CS 5350/6350: Machine Learning Spring 2019

Homework 3

Handed out: 25 Feb, 2019
Due date: 11:59pm, 9 Mar, 2019

- 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 [40 points + 10 bonus]

1. [7 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) [3 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?)

[Ans]: *Yes, there is a margin for the dataset since no points lie on the hyperplane and all points are correctly classified. Use the formula* $d(x,h) = \frac{|\boldsymbol{w}^T\boldsymbol{x}+b|}{|\boldsymbol{w}|}$ *, we find the minimum margin is* $\frac{1}{\sqrt{13}}$.

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?

[Ans]: *There is no margin since the last training example is misclassified.*

2. [7 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!

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

Table 3: Dataset 3

(a) [3 points] Given the dataset in Table 3, can you calculate its margin? If you cannot, please explain why.

[Ans]: *By observation, the margin of the dataset exists. For eg.,* $h: 2x_1+2x_2+1 = 0$ *perfectly separates the date. It can be easily seen that the margin is* $\frac{\sqrt{2}}{2}$.

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.

[Ans]: *This dataset is not separable. We prove it by contradiction. Assume that* $h : w_1x_1 + x_2w_2 + b = 0$ *separates the data, then we have*

$$\begin{aligned} -w_1 + b &\leq 0 \quad (1) \\ -w_2 + b &> 0 \quad (2) \\ w_1 + b &\leq 0 \quad (3) \\ w_2 + b &> 0 \quad (4) \end{aligned}$$

yielding $b > 0$ *and* $b \leq 0$ *at the same time. Hence, the data is not separable.*

3. [8 points] Let us review the Mistake Bound Theorem for Perceptron discussed in our lecture.

(a) [3 points] 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.

[Ans] *The number of mistakes made by the perceptron in upper bounded by* $\frac{R}{\gamma^2}$. *If* $\boldsymbol{u}$ *is not unit, than the bound becomes* $\frac{R^2||\boldsymbol{u}||_2^2}{\gamma^2}$ *due to the following fact: after* t *mistakes,* $\boldsymbol{u}^T\boldsymbol{w}_t \geq t\gamma$ *and* $||\boldsymbol{w}_t||^2 \leq tR^2$. *Using* $\boldsymbol{u}^T\boldsymbol{w}_t = ||\boldsymbol{u}||||\boldsymbol{w}_t|| \cos\theta$, *we get*

$$\sqrt{t}R \geq ||\boldsymbol{w}_t|| \geq \frac{\boldsymbol{u}^T\boldsymbol{w}_t}{||\boldsymbol{u}||} \geq \frac{t\gamma}{||\boldsymbol{u}||} \Rightarrow t \leq \frac{R^2||\boldsymbol{u}||^2}{\gamma^2}$$

.

(b) [3 points] Following (a), if we do NOT assume $\mathbf{u}$ is a unit vector, and we still want to obtain the same upper bound introduced in the lecture, how should we change the inequalities in the second assumption?

[Ans] *We should increase the margin requirement, i.e.,* $\forall i, y_i(\boldsymbol{u}^T\boldsymbol{x}_i) \geq ||u||\gamma$.

(c) [2 points] Now, let us state the second assumption in another way: Suppose there is a hyperplane that can correctly separate all the positive examples from the negative examples in the data, and the margin for this hyper plane is γ. What is the upper bound for the number of mistakes made by Perceptron algorithm?

[Ans] *The original statement of the second assumption is equivalent to saying that the there exists* $h : \boldsymbol{w}^T\boldsymbol{x} = 0$ *which can separate the data with margin* γ, *i.e.,* $\frac{|\boldsymbol{w}^T\boldsymbol{x}^{(i)}|}{||\boldsymbol{w}||} = \frac{y_i(\boldsymbol{w}^T\boldsymbol{x}^{(i)})}{||\boldsymbol{w}||} \geq \gamma$, *which is the same as the statement here. So that the mistake bound is still* $\left(\frac{R}{\gamma}\right)^2$.

4. [6 points] We want to use Perceptron to learn a disjunction as follows,

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

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.
[Ans] *In this case, $R = \sqrt{n}$. For the disjunction f, we can choose the unit vector*

$$\boldsymbol{u} = (\underbrace{-\frac{1}{\sqrt{2k}}, -\frac{1}{\sqrt{2k}}, \cdots, -\frac{1}{\sqrt{2k}}}_{k\ terms}, \underbrace{\frac{1}{\sqrt{2k}}, \frac{1}{\sqrt{2k}}, \cdots, \frac{1}{\sqrt{2k}}}_{k\ terms}, \underbrace{0, 0, \cdots, 0}_{n-2k\ terms})$$

Assuming the binary entries $x_i \in \{+1, -1\}$, then for all positive samples, we have $y_i(\boldsymbol{u}^T\boldsymbol{x}^{(i)}) \geq \frac{1}{\sqrt{2k}}$ and for all negative samples, we have $y_i(\boldsymbol{u}^T\boldsymbol{x}^{(i)}) \geq \sqrt{2k}$. Choose $\gamma = \frac{1}{\sqrt{2k}}$, then it is guaranteed that for any sample $\boldsymbol{x}^{(i)}, y_i(\boldsymbol{u}^T\boldsymbol{x}^{(i)}) \geq \gamma$. Hence, the number of mistakes is upper bounded by $\left(\frac{R}{\gamma}\right)^2 = \left(\frac{\sqrt{n}}{\frac{1}{\sqrt{2k}}}\right)^2 = 2kn = \mathcal{O}(n)$.

