Theory Homework

CSE 493S/599S: Advanced Machine Learning
Instructor: Sewoong Oh

The goal of this homework is to help you better understand the ideas from theoretical machine learning we have covered in class.

Notes:

- You will be assigned a subset of the problems for each homework. Please submit the homework to Gradescope and link each page of your work to the corresponding problem.
- Please typeset your work using LaTeX.
- List every person with whom you discussed any problem in any depth, and every reference (outside of our course slides, lectures, and textbook) that you used.
- You may spend an arbitrary amount of time discussing and working out a solution with your listed collaborators, but **do not take notes, photos, or other artifacts of your collaboration**. Erase the board you were working on, and once you're alone, write up your answers yourself.
- The homework problems have been carefully chosen for their pedagogical value and hence might be similar or identical to those given out in similar courses at UW or other schools. Using any pre-existing solutions from these sources, from the Web or other textbooks constitutes a violation of the academic integrity expected of you and is strictly prohibited.

Version history:

V1 Initial version.

1 ERM and axis aligned rectangles (taught in the 1st theory lecture)

An axis aligned rectangle classifier in the plane is a classifier that assigns the value 1 to a point if and only if it is inside a certain rectangle. Formally, given real numbers $a_1 \leq b_1$, $a_2 \leq b_2$, define the classifier $h_{(a_1,b_1,a_2,b_2)}$ by

$$h_{(a_1,b_1,a_2,b_2)} = \begin{cases} 1 & \text{if } a_1 \leq x_1 \leq b_1 \text{ and } a_2 \leq x_2 \leq b_2 \\ 0 & \text{otherwise} \end{cases}.$$

The class of all axis aligned rectangles in the plane is defined as

$$\mathcal{H}^2_{\text{rec}} = \{h_{(a_1,b_1,a_2,b_2)} : a_1 \leq b_1 \text{ and } a_2 \leq b_2\}$$

Note that this is an infinite sized hypothesis class. Throughout this exercise we rely on the realizability assumption.

(1) Let A be the algorithm that returns the smallest rectangle $\hat{R}(S)$ enclosing all positive examples in the training set S. Show that $\hat{R}(S)$ minimizes the empirical risk. **[5 points]**

(2) Show that if A receives a training set S of size $\geq \frac{4\log(4/\delta)}{\epsilon}$ then, with probability of at least $1-\delta$ it returns a hypothesis with error of at most ϵ. **[10 points]**

Hint: Fix some distribution D over $\mathcal{X}$, let

$$R^* = R(a_1^*, b_1^*, a_2^*, b_2^*) \triangleq \{(x,y) \mid a_1^* \leq x_1 \leq b_1^*, a_2^* \leq x_2 \leq b_2^*\}$$

be the rectangle that generates the labels, and let f be the corresponding hypothesis. Let $a_1 \geq a_1^*$ be a number such that the probability mass (with respect to D) of the rectangle $R_1 = R(a_1^*, a_1, a_2^*, b_2^*)$ is exactly $\epsilon/4$. Similarly, let b_1, a_2, b_2 be numbers such that the probability masses of the rectangles $R_2 = R(b_1, b_1^*, a_2^*, b_2^*)$, $R_3 = R(a_1^*, b_1^*, a_2^*, a_2)$, $R_4 = R(a_1^*, b_1^*, b_2, b_2^*)$ are all exactly $\epsilon/4$. Let $\hat{R}(S)$ be the rectangle returned by A. See illustration in Fig. 1.

(a) Show that $\hat{R}(S) \subseteq R^*$.

(b) Show that if S contains (positive) examples in all of the rectangles R_1, R_2, R_3, R_4, then the hypothesis returned by A has error of at most ϵ.

(c) For each $i \in \{1, \ldots, 4\}$, upper bound the probability that S does not contain an example from R_i.

(d) Use the union bound to conclude the argument.

