CS M146, Winter 2019
Problem Set 2: Perceptron and Regression
Due Feb 12, 2019

Tian Ye
Collaborator: Derek Chu

February 11, 2019

1 Perceptron

(a) **Solution:** We start with the truth table for the OR function:

x_1	x_2	$x_1 \vee x_2$
0	0	0
1	0	1
1	1	1
0	1	1

From the table we can now start solving for w_1 and w_2:

$$w_1 * 0 + w_2 * 0 - b < 0 \tag{1}$$
$$w_1 * 1 + w_2 * 0 - b > 0 \tag{2}$$
$$w_1 * 1 + w_2 * 1 - b > 0 \tag{3}$$
$$w_1 * 0 + w_2 * 1 - b > 0 \tag{4}$$

From this we yield $b > 0$, and we can create a valid perceptron, such as by using $w_1 = 2$, $w_2 = 2$, and $b = 1$.
We can show that the perceptron is not unique, by creating a second perceptron using the following parameters:
$w_1 = 2$, $w_2 = 2$, and $b = 1.5$.

(b) **Solution:** We start with the truth table for the XOR function:

x_1	x_2	$x_1 \oplus x_2$
0	0	0
1	0	1
1	1	0
0	1	1

From the table we can now start solving for w_1 and w_2:

$$w_1 * 0 + w_2 * 0 - b < 0 \tag{5}$$
$$w_1 * 1 + w_2 * 0 - b > 0 \tag{6}$$
$$w_1 * 1 + w_2 * 1 - b < 0 \tag{7}$$
$$w_1 * 0 + w_2 * 1 - b > 0 \tag{8}$$

From this we yield $b > 0$, and we can see that there cannot be a valid perceptron as when we combine equations 6 and 8 we yield $w_1 * 0 + w_2 * 1 - 2b > 0$, yet equation 7 states that $w_1 * 1 + w_2 * 1 - b < 0$. Since $b > 0$, we have a contradiction. Therefore there does not exist a perceptron.

2 Logistic Regression

(a) **Solution:**

$$J(\theta) = -\sum_{n=1}^{N}[y_n \log(\sigma(\theta^T x_n)) + (1-y_n)\log(1-\sigma(\theta^T x_n))] \tag{9}$$

$$\frac{\partial J(\theta)}{\partial \theta_i} = -\sum_{n=1}^{N}\left[\frac{y_n(\sigma(\theta^T x_n))(1-\sigma(\theta^T x_n))x_{n,i}}{\sigma(\theta^T x_n)} - \frac{(1-y_n)(\sigma(\theta^T x_n))(1-\sigma(\theta^T x_n))x_{n,i}}{(1-\sigma(\theta^T x_n))}\right] \tag{10}$$

$$\frac{\partial J(\theta)}{\partial \theta_i} = -\sum_{n=1}^{N}[y_n(1-\sigma(\theta^T x_n)) - (1-y_n)(\sigma(\theta^T x_n))]x_{n,i} \tag{11}$$

$$\frac{\partial J(\theta)}{\partial \theta_i} = -\sum_{n=1}^{N}[y_n - \sigma(\theta^T x_n)]x_{n,i} \tag{12}$$

Where $x_{n,i}$ is the nth row, ith column.

(b) **Solution:**

$$\frac{\partial^2 J}{\partial \theta_i \partial \theta_k} = \sum_{n=1}^{N} x_{n,i}(\sigma(\theta^T x_n))(1-\sigma(\theta^T x_n))(\tfrac{\partial}{\partial \theta_k}\theta^T x_n) \tag{13}$$

$$\frac{\partial^2 J}{\partial \theta_i \partial \theta_k} = \sum_{n=1}^{N} x_{n,i}x_{n,k}(\sigma(\theta^T x_n))(1-\sigma(\theta^T x_n)) \tag{14}$$

$$\frac{\partial^2 J}{\partial \theta_i \partial \theta_k} = \sum_{n=1}^{N} x_{n,i}x_{n,k}(h_\theta(x_n))(1-h_\theta(x_n)) \tag{15}$$

This is a valid formula for calculating the Hessian as the Hessian is given by $H = \sum_{n=1}^{N}(h_\theta(x_n))(1-h_\theta(x_n))x_n x_n^T$ and if we expand $x_{n,i}x_{n,k}$, then we get that the $H_{i,k}$ element is $x_{n,i} \bullet x_{n,k}$.

