$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 0.5 \end{bmatrix} = -1 + 0.5 = -0.5$

$\begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 0.5 \end{bmatrix} = 1 + 0.5 = 1.5$

$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 0.5 \end{bmatrix} = \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix}$

$\begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -1.5 \end{bmatrix}$

CS/ECE/ME532 Activity 19

*Estimated Time: 15 mins for P1, 20 mins for P2, 20 mins for P3*

1. You have two feature vectors $x_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $x_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and corresponding labels

   $d_1 = -1, d_2 = 1$. The linear classifier is $\text{sign}\{x_i^T w\}$ where $w = \begin{bmatrix} -1 \\ 0.5 \end{bmatrix}$.

   a) Find the squared error loss for this classifier. $\left\| \begin{bmatrix} -0.5 \\ -1.5 \end{bmatrix} \right\|_2^2 = 0.25 + 2.25 = \underline{2.5}$

   b) Find the hinge loss for this classifier. $1 - (-1)(-0.5) = 0.5$

   $1 - (1)(1.5) = -0.5 \rightarrow 0$ $\Big\} \rightarrow \underline{0.5}$

2. You have four data points $x_1 = 2, x_2 = 1.5, x_3 = 1/2, x_4 = -1/2$ and corresponding labels $y_1 = 1, y_2 = 1, y_3 = -1, y_4 = -1$.
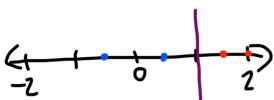
   a) Find a maximum margin linear classifier for this data. *Hint:* Graph the data.
      classifier is at $x=1$. $\quad \text{sign}(x-1)$

   b) Use squared-error loss to train the classifier (with the help of Python). Does this classifier make any errors? No

   c) Find a classifier with zero hinge loss. *Hint:* Use what you've learned about hinge loss, not computation. Does this classifier make any errors?
 $\text{sign}(2x-2)$, there is no errors

   d) Now suppose $x_4 = -5$. Use squared-error loss to find the classifier (with the help of Python). Does this classifier make any errors? Yes

   e) Can you still find a classifier with zero hinge loss when $x_4 = -5$? Does it make any errors? $\text{sign}(2x-2)$ still gives 0 hinge loss with no errors.

3. Previously, we examined the performance of classifiers trained using the squared error loss function (i.e, trained using least squares). This problem uses an *off-the-shelf* Linear Support Vector Machine to train a binary linear classifier.

   The data set is divided into training and test data sets. In order to represent a decision boundary that may not pass through the origin, we can consider the feature vector $x^T = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix}$.

   a) *Classifier using off the shelf SVM.* Code is provided to train a classifier using an off the shelf SVM with hinge loss. Run the code to find the linear classifier weights. Next, uses the weights to predict the class of the test data. How many classification errors occur? Errors: 1213

predicted class on eval data

errors
Errors: 1213

errors

predicted class on eval data
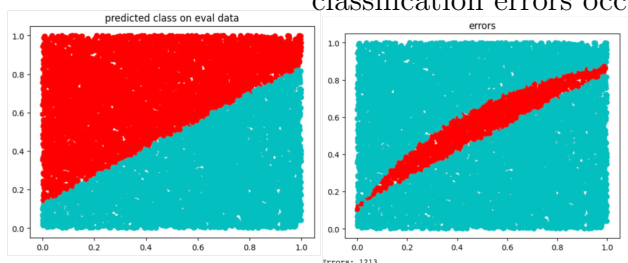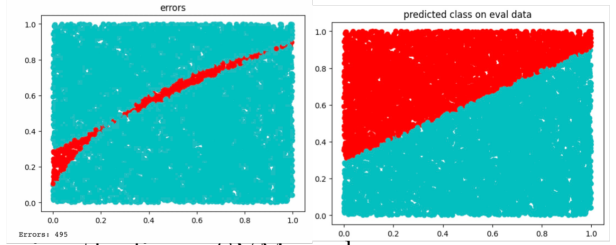
Errors: 495

b) Comment out the code that trains the classifier using the linear SVM, and uncomment the code that train the classifier using least squares (i.e, $\boldsymbol{w}_{opt} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$). How many errors occur on the test set? *Errors: 495*

predicted class on eval data

errors

Errors: 1213

c) Training a classifier using the *squared error as a loss function* can fail when correctly labeled data points lie far from the decision boundary. Linear SVMs trained with hinge loss are not susceptible to the same problem. A new dataset consisting of the first dataset, plus 1000 (correctly labeled) datapoints at $x_1 = 0, x_2 = 10$ is created. What happens to the decision boundary when these new data points are included in training the linear SVM? *The decision boundary doesn't move, and the errors stay the same. Errors = 1213*

d) How does this compare with the error rate of the linear classifier trained with the new data points? Why is the such a difference in performance?

Errors increase greatly with least squares. Errors is now 2668.

These errors occur because the 1000 easy to classify points move the decision boundary/weights so that the squared loss is also minimized for them, even though they are easy to classify

predicted class on eval data

errors

Errors: 2668