(3) Repeat the previous question for the class of axis aligned rectangles in $\mathbb{R}^d$. **[10 points]**

(4) Show that the runtime of applying the algorithm A mentioned in part (3) is polynomial in d, $1/\epsilon$, and in $\log(1/\delta)$. **[5 points]**

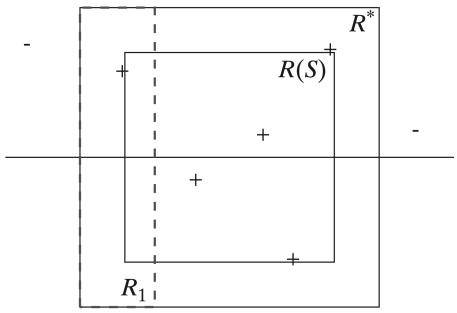

Figure 1: Axis aligned rectangles

[30 points]

---

2 The Bayes optimal predictor (taught in the 2nd theory lecture)

Show that for every probability distribution $\mathcal{D}$, the Bayes optimal predictor $f_{\mathcal{D}}$ is optimal, in the sense that for every classifier g from $\mathcal{X}$ to $\{0,1\}$, $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$.

Hint: For $x \in \mathcal{X}$, let α_x denote the conditional probability of a positive label given x. Show that $\mathbb{P}[f_{\mathcal{D}}(X) \neq y \,|\, X = x] = \min\{\alpha_x, 1-\alpha_x\}$ and that for any classifier $g : \mathcal{X} \to \{0,1\}$, we have $\mathbb{P}[g(X) \neq y \,|\, X = x] \geq \min\{\alpha_x, 1-\alpha_x\}$. Finally, conclude that $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$.

[20 points]

---

3 VC-dimension of axis aligned rectangles (taught in the 4th theory lecture)

Let $\mathcal{H}^d_{\text{rec}}$ be the class of axis aligned rectangles in $\mathbb{R}^d$. Prove that $\text{VCdim}(\mathcal{H}^d_{\text{rec}}) = 2d$.

[20 points]

---

4 Infinite VC-dimension with one parameter (taught in the 4th theory lecture)

It is often the case that the VC-dimension of a hypothesis class equals (or can be bounded above by) the number of parameters one needs to set in order to define each hypothesis in the class. For instance, if $\mathcal{H}$ is the class of axis aligned rectangles in $\mathbb{R}^d$, then $\text{VCdim}(\mathcal{H}) = 2d$, which is equal to the number of parameters used to define a rectangle in $\mathbb{R}^d$. Here is an example that shows that this is not always the case. We will see that a hypothesis class might be very complex and even not learnable, although it has a small number of parameters.

Consider the domain $\mathcal{X} = \mathbb{R}$, and the hypothesis class

$$\mathcal{H} = \{x \mapsto \lceil \sin(\theta x) \rceil : \theta \in \mathbb{R}\}$$

(here, we take $\lceil -1 \rceil = 0$). Prove that $\text{VCdim}(\mathcal{H}) = \infty$.

Hint: There is more than one way to prove the required result. One option is by applying the following lemma: If $0.x_1x_2x_3\ldots$, is the binary expansion of $x \in (0, 1)$, then for any natural number m, $\lceil \sin(2^m \pi x) \rceil = (1 - x_m)$, provided that $\exists k \geq m$ s.t. $x_k = 1$.

[30 points]

---

5 The perceptron (taught in the 6th theory lecture)

The perceptron is a classical algorithm for learning a separating hyperplane of a dataset $S = \{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \{-1, 1\}$. It is presented in the following psuedocode:

Algorithm 1: Perceptron

$w_0 \leftarrow 0$
while w_t *does not separate the data* **do**
  Select a random index $i_t \in \{1, \dots, n\}$
  if $y_{i_t} w_t^\top x_{i_t} < 1$ **then**
    $w_{t+1} \leftarrow w_t + y_{i_t} x_{i_t}$ `// correct a margin mistake`
  else
    $w_{t+1} \leftarrow w_t$
  $t \leftarrow t + 1$
end
return w_t

The perceptron happens to be equivalent to learning a linear separator using SGD and the hinge loss! In this problem, we will prove some famous results about the perceptron.

