Machine Learning - CS 5350
Homework 5

Aaron Templeton
U0734119

November 27, 2019

1 Warm Up: Margins

(1) we know that XOR if not linearly separable in the plane. Nevertheless, if we consider the Boolean values True and False as 1 and -1 respectively, the feature transformation function $\phi : R^2 \longrightarrow R^2$ defined as:

$$\phi([x_1, x_2]) = [x_1, x_1 \cdot x_2]$$

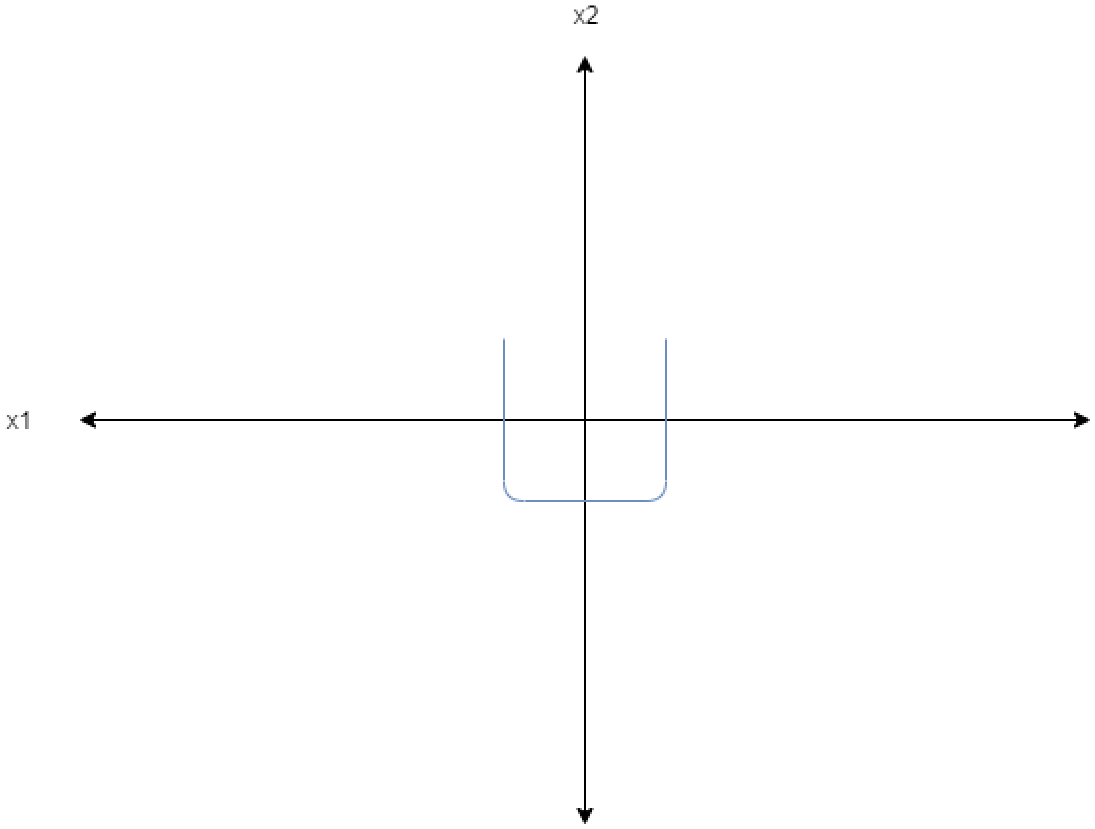
makes the XOR function linearly separable in the new space. What would be the ideal margin of the linear classifier resulting from learning the XOR function as a hard SVM learning problem in the new space? Draw the obtained line corresponding to the learned classifier in the original Euclidean plane (input space).

Answer:

some example points would map from $[x_1, x_2]$ to $[x_1, x_1x_2]$ as follows:

$[-1, -1]$ maps to $[-1, 1]$
$[-1, 1]$ maps to $[-1, -1]$
$[1, -1]$ maps to $[1, -1]$
$[1, 1]$ maps to $[1, 1]$

the maximum (ideal) margin seperator is the line $x_1x_2 = 0$ with a margin of 1. it corresponds to the $x_1 = 0$ and $x_2 = 0$ axes in the original space.



(2) Consider the following collection of points in Table 1:

Point	coordinate	label	Point	coordinate	label
x_1	(0, 0)	+	x_5	(1, 0)	−
x_2	(0, 1)	+	x_6	$(\frac{1}{2}, \frac{\sqrt{3}}{2})$	−
x_3	(1, 1)	+	x_7	$(\frac{3}{2}, 0)$	−
x_4	$(\frac{1}{2}, 0)$	+	x_8	$(1, \frac{1}{2})$	−

Table 1: A collection of points

Let us consider the following training sets consisting of points from the given collection. We have,

$$D_1 = \{x_1, x_2, x_3, x_5, x_7\}$$

$$D_2 = \{x_1, x_5, x_6, x_8\}$$

$$D_3 = \{x_3, x_4, x_5, x_7\}$$

(a) [9 points] Give the corresponding maximum margin for each one of the sets D_1, D_2 and D_3.
to find the max margin we take the distance between the closest positive and negative point and divide by 2.

