Introduction to Machine Learning (67577)

Exercise 4
PAC & Ensemble Methods

Second Semester, 2022

Contents

1 Submission Instructions

Please make sure to follow the general submission instructions available on the course website. In addition, for the following assignment, submit a single `ex4_ID.tar` file containing:

- An `Answers.pdf` file with the answers for all theoretical and practical questions (include plotted graphs *in* the PDF file).
- The following python files (without any directories): `decision_stump.py`, `adaboost.py`, `adaboost_scenario.py`

The `ex4_ID.tar` file must be submitted in the designated Moodle activity prior to the date specified *in the activity*.

- Late submissions will not be accepted and result in a zero mark.
- Plots included as separate files will be considered as not provided.
- Do not forget to answer the Moodle quiz of this assignment.

2 Theoretical Part

2.1 PAC Learnability

1. For $\mathcal{A}$ some learning algorithm, $\mathcal{D}$ a probability distribution over $\mathcal{X}$ and the 0-1 loss function (i.e misclassification), prove the following are equivalent:

$$\begin{array}{ll} (a) & \forall \varepsilon, \delta > 0 \quad \exists m(\varepsilon,\delta) \quad \text{s.t} \quad \forall m \geq m(\varepsilon,\delta) \quad \mathbb{P}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(\mathcal{A}(S)) \leq \varepsilon] \geq 1-\delta \\ (b) & \lim_{m\to\infty} \mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] = 0 \end{array}$$

Hints:

- For $(a) \Rightarrow (b)$ show that $\forall \varepsilon, \delta > 0$ and $\forall m \geq m(\varepsilon,\delta)$ it holds that$\mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \leq \varepsilon + \delta$.
- For $(b) \Rightarrow (a)$ use Markov's inequality

2. Let $\mathcal{X} := \mathbb{R}^2$, $\mathcal{Y} := \{0,1\}$ and let $\mathcal{H}$ be the class of concentric circles in the plane, i.e.,

$$\mathcal{H} := \{h_r : r \in \mathbb{R}_+\} \quad \text{where} \quad h_r(\mathbf{x}) = \mathbb{1}_{[||\mathbf{x}||_2 \leq r]}$$

Prove that $\mathcal{H}$ is PAC-learnable and its sample complexity is bounded by

$$m_{\mathcal{H}}(\varepsilon,\delta) \leq \frac{\log(1/\delta)}{\varepsilon}$$

When proving, *do not* use a VC-Dimension argument. Instead prove the claim directly from the PAC learnability definition by showing a specific algorithm and analyzing its sample complexity.
Hint: Remember that for every $\varepsilon > 0$ it holds that $1-\varepsilon \leq e^{-\varepsilon}$

2.2 VC-Dimension

3. Let $\mathcal{X} = \{0,1\}^n$ and $\mathcal{Y} = \{0,1\}$, for each $I \subseteq [n]$ define the parity function:

$$h_I(\mathbf{x}) = \Big(\sum_{i\in I} x_i\Big) \bmod 2.$$

What is the VC-dimension of the class $\mathcal{H}_{parity} = \{h_I \mid I \subseteq [n]\}$? Prove your answer. You may use results from question 3.

4. Given an integer k, let $([a_i,b_i])_{i=1}^{k}$ be any set of k intervals on $\mathbb{R}$ and define their union $A = \cup_{i=1}^{k}[a_i,b_i]$. The hypothesis class $\mathcal{H}_{k-intervals}$ includes the functions: $h_A(x) := \mathbb{1}_{[x\in A]}$, for all choices of k intervals. Find the VC-dimension of $\mathcal{H}_{k-intervals}$ and prove your answer. Show that if we let A be any finite union of intervals (i.e. k is unlimited), then the resulting class $\mathcal{H}_{intervals}$ has VC-dimension ∞.

