EE 518: Machine Learning Theory & Algorithms **Spring 2024**

Homework 3
Due: May 3, 2024, 11:59PM PT

Student Name: *Instructor Name: John Lipor*

Problem 1 AdaBoost with Decision Stumps (10 pts, 5 pts, 5 pts)

Your task in this problem will be to implement the AdaBoost algorithm using decision stumps (DS) as your base classifier. You will test your implementation on both synthetic data and the Titanic dataset.

(a) Implement ERM for decision stumps by completing the `DecisionStump` class. Note that the description in the textbook only accounts for the case where $b = +1$ but you need to account for $b = -1$ as well. **Turn in** your code, your training error on **both** datasets generated in `prob1a.py`, and a scatter plot of each dataset with color corresponding to your predicted label.

(b) Implement AdaBoost with DS as the base classifier. **Turn in** your code, training error, number of boosting rounds needed to get this error, and scatter plots for **both** datasets generated in `prob1b.py`.

(c) Apply your implementation of AdaBoost to your processed data from MP1, selecting the parameters however you would like. Report the error for both your best linear classifier and AdaBoost.

Problem 2 DSS: XGBoost (5 pts each)

(*DSS rules apply.*) Aside from `sklearn`, the other go-to package for algorithms that don't involve neural networks is `XGBoost`. Note that in this case the "boost" stands for gradient boosting, which is another approach to ensemble learning with decision trees. XGBoost is extremely popular, both because it achieves excellent empirical performance and very low computational complexity. It is fair to say that XGBoost is a good first choice of nonlinear method for classification or regression. In this problem, you will apply XGBoost to the MNIST dataset.

(a) Apply XGBoost to the MNIST dataset using a multiclass softmax loss. For an initial validation, use 10 rounds of boosting and a max depth of 2. **Turn in** your training and test error and which API you used.

(b) Your above error should be worse than what you achieved with the multiclass LR from Homework 2. Vary the parameters to try to get an improvement over LR. **Turn in** your ultimate training and test error, as well as a description of each parameter that you changed and why you think this setting resulted in an improvement.

(c) Apply XGBoost to your processed data from MP1, selecting the parameters however you would like. Report the error for both your best linear classifier and XGBoost.

Problem 3 VC-Dimension of Decision Stumps (5 pts)

Prove that the VC-dimension of decision stumps on $\mathcal{X} = \mathbb{R}$ is 2. DSS rules do not apply.

Problem 4 AdaBoost Step Size (10 pts)

AdaBoost and XGBoost can both be viewed through the more general framework of functional gradient descent. In this case, the loss we aim to minimize is

$$F(w) = \sum_{i=1}^{m} e^{-y_i \sum_{t=1}^{T} w_t h_t(x_i)},$$

where $w \in \mathbb{R}^T$ is a weight vector. Letting $f_T = \frac{1}{m}\sum_{t=1}^{T} w_t h_t$, we see that this is exactly the term that we upper bounded in the proof of Thm. 10.2. Since we proceed one coordinate of w at a time, our goal at round $t+1$ is to minimize

$$F(w_{t+1}) = \sum_{i=1}^{m} e^{-y_i (f_t(x_i) + w_{t+1} h_{t+1}(x_i))}. \tag{1}$$

Prove that the minimum of (1) yields the AdaBoost weight

$$w_{t+1} = \frac{1}{2} \log\left(\frac{1}{\varepsilon_{t+1}} - 1\right).$$

Hint: Several steps from the proof of Thm. 10.2 are useful in simplifying the derivative of the above.