5. [6 points] Suppose we have a finite hypothesis space $\mathcal{H}$.

   (a) [3 points] Suppose $|\mathcal{H}| = 2^{10}$. What is the VC dimension of $\mathcal{H}$?
   [Ans] *We have $VC(\mathcal{H}) \leq 10$.*

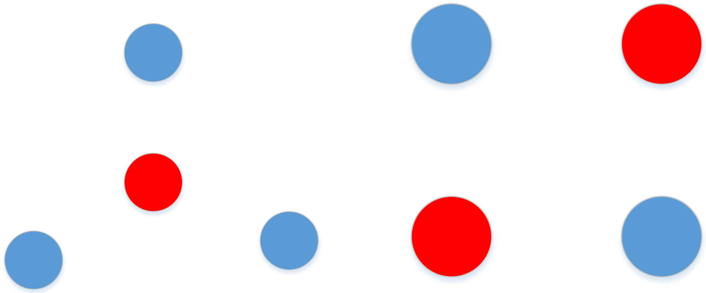
   (b) [3 points] Generally, for any finite $\mathcal{H}$, what is VC($\mathcal{H}$) ?
   [Ans] *In general, $VC(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$. Since for a a set of m data points, there are 2^m possible labels distributions in total, each corresponding to the output of a specific concept $h \in \mathcal{H}$. If $2^m > |\mathcal{H}|$, then there are at most $|\mathcal{H}|$ distinct concept outputs. However, the number of possible partitions is 2^m, which is greater than the number of functions contained in $\mathcal{H}$. Hence, there is no way to shatter there data points if $m > \log_2(|\mathcal{H}|)$, implying $VC(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$.*

6. [6 points] Prove that linear classifiers in a plane cannot shatter any 4 distinct points.

   [Proof] *We need to show that for any arrangement of four points, there exists at least one splitting (labeling) that can not be shattered by a line. The arrangement of 4 points can only be of the following forms: 1) One point lies within the convex hull of the other three points; 2) All 4 points are corner points of a convex hull. Consider two splittings shown in Fig. 1. It is clear the first splitting can not be shattered be a line since any line that correctly classifies the points with blue label with misclassify the red one. The second splitting is also not separable similar to the XOR function. Hence, linear classifiers in a plane can not shatter 4 points.*

7. [**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($\mathcal{H}$)?

   [Ans] *The VC dimension is 7. We prove this by showing that 1) there exists a set of $m = 7$ points that can be shattered with any label splitting and 2) there does not exist any set of $m = 8$ points that can be shattered.*
   Let us first consider $m = 7$ points where the 7 points are uniformly distributed on a circle. it is easy to check that for any subset of $g = 1, 2, 3, 4, 5, 6$ points with positive labels can be contained in a rectangle and the negative labels are outside the rectangle. Then



(a) Splitting for arrangement 1 (b) Splitting for arrangement 2

Figure 1: label splitting for two types arrangement of points.

we can show that there exists some specific label splitting containing 5 positive labels can not be separated for any arrangement of the 8 points. Hence, the VC dimension is 7.

2 Practice [60 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.

   [**Ans**] *Check out*: `https://github.com/xiangzhang-122/Machine_Learning`.

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.

   [**Ans:**] *The learned weight vector, and the average prediction error on the test* *dataset are (choose rate $r = 1$, the result is formulated as $[\boldsymbol{w}, b, error]$):*

$$\begin{aligned}
T &= 1, [array([-52.211826, -31.37905, -35.113615, -6.34541]), 45.0, 0.012] \\
T &= 2, [array([-35.026716, -22.52003, -26.856995, -10.4383856]), 30.0, 0.026] \\
T &= 3, [array([-42.803024, -28.17923, -32.24132, -12.649349]), 36.0, 0.012] \\
T &= 4, [array([-45.2239555, -30.81784, -35.3098926, -14.291459]), 43.0, 0.01] \\
T &= 5, [array([-45.5964275, -34.1163, -31.127735, -9.3000546]), 46.0, 0.014] \\
T &= 6, [array([-42.1330176, -34.23436972, -36.41576, -12.8472406]), 49.0, 0.008] \\
T &= 7, [array([-51.650622, -34.56892, -34.0090636, -11.878567]), 43.0, 0.014] \\
T &= 8, [array([-51.054411, -31.16544, -33.196195, -8.615503]), 48.0, 0.012] \\
T &= 9, [array([-59.876397, -31.89125, -40.500546, -10.562464]), 51.0, 0.008] \\
T &= 10, [array([-56.9034899, -43.21666, -39.9278306, -4.181065]), 57.0, 0.014]
\end{aligned}$$

(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.

[Ans:] *Refer to the text file* `part(b)_results.txt` *for the results of voted perceptron.*

(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.

[Ans:] *Refer to the text file* `part(c)_results.txt` *for the results of voted perceptron.*

(d) [10 points] Compare the average prediction errors for the three methods. What do you conclude?

[Ans:] *We got the error(%) table as follows:*

T	1	2	3	4	5	6	7	8	9	10
std	1.2	2.6	1.2	1.0	1.4	0.8	1.4	1.2	0.8	1.4
voted	3.6	2.2	1.6	1.6	1.4	1.4	1.4	1.4	1,4	1.4
aver.	6.8	2.8	1.8	2.0	1.6	1.8	1.8	1.6	1.8	1.4

Table 5: Error table for three types of perceptrons

It can be seen that (1) with the increase of number of epochs, the test error decreases. (2) In general, averaged perceptron has larger test error than the other two methods.