CIS5200: Machine Learning Spring 2023

Homework 4

Release Date: April 4, 2023 *Due Date: April 14, 2023*

- HW4 will count for 8% of the grade. This grade will be split between the written (30 points) and programming parts. The programming part will be released on April 7, 2023. Note that this will be shorter than previous homework, hence the shorter time span and lower total grade.
- All written homework solutions are required to be formatted using LaTeX. Please use the template here. Do not modify the template. This is a good resource to get yourself more familiar with LaTeX, if you are still not comfortable.
- You will submit your solution for the written part of HW4 as a single PDF file via Gradescope. The deadline is **11:59 PM ET**. Contact TAs on Ed if you face any issues uploading your homeworks.
- Collaboration is permitted and encouraged for this homework, though each student must understand, write, and hand in their own submission. In particular, it is acceptable for students to discuss problems with each other; it is not acceptable for students to look at another student's written Solutions when writing their own. It is also not acceptable to publicly post your (partial) solution on Ed, but you are encouraged to ask public questions on Ed. If you choose to collaborate, you must indicate on each homework with whom you collaborated.
- **Bonus Questions**: We have added two bonus questions in this homework for extra credit. These are intended to be more challenging than the non-bonus homework questions.

Please refer to the notes and slides posted on the website if you need to recall the material discussed in the lectures.

1 Written Questions (30 points + 6 bonus points)

Problem 1: Decision Trees (8 points)

Recall that when building a decision tree for classification problems, we used the notion of node purity instead of classification error when deciding how to grow the decision tree. Consider 3-dimensional binary data $x \in \mathbb{R}^3$ where the label $y \in \{-1, 1\}$ is generated from the first two features as $f(x) = x_1 \wedge \neg x_2$. In other words, the entire dataset of all possible examples can be written as:

x_1	x_2	x_3	y
0	0	0	-1
0	0	1	-1
0	1	0	-1
0	1	1	-1
1	0	0	1
1	0	1	1
1	1	0	-1
1	1	1	-1

1.1 (1 point) What is the best possible classification error that a 1-leaf decision tree (i.e. only the root node predicting a constant value) can achieve?

1.2 (2 points) Consider all three possible splits for growing to a 2-leaf decision tree. Does there exist a split that achieves lower classification error than a 1-leaf decision tree? If so, what is it? If not, why is that?

1.3 (1 point) What is the entropy and Gini index of the 1-leaf decision tree?

1.4 (2 point) Consider all three possible splits for growing to a 2-leaf decision tree. Does there exist a split that achieves lower entropy than a 1-leaf decision tree? Find the split that results in the lowest possible entropy, and calculate the corresponding entropy of the split.

1.5 (2 point) Consider all three possible splits for growing to a 2-leaf decision tree. Does there exist a split that achieves lower Gini index than a 1-leaf decision tree? Find the split that results in the lowest possible Gini index, and calculate the corresponding Gini index of the split.

Problem 2: PAC Learning (13 points + 3 bonus points)

2.1 (4 points) Consider a different version of PAC learning in which the learner is allowed access to distribution $\mathcal{D}_+$ of inputs with positive labels and $\mathcal{D}_-$ of inputs with negative labels. In this new PAC learning model, the learner has to perform well with respect to both $\mathcal{D}_+$ and $\mathcal{D}_-$, that is, we want the learner to output a predictor $\hat{f}$ such that: with probability $1-\delta$,

$$\Pr_{x\sim\mathcal{D}_+}[\hat{f}(x) \neq f(x)] \leq \epsilon \text{ and } \Pr_{x\sim\mathcal{D}_-}[\hat{f}(x) \neq f(x)] \leq \epsilon.$$

Show that if a function class if PAC learnable in the original model with sample complexity $m(\epsilon, \delta)$, then it is PAC learnable in the new model with sample complexity $2m(\epsilon, \delta)$.

2.2 (5 points) Consider a finite domain set $\mathcal{X} = \{x_1, \ldots, x_n\}$ of size n. Now consider the function class $\mathcal{F}_k := \{f : \mathcal{X} \to \{-1, 1\} : |\{x : f(x) = 1\}| = k\}$, that is, the set of all functions that assigns exactly k elements of $\mathcal{X}$ to the value 1 and the rest to -1. Compute the VC dimension of this class.

2.3 (4 points) Consider the function class of circles in a 2-dimensional plane, that is, $\mathcal{F} := \{x \mapsto \text{sign}\,(\|x-\mu\|_2 \leq R) : \mu \in \mathbb{R}^2, R \in \mathbb{R}, R \geq 0\}$. Compute the VC dimension of this class.

Bonus (3 points) In class we discussed how the number of parameters is a good proxy for the VC dimension of the class, for example, linear classifiers. However, this is not always true. In particular, the function class $\mathcal{F} = \{x \mapsto \text{sign}\,(\cos(\alpha x + \beta)) : \alpha, \beta \in \mathbb{R}\}$ with only two parameters has infinite VC dimension. Give a proof for this.

Problem 3: Boosting (9 points + 3 bonus points)

In this problem, we will further refine our understanding of the AdaBoost algorithm.

Note: Refer to the lecture notes for the notation.

3.1 (5 point) At iteration t, AdaBoost re-weighted the examples using the weak learner f_t, increasing the weights on points on which f_t was incorrect and decreasing the weight on which it is correct. Show that the error of f_t on the new distribution μ_{t+1} is as good as random guessing, in particular,

$$\Pr_{i \sim \mu_{t+1}} [f_t(x_i) \neq y_i] = 0.5.$$

3.2 (4 points) Recall the class of decision stumps (this time on scalar inputs $x \in \mathbb{R}$), that is, $\mathcal{F} = \{x \mapsto b \operatorname{sign}(x - \theta) : \theta \in \mathbb{R}, b \in \{-1, 1\}\}$. Show that the final classifier learned by this weak learning class $f(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t b_t \operatorname{sign}(x - \theta_t)\right)$ is equivalent to

$$g(x) = \sum_{r=1}^{R} \beta_r \mathbb{1}[x \in (\eta_{r-1}, \eta_r]]$$

for some $R > 0$, $\beta_r \in \mathbb{R}$ for $r \in [R]$ and $-\infty = \eta_0 < \eta_1 < \ldots < \eta_R = \infty$. *Note that g is a piece-wise constant function, that is, it takes a constant value in each interval.*

Bonus (3 points) Describe an algorithm to efficiently find f_t at each step t of AdaBoost using decision stumps (as described above) as the weak learning class. Also give the runtime of the algorithm in terms of m (the total number of training examples).