DS 5220: Supervised Machine Learning and Learning Theory (Spring 2023) Dr. Roi Yehoshua
Student name: (Due) March 30, 2023

PS3: Decision Trees, Ensemble Methods

1 Decision Trees on a Toy Dataset (20%)

Master Yoda is concerned about the number of Jedi apprentices that have turned to the Dark Side, so he's decided to train a decision tree on some historical data to help identify problem cases in the future. The following table summarizes whether or not each of the 12 initiates turned to the Dark Side based on their age when their Jedi training began, whether or not they completed their training, their general disposition, and their species.

Dark Side	Age Started Training	Completed Training	Disposition	Species
0	5	1	Happy	Human
0	9	1	Happy	Gungan
0	6	0	Happy	Wookiee
0	6	1	Sad	Mon Calamari
0	7	0	Sad	Human
0	8	1	Angry	Human
0	5	1	Angry	Ewok
1	9	0	Happy	Ewok
1	8	0	Sad	Human
1	8	0	Sad	Human
1	6	0	Angry	Wookiee
1	7	0	Angry	Mon Calamari

1. What is the initial entropy of Dark Side?
2. Which attribute would the decision tree building algorithm choose to use for the root of the tree?
3. What is the information gain of the attribute you chose to split on in the previous question?
4. Draw the full decision tree that would be learned for this data (with no pruning).
   Hint: The correct split at each point should be clear from just the groups it splits the data into, without having to actually compute the information gain for each possible split.

2 Decision Trees Implementation (30%)

1. Implement the decision tree learning algorithm described in class (without post-pruning). Your program can assume that all features/attributes are real-valued attributes (no categorical attributes), except for the target attribute that is discrete and may take on a finite set of possible values. You may also assume that the data contains no missing attributes. Use information gain as your splitting criterion and allow only binary splits.

   Your algorithm should accept an optional parameter that sets the limit of the depth of the tree that will be generated when training. If the depth limit has not been set, the entire tree should be learned. If the depth limit has been set, only nodes up to that depth limit should be built.

2. Run your algorithm on the Iris flower data set (available at `sklearn.datasets`), and display the resulting decision tree.

3. Run `DecisionTreeClassifier` from scikit-learn on the same data set, and compare its generated decision tree to the one built by your algorithm.

3 AdaBoost on a Toy Dataset (20%)

Now we will apply AdaBoost to classify a toy data set. Consider the data set shown in Figure 1a. The data set consists of 4 points: $(0, -1, -), (1, 0, +), (-1, 0, +)$ and $(0, 1, -)$. For this part, you may find it helpful to use python as a calculator rather than doing the computations by hand. (You do not need to submit any code though.)

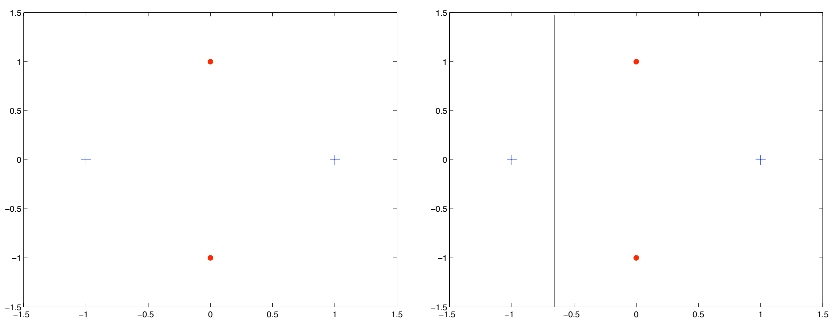

Figure 1: a) Toy data set. b) Decision stump h_1.

1. For $T = 4$, show how AdaBoost works for this data set, using simple decision stumps (depth-1 decision trees that simply split on a single variable once) as weak classifiers. For each time step compute the following:

$$\epsilon_t, \alpha_t, Z_t, D_t(i) \ \forall i$$

   Also for each time step draw your weak classifier. For example, h_1 can be as shown in Figure 1b.

2. What is the training error of AdaBoost for this toy data set?

3. Is the above data set linearly separable? Explain why AdaBoost does better than a decision stump on the above data set.

