CSCI 567, Fall 2018
Haipeng Luo

# Homework #3 Written Assignments

Due: 11:59 pm, October 21, 2018

## 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 instructions outlined here.

You can submit multiple times, but only *the last submission* counts. As a results, if you finish some problems, you might want to submit them first, and update later when you finish the rest. You are encouraged to do so. This way, if you forget to finish the homework on time or something happens (remember Murphy's Law), 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. It is strongly recommended that you typeset with LaTeX. There are many free integrated LaTeX editors that are convenient to use (e.g Overleaf, ShareLaTeX). Choose the one(s) you like the most. This tutorial Getting to Grips with LaTeX is a good start if you do not know how to use LaTeX yet.

Please also follow the rules below:

- The file should be named as `firstname_lastname_USCID.pdf` e.g., `Don_Quijote_de_la_Mancha_8675309045.pdf`).
- Do not have any spaces in your file name when uploading it.
- Please include your name and USC ID in the header of the report as well.

**Collaboration:** You may discuss with your classmates. However, you need to write your own solutions and submit separately. Also in your report, you need to list with whom you have discussed for each problem. Please consult the syllabus for what is and is not acceptable collaboration. Review the rules on academic conduct in the syllabus: a single instance of plagiarism can adversely affect you significantly more than you could stand to gain.

**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

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

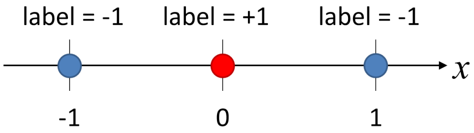

Figure 1: Three data points considered in Problem 1

**1.1** Can three points shown in Figure 1, in their current one-dimensional feature space, be perfectly separated with a linear separator? Why or why not?

**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 (use any package you prefer for the plot, e.g., Matplotlib, PowerPoint, or simply hand draw it). Is there a linear model that can correctly separate the points in this new feature space? Why or why not?

**1.3** Given the feature mapping $\boldsymbol{\phi}(x) = [x, x^2]^T$, write down the corresponding kernel function $k(x, x')$. Moreover, write down the $3 \times 3$ kernel (or Gram) matrix $\mathbf{K}$ of the three data points. Finally verify that $\mathbf{K}$ is a positive semi-definite (PSD) matrix. You may want to show this by the definition of PSD matrices: a symmetric $N \times N$ real matrix $\mathbf{M}$ is said to be positive semi-definite if the scalar $\mathbf{z}^T\mathbf{M}\mathbf{z}$ is non-negative for every column vector $\mathbf{z}$ of $N$ real numbers.

**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 ($\xi$) in the primal formulation have to be 0 (and thus can be removed from the optimization).

**1.5** Next, solve the dual formulation. Based on that, derive the primal solution.

**1.6** Plot the decision boundary (which is a line) of the linear model $\mathbf{w}^{*T}\boldsymbol{\phi}(\mathbf{x}) + b^*$ 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 two points).

## Problem 2 Decision trees

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. (Note: the value/prediction of each leaf is the majority class among all examples that belong to that leaf.)

**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.

## Problem 3 Boosting

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

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

where $\epsilon_t$ 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.

**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 $h_t$.