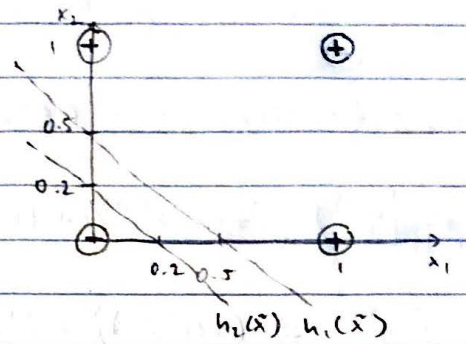
Christopher Ngai
404775704

CSM146 Problem Set 2

① Perception

A) OR

x_1	x_2	y
0	0	-1
0	1	+1
1	0	+1
1	1	+1



$h(\tilde{x}) = \tilde{w}^T \tilde{x}$

$x_1 + x_2 - 0.5 = 0$

$\tilde{x} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \quad \tilde{w} = \begin{pmatrix} -0.5 \\ 1 \\ 1 \end{pmatrix}$

$h_1(\tilde{x}) = \begin{pmatrix} -0.5 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} = -0.5 + x_1 + x_2 = 0$

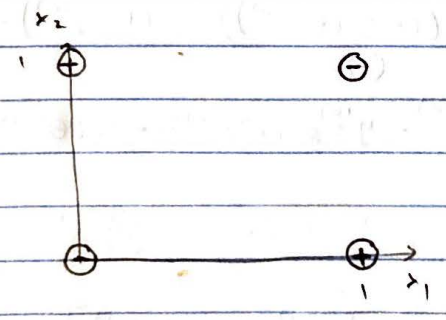
$h_2(\tilde{x}) = \begin{pmatrix} -0.2 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$

$= -0.2 + x_1 + x_2 = 0$

$\tilde{x} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \quad \tilde{w} = \begin{pmatrix} -0.2 \\ 1 \\ 1 \end{pmatrix}$

B) XOR

x_1	x_2	y
0	0	-1
0	1	+1
1	0	+1
1	1	-1



No perceptron exists because the data is NOT linearly separable. Thus, no hyperplane exists.

② Logistic Regression

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

A) $h_\theta(x^i) = \sigma(a(x^i)) = \sigma(\theta^T x^i)$ $\quad \sigma(a) = \frac{1}{1+e^{-a}}$

$$J(\theta) = -\sum_{n=1}^{N} [y^{(n)} \log(\sigma(\theta^T x^{(n)})) + (1-y^{(n)}) \log(1-\sigma(\theta^T x^{(n)}))]$$

$$\frac{\partial J}{\partial \theta_j} = -\sum_{n=1}^{N} \left[\frac{y^{(n)}}{\sigma(\theta^T x^{(n)})} \frac{\partial}{\partial \theta}(\sigma(\theta^T x^{(n)})) - \frac{(1-y^{(n)})}{(1-\sigma(\theta^T x^{(n)}))} \frac{\partial}{\partial \theta}(\sigma(\theta^T x^{(n)})) \right]$$

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

$$= \sigma(a)\left(\frac{1+e^{-a}}{1+e^{-a}} - \frac{1}{1+e^{-a}}\right)$$

$$= \sigma(a)(1-\sigma(a))$$

$$= -\sum_{n=1}^{N} \left[\frac{y^{(n)}}{\sigma(\theta^T x^{(n)})} \sigma(\theta^T x^{(n)})(1-\sigma(\theta^T x^{(n)})) x^{(n)} - \frac{(1-y^{(n)})}{(1-\sigma(\theta^T x^{(n)}))} \sigma(\theta^T x^{(n)})(1-\sigma(\theta^T x^{(n)})) x^{(n)} \right]$$

$$= -\sum_{n=1}^{N} \left[y^{(n)}(1-\sigma(\theta^T x^{(n)})) - (1-y^{(n)})\sigma(\theta^T x^{(n)}) \right] x^{(n)}$$

$$= -\sum_{n=1}^{N} \left[y^{(n)} - y^{(n)}\sigma(\theta^T x^{(n)}) - \sigma(\theta^T x^{(n)}) + y^{(n)}\sigma(\theta^T x^{(n)}) \right] x^{(n)}$$

$$= -\sum_{n=1}^{N} \left[y^{(n)} - \sigma(\theta^T x^{(n)}) \right] x^{(n)}$$

$$\boxed{\frac{\partial J}{\partial \theta_j} = \sum_{n=1}^{N} [y^{(n)} - h_\theta(x^{(n)})] x_j^{(n)}}$$