4 Training Error of Boosting (30%)

Consider the AdaBoost algorithm you saw in class. In this question we will try to analyze the training error of boosting.

1. Given a set of n examples, $(\mathbf{x}_i, y_i)$ (y_i is the class label of $\mathbf{x}_i$), $i = 1, \ldots, n$, let $h_t(\mathbf{x})$ be the weak classifier obtained at step t, and let α_t be its weight. Recall that the final classifier is

$$H(\mathbf{x}) = \mathrm{sign}(f(\mathbf{x})), \text{ where } f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}).$$

Show that the training error of the final classifier can be bounded from above by an exponential loss function:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(H(\mathbf{x}_i) \neq y_i) \leq \frac{1}{n}\sum_{i=1}^{n} \exp(-f(\mathbf{x}_i)y_i), \tag{1}$$

where $\mathbb{1}(a = b)$ is the indicator function. It is equal to 1 if $a = b$, and 0 otherwise.

Hint: $e^{-x} \geq 1 \Leftrightarrow x \leq 0$.

2. Remember that

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{Z_t}, \tag{2}$$

where Z_t is the normalization factor for distribution D_{t+1}:

$$Z_t = \sum_{i=1}^{n} D_t(i)\exp(-\alpha_t y_i h_t(\mathbf{x}_i)) \tag{3}$$

Use this recursive definition to prove the following:

$$\frac{1}{n}\sum_{i=1}^{n} \exp(-f(\mathbf{x}_i)y_i) = \prod_{t=1}^{T} Z_t. \tag{4}$$

Hint: Remember that $e^{\sum_i g_i} = \prod_i e^{g_i}$, $D_1(i) = \frac{1}{n}$, and that $\sum_i D_{t+1}(i) = 1$.

3. Equation (4) suggests that the training error can be reduced rapidly by greedily optimizing Z_t at each step. You have shown that the error is bounded from above:

$$\epsilon_{training} \leq \prod_{t=1}^{T} Z_t.$$

Observe that $Z_1, \ldots, Z_{t-1}$ are determined by the first $(t-1)$ rounds of boosting, and we cannot change them on round t. A greedy step we can take to minimize the training error bound on round t is to minimize Z_t.

In this question, you will prove that for binary weak classifiers, Z_t from Equation (3) is minimized by picking α_t as:

$$\alpha_t^* = \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right), \tag{5}$$

where ϵ_t is the training error of weak classifier h_t for the weighted data set:

$$\epsilon_t = \sum_{i=1}^{n} D_t(i)\mathbb{1}(h_t(\mathbf{x}_i) \neq y_i).$$

For this proof, only consider the simplest case of binary classifiers, i.e. the output of $h_t(x)$ is binary, $\{-1, +1\}$.

For this special class of classifiers, first show that the normalizer Z_t can be written as:

$$Z_t = (1-\epsilon_t)\exp(-\alpha_t) + \epsilon_t \exp(\alpha_t).$$

Hint: Consider the sums over correctly and incorrectly classified examples separately.

4. Now, prove that the value of α_t that minimizes this definition of Z_t is given by Equation (5).

5. Prove that for the above value of α_t we have $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$.

6. Furthermore, let $\epsilon_t = \frac{1}{2} - \gamma_t$ and prove that $Z_t \leq \exp(-2\gamma_t^2)$. Therefore, we will know

$$\epsilon_{training} \leq \prod_t Z_t \leq \exp(-2\sum_t \gamma_t^2). \tag{6}$$

   Hint: $\log(1-x) \leq -x$ for $0 < x \leq 1$.

7. If each weak classifier is slightly better than random, so that $\gamma_t \geq \gamma$, for some $\gamma > 0$, then the training error drops exponentially fast in T, i.e.

$$\epsilon_{training} \leq \exp(-2T\gamma^2).$$

   Argue that in each round of boosting, there always exists a weak classifier h_t such that its training error on the weighted data set $\epsilon_t \leq 0.5$.

8. Show that for $\epsilon_t = 0.5$ the training error can get "stuck" above zero. *Hint*: $D_t(i)$s may get stuck.