ECE523: Engineering Applications of Machine Learning and Data Analytics

I acknowledge that this exam is solely my effort. I have done this work by myself. I have not consulted with others about this exam in any way. I have not received outside aid (outside of my own brain) on this exam. I understand that violation of these rules contradicts the class policy on academic integrity.

Name: _____

Signature: _____

Date: _____

Instructions: There are four problems. You have 50 minutes to complete the exam. Partial credit is given for answers that are partially correct. No credit is given for answers that are wrong or illegible. Write neatly.

Problem 1: _____

Problem 2: _____

Problem 3: _____

Problem 4: _____

Total: _____

Problem #1 – Support Vector Machines (10 Points)

Using a kernel function is equivalent to mapping data into a higher dimensional space then taking the linear dot product in that space. (1) Please explain what is the advantage of using the kernel function compared to doing the explicit mapping. (2) Given that we have an expression for $\mathbf{w}$, do we still need to explicitly compute the mapping at the time of classification? Why or why not? (3) What is a support vector? Feel free to support your conclusions with equations.

Problem #1 cont'd

Problem #2 – AdaBoost (15 Points)

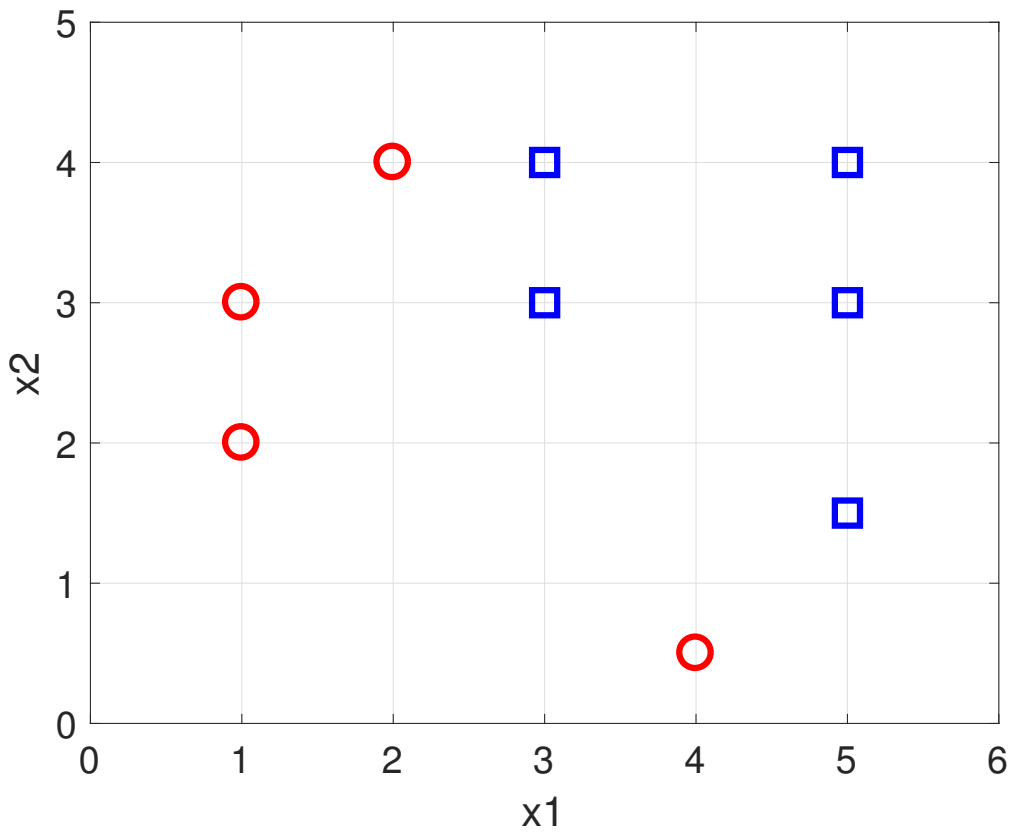

Figure 1: Labeled training points for Problem 2.

Consider the labeled training points in Figure 1, where ∘ and □ denote positive and negative labels, respectively. We wish to apply AdaBoost with a threshold classifier (i.e., pick an axis then pick a threshold to label the data). In each boosting iteration, we select the threshold that minimizes the weighted training error, breaking ties arbitrarily. Use the AdaBoost pseudo-code to help with this question

1. In Figure 1, draw a decision boundary on x_1-axis (i.e., vertical line) corresponding to the first threshold that the boosting algorithm could choose. Label this boundary (1), and also indicate +/- side of the decision boundary.