B) $\frac{\partial J}{\partial \theta_j} = \sum_{n=1}^{N} \left[y^{(n)} - h_\theta(x^{(n)}) \right] x_j^{(n)}$

$$= \sum_{n=1}^{N} \left[h_\theta(x^{(n)}) - y^{(n)} \right] x_j^{(n)}$$

$$\frac{\partial^2 J}{\partial \theta_j \partial \theta_k} = \frac{\partial J}{\partial \theta} \sum_{n=1}^{N} \left[h_\theta(x^{(n)}) - y^{(n)} \right] x_j^{(n)}$$

$$= \frac{\partial J}{\partial \theta} \sum_{n=1}^{N} \left[\sigma(\theta^T x^{(n)}) - y^{(n)} \right] x_j^{(n)}$$

$$= \sum_{n=1}^{N} \frac{\partial}{\partial \theta} \sigma(\theta^T x^{(n)}) \, x_j^{(n)} \qquad \frac{\partial \sigma(a)}{\partial a} = \sigma(a)(1-\sigma(a))$$

$$= \sum_{n=1}^{N} \sigma(\theta^T x^{(n)}) (1 - \sigma(\theta^T x^{(n)}) \, x_j^{(n)} x_j^{(n)T}$$

$$H = \sum_{n=1}^{N} h_\theta(x^{(n)}) \left(1 - h_\theta(x^{(n)})\right) x_j^{(n)} x_j^{(n)T}$$

c) $z^T H z = \sum_{j,k} z_j z_k H_{jk} \geq 0$

$$H = \sum_{n=1}^{N} h_\theta(x^{(n)}) \left(1 - h_\theta(x^{(n)})\right) x_j^{(n)} x_j^{(n)T}$$

$\rightarrow \sum_{n=1}^{N} x_j^{(n)} x_j^{(n)T}$ can be combined into vectors XX^T

$h_\theta(x^{(n)}) (1 - h_\theta(x^{(n)}))$ is a scalar ≥ 0

$0 \leq h_\theta(x^{(n)}) \leq 1$

$\Rightarrow$ thus $h_\theta(x^{(n)}) (1 - h_\theta(x^{(n)})) \geq 0$

$$z^T H z = \sum_{n=1}^{N} h_\theta(x^{(n)}) (1 - h_\theta(x^{(n)})) \, z^{(n)T} x^{(n)} x^{(n)T} z^{(n)}$$

$$= \sum_{n=1}^{N} \underbrace{h_\theta(x^{(n)}) (1 - h_\theta(x^{(n)}))}_{\geq 0} \underbrace{\left(z^{(n)T} x^{(n)}\right)^2}_{\geq 0}$$

$z^T H z \geq 0$

$\Rightarrow$ $H \succeq 0$, J is a convex function

(3) **Locally Weighted Linear Regression**

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

A) $$\frac{\partial J}{\partial \theta_0} = w_n (2(\theta_0 + \theta_1 x_{n,1} - y_n))$$

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

$$\frac{\partial J}{\partial \theta_1} = w_n (2(\theta_0 + \theta_1 x_{n,1} - y_n) \cdot x_{n,1}$$

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

B) $\frac{\partial J}{\partial \theta_0} = \cancel{2} \sum_n w_n (\theta_0 + \theta_1 x_n - y_n) = 0$

$$\theta_0 \sum_n w_n + \theta_1 \sum w_n x_n = \sum w_n y_n$$

$$\theta_0 = \frac{\sum_n w_n y_n - \theta_1 \sum w_n x_n}{\sum_n w_n}$$

$$\frac{\partial J}{\partial \theta_1} = \cancel{2} \sum_n w_n x_{n,1} (\theta_0 + \theta_1 x_{n,1} - y_n) = 0$$

$$= \theta_0 \sum_n w_n x_n + \theta_1 \sum w_n x_n^2 = \sum_n w_n x_n y_n$$

$$\left(\frac{\sum w_n y_n - \theta_1 \sum w_n x_n}{\sum w_n} \right) \sum_n w_n x_n + \theta_1 \sum w_n x_n^2 = \sum_n w_n x_n y_n$$

