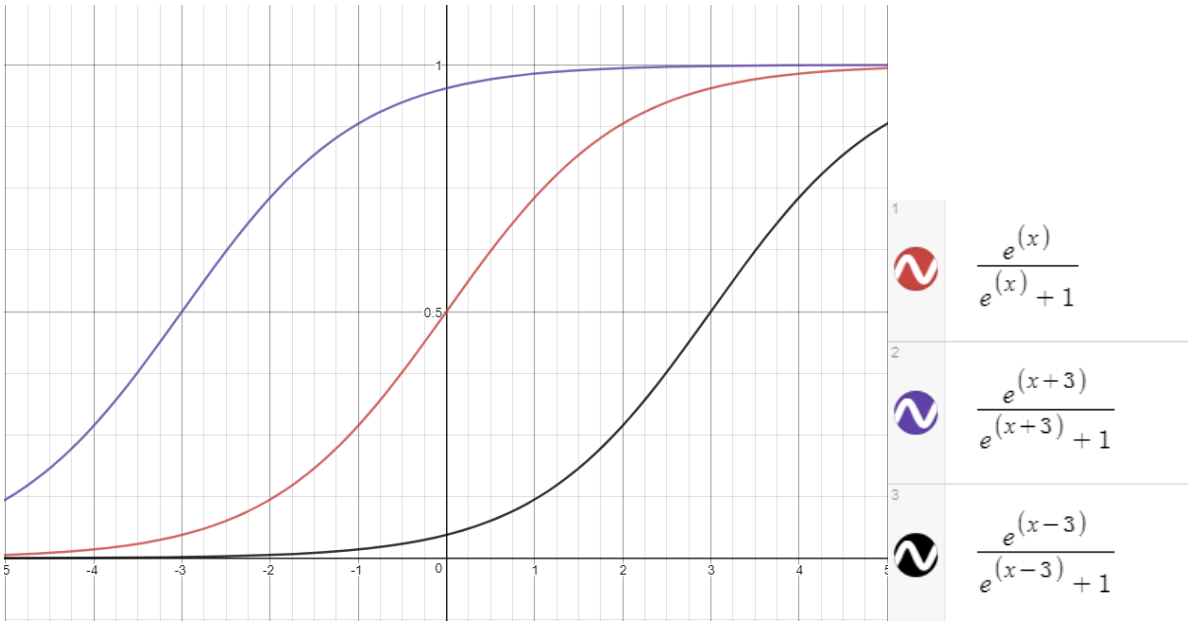
Theodore Kim
CS 4563-A: Introduction to Machine Learning
Homework #4

1. How does the logistic function change when w_0 changes?

Changing w_0 would cause the delimiting line (i.e. the line that separates the two populations) of the classifier to shift either upwards or downwards on the graph: if w_0 is increased the line would shift upwards, whereas the opposite is true of reducing w_0.

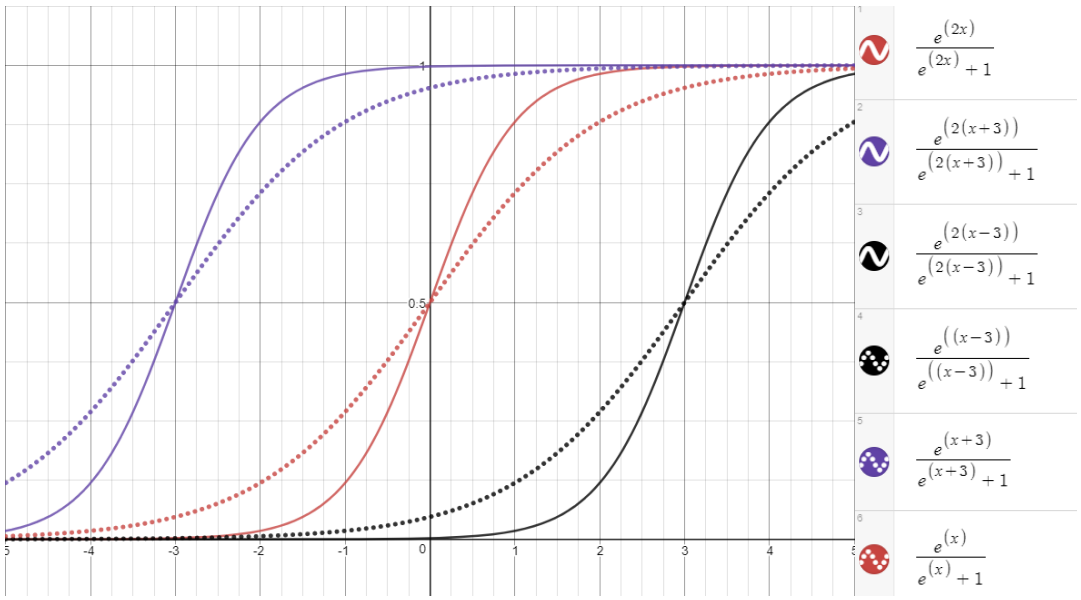
This shift in the line affects the logistic function as it affects the probability of the classification prediction. If the delimiter line is shifted upwards, there exists more "sample space" or possible samples that would be classified under the line and less samples that could possibly be classified above the line. Therefore, the "median" of the logistic function (i.e. the value of x at which the probability of prediction is 0.5 (i.e. the x value that lies ON the delimiter) would shift to the left if w_0 was increased as the portion of the domain that falls below 0.5 would decrease. The opposite is true if w_0 was decreased.



