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: Solution

Signature: ____________________

Date: ____________________

Instructions: There are five 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: __________

Problem 5: __________

Total: __________

Problem #1 – Principal Component Analysis (10 Points)

In class, we showed two different approaches that we could arrive at a solution to PCA: one with linear algebra and one with optimization. This problem asks you to use both of what you know about the PCA projection and task of optimization. Use these facts:

- The projection is performed with $z = \mathbf{w}^\mathsf{T}\mathbf{x}$. Note that z is a scalar because we are only looking for one principal axis.
- I am not too concerned with the magnitude of $\mathbf{w}$, but I am concerned with its direction.
- You need to maximize the variance of z.

Use these facts to find $\mathbf{w}$. It maybe a good idea to let $\mathbf{X} \in \mathbb{R}^{p\times n}$ be the matrix of data. Then the covariance matrix is given by $\frac{1}{n-1}\mathbf{X}\mathbf{X}^\mathsf{T} = \Sigma$. This approach is similar to how we discussed PCA from a linear algebra perspective.

Note that if you simply write out the lecture notes from class then you'll receive zero credit.

Solution

This one is verbatim from your text book (and the in class notes would have been sufficient too)! The projection is performed with $z = \mathbf{w}^\mathsf{T}\mathbf{x}$ and we know that $\mathsf{Var}(X) = \frac{1}{n-1}\mathbf{X}\mathbf{X}^\mathsf{T} = \Sigma$. Then we have

$$\mathsf{Var}(z) = \mathbf{w}^\mathsf{T}\Sigma\mathbf{w}$$

We seek to fin the $\mathbf{w}$ such that $\mathsf{Var}(z)$ is maximized as asked in the question; however, we have the constraint that $\|\mathbf{w}\|_2^2 = 1$[1]. As we saw with other problems we've encountered in the homework and lecture, this is a constrained optimization problem where we want to maximize $\mathbf{w}^\mathsf{T}\Sigma\mathbf{w}$ subject to $\|\mathbf{w}\|_2^2 = 1$ Writing this in the form of the Lagrangian gives us

$$L(\mathbf{w}, \eta) = \mathbf{w}^\mathsf{T}\Sigma\mathbf{w} - \eta(\|\mathbf{w}\|_2^2 - 1)$$

which is exactly what we had in class! Taking the derivative w.r.t. $\mathbf{w}$, we find that $\Sigma\mathbf{w} = \eta\mathbf{w}$, which means that $\mathbf{w}$ is an eigenvector of Σ.

[1] You could even assume that $\|\mathbf{w}\|_2^2 = d$, where $d > 0$.

Problem #2 – AdaBoost (15 Points)

