CSCI 567, Fall 2021
Haipeng Luo

Written Assignment #3

Due: Oct 26, 2021, 11:59 pm, PT

Instructions

Submission: Assignment submission will be via `courses.uscden.net`. By the submission date, there will be a folder set up in which you can submit your files. Please be sure to follow all directions outlined here.

You can submit multiple times, but only *the last submission* counts. That means if you finish some problems and want to submit something first and update later when you finish, that's fine. In fact you are encouraged to do this: that way, if you forget to finish the homework on time or something happens, you still get credit for whatever you have turned in.

Problem sets must be typewritten or neatly handwritten when submitted. In both cases, your submission must be a single PDF. Please also follow the rules below:

- The file should be named as `firstname_lastname_USCID.pdf` (e.g., Joe_Doe_1234567890.pdf).
- Do not have any spaces in your file name when uploading it.
- Please include your name and USCID in the header of the report as well.

Total points: 40 points

Notes on notation:

- Unless stated otherwise, scalars are denoted by small letter in normal font, vectors are denoted by small letters in bold font and matrices are denoted by capital letters in bold font.
- $\|.\|$ means L2-norm unless specified otherwise i.e. $\|.\| = \|.\|_2$

Problem 1 Support Vector Machines (19 points)

Consider a dataset consisting of points in the form of (x, y), where x is a real value, and $y \in \{-1, 1\}$ is the class label. There are only three points $(x_1, y_1) = (-1, -1)$, $(x_2, y_2) = (1, -1)$, and $(x_3, y_3) = (0, 1)$, shown in Figure 1.

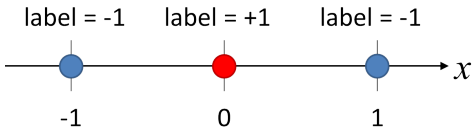

Figure 1: Three data points considered in Problem 1

1.1 Can these three points in their current one-dimensional feature space be perfectly separated with a linear classifier? Why or why not? **(2 points)**

1.2 Now we define a simple feature mapping $\boldsymbol{\phi}(x) = [x, x^2]^T$ to transform the three points from one-dimensional to two-dimensional feature space. Plot the transformed points in the new two-dimensional feature space. Is there a linear model $\mathbf{w}^\top \mathbf{x} + b$ for some $\mathbf{w} \in \mathbb{R}^2$ and $b \in \mathbb{R}$ that can correctly separate the three points in this new feature space? Why or why not? **(3 points)**

1.3 Given the feature mapping $\boldsymbol{\phi}(x) = [x, x^2]^T$, write down the 3×3 kernel/Gram matrix $\mathbf{K}$ for this dataset. **(2 points)**

1.4 Now write down the primal and dual formulations of SVM for this dataset in the two-dimensional feature space. Note that when the data is separable, we set the hyperparameter C to be $+\infty$ which makes sure that all slack variables (ξ) in the primal formulation have to be 0 (and thus can be removed from the optimization). **(4 points)**

1.5 Next, solve the dual formulation exactly (note: while this is not generally feasible as discussed in the lecture, the simple form of this dataset makes it possible). Based on that, calculate the primal solution. **(5 points)**

1.6 Plot the decision boundary (which is a line) of the linear model $\mathbf{w}^{*T}\mathbf{x} + b^*$ in the two-dimensional feature space, where $\mathbf{w}^*$ and b^* are the primal solution you got from the previous question. Then circle all support vectors. Finally, plot the corresponding decision boundary in the original one-dimensional space (which are just all the points x such that $\mathbf{w}^{*T}\boldsymbol{\phi}(x) + b^* = 0$). **(3 points)**

Problem 2 Decision trees (12 points)

Consider a binary dataset with 400 examples, where half of them belongs to class A and another half belongs to class B.

Next consider two decision stumps (i.e. trees with depth 1) $\mathcal{T}_1$ and $\mathcal{T}_2$, each with two children. For $\mathcal{T}_1$, its left child has 150 examples in class A and 50 examples in class B; for $\mathcal{T}_2$, its left child has 0 example in class A and 100 examples in class B. (You can infer what are in the right child.)

2.1 For each leaf of $\mathcal{T}_1$ and $\mathcal{T}_2$, compute the corresponding classification error, entropy (base e) and Gini impurity. You can either exactly express the final numbers in terms of fractions and logarithms, or round them to two decimal places. (Note: the value/prediction of each leaf is the majority class among all examples that belong to that leaf.) **(6 points)**

2.2 Compare the quality of $\mathcal{T}_1$ and $\mathcal{T}_2$ (that is, the two different splits of the root) based on classification error, conditional entropy (base e), and weighted Gini impurity respectively. **(6 points)**

Problem 3 Boosting (9 points)

3.1 We discussed in class that AdaBoost minimizes the exponential loss greedily. In particular, the derivation of β_t is by finding the minimizer of

$$\epsilon_t(e^{\beta_t} - e^{-\beta_t}) + e^{-\beta_t}$$

where ϵ_t is the weighted classification error of h_t and is fixed. Show that $\beta_t^* = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ is the minimizer. (You can use the fact that the function above is convex.) **(3 points)**

3.2 Recall that at round t of AdaBoost, a classifier h_t is obtained and the weighting over the training set is updated from D_t to D_{t+1}. Prove that h_t is only as good as random guessing in terms of classification error weighted by D_{t+1}. That is

$$\sum_{n:h_t(\mathbf{x}_n)\neq y_n} D_{t+1}(n) = \frac{1}{2}.$$

In other words, the update is so that D_{t+1} is the "hardest" weighting for h_t. **(6 points)**