2. In the same figure also circle the point(s) that have the highest weight after the first boosting iteration.

3. What is the weighted error of the first threshold after the first boosting iteration, i.e., after the points have been re-weighted?

4. Draw a decision boundary corresponding to the second threshold using the weights, again in Figure , and label it with (2), also indicating the +/- side of the boundary. For clarity grading exams draw a decision boundary on x_1 (i.e., vertical line).

Problem #3 – Random Short Answer (15 Points)

(SA:1) AdaBoost has a weight for every sample in the training data set, $\mathcal{D}_t(i)$ (where t is the learning round on the ith instance), and we said that this weight: (a) provided a measure of the relative difficulty to classify the point; and (b) was used to build the next classifier h_{t+1}. Describe two methods how these weights can be used to train h_{t+1}.

(SA:2) What is an appropriate way to train a deep neural network? The key word in that sentence is "appropriate".

(SA:3) AdaBoost requires that the error of a weak classifier is at least better than 50%. (1) Why is this requirement needed? (2) What are some of the issues associated with requiring 50% accuracy on any arbitrary data set? Recall that

$$\varepsilon_t = \sum_{i=1}^{n} \mathcal{D}_t(i) \mathbb{1}_{h_t(\mathbf{x}_i) \neq y_i}$$

where $\mathbb{1}$ is the indicator function (i.e., $\mathbb{1}_{\text{expression}}$ is 1 if the expression is true otherwise it is zero).

(SA:4) In the context of multi-armed bandits, what is regret?

Problem #4 – True/False: A Gamblers Ruin (10 Points)

[True/False] (1 point): In class, we derived the classifier weights for AdaBoost then in a later lecture, we derived a set of Bayes optimal weights for a generic ensemble of classifiers, which resulted in the same way to calculate classifier voting weights. To quite different approaches and the same conclusion!

[True/False] (1 point): Bagging works because it assumes the base models have low variance and high bias. That is we do not need base classifiers that are unstable.

[True/False] (1 point): The multi-armed bandit address problems that require exploration of new arms and exploitation of the ones we know perform well.

[True/False] (1 point): The weight correction expression for backpropagation in a neural network are of the same form for both hidden and output nodes. The only difference is how the local gradient is determined.

[True/False] (1 point): One of the disadvantages of deep learning with auto-encoders is that we need a large volume of labeled data to train each layer.

[True/False] (1 point): The first implementation of passive aggressive online learning updates the weights of a linear model at every time step; however, the update is much more aggressive if the model made a mistake.

[True/False] (1 point): A neural network will (likely) find a local minimum for its optimization problem and the same is true for a support vector machine.

[True/False] (1 point): AdaBoost works in successive rounds by having classifiers focus more on the data points that have been misclassified by previous classifiers.

[True/False] (0 point): To the student who answered False last time on the question about free points: No more free points!

[True/False] (1 point): The theory behind AdaBoost proves that the error on the testing data is upper bouned by

$$\widehat{\text{err}}(H) \leq 2^T \prod_{t=1}^{T} \sqrt{\varepsilon_t(1-\varepsilon_t)}$$

[Accept/Reject] (1 point): "I decided to use the UCB1 multi-armed bandit algorithm for the task of selecting buyers in stock. We felt this is the best selection since the problem relies on modeling consumer interests, which we assumed to be a stationary problem."

Cheat Sheet

Algorithm 1 Adaboost (Adaptive Boosting)

Input: $\mathcal{S} := \{x_i, y_i\}_{i=1}^n$, learning rounds T, and hypothesis class $\mathcal{H}$

Initialize: $\mathcal{D}_1(i) = 1/n$

1: **for** $t = 1, \ldots, T$ **do**
2: $h_t = \arg\min_{h \in \mathcal{H}} \widehat{\text{err}}(h, \mathcal{S}, \mathcal{D}_t)$
3: $\epsilon_t = \sum \mathcal{D}_t(i) \mathbb{1}_{h(\mathbf{x}_i) \neq y_i}$
4: $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
5: $\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i)}{Z_t} \exp\left(-\alpha_t y_i h_t(x_i)\right)$
6: **end for**
7: **Output:** $H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$

Scratch Paper