$$\left(\sum_n w_n y_n - \theta_1 \sum_n w_n x_n \right) \sum_n w_n x_n + \theta_1 \sum w_n x_n^2 \sum_n w_n = \sum_n w_n x_n y_n \sum_n w_n$$

$$\theta_1 \left(\sum_n w_n x_n^2 \sum_n w_n - \left(\sum_n w_n x_n \right)^2 \right) = \sum_n w_n x_n y_n \sum_n w_n - \sum_n w_n y_n \sum_n w_n x_n$$

$$\boxed{\theta_1 = \frac{\sum_n w_n x_n y_n \sum_n w_n - \sum_n w_n y_n \sum_n w_n x_n}{\sum_n w_n x_n^2 \sum w_n - \left(\sum w_n x_n \right)^2}}$$

$$\boxed{\theta_0 = \frac{\sum_n w_n y_n - \sum_n w_n x_n y_n \sum_n w_n \sum_n w_n x_n - \sum_n w_n y_n \left(\sum_n w_n x_n \right)^2}{\sum_n w_n x_n^2 \left(\sum_n w_n \right)^2 - \left(\sum w_n x_n \right)^2 \sum w_n}}$$

1×1 1×N N×N N×1

c) $J(\theta) = (X\theta - y)^T W (X\theta - y)$ D=1

$$X = \begin{pmatrix} 1 & x_{1,1} \\ 1 & x_{2,1} \\ \vdots & \vdots \\ 1 & x_{N,1} \end{pmatrix} \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \quad W = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & w_N \end{pmatrix}$$

$N \times (D+1)$ $N \times 1$ $N \times N$

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}$$

$(D+1) \times 1$

$J(\theta) = (X\theta - y)^T W (X\theta - y)$ $W = \sqrt{W}\sqrt{W}$

$= (X\theta - y)^T \sqrt{W}\sqrt{W} (X\theta - y)$ ↳ diagonal matrix

$= (X\theta - y)^T \sqrt{W}^T \sqrt{W} (X\theta - y)$ $\sqrt{W}^T = \sqrt{W}$

$= (\sqrt{W}(X\theta - y))^T (\sqrt{W}(X\theta - y))$

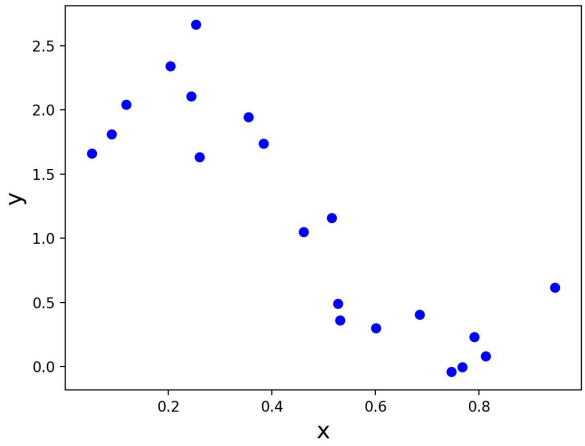
$= \sum_{n=1}^{N} (\sqrt{w_n}(\theta^T x_n - y_n))^2$

$= \sum_{n=1}^{N} w_n (\theta^T x_n - y_n)^2$

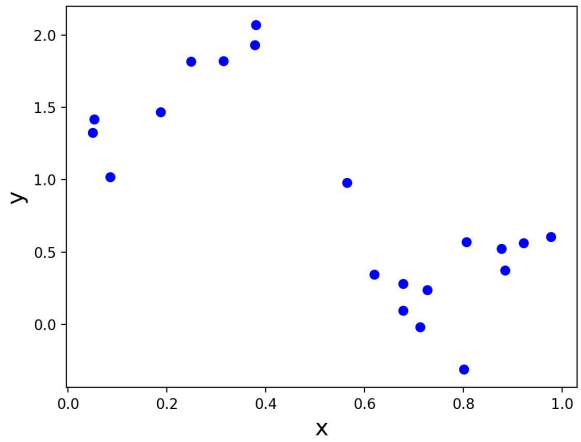
$= \sum_{n=1}^{N} w_n (\theta_0 + \theta_1 x_{n,1} - y_n)^2$ ✓

4 Implementation: Polynomial Regression

(a)



Training Data



Test Data

From these scatter plots, I notice that the training data follows a more negative linear path and has a stronger correlation between the x and y values. While the test data also somewhat follows a negative linear path, the correlation isn't as strong since the data points are more spread out and would most likely have a larger RSS than the training data. Based on these observations, I think linear regression would predict the data more effectively for the training data, but the test data may not necessarily fit that same regression line.