(c) **Solution:**

$$z^T H z = \sum_{i,k} z_i z_k H_{i,k} \geq 0 \tag{16}$$

$$z^T H z = \sum_{i,k} \sum_{n=1}^{N} z_i z_k H_{i,k} x_{n,i} x_{n,k} (h_\theta(x_n))(1 - h_\theta(x_n)) \tag{17}$$

Since $h_\theta(x_n) = \sigma(\theta^T x_n) = (1 + e^{-\theta^T x_n})^-1$, $h_\theta(x_n) > 0$. Furthermore, since $h_\theta(x_n) < 1$, $1 - h_\theta(x_n) > 0$ as well. We can then remove those two components from Equation 17 as they are always positive:

$$z^T H z = z^T \sum_{n=1}^{N} x_n x_n^T z \tag{18}$$

$$z^T H z = z^T \sum_{n=1}^{N} X X^T z \tag{19}$$

$$\tag{20}$$

Where X is the full matrix. Therefore $(X^T z)^T (X^T z) > 0$.

3 Locally Weighted Linear Regression

(a) **Solution:**

$$\tfrac{\partial J}{\partial \theta_0} = \sum_{n=1}^{N} 2w_n(\theta_0 + \theta_1 x_{n,1} - y_n) \tag{21}$$

$$\tfrac{\partial J}{\partial \theta_1} = \sum_{n=1}^{N} 2w_n x_{n,1}(\theta_0 + \theta_1 x_{n,1} - y_n) \tag{22}$$

(b) **Solution:**

$$\sum_{n=1}^{N} 2w_n(\theta_0 + \theta_1 x_{n,1} - y_n) = 0 \tag{23}$$

$$\theta_0 = -\frac{\sum_{n=1}^{N} w_n(\theta_1 x_{n,1} - y_n)}{\sum_{n=1}^{N} w_n} \tag{24}$$

$$\sum_{n=1}^{N} 2w_n x_{n,1}(\theta_0 + \theta_1 x_{n,1} - y_n) = 0 \tag{25}$$

$$\theta_1 = \frac{\sum_{n=1}^{N} w_n x_{n,1}(\theta_1 x_{n,1} - y_n)}{\sum_{n=1}^{N} w_n x_{n,1}^2} \tag{26}$$

(c) **Solution:**

$$X\theta - y = \begin{bmatrix} \theta_0 + \theta_1 x_{1,1} - y_1 \\ \theta_0 + \theta_1 x_{2,1} - y_2 \\ \vdots \\ \theta_0 + \theta_1 x_{N,1} - y_N \end{bmatrix} \tag{27}$$

$$(X\theta - y)^T = \begin{bmatrix} \theta_0 + \theta_1 x_{1,1} - y_1 \dots \theta_0 + \theta_1 x_{N,1} - y_N \end{bmatrix} \tag{28}$$

$$W = \begin{bmatrix} w_1 & 0 & \dots & \dots & 0 \\ 0 & w_2 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & w_{N-1} & 0 \\ 0 & \dots & \dots & 0 & w_N \end{bmatrix} \tag{29}$$

$$(X\theta - y)^T W = \begin{bmatrix} w_1(\theta_0 + \theta_1 x_{1,1} - y_1) \dots w_N(\theta_0 + \theta_1 x_{N,1} - y_N) \end{bmatrix} \tag{30}$$

$$(X\theta - y)^T W (X\theta - y) = \begin{bmatrix} w_1(\theta_0 + \theta_1 x_{1,1} - y_1)^2 \dots w_N(\theta_0 + \theta_1 x_{N,1} - y_N)^2 \end{bmatrix} \tag{31}$$

$$(X\theta - y)^T W (X\theta - y) = \sum_{n=1}^{N} w_n(\theta_0 + \theta_1 x_{n,1} - y_n)^2 \tag{32}$$