A simulation of the logistic function with altered w_0 values

*2. How does the logistic function change when if you use **w' = 2w** instead of **w**?*

Using **2w** rather than **w** would cause both the slope and intercept of the delimiter line to be doubled. Since both the slope and the intercept of the line is being altered by the same factor, the line is not translated in any direction (the altered lines will simply rotate by a fixed point). However, the line will become progressively more vertical, and therefore will result in a reduced "uncertainty" when choosing a prediction from a given x-value, in other words if a vertical line was used as the delimiter, there would be little uncertainty when determining "which side of the line" that a given x value belonged. Consequentially, the logistic function becomes more steep (there are less values of x that result in a y value between in the range [-0.9, 0.9]) and as there are less values of x that are not "extremely" far from the delimiter, therefore, not explicitly 1 or 0.



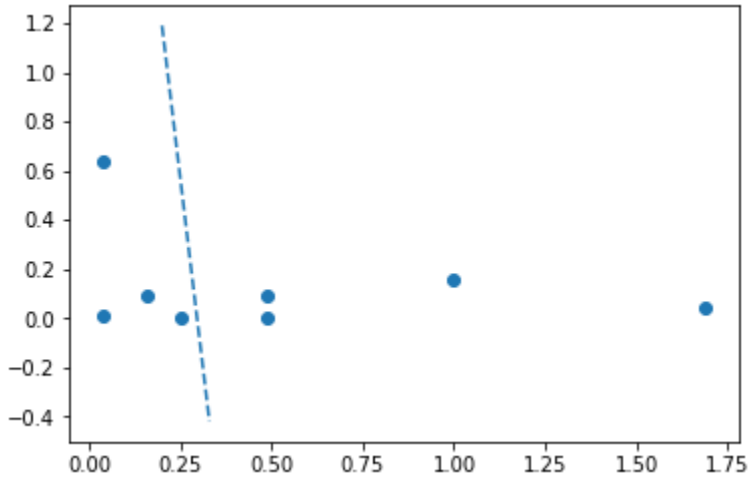
A simulation of the logistic function using 2w instead of w

3. Training a logistic classifier results in the following values for w^T = [0.66, -0.24, -0.18]

***a**. Create a confusion matrix for a threshold of 0.5:*

n = 8	**Predicted TRUE**	**Predicted FALSE**	
Actual TRUE	3	1	4
Actual FALSE	1	3	4
	4	4	8

***b**. Graph the points and the delimiter:*



***c** and **d**. Find the FPR and TPR*

$$TPR = \frac{\#TP}{\#FN + \#TP}$$
$$TPR = \frac{3}{1+3} = 0.75$$

$$FPR = \frac{\#FP}{\#FP + \#TN}$$
$$FPR = \frac{1}{1+3} = 0.75$$

***e**, **f**, **g**. Calculate accuracy, precision and recall*

$$accuracy = \frac{TP + TN}{N} = \frac{6}{8} = \mathbf{0.75}$$

$$precision = \frac{TP}{TP + FP} = \frac{3}{3+1} = \mathbf{0.75}$$

$$recall = \frac{TP}{TP + FN} = \frac{3}{3+1} = \mathbf{0.75}$$

***h**. How likely is the **w** for the examples above?*

$$L(w) = \prod_{i=1}^{N} h(x_i)^{y_i}\left(1 - h(x_i)\right)^{(1-y_i)}$$

$$L(w) = [(0.389^0)(1-0.389)^1][(0.042)^0(1-0.042)^1][(0.613)^0(1-0.613)^1][(0.167)^0(1 - 0.167)^1][(0.572)^1(1-0.572)^0][(0.526)^1(1 - 0.526)^0][(0.393)^1(0.393)^0][(0.638)^0(1-0.638)^1]$$
$$L(w) = 0.014$$

i. *Given* ***w*** *as described above and* ***w'*** *= (1.33, -2.96, -2.77), which one is more likely to be the correct decision boundary given access to the data above.*