the closest points in D_1 are x_1, x_5 and x_3, x_5 so max margin is

$$\frac{\sqrt{(1-0)^2+(0-0)^2}}{2} = \frac{1}{2}$$

the closest points in D_2 are either x_1, x_5 or x_1, x_6 so max margin is:

$$\frac{\sqrt{(1-0)^2+(0-0)^2}}{2} = \frac{1}{2}$$

the closest points in D_3 are x_4, x_5 so max margin is:

$$\frac{\sqrt{(1-1/2)^2+(0-0)^2}}{2} = \frac{\sqrt{1/4}}{2}$$

(b) [3 points] What is the Perceptron mistake bound for each dataset? Which one has the highest bound?.

to find the perceptron mistake boud for each dataset we will find the centroid of each set. after we find the centroid the radius of the smallest enclosing circle will be the distance from the centroid to the furthest away point in the dataset. Then we can use the formula $(R/\gamma)^2$ where gamma is the margin and R is the radius, to find the perceptron mistake bound.

$$\begin{aligned} &\text{for } D_1 \\ \text{centroid } &= \left(\frac{0+0+1+1+3/2}{5}, \frac{0+1+1+0+0}{5}\right) = (0.7, 0.4) \\ &\text{the furtherst point from the centroid is } x_2 \\ &\text{distance from centroid to } x_2 = 0.92 = R \\ &(R/\gamma)^2 = (0.92/1/2)^2 = 3.38 \end{aligned}$$

$$\begin{aligned} &\text{for } D_2 \\ \text{centroid} = &\left(\frac{0+1+1/2+1}{4}, \frac{0+0+\sqrt{3}/2+1/2}{4}\right) \\ &= (0.625, 0.34) \\ \text{distance from centroid to furthest point } x_1 &= 0.71 = R \\ (R/\gamma)^2 &= (0.71/0.5)^2 = 2.01 \end{aligned}$$

$$\begin{aligned} &\text{for } D_3 \\ \text{centroid } &= \left(\frac{1+0.5+1+3/2}{4}, \frac{1+0+0+0}{4}\right) \\ &= (1, 1/4) \\ &\text{distance from centroid to } x_3 = .75 \\ &(R/\gamma)^2 = (0.75/\frac{\sqrt{1/4}}{2})^2 = 9 \end{aligned}$$

D_3 has the highest perceptron mistake bound

(c) [3 points] Briefly explain why is a dataset “better” than the other among D_1, D_2 and D_3.

a dataset is better than another data based on the maximum margin and the perceptron mistake bound. A bigger margin allows for a higher chance that data will not be misclassified and a small mistake bound margin means that there is a less chance that perceptron will make a mistake. The dataset with the biggest margin and lowest mistake will likely perform better than other datasets.

2 Experiments

	Best hyper-parameters	Average cross-validation accuracy	Training accuracy	Test Accuracy
SVM	best hp = 0.1	41.8	54.80	80.5
Naive Bayes	best smoothing = 0.5	41.87	42.80	46.5
Bagged Forests	of trees = 100	89	96.37	96.32
SVM over trees	trees 10	19.89	0.0	68.0

Table 2: Result table for semeion

	Best hyper-parameters	Average cross-validation accuracy	Training accuracy	Test Accuracy
SVM	best hp = .001	38.67	50.05	63.33
Naive Bayes	smoothing =2	40.23	42.5	43.61
Bagged Forests	trees = 100	98.93	99.74	99.75
SVM over trees	Trees = 100	20.1	53	61.2

Table 3: Result table for madelon

Explanations of implementations

1. SVM for the svm i followed the slides and implemented the cross validation and necessary data transformations. I made a train function that trains the svm for 10 epochs and it updates the learning rate every epoch per the homework. It is not perfect, but it achieves a decent result

2. Naive Bayes this one was hard for me to understand at first so i dug around online for a bit to try and understand what i needed to do to implement this. i found that i needed a W for positive results and negative results and a list of prior positives and negatives. i then implemented the probabilities

3. random forest random forest took some time because i wanted to redo my tree and i started a new decision tree implementation from scratch. however the forest was straight forward. once i had my tree done i created ”k” trees and got the accuracy from the k trees

4. svm over trees my svm over trees is not perfect. i believe i have an error with the data transformation from the trees to the svm but the test accuracy was not all that bad. it was just a simple data transformation pulling data prediction from the k trees and then passing the data to the svm for training

(5) Extra Credit. this has to do with model stacking. is a model ensembling technique used to combine information from multiple predictive models to generate a new model. In general, stacking produces small gains with a lot of added complexity not worth it for most businesses. But Stacking is almost always fruitful so its almost always used in top solutions. In fact, stacking is really effective when you have a team of people trying to collaborate on a model. A single set of folds is agreed upon and then every team member builds their own model(s) using those folds. Then each model can be combined using a single stacking script. This is great because it prevents team members from stepping on each others toes, awkwardly trying to stitch their ideas into the same code base. the end result is a model that is less likely to overfit or fail to generalize data.

Comments to graders: this assignment was an extremely tedious one. I hope that there can be some leniency in the grading. i had a hard time because i work 30+ hours a week and I am taking 5 courses. I do not have 20+ hours a week to dedicate to this course alone and i have done my best with the time that i have. Furthermore, my power went out last night and I was unable to work on the assignment. I am not trying to make excuses, we are all in different positions and I believe that this assignment, and overall, this course has been very rigorous especially now with the assignments and final projects. Please show some leniency.