CS 5350/6350: Machine Learning Fall 2021

Homework 3

Handed out: 19 Oct, 2021
Due date: 11:59pm, 2 Nov, 2021

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free to discuss the homework with the instructor or the TAs.
- Your written solutions should be brief and clear. You do not need to include original problem descriptions in your solutions. You need to show your work, not just the final answer, but you do *not* need to write it in gory detail. Your assignment should be **no more than 15 pages**. Every extra page will cost a point.
- Handwritten solutions will not be accepted.
- *Your code should run on the CADE machines.* **You should include a shell script, `run.sh`, that will execute your code in the CADE environment. Your code should produce similar output to what you include in your report.**

  You are responsible for ensuring that the grader can execute the code using only the included script. If you are using an esoteric programming language, you should make sure that its runtime is available on CADE.
- Please do not hand in binary files! We will *not* grade binary submissions.
- The homework is due by **midnight of the due date**. Please submit the homework on Canvas.

1 Paper Problems [36 points + 15 bonus]

1. [8 points] Suppose we have a linear classifier for 2 dimensional features. The classification boundary, i.e., the hyperplane is $2x_1 + 3x_2 - 4 = 0$ (x_1 and x_2 are the two input features).

x_1	x_2	label
1	1	1
1	-1	-1
0	0	-1
-1	3	1

Table 1: Dataset 1

(a) [4 points] Now we have a dataset in Table 1. Does the hyperplane have a margin for the dataset? If yes, what is the margin? Please use the formula we discussed in the class to compute. If no, why? (Hint: when can a hyperplane have a margin?)
Answer: The hyper plane correctly guesses each of the examples so there is a margin that can be calculated.

$$\text{Example 1: } \frac{|\mathbf{w}^T\mathbf{x}_1|}{||\mathbf{w}||} = .277$$

$$\text{Example 2: } \frac{|\mathbf{w}^T\mathbf{x}_2|}{||\mathbf{w}||} = 1.387$$

$$\text{Example 3: } \frac{|\mathbf{w}^T\mathbf{x}_3|}{||\mathbf{w}||} = 1.109$$

$$\text{Example 4: } \frac{|\mathbf{w}^T\mathbf{x}_4|}{||\mathbf{w}||} = .832$$

The smallest distance between the hyperplane and an instance in the dataset is .277 so that is the margin for the hyperplane.

x_1	x_2	label
1	1	1
1	-1	-1
0	0	-1
-1	3	1
-1	-1	1

Table 2: Dataset 2

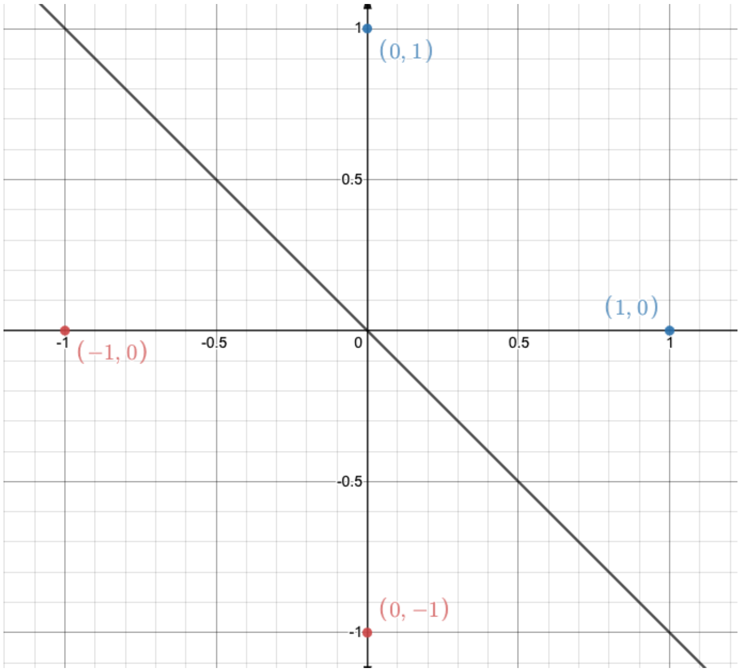
(b) [4 points] We have a second dataset in Table 2. Does the hyperplane have a margin for the dataset? If yes, what is the margin? If no, why?
Answer: We do not have a margin for this dataset since the data is not linearly separable with the hyperplane given. If we plugin $[-1, -1]$ for our two features we will get $2(-1) + 3(-1) - 4 = -9 \leq 0$ which means that our label should be negative but it is positive in the dataset.