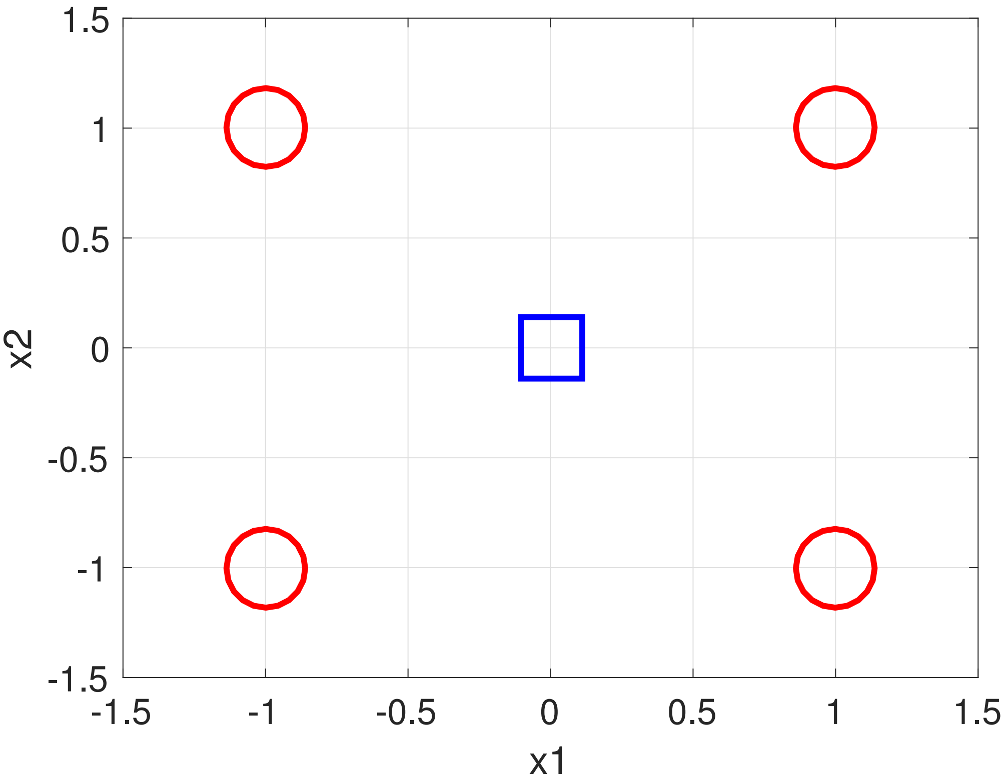

Figure 1: Labeled training points for Problem 2.

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

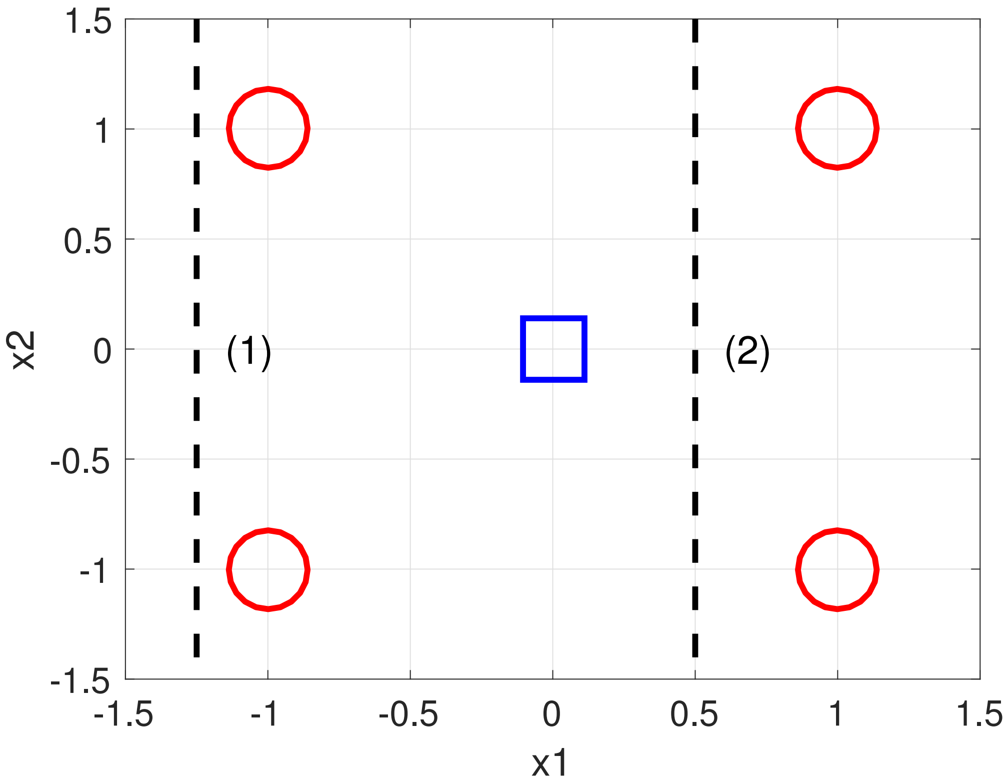

Figure 2: Solution for Problem 2.