2.3 Monotonicity

5. Let $\mathcal{H}$ be a hypothesis class for a binary classification task. Suppose that $\mathcal{H}$ is PAC learnable and its sample complexity is given by $m_{\mathcal{H}}(\cdot,\cdot)$. Show that $m_{\mathcal{H}}$ is monotonically non-increasing in each of its parameters. That is:
   - Show that given $\delta \in (0,1)$, and given $0 < \varepsilon_1 \leq \varepsilon_2 < 1$, we have that $m_{\mathcal{H}}(\varepsilon_1,\delta) \geq m_{\mathcal{H}}(\varepsilon_2,\delta)$.
   - Similarly, show that given $\varepsilon \in (0,1)$, and given $0 < \delta_1 \leq \delta_2 < 1$, we have that $m_{\mathcal{H}}(\varepsilon,\delta_1) \geq m_{\mathcal{H}}(\varepsilon,\delta_2)$.
6. Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two classes for binary classification, such that $\mathcal{H}_1 \subseteq \mathcal{H}_2$. Show that $VC-dim(\mathcal{H}_1) \leq VC-dim(\mathcal{H}_2)$.

2.4 Agnostic-PAC

7. Prove that if $\mathcal{H}$ has the uniform convergence property with function $m_{\mathcal{H}}^{UC} : (0,1)^2 \to \mathbb{N}$, then $\mathcal{H}$ is Agnostic-PAC learnable with sample complexity $m_{\mathcal{H}}(\varepsilon,\delta) \leq m^{UC}(\varepsilon/2,\delta)$.
8. Let $\mathcal{H}$ be a hypothesis class over a domain $\mathcal{Z} = \mathcal{X} \times \{\pm 1\}$, and consider the 0-1 loss function. Assume that there exists a function $m_{\mathcal{H}}$, for which it holds that for every distribution $\mathcal{D}$ over $\mathcal{Z}$ there is an algorithm $\mathcal{A}$ with the following property: when running $\mathcal{A}$ on $m \geq m_{\mathcal{H}}$ i.i.d. examples drawn from $\mathcal{D}$, it is guaranteed to return, with probability at least $1-\delta$, a hypothesis $h_S : \mathcal{X} \to \{\pm 1\}$ with $L_{\mathcal{D}}(h_S) \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \varepsilon$. Is $\mathcal{H}$ agnostic PAC learnable? Prove or show a counter example.

3 Practical Part

In the following part you will implement a decision stump (decision tree of depth 1) and the AdaBoost algorithm. Be sure to have pulled the latest version of the GreenGilad/IML.HUJI repository.

3.1 Boosting - Separate the inseparable

- Implement the `DecisionStump` class in the `decision_stump.py` file as described in class documentation.
- Implement the `AdaBoost` class in the `learners\metalearners\adaboost.py` file as described in class documentation. When implementing the `predict` and `partial_predict`, and the `loss` and `partial_loss` functions avoid code duplication.

Then, implement the in the `adaboost_scenario.py` file and answer the following:

1. Use the provided `generate_data` function to generate 5000 train samples and 500 test samples, both with no noise (i.e. `noise_ratio = 0`). Train an Adaboost ensemble of size 250 (i.e passing 250 as the number of iterations) using your implementation of the `DecisionStump` as weak learner. Plot, in a single figure, the training- and test errors as a function of the number of fitted learners. Explain your results
2. Using the previously fitted ensemble, plot the decision boundary obtained by using the the ensemble up to iteration 5, 50, 100 and 250. Use your implementation of the `partial_predict` to obtain the predictions of the ensemble up to the specified size. In each of these plots also add the test set (colored and/or shaped by the actual labels). Explain your results.

   Are can use the `decision_surface` function in `IML.HUJI\utils.py`. Revisit lab 05 of comparing classifiers to see how to create such plots.
3. Using the previously fitted ensemble, which ensemble size achieved the lowest test error? Plot the decision surface of that ensemble as well as the test set data points. In the plot's title provide the ensemble size and its accuracy.
4. Using the previously fitted ensemble, use the weights of the last iteration (i.e. $\mathcal{D}^T$) to plot the *training* set with a point size proportional to it's weight and color (and/or shape) indicating its label.
   - As previously, plot the decision surface (using the full ensemble).
   - As the weights are of very small numbers normalize and transform then as follows: `D = D/np.max(D) * 5`

   Explain your results, and specifically explain which samples are "easier" and which are "challenging" for the classifier.
5. Repeat the steps above (while avoiding code repetition) for train- and test sets generated with noise levels of 0.4. Show graphs as in and . Explain the results. In your answer explain what is seen in the plot of the loss in terms of the bias-variance tradeoff.