2. [8 points] Now, let us look at margins for datasets. Please review what we have discussed in the lecture and slides. A margin for a dataset is not a margin of a hyperplane!

(a) [4 points] Given the dataset in Table 3, can you calculate its margin? If you cannot, please explain why.
Answer: If we look at the points plotted on the graph with + points labeled blue and - points labeled red we get

x_1	x_2	label
-1	0	-1
0	-1	-1
1	0	1
0	1	1

Table 3: Dataset 3



The black line is the linear classifier $x_1+x_2=0$. Visually we see that this classifier will give us the margin for the dataset. Any other classifiers will give us smaller margins. Now we just need to plug in the values for the distance between any of the points and the hyperplane.

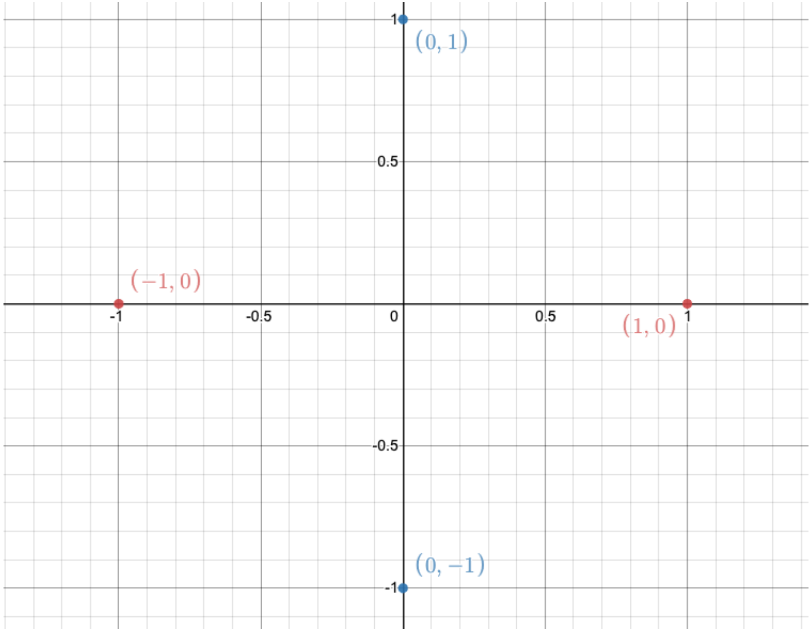
$$= \frac{|\mathbf{w}^T\mathbf{x} + b|}{\|\mathbf{w}\|} = \frac{1}{1.41} = .707$$

x_1	x_2	label
-1	0	-1
0	-1	1
1	0	-1
0	1	1

Table 4: Dataset 4

(b) [4 points] Given the dataset in Table 4, can you calculate its margin? If you cannot, please explain why.

Answer: The margin cannot be calculated since the data is not linearly separable.



3. [**Bonus**] [5 points] Let us review the Mistake Bound Theorem for Perceptron discussed in our lecture. If we change the second assumption to be as follows: Suppose there exists a vector $\mathbf{u} \in \mathbb{R}^n$, and a positive γ, we have for each $(\mathbf{x}_i, y_i)$ in the training data, $y_i(\mathbf{u}^\top \mathbf{x}_i) \geq \gamma$. What is the upper bound for the number of mistakes made by the Perceptron algorithm? Note that $\mathbf{u}$ is unnecessary to be a unit vector.
Answer: For proof 1/3 and 2/3 we will get the same conclusions as before i.e. (1)$\mathbf{u}^\top \mathbf{w}_t \geq t\gamma$ and (2)$\|\mathbf{w}_t\|^2 \leq tR^2$ ($\mathbf{u}$ not being a unit vector will not affect these proofs) $\therefore$

$$R\sqrt{t} \geq \|\mathbf{w}_t\|$$

Now let's divide (1) by $\|\mathbf{u}\|$ on both sides giving us

$$\frac{\mathbf{u}^\top \mathbf{w}_t}{\|\mathbf{u}\|} \geq \frac{t\gamma}{\|\mathbf{u}\|}$$

$$R\sqrt{t} \geq \|\mathbf{w}_t\| \geq \frac{\mathbf{u}^\top \mathbf{w}_t}{\|\mathbf{u}\|}$$

