4721 hw3

Yuning Ding yd2617

March 2022

1 Ridge regression and gradient descent

Problem 1.1

It can be proved since

$$(A^\top A)^\top = A^\top (A^\top)^\top = A^\top A. \tag{1}$$

It implies that all eigenvalues are real and non-negative.

Problem 1.2

Let $\mathbf{v}$ be the unit eigenvector, we have

$$A^\top A\mathbf{v} = \lambda \mathbf{v}. \tag{2}$$

We can further deduce that

$$\lambda = \lambda \mathbf{v}^\top \mathbf{v} = \mathbf{v}^\top A^\top A\mathbf{v} = (A\mathbf{v})^\top A\mathbf{v} \geq 0 \tag{3}$$

Problem 1.3

Let $\mathbf{v}_i$ be one of the eigenvectors of $A^\top A + \lambda I$, we have

$$\begin{aligned} (A^\top A + \lambda I)\mathbf{v}_i &= \gamma_i \mathbf{v}_i \\ A^\top A\mathbf{v}_i &= (\gamma_i - \lambda)\mathbf{v}_i \end{aligned} \tag{4}$$

where γ_i is one of the eigenvalue of $A^\top A + \lambda I$, in which $i = 1, 2, \cdots, d$.

Let λ_i be one of the eigenvalues of $A^\top A$. We have

$$\gamma_i = \lambda_i + \lambda. \tag{5}$$

Problem 1.4

Let $\det(A^\top A - \lambda I) = 0$ and $\det(A^\top A - (\lambda + 1)I) = 0$, we can get the eigenvalues of $A^\top A$ is $\lambda_1 = 0$, $\lambda_2 = 4$, $\lambda_3 = 7$. And the eigenvalues of $A^\top A + I$ is $\lambda_1 = 1$, $\lambda_2 = 5$, $\lambda_3 = 8$, which verifies the properties from the previous problems.

Problem 1.5

For every $i = 1, 2, \cdots, n$, we have

$$(A^\top A + \lambda I)\mathbf{v}_i = (\lambda_i + \lambda)\mathbf{v}_i \tag{6}$$

Since $A^\top A + \lambda I$ is the symmetric matrix, it is invertible. We have

$$(A^\top A + \lambda I)^{-1}\mathbf{v}_i = \frac{1}{\lambda_i + \lambda}\mathbf{v}_i, \tag{7}$$

which means that the eigenvalue of $(A^\top A + \lambda I)^{-1}$ is $\frac{1}{\lambda_i+\lambda}$ and the eigenvector of $(A^\top A + \lambda I)^{-1}$ is $\mathbf{v}_i$. $(A^\top A + \lambda I)^{-1}$ is also symmetric, so we have

$$(A^\top A + \lambda I)^{-1} = \sum_{i=1}^{d} \frac{1}{\lambda_i + \lambda}\mathbf{v}_i\mathbf{v}_i^\top \tag{8}$$

The orthonormal basis of $A^\top A$ is

$$\begin{aligned}
\mathbf{v}_1 &= (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0)^\top \\
\mathbf{v}_2 &= (\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}})^\top \\
\mathbf{v}_3 &= (-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})^\top,
\end{aligned} \tag{9}$$

with corresponding eigenvalue $\lambda_1 = 0$, $\lambda_2 = 4$, $\lambda_3 = 7$.