$$L(w) = \prod_{i=1}^{N} h(x_i)^{y_i}\big(1-h(x_i)\big)^{(1-y_i)}$$
$$L(w) = \prod_{i=1}^{N}\left(\frac{1}{1+e^{-w_0-w_{1:k}^T}}\right)^{y_i}\left(1-\frac{1}{1+e^{-w_0-w_{1:k}^T}}\right)^{1-y_i}$$
$$L(w) = [(0.409^0)(1-0.409)^1][(0.022)^0(1-0.022)^1][(0.363)^0(1-0.363)^1][(0.112)^0(1 - 0.112)^1][(0.647)^1(1-0.647)^0][(0.643)^1(1 - 0.643)^0][(0.470)^1(0.470)^0][(0.766)^0(1-0.766)^1]$$
$$L(w) = 0.049$$

w' is more likely to the be the correct decision boundary.

j. *Perform one step of gradient descent using the* ***w*** *above.* ***Assuming learning rate is 0.1***

$$w_j^{new} = w_j + \frac{\alpha}{N}\frac{\partial}{\partial w_j}l(w) = w_j + \frac{\alpha}{N}\sum_{i=1}^{N}\big(y_i - h(x_i)\big)x_{ij}$$

$$w_0^{new} = w_0 + \frac{0.1}{8}\sum_{i=1}^{8}\big(y_i - h(x_i)\big)$$
$$= 0.66 + \frac{0.1}{8}\big((0-0.389)+(0-0.042)+(0-0.613)+(0-0.167)+(1-0.572) + (1-0.526)+(1-0.393)+(1-0.638)\big) = \mathbf{0.66825}$$

$$w_1^{new} = w_1 + \frac{0.1}{8}\sum_{i=1}^{8}\big(y_i - h(x_i)\big)x_{i1}$$
$$= -2.24 + \frac{0.1}{8}\big((0-0.389)0.49+(0-0.042)1.69+(0-0.613)0.04+(0-0.167)1 + (1-0.572)0.16+(1-0.526)0.25+(1-0.393)0.49+(1-0.638)0.04\big)$$
$$= \mathbf{-2.23943}$$

$$w_2^{new} = w_2 + \frac{0.1}{8}\sum_{i=1}^{8}\left(y_i - h(x_i)\right)x_{i1}$$
$$= -0.18 + \frac{0.1}{8}\big((0-0.389)0.09 + (0-0.042)0.04 + (0-0.613)0.64 + (0-0.167)0.16 + (1-0.572)0.09 + (1-0.526)0 + (1-0.393)0 + (1-0.638)0.01\big)$$
$$= \mathbf{-0.18517}$$

k. *How did the data points near the decision boundary contribute to the new value of w?*

The decision boundary is more strictly adherent to the closer data points (that is they affected the values of the predictor's parameters more closely), specifically, they altered the value of w_1 the greatest, which means that it made the slope of the predictor boundary to be steeper and better fit the actually boundary between the two datasets.

l*. How did the data points which were correctly classified and far away from the boundary contribute to the new value of w?*

These points had very little difference between their yhat and y values as they *were* correctly classified by the model. Therefore, their values did not greatly influence the value of w_0, which was only dependent on that difference. Similar is true of the w_1 and w_2 values, as the difference was negligible, overall they did little to influence the new value of each parameter. However, since the difference between the actual and predicted value was small, the weight of each point was accounted for by the value of x_j, therefore, they did slightly alter the slopes of the predictor line to more closely match their values and create a tighter boundary.

m*. How did incorrectly classified points contribute to the new value of w?*

These points did not influence the value of w_0 very significantly. While the difference between y and yhat was the greatest for these points, the two incorrect points were equal in absolute value, but had the opposite sign, so when summed, cancelled each other out. However, they did greatly influence the values of w_1 and w_2 as the difference between the two incorrect points was the greatest compared to the difference between the correctly classified point. For w_1 the false positive was much greater than the false negative, where as for w_2 the inverse was true. Given that the correctly classified points did not have much influence on these features of w, the incorrectly classified points were the most significant.