We know the above is true looking at slide 60 from the lectures.

$$R\sqrt{t} \geq \|\mathbf{w}_t\| \geq \frac{\mathbf{u}^\top \mathbf{w}_t}{\|\mathbf{u}\|} \geq \frac{t\gamma}{\|\mathbf{u}\|}$$

$$R\sqrt{t} \geq \frac{t\gamma}{\|\mathbf{u}\|}$$

$$t \leq \frac{R^2\|\mathbf{u}\|^2}{\gamma^2}$$

4. [10 points] We want to use Perceptron to learn a disjunction as follows,

$$f(x_1, x_2, \ldots, x_n) = \neg x_1 \vee \neg \ldots \neg x_k \vee x_{k+1} \vee \ldots \vee x_{2k} \quad \text{(note that } 2k < n).$$

The training set are all 2^n Boolean input vectors in the instance space. Please derive an upper bound of the number of mistakes made by Perceptron in learning this disjunction.
Answer: Our equivalent classifier would be

$$(1 - x_1) + \ldots + (1 - x_k) + x_{k+1} + \ldots + x_{2k} - 1 = 0$$

We don't want to use this classifier though since some points will lie on it. $\therefore$ we use

$$(1 - x_1) + \ldots + (1 - x_k) + x_{k+1} + \ldots + x_{2k} - 0.5 = 0$$

The closest point to this hyperplane is a vector where half of the beginning elements are 0's and the rest are 1's. We now can calculate γ

$$\gamma = \sqrt{\frac{1}{(2k+1)^2}} = \frac{1}{2k+1}$$

$$R = \sqrt{n+1}$$

Let's calculate the upper bound now

$$t = \frac{R^2}{\gamma^2} = (n+1)(2k+1)^2$$

5. [10 points] Prove that linear classifiers in a plane cannot shatter any 4 distinct points.
Answer: We see this in 2b

6. [**Bonus**] [10 points] Consider our infinite hypothesis space $\mathcal{H}$ are all rectangles in a plain. Each rectangle corresponds to a classifier — all the points inside the rectangle are classified as positive, and otherwise classified as negative. What is VC($\mathcal{H}$)?

2 Practice [64 points]

1. [2 Points] Update your machine learning library. Please check in your implementation of ensemble learning and least-mean-square (LMS) method in HW1 to your GitHub repository. Remember last time you created the folders "Ensemble Learning" and "Linear Regression". You can commit your code into the corresponding folders now. Please also supplement README.md with concise descriptions about how to use your code to run your Adaboost, bagging, random forest, LMS with batch-gradient and stochastic gradient (how to call the command, set the parameters, etc). Please create a new folder "Perceptron" in the same level as these folders.

2. We will implement Perceptron for a binary classification task — bank-note authentication. Please download the data "bank-note.zip" from Canvas. The features and labels are listed in the file "bank-note/data-desc.txt". The training data are stored in the file "bank-note/train.csv", consisting of 872 examples. The test data are stored in "bank-note/test.csv", and comprise of 500 examples. In both the training and testing datasets, feature values and labels are separated by commas.

   (a) [16 points] Implement the standard Perceptron. Set the maximum number of epochs T to 10. Report your learned weight vector, and the average prediction error on the test dataset.
   Answer: Please run the `run.sh` file and go to the weights and error directory to view the weights and the error.

   (b) [16 points] Implement the voted Perceptron. Set the maximum number of epochs T to 10. Report the list of the distinct weight vectors and their counts — the number of correctly predicted training examples. Using this set of weight vectors to predict each test example. Report the average test error.
   Answer: Please run the `run.sh` file and go to the weights and error directory to view the weights and the error.

   (c) [16 points] Implement the average Perceptron. Set the maximum number of epochs T to 10. Report your learned weight vector. Comparing with the list of weight vectors from (b), what can you observe? Report the average prediction error on the test data.
   Answer: Please run the `run.sh` file and go to the weights and error directory to view the weights and the error.

   (d) [14 points] Compare the average prediction errors for the three methods. What do you conclude?
   Answer: That the error for voted and averaged perceptron are fairly low and reliable. Standard perceptron can have lower error but it is less reliable because of the shuffling of the data.