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 five problems. You have 50 minutes to complete the exam. You may use handwritten notes on both sides of one 8.5" × 11" piece of paper. Use of any other notes, textbooks, or any other form of outside help is strictly forbidden. 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: _____

Problem 5: _____

Total: _____

Problem #1 – Feature Selection / Miscellaneous (10 Points)

Consider a dataset with 1000 features total. 50 of them are truly informative about class. Another 50 features are almost direct copies of the first 50 features. The final 900 features are not informative. Assume there is enough data to reliably assess how useful features are. Use this information to answer the questions below and provide your reasoning about how you came to your decision.

Question 1 How many features will be selected by mutual information filtering?

Question 2 How many features will be selected by using an approach such as mRMR?

Question 3 Consider k-fold cross-validation. Let us consider the tradeoffs of larger or smaller k (the number of folds). With a higher number of folds, the estimated error will be, on average higher, lower, about the same, or don't know.

Question 4 Consider the logistic regression classifier with an output of $\sigma(\mathbf{x}) = 1/(1 + \exp(-\mathbf{w}^\mathsf{T}\mathbf{x}))$. We minimize the cross-entropy function plus a term for L_2 regularization on $\mathbf{w}$. Is this still a convex optimization problem?

Question 5 Suppose you are dealing with 4 class classification problem and you want to train a SVM model on the data for that you are using One-vs-all method. How many times we need to train our SVM model in such case?

Problem #2 – Boosting (10 Points)

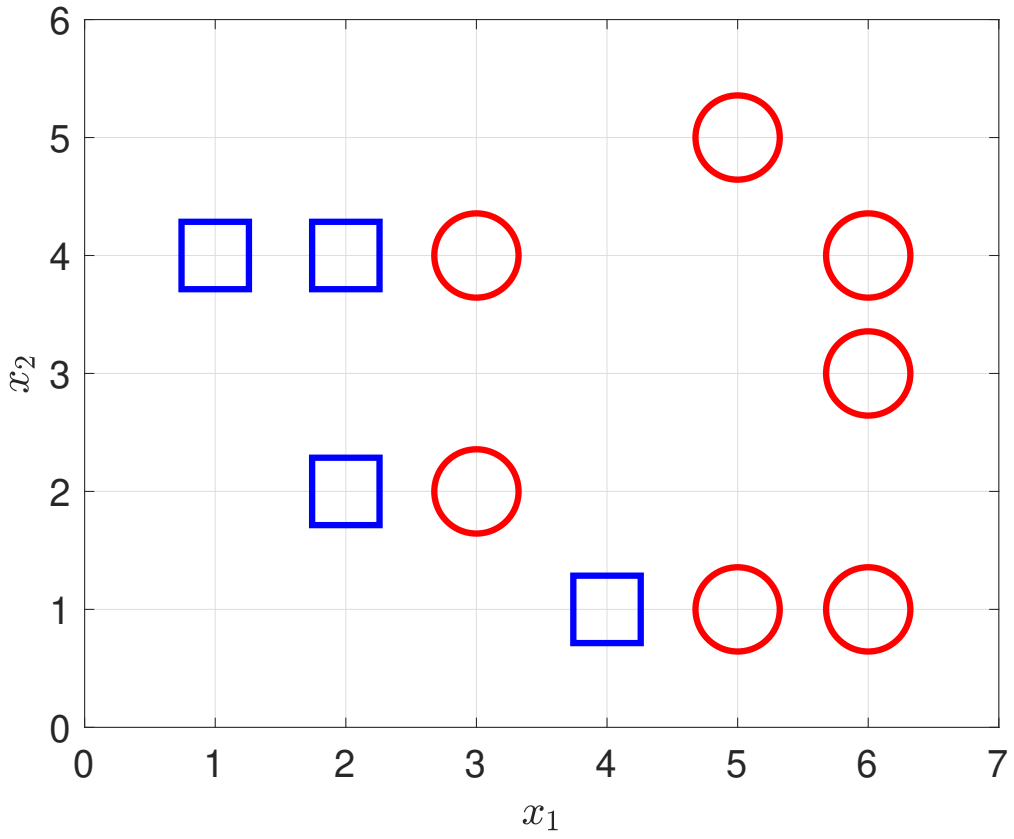

Figure 1: Labeled training points for Problem 2.

