417T Homework 3

Matthew DeSantis

2023-10-16

1

(a) LFD 4.8

When $\Gamma = 1$,

$E_{aug}(w) = E_{in}(w) + \lambda w^T w$

Take derivative with respect to w

$\nabla E_{aug}(w) = \nabla E_{in}(w) + \nabla \lambda w^T w$
$\nabla E_{aug}(w) = \nabla E_{in}(w) + 2\lambda w$

The update rule is thus:
$w(t+1) = w(t) - \eta \nabla E_{aug}(w(t))$
$w(t+1) = w(t) - \eta(\nabla E_{in}(w(t)) + 2\lambda w(t))$
$w(t+1) = w(t) - \eta \nabla E_{in}(w(t)) - 2\eta\lambda w(t)$
$w(t+1) = (1 - 2\eta\lambda)w(t) - \eta \nabla E_{in}(w(t))$

(b)

$E_{aug}(w) = E_{in}(w) + ||w||$

Take derivative with respect to w

$\nabla E_{aug}(w) = \nabla E_{in}(w) + \nabla \lambda ||w||$
$\nabla E_{aug}(w) = \nabla E_{in}(w) + \lambda sign(w)$
note that sign(x) returns 1 if x>0, -1 if x<0, and 0 if x=0.

The update rule is thus:
$w(t+1) = w(t) - \eta \nabla E_{aug}(w(t))$
$w(t+1) = w(t) - \eta(\nabla E_{in}(w) + \lambda sign(w))$
$w(t+1) = w(t) - \eta \nabla E_{in}(w) - \eta\lambda sign(w)$

(c)

```
hw3.main(reg="L1", lam = 0, data = 'hw3')
hw3.main(reg="L1", lam = 0.0001, data = 'hw3')
hw3.main(reg="L1", lam = 0.001, data = 'hw3')
hw3.main(reg="L1", lam = 0.005, data = 'hw3')
hw3.main(reg="L1", lam = 0.01, data = 'hw3')
hw3.main(reg="L1", lam = 0.05, data = 'hw3')
hw3.main(reg="L1", lam = 0.1, data = 'hw3')
```

[5] ✓ 34.2s

```
starting experiment using L1 regularization with lambda = 0
binary classification error (test data):  0.43457943925233644
number of 0s in the weight vector:  0
starting experiment using L1 regularization with lambda = 0.0001
binary classification error (test data):  0.43457943925233644
number of 0s in the weight vector:  0
starting experiment using L1 regularization with lambda = 0.001
binary classification error (test data):  0.4532710280373832
number of 0s in the weight vector:  7
starting experiment using L1 regularization with lambda = 0.005
binary classification error (test data):  0.48130841121495327
number of 0s in the weight vector:  9
starting experiment using L1 regularization with lambda = 0.01
binary classification error (test data):  0.5747663551401869
number of 0s in the weight vector:  17
starting experiment using L1 regularization with lambda = 0.05
binary classification error (test data):  0.5607476635514019
number of 0s in the weight vector:  5
starting experiment using L1 regularization with lambda = 0.1
binary classification error (test data):  0.4579439252336449
number of 0s in the weight vector:  7
```

```
hw3.main(reg="L2", lam = 0, data = 'hw3')
hw3.main(reg="L2", lam = 0.0001, data = 'hw3')
hw3.main(reg="L2", lam = 0.001, data = 'hw3')
hw3.main(reg="L2", lam = 0.005, data = 'hw3')
hw3.main(reg="L2", lam = 0.01, data = 'hw3')
hw3.main(reg="L2", lam = 0.05, data = 'hw3')
hw3.main(reg="L2", lam = 0.1, data = 'hw3')
```

[6] ✓ 22.6s

```
starting experiment using L2 regularization with lambda = 0
binary classification error (test data):  0.43457943925233644
number of 0s in the weight vector:  0
starting experiment using L2 regularization with lambda = 0.0001
binary classification error (test data):  0.43457943925233644
number of 0s in the weight vector:  0
starting experiment using L2 regularization with lambda = 0.001
binary classification error (test data):  0.43457943925233644
number of 0s in the weight vector:  0
starting experiment using L2 regularization with lambda = 0.005
binary classification error (test data):  0.4532710280373832
number of 0s in the weight vector:  0
starting experiment using L2 regularization with lambda = 0.01
binary classification error (test data):  0.4672897196261683
number of 0s in the weight vector:  0
starting experiment using L2 regularization with lambda = 0.05
binary classification error (test data):  0.5186915887850467
number of 0s in the weight vector:  0
starting experiment using L2 regularization with lambda = 0.1
```

I did my best and troubleshot this problem for a long, long time, including several hours with TAs at office hours who also

couldn't figure it out and posting on Piazza, but could not figure out what the problem with my code was and why I wasn't getting the results I should have been. Please check my code to see that my implementation is at least mostly correct. I would love to fix this issue but I simply do not have any more time.

2 LFD Exercise 4.5

(a)

$\sum_{q=0}^{Q} w_q^2 \leq C$ is equivalent to $w^T w \leq C$, so Γ should be I, the identity matrix.

(b)

$(\sum_{q=0}^{Q} w_q)^2 \leq C$ is the same as $w^T[1, 1, ..., 1][1, 1, ..., 1]^T w$, so Γ should be a column vector of ones equal in length to w.

3 LFD Problem 4.25

(a)

No, the size of the validation size that they used is also an important consideration. The VC bound states that the out of training error is bounded by the validation error plus the term $O(\sqrt{\frac{\ln M}{2K}})$ where K is the size of the validation set. Imagine, for example, that your validation set was only one data point. This would clearly not make for a very convincing estimation of E_{out}, and this is reflected in the aforementioned VC bound.

(b)

If they all used the same validation set, then the K term is a constant, and the previous issue goes away.

(c)

Assuming that m^* is the learner with the lowest validation loss, then $\mathbb{P}[E_{out}(m^*) > E_{val}(m^*) + \epsilon$ is equal to the probability that at least one of the m has a an E_{out} less than ϵ, as m^*, as it has the lowest expected E_{out} (because of the result of part (b)). This is equivalent to saying that
$P[E_{out}(m*) > E_{val}(m^*) + \epsilon] = P[E_{out}(m_1) > E_{val}(m_1) + \epsilon]$ or $E_{out}(m_2) > E_{val}(m_2) + \epsilon]$ or $\ldots$ or $E_{out}(m_M) > E_{val}(m_M) + \epsilon$
This, in turn, is $\leq \sum_{i=1}^{M} P[E_{out}(m_i) > E_{val}(m) + \epsilon]$
$\leq \sum_{i=1}^{M} e^{-2\epsilon^2 K_{m_i}}$
$\leq \sum_{i=1}^{M} e^{-2\epsilon^2 k(\epsilon)}$
$= Me^{-2\epsilon^2 k(\epsilon)}$

4 LFD Problem 5.4

(a)

(i)

The issue is sampling bias. Since we limited ourselves to the S&P 500, we essentially only picked the highest performing stocks. This is similar to picking the coin that flips the greatest number of heads in the simulation from before.

(ii)

A better estimate would use M=50000 (all stocks ever). Using this, we can see that the new bound is about 4.5, which tells us nothing about the profitability of the stock.

(b)

(i)

Again, we have the issue of sampling bias. Because we're only looking at the S&P 500, we can't generalize to the general population of stocks.

(ii)

We can bound the performance of buy and hold trading with the performance of buy and hold trading in the S&P 500, but that's about it.