CSE 4309 Assignment 3

Inshaad Merchant - 1001861293

Due Date: 26 September 2024

Task 1:

The Python file is attached to this document.

1 Output when degree = 1 and lambda = 0

Learned weights:
w0=40.2937
w1=-85.3182
w2=40.5272
w3=2.8325
w4=2934.2841
w5=-14575.7107
w6=2403.3571
w7=5.3809
w8=-1217.1594
w9=238.0055
w10=-8.3754
w11=-641.5481
w12=6.1993
w13=-395.2040
ID=102, output=25.3395, target value=25.0000, squared error=0.1153

2 Output when degree = 1 and lambda = 1

Learned weights:
w0=25.4349
w1=-4.8778
w2=19.3216
w3=-4.4155
w4=0.1931
w5=-0.0541
w6=1.9160

w7=-9.8849
w8=-0.5154
w9=1.5211
w10=-12.1479
w11=-3.2372
w12=9.9838
w13=-16.4487
ID=102, output=20.2186, target value=25.0000, squared error=22.8615

3 Output when degree = 2 and lambda = 0

Learned weights:
w0=166.3681
w1=-298.2493
w2=1754.4640
w3=-43.9698
w4=412.4657
w5=-58.9031
w6=2286.8364
w7=3101.1087
w8=4.3616
w9=-17152.5014
w10=389.2442
w11=-15204.6413
w12=970080.3976
w13=-14.4500
w14=100.9698
w15=-1787.9444
w16=65442.1147
w17=387.9337
w18=-4914.2598
w19=-23.3693
w20=15.3658
w21=-3571.2823
w22=62091.1718
w23=17.6688
w24=-22.6487
w25=-1020.4161
w26=12937.5971
ID=102, output=25.0664, target value=25.0000, squared error=0.0044

4 Output when degree = 2 and lambda = 1

Learned weights:

w0=24.9543
w1=-4.7855
w2=-0.3745
w3=19.1369
w4=2.1786
w5=-4.3255
w6=-0.1117
w7=0.1936
w8=0.0003
w9=-0.0523
w10=-0.0001
w11=1.9109
w12=0.0367
w13=-9.6087
w14=-1.7549
w15=-0.5378
w16=-0.0086
w17=1.6619
w18=0.0522
w19=-10.4334
w20=-1.2720
w21=-3.2279
w22=-0.1585
w23=7.3811
w24=4.7731
w25=-16.3536
w26=-0.6014
ID=102, output=19.9777, target value=25.0000, squared error=25.2235

Task 2:

we are given the following training examples for a linear regression problem:

$$x_1 = 5.3, \quad t_1 = 9.6$$

$$x_2 = 7.1, \quad t_2 = 4.2$$

$$x_3 = 6.4, \quad t_3 = 2.2$$

We want to find the 2-dimensional vector $\mathbf{w}$ that minimizes $E_D(\mathbf{w})$, when λ approaches positive infinity. The expression for $E_D(\mathbf{w})$, including the regularization term, is:

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^T\phi(x_n))^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

The solution for $\mathbf{w}$ derived from the normal equation with regularization is:

$$\mathbf{w} = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T\mathbf{t}$$

As λ approaches infinity, the matrix $(\Phi^T\Phi + \lambda I)$ converges to λI, and its inverse approaches $\frac{1}{\lambda I}$. Therefore, the limit of $\mathbf{w}$ as $\lambda \to \infty$ is:

$$\lim_{\lambda\to\infty} \mathbf{w} = \mathbf{0}$$

Task 3

we are given the linear regression problem with the following training examples:

$$x_1 = 5.3, \quad t_1 = 9.6$$

$$x_2 = 7.1, \quad t_2 = 4.2$$

$$x_3 = 6.4, \quad t_3 = 2.2$$

and the two potential models:

$$f_1(x) = 3.1x + 4.2$$

$$f_2(x) = 2.4x - 1.5$$

We will use the sum-of-squares criterion $E_D(\mathbf{w})$, defined as:

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(t_n - f(x_n))^2$$

Calculations:
For the model $f_1(x) = 3.1x + 4.2$:
f(x$_1$) $= (3.1 * 5.3) + 4.2 = 20.43$
f(x$_2$) $= (3.1 * 7.1) + 4.2 = 26.01$
f(x$_3$) $= (3.1 * 6.4) + 4.2 = 23.84$
E$_{D1}$ $= \frac{1}{2}\left[(9.6 - 20.43)^2 + (4.2 - 26.01)^2 + (2.2 - 23.84)^2\right]$
E$_{D1}$ $= \frac{1}{2}(117.3489 + 475.1281 + 470.5956)$
E$_{D1}$ $= 531.5363$
For the model $f_2(x) = 2.4x - 1.5$:
f(x$_1$) $= (2.4 * 5.3) - 1.5 = 11.22$
f(x$_2$) $= (2.4 * 7.1) - 1.5 = 15.54$
f(x$_3$) $= (2.4 * 6.4) - 1.5 = 13.26$
E$_{D2}$ $= \frac{1}{2}\left[(9.6 - 11.22)^2 + (4.2 - 15.54)^2 + (2.2 - 13.26)^2\right]$
E$_{D2}$ $= \frac{1}{2}(2.6224 + 128.5956 + 122.2596)$
E$_{D2}$ $= 126.7388$
Conclusion:

The model $f_2(x) = 2.4x - 1.5$ results in a lower sum-of-squares error $E_D(\mathbf{w}) = 126.7388$, compared to $f_1(x)$. Therefore, $f_2(x)$ is the better solution according to the sum-of-squares criterion.

Task 4:

Benefits of Bob's Algorithm:

- **Automation:** It removes the need for manual selection of λ, potentially speeding up the model tuning process.
- **Optimization:** By computing both $\mathbf{w}$ and λ simultaneously, it may find a better balance between fit and regularization, potentially leading to better generalization on unseen data.

Potential Drawbacks:

- **Complexity:** The algorithm's internal mechanism for determining λ may add computational complexity or require more sophisticated software infrastructure.
- **Overfitting:** If not properly validated, automatic adjustment of λ might overfit to the training data.

Recommendation: Given that the algorithm works correctly and optimizes both $\mathbf{w}$ and λ effectively, I recommend adopting Bob's algorithm for our projects. However, there is room for further cross-validation as a way of ensuring that it does well on varied sets of data sets, and avoids overfitting

Conclusion: Bob's approach introduces an innovative step forward in automating model regularization, which could significantly enhance our modeling capabilities and efficiency. Thus, it is a worth adding to our data analytics toolkit, provided that ongoing monitoring and validation are maintained.