Mathematics of Machine Learning
Homework 12 - Solutions

Instructors: Afonso S. Bandeira & Nikita Zhivotovskiy
Course Coordinator: Pedro Abdalla Teixeira

May 28, 2021

Try to solve the questions before looking to the answers. Every item must be proved rigorously. Starred problems are harder.

Problem 1

Let $L^* := \inf_{f: X \to \{0,1\}} \mathbb{P}(f(X) \neq Y)$ denotes the Bayes error and let f^* be the optimal Bayes classifier.

(a) Write L^* in terms of $\eta(x) = \mathbb{P}(Y = 1 | X = x)$.

(b) Let $\tilde{\eta}(x)$ be a nonnegative function that ε-approximates $\eta(x)$ in the $L1$-sense, precisely $\mathbb{E}|\eta(X) - \tilde{\eta}(X)| \leq \varepsilon$. Prove that the classifier f defined as

$$f(x) := \begin{cases} 1, \tilde{\eta}(x) \geq \frac{1}{2} \\ 0, \text{otherwise} \end{cases}$$

satisfies the following error bound

$$\mathbb{P}(f(X) \neq Y) - L^* \leq 2\varepsilon.$$

Solution 1

(a) Observe that $L^* = \mathbb{E}_X \mathbb{P}(f(x) \neq Y | X = x)$. By the Bayes optimal estimator, $\mathbb{P}(f(x) \neq Y | X = x) = \min\{\eta(x), 1 - \eta(x)\}$, therefore we can write $L^* = \mathbb{E}_X \min\{\eta(X), 1 - \eta(X)\}$. We can simplify a bit by using that $\min(a, b) = \frac{a+b-|a-b|}{2}$, so the Bayes risk becomes $L^* = \frac{1}{2}(1 - \mathbb{E}|2\eta(X) - 1|)$.

(b) We start by writing

$$\begin{aligned}
&\mathbb{P}(f(x) \neq Y | X = x) - \mathbb{P}(f^*(x) \neq Y | X = x) \\
&= (2\eta(x) - 1)(\mathbb{1}_{f^*(x)=1} - \mathbb{1}_{f(x)=1}) \\
&= (2\eta(x) - 1)\mathbb{1}_{f(x) \neq f^*(x)}
\end{aligned}$$

So we take the expectation on both sides to obtain

$$\mathbb{P}(f(X) \neq Y) - L^* = \mathbb{E}_X (2\eta(X) - 1)\mathbb{1}_{f(x) \neq f^*(x)} \leq 2\mathbb{E}|\eta(X) - \tilde{\eta}(X)|.$$

We used that if $f^*(x) \neq f(x)$ then $|\eta(x) - \frac{1}{2}| \leq |\eta(x) - \tilde{\eta}(x)|$.

Problem 2

In the lecture, we proved that the Halving algorithm does not make more than $\log_2 |\mathcal{F}|$ mistakes, where $|\mathcal{F}|$ denotes the size of the class. Prove that this bound is tight in the following sense: for any integer n there is a class of size 2^n such that halving algorithms makes at least n mistakes when run on a particular sequence.

Solution 2

We define, for $j \in \{1, \dots, 2^n\}$, the classifier f_j by

$$f_j(x) := \begin{cases} -1, x \geq j \\ 1, x < j \end{cases}$$

We assume an adversarial situation, when there is a draw the algorithm returns the wrong answer. Suppose the input class is $\mathcal{F}$ formed by all classifiers f_j, clearly $|\mathcal{F}| = 2^n$. Observe that if the first sample is $(x_1, y_1) = (2^{n-1}, 1)$, the majority vote is -1 and, by the observation above, we have a mistake. Now if the second sample is $(x_2, y_2) = (2^{n-2}, 1)$, the majority vote is -1, we have a mistake again. We proceed this way, it is easy to check by induction that the algorithm makes mistakes unless we have only one possible classifier in the remaining class, so we need to make k mistakes such that $2^{n-k} \leq 1$, i.e, we have $k \geq n$ mistakes. Since the number of mistakes k is at most n, we finish the proof.

Problem 3

Prove that the Halving algorithm determines a sample compression scheme of finite size for any finite class. Give an upper bound on its sample complexity in the PAC learning setup. Does the output classifier belong to $\mathcal{F}$?

Solution 3

The sample compression scheme can be constructed as follows: The compression function saves all points, in the default order, in which the Halving algorithm made a mistake. The reconstruction function is defined as follows: For an input point x, the Halving algorithm runs for all points in the compressed set that appear before (with respect to the default order) the point x. The above sample-compression scheme has size at most $\log_2 |\mathcal{F}|$.
We know that sample compression schemes imply in PAC learnability, so we just apply Theorem 14.5 in the lecture notes, observe that, since we fixed an order, the reconstruction function is permutation invariant, so we guarantee that the hypothesis in Theorem 14.5 holds. We conclude that the sample complexity $n(\varepsilon, \delta)$ in the PAC learning setup is at most $l + \left\lceil \frac{l}{\varepsilon} \log(\frac{en}{l}) + \frac{1}{\varepsilon} \log(\frac{1}{\delta}) \right\rceil$ where $l := \log_2 |\mathcal{F}|$. Finally observe that the output classifier does not necessarily belong to the class $\mathcal{F}$, because the majority vote classifier may not belong to the class.