CM146, Winter 2020
Problem Set 2: Perceptron and regression
Due Feb 8, 2020

1 Splitting Heuristic for Decision Trees

Solution:

(a) Such perceptron could be $\boxed{\theta = \begin{pmatrix}1\\1\end{pmatrix}, b = 1}$. Lets see what this perceptron do to an input vector $x = \begin{pmatrix}x_1\\x_2\end{pmatrix}$:

$$y = sign(\theta^T x + b) = sign(x_1 + x_2 + 1)$$

Note that x_1, x_2 can only take the values ± 1. Thus, if either x_1 or x_2 (or both) are $+1$, the result would be $y = sign(x_1 + x_2 + 1 \geq 1) = 1$. Otherwise, if $x_1 = -1, x_2 = -1$, the result would be -1. Thus, this perceptron correctly classify OR.

Another possible perceptron would be $\boxed{\theta = \begin{pmatrix}0.5\\1\end{pmatrix}, b = 0.7}$, in which case $y = sign(0.5x_1 + x_2 + 0.7)$. Thus, only when both x_1 and x_2 are -1, the perceptron classifies -1, and otherwise it classifies $+1$. Thus, this perceptron also correctly classifies OR.

(b) No perceptron exists because XOR is not linearly separable.

2 Logistic Regression

Solution:

(a) As we know from class: $h_\theta(x_n) = \sigma(\theta^T \boldsymbol{x_n} + b) = \sigma(\theta_1 x_1^{(n)} + \ldots + \theta_k x_k^{(n)} + b)$. Thus, by using the chain rule, we get: $\frac{\partial h_\theta(x_n)}{\partial \theta_j} = \frac{\partial \sigma(\theta^T x_n + b)}{\partial(\theta^T x_n + b)} \frac{\partial(\theta^T x_n + b)}{\partial \theta_j} = \sigma(\theta^T x_n + b)[1 - \sigma(\theta^T x_n + b)] x_j^{(n)} = h_\theta(x_n)[1 - h_\theta(x_n)] x_j^{(n)}$. Lets plug this to our objective function and use the chain rule again:

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

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

$$= \boxed{\sum_{n=1}^{N}[h_\theta(x_n)x_j^{(n)} - y_n x_j^{(n)}]}$$

(b) To find the second derivatives, we will use the following derivative that we already found before: $\frac{\partial h_\theta(x_n)}{\partial \theta_j} = h_\theta(x_n)[1-h_\theta(x_n)]x_j^{(n)}$:

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

$$= \boxed{\sum_{n=1}^{N} h_\theta(x_n)[1-h_\theta(x_n)]x_k^{(n)} x_j^{(n)}}$$

By definition, this is the element that belongs to the jth row and kth column of the Hessian matrix. Therefore, it's straightforward to see that:

$$\boldsymbol{H} = \sum_{n=1}^{N} h_\theta(x_n)[1-h_\theta(x_n)]x_n x_n^T$$

because $x_n x_n^T$ is a matrix that contains $x_j^{(n)} x_k^{(n)}$ in the jth row and kth column.

(c) We will prove that $\boldsymbol{H}$ is positive semi-definite:

$$z^T H z = z^T \sum_{n=1}^{N} h_\theta(x_n)[1-h_\theta(x_n)]x_n x_n^T z = \sum_{n=1}^{N} h_\theta(x_n)[1-h_\theta(x_n)]z^T x_n x_n^T z$$

We know that:

$$0 \leq h_\theta(x_n) \leq 1 \Rightarrow 0 \leq 1 - h_\theta(x_n) \leq 1$$

$$\Rightarrow 0 \leq h_\theta(x_n)[1-h_\theta(x_n)] \leq 1$$

Thus, we only need to prove that $z^T x_n x_n^T z \geq 0$. Indeed, since z and x_n are vectors, we can write:

$$z^T x_n x_n^T z = x_n^T z x_n^T z = (x_n^T z)^2 \geq 0$$

Therefore, the Hessian matrix is positive semi-definite, and so J is convex.

3 Maximum Likelihood Estimation

Solution:

(a) $L(\theta) = P(X_1, ..., X_n; \theta) \underbrace{=}_{independent} P(X_1; \theta) \dots (X_n; \theta) =$ $\theta^{X_1}(1-\theta)^{1-X_1} \dots \theta^{X_n}(1-\theta)^{1-X_n} = \boxed{\Pi_{i=1}^n \theta^{X_i}(1-\theta)^{1-X_i}}$. We can see that the likelihood doesn't depend on the order in which the random variables are observed.