By calculation, we find that

$$(A^\top A + \lambda I)^{-1} = \begin{pmatrix} 0.575 & -0.425 & 0.025 \\ -0.425 & 0.575 & 0.025 \\ 0.025 & 0.025 & 0.175 \end{pmatrix} = \sum_{i=1}^{d} \frac{1}{\lambda_i + \lambda}\mathbf{v}_i\mathbf{v}_i^\top \tag{10}$$

Problem 1.6

The gradient of J is

$$\frac{\partial J}{\partial w} = 2(A^\top A + \lambda I)\mathbf{w} - 2A^\top \mathbf{b} \tag{11}$$

Problem 1.7

The gradient of J can be also written as

$$\frac{\partial J}{\partial w} = 2(A^\top (A\mathbf{w} - b) + \lambda \mathbf{w}). \tag{12}$$

Then the update would be

$$\mathbf{w}^t = \mathbf{w}^{(t-1)} - \eta \frac{\partial J}{\partial \mathbf{w}^{(t-1)}} = \mathbf{w}^{(t-1)} - 2(A^\top (A\mathbf{w}^{(t-1)} - b) + \lambda \mathbf{w}^{(t-1)}) \tag{13}$$

Therefore, the pseudocode can be written as

Algorithm 1 Update(w, x[i], y[i], lambda, eta)

```
error = zero(n)
ridge = w
grad = zero(d)
for i = 1 to n do
  error[i] = ip(x[i], w) - y[i]
end for
for j = 1 to d do
  for i = 1 to n do
    grad[j] = grad[j] + x[i][j] * error[i]
  end for
end for
scale(ridge, lambda)
add(grad, 1, ridge)
add(w, -2*eta, grad)
```

Problem 1.8

Initialization needs $O(\max(n, d))$ time. The first loop needs $O(nd)$ time. The next nested loop needs $O(nd)$ time. Hence the running time of gradient descent is $O(nd)$.

Problem 1.9

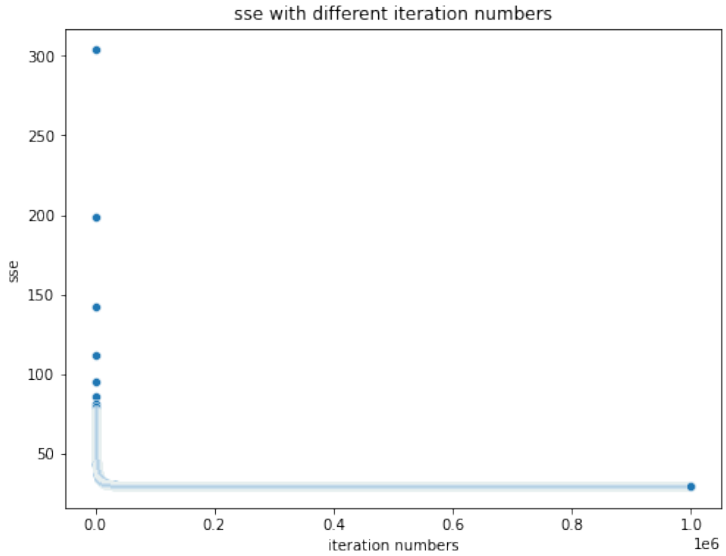
(a)

The coefficients in the weight vector are [0.57464973, 0.6242766, -0.0187258, 0.14278682, 0.73856301, -0.20647887, -0.01632126, 0.00923619]. The intercept term is 0.29332229.

(b)

The SSE is 29.430.

(c)



2 Cost-weighted logarithmic losses

Problem 2.1

To prove (2) we first prove the following lemma

$$\ln x \leq x - 1. \tag{14}$$

Let $g(x) = 1 + \ln x - x$, we have

$$g'(x) = \frac{1}{x} - 1. \tag{15}$$

Let (15) equals to zero, we find that when $x = 1$, $g(1) = 0$ is the minimum value of $g(x)$, which means

$$g(x) \leq 0 \longrightarrow \ln x \leq x - 1. \tag{16}$$

Let $x = \frac{t}{t_0}$, by (14) we have

$$\ln \frac{t}{t_0} \leq \frac{t}{t_0} - 1 = -f'(t_0) \times (t - t_0), \tag{17}$$

since $f'(t_0) = \frac{1}{t_0}$.