4. **SEE JUPYTER NOTEBOOK SUBMISSION**

5. Select possible response variables and predictors for the following classification problems, and indicate how many classes there are.

a. *Given an audio sample, to detect the gender of the voice*

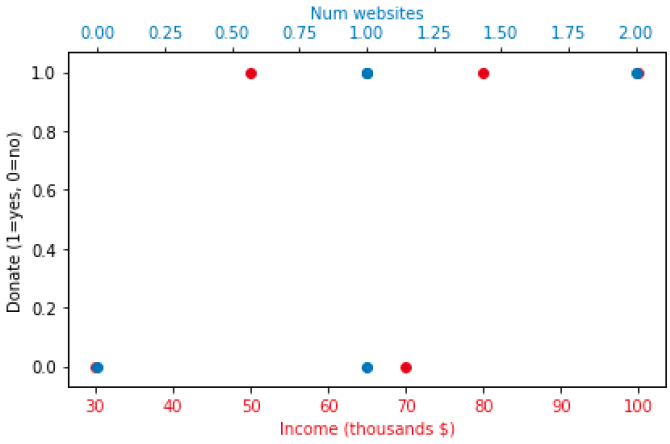
The response variable would be gender, which *technically* is a two class classification (going on purely biological gender). The predictor variables would be the pitch, timbre and volume.

b. *A electronic writing pad records motion of a stylus and it is desired to determine which letter or number was written. Assume a segmentation algorithm is already run which indicates very reliably the beginning and end time of the writing of each character.*

The response variable would be the character that was written, which, if the language is English, includes 62 possible characters (10 numerals, 26 lowercase, 26 uppercase). The predictor variables may include how long the individual takes to write out the character, how many times the individual lifts the stylus off the writing pad, and speed of the actual stylus (does the stylus move a longer distance in a specific period of time, i.e. simple letters like 'l' and 'i' would be quick to write.

6. Given data about political donation prediction.

a. *Make a scatter plot of the data*



b. *Train a logistic classifier on the data that makes at most one error and state the weights* ***w****.*

I trained this logistic model using the python module sklearn with the following code:

```
from sklearn.linear_model import LogisticRegression
from sklearn import preprocessing
import numpy as np

x = np.array([[30, 0], [50, 1], [70, 1], [80, 2], [100, 1]])
y = np.array([0, 1, 0, 1, 1])

x_scale = preprocessing.scale(x)

logreg = LogisticRegression(C=10000)
logreg = logreg.fit(x_scale, y)
w = logreg.coef_
intercept = logreg.intercept_

print(w)
print(intercept)

```

My final value for the weight vector is **w = (0.615, 0.292, 5.351)**

The final prediction vector (i.e. classification based on the trained logistic model) was (0, 1, **1**, 1, 1) where the highlighted value was incorrect.

c. *Determine which sample point was least likely to fit the trained model*

SCALED DATA:

Income	-1.4896	-0.6621	0.1655	0.5793	1.4069
Num. Websites	-1.5811	0	0	1.5811	0

$$P(y_i = 1|x_i) = \frac{1}{1 + e^{-z_i}}, z_i = w_{1:k}^T x_i + w_0$$

$$z_i = [0.292 \quad 5.351]x_i + 0.615$$

$$z_1 = [0.292 \quad 5.351][-1.4896 \quad -1.5811] + 0.615 = \mathbf{-7.817}$$

$$z_2 = [0.292 \quad 5.351][-0.6621 \quad 0] + 0.615 = \mathbf{-2.928}$$

$$z_3 = [0.292 \quad 5.351][0.1655 \quad 0] + 0.615 = \mathbf{1.501}$$

$$z_4 = [0.292 \quad 5.351][0.5793 \quad 1.5811] + 0.615 = \mathbf{4.177}$$

$$z_5 = [0.292 \quad 5.351][1.4069 \quad 0] + 0.615 = \mathbf{8.143}$$

$$P(y_1 = 1|x_1) = \frac{1}{1 + e^{7.817}} = \mathbf{0.0004}$$

$$P(y_2 = 1|x_2) = \frac{1}{1 + e^{2.928}} = \mathbf{0.0508}$$

$$P(y_3 = 1|x_3) = \frac{1}{1 + e^{-1.501}} = \mathbf{0.8177}$$

$$P(y_4 = 1|x_4) = \frac{1}{1 + e^{-4.177}} = \mathbf{0.9849}$$

$$P(y_5 = 1|x_5) = \frac{1}{1 + e^{-8.143}} = \mathbf{0.9997}$$

X_5 is the most likely sample point

d. *Would altering the value of w, so that w' = aw, where a is some positive constant, would it change the values of yhat or the likelihoods of part c?*

It would change the value of the predicted values as the z_i values would be scaled by that value alpha. Each w would be scaled up by the same value, therefore, by the distributive property, the final z_i would be:

$$z_{i_{new}} = \alpha z_{i_{old}}$$

Since all the values of the z_i are similarly adjusted, the relative "correctness" of the model would be unaffected as the values. In other words, the negative values are still negative, and the positive values would remain positive. Therefore yhat would be unaffected.

The value of the likelihood would be influenced by the scaled z_i values. The z_i would increased in absolute value: the negative values would move right on the real number line and the positive values would move left on the real number line. As a result, the likelihood values would become more "spread out." The newly scaled positive values would become MORE likely, while the negative values would become less likely.