5.1 Mistake bound

First, we will show that the perceptron performs well on the training data (denoted S) using a *mistake bound*. In particular, we will show that if there exists a linear separator of the training data, then the perceptron will find it provided the margin of S is not too small.

The margin is first defined for a particular hyperplane $\mathcal{H}_w = \{x : w^\top x = 0\}$ corresponding to a vector $w \in \mathbb{R}^d$. Supposing that $\mathcal{H}_w$ perfectly separates S, we define the margin $\gamma(S, w)$ as the smallest distance between a point in S and a point in $\mathcal{H}_w$:

$$\gamma(S, w) = \text{dist}(S, \mathcal{H}_w)$$

where $\text{dist}(A, B) = \min(\|a - b\| : a \in A, b \in B)$.

The margin of S is then defined as the largest margin achievable by any w:

$$\gamma(S) = \max_{\|w\|=1} \gamma(S, w).$$

Additionally, define the diameter of S to be $D(S) = \max_{(x,y)\in S} \|x\|$.

Our goal in this section is to prove the following theorem:

Theorem 5.1. Algorithm 1 makes at most $(2 + D(S)^2)/\gamma(S)^2$ mistakes on any sequence of examples S that can be perfectly linearly separated.

The proof of this theorem can be broken into parts:

(1) First, we will upper bound $\|w_t\|$. In particular, show that

$$\|w_{t+1}\|^2 \leq \|w_t\|^2 + 2 + D(S)^2.$$

Now, let m_t be the total number of mistakes made by Algorithm 1 during the first t iterations. Use the previous result to show that

$$\|w_t\| \leq \sqrt{m_t(2 + D(S)^2)}.$$

[8 points]

(2) Next we will lower bound $\|w_t\|$. Start by showing that for any unit vector w that perfectly separates S, we have

$$\langle w, w_{t+1} - w_t \rangle \geq \gamma(S, w).$$

when we make a mistake at iteration t.

Let unit vector $w^\star$ denote the hyperplane achieving the maximum margin $\gamma(S)$. Use the previous result to show that

$$\langle w^\star, w_t \rangle \geq m_t \gamma(S).$$

Use this to obtain a lower bound for $\|w_t\|$. **[8 points]**

(3) Combine the two bounds to obtain a bound on the number of mistakes m_t. **[4 points]**

[total 20 points]

5.2 Generalization bound

Let us assume that the data S was drawn i.i.d. from a fixed underlying distribution $\mathcal{D}$ which is linearly separable. In the previous section, we saw that Algorithm 1 finds a linear predictor for S. Now we will show that this predictor also works on new data drawn from $\mathcal{D}$! In particular, use the result from the previous subsection to give a proof of the following theorem:

Theorem 5.2. Let S_n denote a set of n i.i.d. samples from $\mathcal{D}$. Let $w(S)$ be the output of Algorithm 1 on dataset S. Let $Z = (X, Y)$ be an additional independent sample from $\mathcal{D}$. Then,

$$\mathbb{P}[Y w(S_n)^\top X < 1] \leq \frac{1}{n+1} \mathbb{E}_{S_{n+1}} \left[\frac{2 + D(S_{n+1})^2}{\gamma(S_{n+1})^2} \right].$$

Hint: Note that:

(i) Z can be swapped with any entry of S_n without changing the distribution of outcomes.

(ii) If for some S, the perceptron never makes a mistake on example $s \in S$, then $w(S) = w(S \setminus s)$. This implies that $w(S \setminus s)$ will predict s correctly.

Use the mistake bound from earlier to show that the reasoning in (ii) will apply to many examples.

[20 points]