(d)

Step Size	Coefficients	# Iterations Until Convergence	Final Value of Objective Function	Time Until Convergence (s)
10^{-4}	[1.91573585 -1.74358989]	10000	5.493565588736025	0.220139
10^{-3}	[2.40629415 -2.73526439]	3338	3.921609713738656	0.207232
10^{-2}	[2.43388002 -2.79102983]	437	3.9134574080364732	0.041015
0.0407	[2.44053241 -2.80447778]	122	3.9127701581293137	0.010025

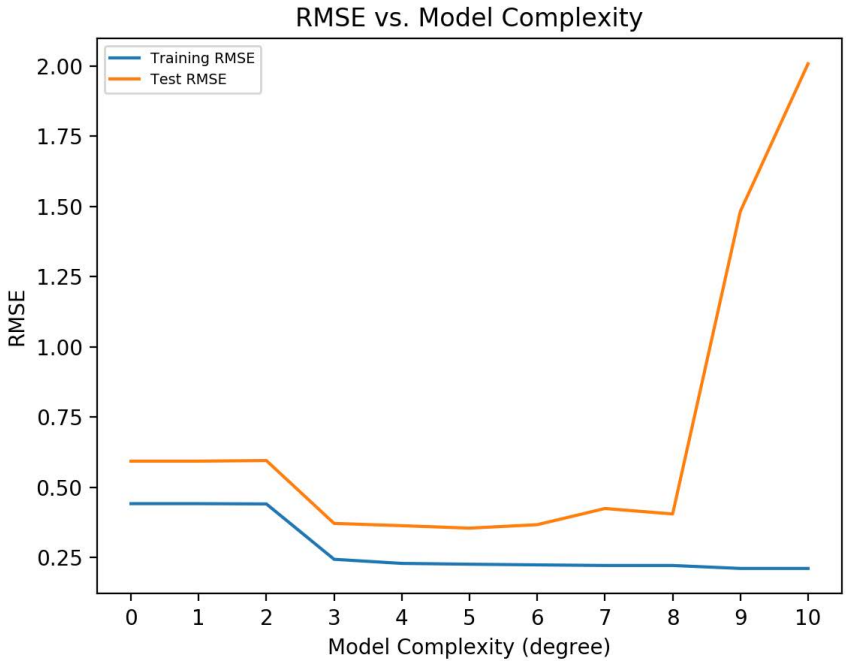
The coefficients of 10-4 are quite different compared to the coefficients of 10-3, 10-2, and 0.0407. The times until convergence are listed in the table above.

(e) The closed-form solution is [2.44640709 -2.81635359] with a cost of 3.9125764057914636. Compared to the coefficients and cost obtained by GD, they are very similar down to the decimal points. The algorithm runs in 0.000293 seconds, which is quicker than any of the GD algorithms.

(f) With the new learning rate, the algorithm has coefficients [2.45047853 -2.82461535] and cost of 3.9126700415047053. It took only 27 iterations to converge.

(h) I think we might prefer RMSE as a metric over J(θ) because the RMSE is the standard deviation of the residuals (or prediction errors). These residuals describe how large the differences are between the predicted and actual values. Intuitively, the RMSE measures how spread out these residuals are and thus illustrate how concentrated the actual values are around the predicted regression line. This is a better metric over the cost function of linear regression because outliers can skew the cost function more than it can affect the RMSE.

(i)



I think that a degree polynomial of 5 would best fit the data because this is the polynomial degree in which the training RMSE and test RMS are at a minimal error level. From this plot, there is evidence of underfitting at the leftmost end of the plot (0 <= M <= 2) since the RMSE is a bit greater than it is towards the middle M values (3 <= M <= 8). There is also evidence of overfitting since the training RMSE continues to decrease as M increases, showing that the model is being shaped more and more to fit the training data. However, at a certain point, the test RMSE suddenly increases which shows that, while the model fits the training data well, it does not generalize all data points well.