4 Implementation: Polynomial Regression

(a) **Solution:**

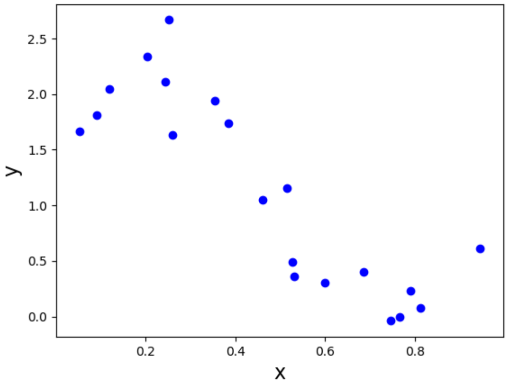


Figure 1: Training Data

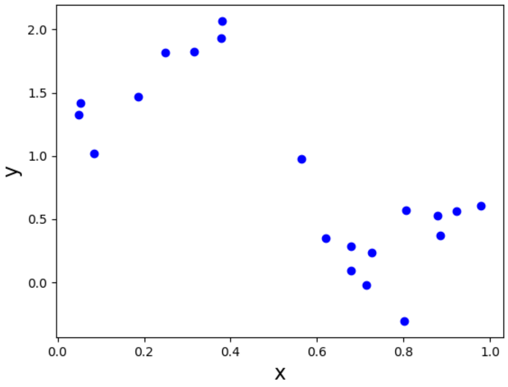


Figure 2: Test Data

Viewing both graphs above, we can see that they are as a whole appear to be relatively sinusoidal. However, since the central portion of both graphs are relatively linear, linear regression would be feasible; however, higher order functions should be used for higher accuracy.

(b) **Solution:** Complete.

(c) **Solution:** Complete.

(d) **Solution:**

Step Size	Coefficients	No. of Iterations	Cost	Time
10^{-4}	[1.91573585 -1.74358989]	10,000	5.493565588736025	0.218999862671
10^{-3}	[2.4463815 -2.81630184]	10,000	3.9125764094699393	0.18799996376
10^{-2}	[2.44640698 -2.81635335]	1,480	3.9125764057915418	0.0310001373291
0.0407	[2.44640706 -2.81635353]	387	3.9125764057914694	0.00799989700317

Viewing the coefficient sizes for 10^{-3}, 10^{-2}, and 0.0407, we see that they are all approximately the same. From this we can say that these learning rates were able to converge and obtain optimal coefficients. However, the 10^{-4} has a different coefficient. Combined with the fact that it reached the max number of iterations, we can say that the learning rate was unable to reach convergence.

Looking at the table, we can say that a smaller step size results in more iterations to reach convergence (While 10^{-3} did also reach the max number of steps, its coefficient indicates that it was nearly at convergence), and larger steps also result in less time taken.

(e) **Solution:**
Closed Form Coefficients: [2.44640709 -2.81635359]
Cost:3.912576405791464
Time: 0.000999912

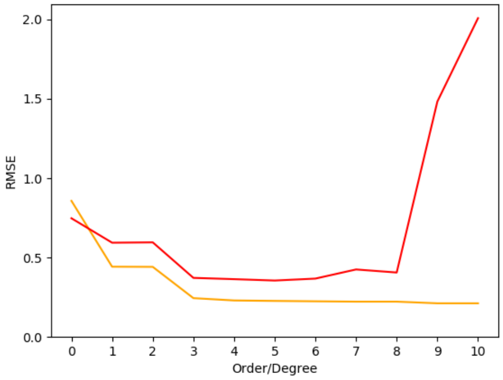
The coefficients and cost are very similar to those of Gradient Descent Convergence; however, the time taken for the closed form solution is much less than that of the Gradient Descent Convergence.

(f) **Solution:** The updated algorithm took 0.198999881744 seconds to run and the full 10,000 iterations; however, the coefficients did converge onto the expected values. This algorithm is slower than that of the larger constant step size algorithms; this is due to the fact that as the number of iterations increase, the steps decrease in size, thereby requiring more steps and time to reach convergence.

(g) **Solution:** Complete.

(h) **Solution:** RSME is preferred as a metric over $J(\theta)$ as it is a measure independent of the training set size. This therefore permits us to compare model costs without needing to take into consideration the number of training instances that were used.

(i) **Solution:**



In the figure above, red coorelates to testing data and orange coorelates to training data. The best degree polynomial would be one in the range of the third degree to the seventh degree. Anything less than that exhibits signs of underfitting as RMSE for both training and test data are relatively high. Anything higher than the seventh degree exhibits signs of overfitting as the RSME for the test data spikes sharply (while the RSME for the training data remains low).