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. *Hint: Find the vertical line that will give you the fewest errors.* Also, note there are two solutions to this question.

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 1, and label it with (2), also indicating the +/- side of the boundary. For clarity grading exams draw a decision boundary on x_2 (i.e., horizontal line).

Problem #3 – A Gamblers Ruin (10 Points)

[True/False] (1 point): Adaboost forces classifiers that are learned to focus more on the classifiers that the ensemble incorrectly classified.

[True/False] (1 point): Increasing the term C in a support vector machine will decrease the number of support vectors.

[True/False] (1 point): Parzen windows are one way to estimate the density, $p_n(x)$, but they cannot be used in classification.

[True/False] (1 point): Cross validation can be used to select the number of iterations in boosting; this procedure may help reduce overfitting.

[True/False] (1 point): Regularization is one way to prevent overfitting and the reason it is so effective is because the regularization term is data-independent. Therefore, the optimization process will "find" the best way to be resilient against overfitting.

[True/False] (1 point): Parzen windows estimate the density of a dataset by growing a volume V until there are k samples that are enclosed on the region $\mathcal{R}$ with volume V.

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

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

[True/False] (1 point): The support vectors in the context of an SVM are the $\mathbf{x}_i \in \mathcal{D}_{\text{train}}$ (i.e., data set) that correspond to $\alpha_i \neq 0$.

$$\max_{\alpha} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^{\mathsf{T}} \mathbf{x}_j \right\}$$
$$\text{s.t.} \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \text{ and } 0 \leq \alpha_i \leq C$$

[Accept/Reject] (1 point): "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for best value of λ, chosen with 10-fold cross validation.)"

[Accept/Reject] (1 point): We did not standardize the features of our data and we did 10-fold cross validation with a k-NN classifier.

Problem #4 – Short Answer (10 Points)

Question 1 SVMs tend to be robust to irrelevant features. Suppose we run SVMs with features $X1, \ldots, X_p$, and then add an irrelevant feature X_{p+1} that cannot help increase the margin. How will SVMs automatically ignore this feature? Justify your answer formally.

Question 2 PCA is method that is known as a decorrelation transform. Explain why this is the case.

Question 3 You have just been given a ten datasets from your company and they want you to compare your new classification method to what they currently have. Describe a using techniques that we have discussed in class to make this comparison.

Question 4 In k-nearest neighbors (KNN), the classification is achieved by majority vote in the vicinity of a data sample $\mathbf{x}$. Suppose there are two classes, where each class has $n/2$ points overlapped to some extent in a 2-D space. Describe what happens to the training error (using all available data) when the neighbor size k varies from n to 1.

Question 5 Explain what a kernel is and the "kernel trick". Why is this useful in the context of a support vector machine.

Problem #5 – The Role for Best Supporting Vector Goes To. . . (10 Points)

Let $\Phi(\mathbf{x})$ be a non-linear map to a higher dimensional space and $\mathbf{z}, \mathbf{x} \in \mathbb{R}^p$ be vectors. Furthermore, let $k(\mathbf{x}, \mathbf{z})$ be a kernel evaluated as a function of $\mathbf{x}$ and $\mathbf{z}$. In class, we showed that you do not need to explicitly compute the vectors $\Phi(\cdot)$. Show that you do not need to compute these high-dimensional vectors when you measure the distance in a Euclidean space. That is show that

$$\|\Phi(\mathbf{x}) - \Phi(\mathbf{z})\|_2^2 \,,$$

can be written in terms of kernels. Can we state that for an RBF kernel that $\|\Phi(\mathbf{x}) - \Phi(\mathbf{z})\|_2^2 \leq 2$, where

$$\Phi(\mathbf{x})^\mathsf{T}\Phi(\mathbf{z}) = k(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{1}{2}\frac{\|\mathbf{x} - \mathbf{z}\|_2^2}{\sigma}\right)$$

where $\sigma \geq 0$.

Bonus – Sampling from a Random Processes(1 Point)

You were just hired as a data scientist at one of the most popular startups in San Francisco and they have a problem for you to address. Assume that you are given a biased coin that appears as heads with probability p (i.e., $p \neq 1/2$). Your first task at this company is the use this coin in a black box to produce another random process with outputs 1's and 0's with probability 1/2. How do you achieve this task?

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:** $H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$