Problem 2.2

$R(P)$ can be rewritten as

$$\begin{aligned} R(p) &= \mathbb{E}[c(Y) \times l_{\log}(Y, p(\vec{X})] \\ &= -c(0)\Pr(Y=0|\vec{X}=\vec{x})\log(1-p(\vec{x})) - c(1)\Pr(Y=1|\vec{X}=\vec{x})\log(p(\vec{x})) \\ &= -c(0)(1-\eta(\vec{x}))\log(1-p(\vec{x})) - c(1)\eta(\vec{x})\log(p(\vec{x})). \end{aligned} \tag{18}$$

Compute $R'(p)$ and set it to zero

$$R'(p) = \frac{c(0)(1-\eta(\vec{x}))}{1-p(\vec{x})} - \frac{c(1)\eta(\vec{x})}{p(\vec{x})} = 0 \tag{19}$$

We have

$$\ln \frac{p(\vec{x})}{1-p(\vec{x})} = \ln \frac{c_{FN}\eta(\vec{x})}{c_{FP}(1-\eta(\vec{x}))}. \tag{20}$$

Problem 2.3

If $p(\vec{x}) > \frac{1}{2}$, we have

$$\frac{p(\vec{x})}{1-p(\vec{x})} > 1. \tag{21}$$

Therefore, we have

$$\frac{c_{FN}\eta(\vec{x})}{c_{FP}(1-\eta(\vec{x}))} = \frac{p(\vec{x})}{1-p(\vec{x})} > 1, \tag{22}$$

i.e.

$$\eta(\vec{x}) > \frac{c_{FP}}{c_{FN}+c_{FP}}. \tag{23}$$

Problem 2.4

The step size is 0.03 and the number of iterations is 5000. I choose 0.03 as the step size because I hope the step size could be larger to make the loss function descend to the minimum as fast as possible. I tried to run the code as long as possible and find that the loss function would be converged around 108.90. When the number of iterations is 5000, the loss would be equal to 108.906685. The final number of false positive mistakes is 34 and the number of false negative mistakes is 0.

Problem 2.5

subinterval	fraction
$[0, 0.1)$	0
$[0.1, 0.2)$	0
$[0.2, 0.3)$	0
$[0.3, 0.4)$	0
$[0.4, 0.5)$	0
$[0.5, 0.6)$	0.2
$[0.6, 0.7)$	0
$[0.7, 0.8)$	0
$[0.8, 0.9)$	0.273
$[0.9, 1]$	0.735

Table 1: slopes and intercepts of affine function of each feature

3 Classifying restaurant reviews

Problem 3.1

The amount of memory is estimated as 29997125000 bytes.

Problem 3.2

The training error rate is 0.133159. The test error rate is 0.135829.

Problem 3.3

Top 10 highest weights: ['perfection', 'gem', 'incredible', 'heaven', 'superb', 'phenomenal', 'amazing', 'worried', 'heavenly', 'perfect']

Top 10 lowest weights: ['mediocre', 'worst', 'meh', 'disappointing', 'lacked', 'underwhelmed', 'flavorless', 'bland', 'poisoning', 'disgusting'].

Problem 3.4

Since we predict true if $\frac{1}{n}\vec{w} \cdot \vec{x} > 0$, we would also predict true if $\vec{w} \cdot \vec{x} > 0$. Multiply n would not change whether $\frac{1}{n}\vec{w} \cdot \vec{x}$ is greater than zero.

Problem 3.5

The training error rate is 0.104548. The test error rate is 0.106856.

Problem 3.6

Top 10 highest weights: ['perfection', 'perfect', 'incredible', 'perfectly', 'gem', 'fantastic', 'delicious', 'amazing', 'excellent', 'disappoint']

Top 10 lowest weights: ['worst', 'mediocre', 'bland', 'meh', 'disappointing', 'awful', 'horrible', 'terrible', 'lacked', 'flavorless']

Problem 3.7

I use 2 passes through the training data and use averaging perceptron.

The training error rate is 0.103399, and the test error rate is 0.105782, which is better than the one in problem 3.5.