(b)

$$\ell(\theta) = \log L(\theta) = \log \Pi_{i=1}^n \theta^{X_i}(1-\theta)^{1-X_i} = \sum_{i=1}^n \log \theta^{X_i}(1-\theta)^{1-X_i}$$

$$= \boxed{\sum_{i=1}^n [X_i \log \theta + (1-X_i)\log(1-\theta)]}$$

$$\Rightarrow \frac{d\ell(\theta)}{d\theta} = \boxed{\sum_{i=1}^n [X_i \frac{1}{\theta} - (1-X_i)\frac{1}{1-\theta}]}$$

$$\Rightarrow \frac{d^2\ell(\theta)}{d\theta^2} = \boxed{\sum_{i=1}^n [-X_i \frac{1}{\theta^2} - (1-X_i)\frac{1}{(1-\theta)^2}]}$$

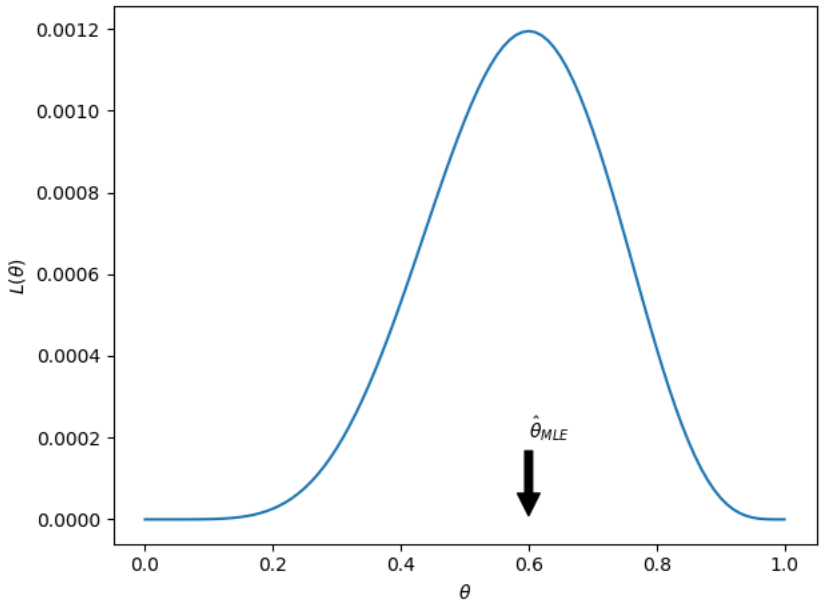
Notice that the second derivative holds $\frac{d^2\ell(\theta)}{d\theta^2} \leq 0$. Thus, $\ell(\theta)$ is concave, and have a single maximum which is also the global maximum. To find this maximum, we can find θ that makes the first derivative 0:

$$0 = \sum_{i=1}^n [\frac{X_i}{\theta} - \frac{1-X_i}{1-\theta}] \Rightarrow \frac{1}{\theta}\sum_{i=1}^n X_i = \frac{1}{1-\theta}\sum_{i=1}^n (1-X_i) \Rightarrow (1-\theta)\sum_{i=1}^n X_i = \theta \sum_{i=1}^n (1-X_i)$$

$$\Rightarrow \boxed{\theta = \frac{1}{n}\sum_{i=1}^n X_i}$$

This θ maximizes $L(\theta)$.

(c) The following is the plot of the likelihood $L(\theta)$:

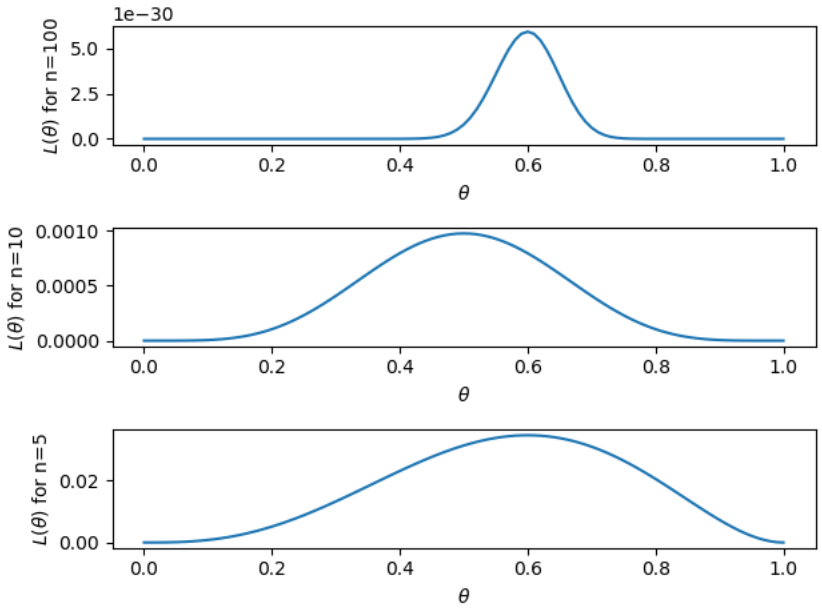


We can see that $\theta = 0.6$ acheives the maximum likelihood. This makes sense since there are 6 examples of $X_i = 1$ and 4 examples of $X_i = 0$, so based on our derivation from before:

$$\hat{\theta}_{MLE} = \frac{6}{10} = 0.6$$

So our answer agrees with the closed form formula.

(d) I received the following plots:

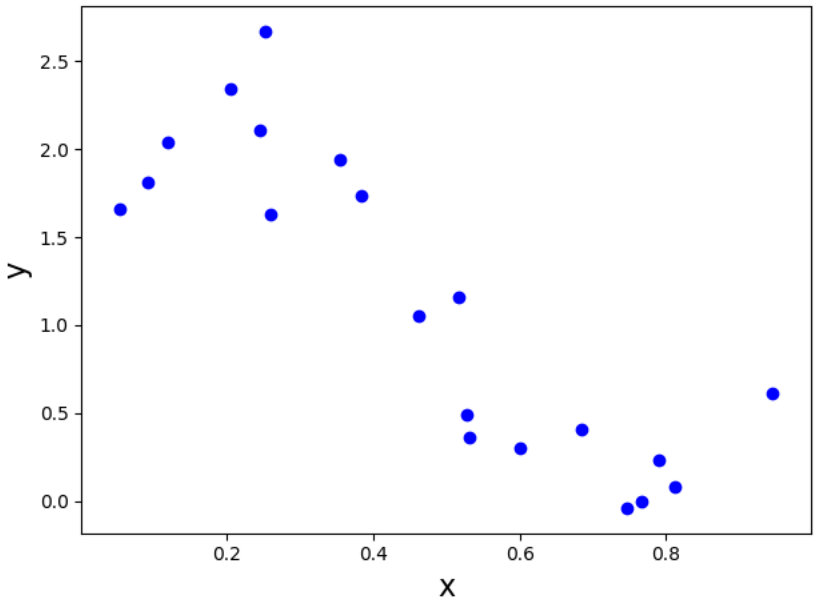


We can see that when there's less data, the variance of the likelihood is larger. More data makes the likelihood function narrower, which means that $\hat{\theta}_{MLE}$ becomes a more confident approximation as the amount of data goes up.

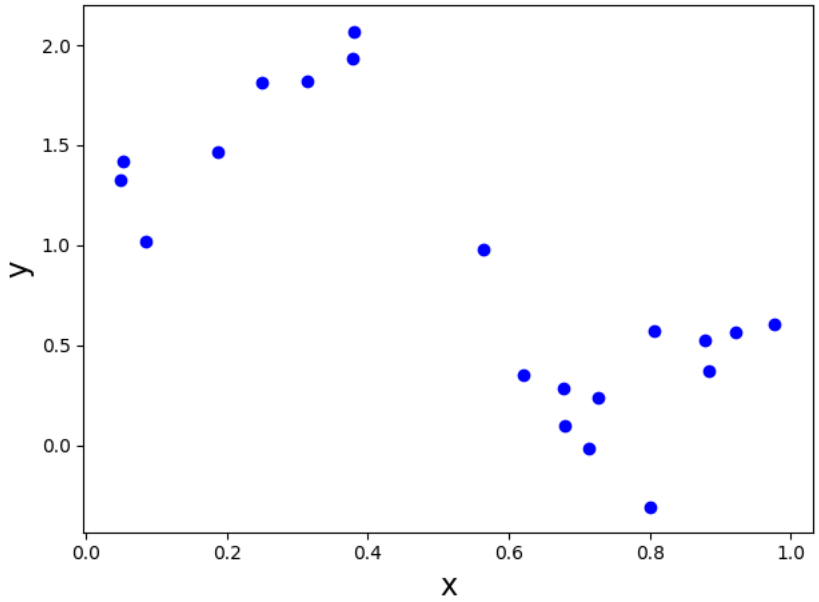
4 Implementation: Polynomial Regression

Solution:

(a) The following plot is the training data:



The following plot is the test data:



We can see that the test data is seperated into two groups, and that the training data is somewhat similar to the test data except for a few points in the middle. A linear regression model that will train on our training data would be a line from the top left corner to the bottom right corner (approximately). Thus, the linear regression model wouldn't be very effective in predicting the test data since there are points in the test data that would be far off from the model's resulted line.

(b) Nothing to submit.

(c) Nothing to submit.

(d) The following table summarizes the results:

θ	number of iterations	final value	η	time[s]
$[2.270447, -2.46064]$	10000	4.0863	0.0001	0.2587
$[2.44640, -2.81635]$	7021	3.912576	0.001	0.1856
$[2.44640, -2.81635]$	765	3.912576	0.01	0.02417
$[-9.404 \cdot 10^{18}, -4.652 \cdot 10^{18}]$	10000	$2.7109 \cdot 10^{39}$	0.0407	0.2656

We can see that:

- For the two cases of convergence, the coefficients ended up being the same.
- We can see that the fastest convergence appeared when $\eta = 0.01$, in which the convergence took 0.02417 seconds.

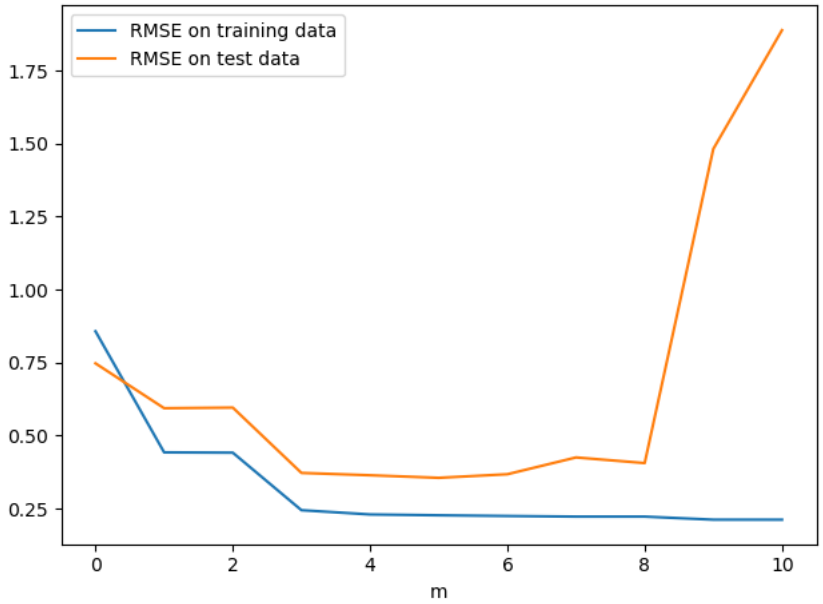
(e) The closed-form solution for the coefficients is $[2.44640706, -2.81635352]$, which is the same as the coefficients obtained by gradient decent by using $\eta = 0.001$ or $\eta = 0.01$ (as expected, since the loss is convex and so the convergence will always be the same). The closed-form solution took 0.0191 seconds, which is faster than both cases in which GD converged.

(f) By using $\eta_k = \frac{1}{1+k}$, the GD algorithm successfully converged in approximately 0.035 seconds, which is still slower than using the closed form solution.

5 Polynomial Regression

(g) $RMSE$ calculates the standard deviation of the errors, and $J(\theta)$ calculated the sum of the squares of the errors. Thus, $RMSE$ is a better measure for overfitting, since a very low standard deviation means that the model overfits the training data (most examples are very close to the model's trajectory).

(h) The following plot is the $RMSE$ on both test and training data, as a function of m (the polynomial degree):



We can see that $m = 5$ (degree 5 polynomial) best fits the data because the test $RMSE$ is the lowest when $m = 5$ (and thus, the model better than any other degree tested). We can see that overfitting starts to occur when $m > 5$, and especially when $m > 8$, because we see that as the training $RMSE$ goes down, the test $RMSE$ goes up (which indicates overfitting).