{x}_i) + (1-y_i)(1-f_{\mathbf{w},b}(\mathbf{x}_i))\right))$$
$$= \sum_{i=1}^{N}(-y_i\log f_{\mathbf{w},b}(\mathbf{x}_i) - (1-y_i)\log(1-f_{\mathbf{w},b}(\mathbf{x}_i)))$$

(c) The answer will be $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla L(\mathbf{w}) = \begin{bmatrix}\frac{\partial L(\mathbf{w},b)}{\partial \mathbf{w_1}} & \frac{\partial L(\mathbf{w},b)}{\partial \mathbf{w_2}} & \dots & \frac{\partial L(\mathbf{w},b)}{\partial \mathbf{w_n}}\end{bmatrix}^T$.

Then, we can find the value of $\frac{\partial L(\mathbf{w},b)}{\partial \mathbf{w_i}}$ for any i:

$$\frac{\partial L(\mathbf{w},b)}{\partial \mathbf{w}_i}$$
$$= \frac{\partial \sum_{j=1}^{N}(-y_j\log f_{\mathbf{w},b}(\mathbf{x}_j) - (1-y_j)\log(1-f_{\mathbf{w},b}(\mathbf{x}_j)))}{\partial \mathbf{w}_i}$$
$$= \sum_{i=j}^{N}(-y_j)\left(\frac{\partial \log f_{\mathbf{w},b}(\mathbf{x}_j)}{\partial \mathbf{w}_i}\right) - \sum_{i=j}^{N}(1-y_j)\left(\frac{\partial \log(1-f_{\mathbf{w},b}(\mathbf{x}_j))}{\partial \mathbf{w}_i}\right)$$

And, $\frac{\partial \log f_{\mathbf{w},b}(\mathbf{x}_j)}{\partial \mathbf{w}_i} = \frac{\partial \log\sigma(\mathbf{w}^T\mathbf{x}_j+b)}{\partial \mathbf{w}_i} = \frac{\partial \log\sigma(\mathbf{w}^T\mathbf{x}_j+b)}{\partial(\mathbf{w}^T\mathbf{x}_j+b)}\frac{\partial(\mathbf{w}^T\mathbf{x}_j+b)}{\partial \mathbf{w}_i}$
$= (1-\sigma(\mathbf{w}^T\mathbf{x}_j+b))\mathbf{x}_{j,i}$

And, $\frac{\partial \log(1-f_{\mathbf{w},b}(\mathbf{x}_j))}{\partial \mathbf{w}_i} = \frac{\partial \log(1-\sigma(\mathbf{w}^T\mathbf{x}_j+b))}{\partial \mathbf{w}_i} = \frac{\partial \log(1-\sigma(\mathbf{w}^T\mathbf{x}_j+b))}{\partial(\mathbf{w}^T\mathbf{x}_j+b)}\frac{\partial(\mathbf{w}^T\mathbf{x}_j+b)}{\partial \mathbf{w}_i}$
$= -\sigma(\mathbf{w}^T\mathbf{x}_j+b)\mathbf{x}_{j,i}$

So,
$\frac{\partial L(\mathbf{w},b)}{\partial \mathbf{w}_i} = \sum_{j=1}^{N}(-y_j)(1-\sigma(\mathbf{w}^T\mathbf{x}_j+b))\mathbf{x}_{j,i} - \sum_{j=1}^{N}(1-y_j)(-\sigma(\mathbf{w}^T\mathbf{x}_j+b)\mathbf{x}_{j,i})$,
which is equal to $\sum_{j=1}^{N}(-y_j\mathbf{x}_{j,i} + \sigma(\mathbf{w}^T\mathbf{x}_j+b)\mathbf{x}_{j,i}) = \sum_{j=1}^{N}\mathbf{x}_{j,i}(f_{\mathbf{w},b}(\mathbf{x}_j) - y_j)$

So, we can derive the answer:

$$
\begin{aligned}
\mathbf{w}^{(t+1)} &= \mathbf{w}^{(t)} - \eta \nabla L(\mathbf{w}) \\
&= \mathbf{w}^{(t)} - \eta \begin{bmatrix} \frac{\partial L(\mathbf{w},b)}{\partial \mathbf{w_1}} & \frac{\partial L(\mathbf{w},b)}{\partial \mathbf{w_2}} & \dots & \frac{\partial L(\mathbf{w},b)}{\partial \mathbf{w_n}} \end{bmatrix}^T \\
&= \mathbf{w}^{(t)} - \eta \begin{bmatrix} \sum_{j=1}^{N} \mathbf{x}_{j,1}(f_{\mathbf{w},b}(\mathbf{x}_j) - y_j) & \dots & \sum_{j=1}^{N} \mathbf{x}_{j,n}(f_{\mathbf{w},b}(\mathbf{x}_j) - y_j) \end{bmatrix}^T \\
&= \mathbf{w}^{(t)} - \eta \sum_{j=1}^{N} \mathbf{x}_j(f_{\mathbf{w},b}(\mathbf{x}_j) - y_j)
\end{aligned}
$$