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? **Solution**: 1/2

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_1 (i.e., vertical line).

Problem #3 – A Gamblers Ruin (10 Points)

[True/False] (1 point): The testing error of 1-NN classifier is guaranteed to be 0.

[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): The depth of a learned decision tree can be larger than the number of training examples used to create the tree.

[True/False] (1 point): We learn a classifier H by boosting weak learners h. The functional form of H's decision boundary is the same as h's, but with different parameters. (e.g., if h was a linear classifier, then H is also a linear classifier).

[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): 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): I don't like True & False questions, but I do like free points.

[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 $\alpha_i = 0$.

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

[Accept/Reject] (1 point): "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for $\lambda = 1.789489345672120002$.)"

[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.)"

Problem #4 – k-NN Classifier (10 Points)

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.

Solution: Each point is its own neighbor, so 1-NN classifier achieves perfect classification on training data. The error will increase to $\frac{1}{2}$ as k increases to n.

Problem #5 – Distances and Kernels (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. Then show that for an RBF kernel that $\|\Phi(\mathbf{x}) - \Phi(\mathbf{z})\|_2^2 \leq 2$, where

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

Solution: One set of distances that can be calculated can be computed in a *feature space* via kernels. A norm in feature space is calculated as

$$\begin{aligned}
\|\Phi(\mathbf{x}) - \Phi(\mathbf{z})\|_2^2 &= \left(\sqrt{(\Phi(\mathbf{x}) - \Phi(\mathbf{z}))^{\mathsf{T}}(\Phi(\mathbf{x}) - \Phi(\mathbf{z}))}\right)^2 \\
&= (\Phi(\mathbf{x}) - \Phi(\mathbf{z}))^{\mathsf{T}}(\Phi(\mathbf{x}) - \Phi(\mathbf{z})) \\
&= \Phi(\mathbf{x})^{\mathsf{T}}\Phi(\mathbf{x}) - \Phi(\mathbf{z})^{\mathsf{T}}\Phi(\mathbf{x}) - \Phi(\mathbf{x})^{\mathsf{T}}\Phi(\mathbf{z}) + \Phi(\mathbf{z})^{\mathsf{T}}\Phi(\mathbf{z}) \\
&= \Phi(\mathbf{x})^{\mathsf{T}}\Phi(\mathbf{x}) + \Phi(\mathbf{z})^{\mathsf{T}}\Phi(\mathbf{z}) - 2\Phi(\mathbf{x})^{\mathsf{T}}\Phi(\mathbf{z}) \\
&= k(\mathbf{x}, \mathbf{x}) + k(\mathbf{z}, \mathbf{z}) - 2k(\mathbf{x}, \mathbf{z})
\end{aligned}$$

Then using the definition of the RBF kernel, we have

$$\begin{aligned}
\|\Phi(\mathbf{x}) - \Phi(\mathbf{z})\|_2^2 &= k(\mathbf{x}, \mathbf{x}) + k(\mathbf{z}, \mathbf{z}) - 2k(\mathbf{x}, \mathbf{z}) \\
&= \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}\|_2^2\right) + \exp\left(-\frac{1}{2}\|\mathbf{z} - \mathbf{z}\|_2^2\right) - 2\exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2\right) \\
&= \exp(0) + \exp(0) - 2\exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2\right) \\
&= 2 - 2\exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2\right) \\
&\leq 2
\end{aligned}$$

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: $\quad h_t = \arg\min_{h \in \mathcal{H}} \widehat{\text{err}}(h, \mathcal{S}, \mathcal{D}_t)$

3: $\quad \epsilon_t = \sum \mathcal{D}_t(i) \mathbb{1}_{h(\mathbf{x}_i) \neq y_i}$

4: $\quad \alpha_t = \frac{1}{2} \log \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$

5: $\quad \mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i)}{Z_t} \exp\left(-\alpha_t y_i h_t(x_i)\right)$

6: **end for**

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