DS-UA 301
Advanced Topics in Data Science
Instructor: Parijat Dube

# Homework 2

Due: October 8, 2023

## General Instructions

This homework must be turned in on both Gradescope and Brightspace by 11:59 pm on the due date. It must be your own work and your own work only—you must not copy anyone's work, or allow anyone to copy yours. This extends to writing code. You may consult with others, but when you write up, you must do so alone. Your homework submission must be written and submitted using Jupyter Notebook (.ipynb). **No handwritten solutions will be accepted**. You should submit:

1. On **Gradescope**: one Jupyter Notebook(.ipynb) containing your solutions for each Problem in this assignment. Each Jupyter Notebook should be named as (*netid*)_hw(*#Homework*)_p(*#Problem*).ipynb.

   For example, *bz2058_hw2_p1.ipynb*.

2. On **Brightspace**: one .zip file containing all the notebooks, datasets, and other supporting materials involved in this homework. The .zip file should be named as (*netid*)_hw(*#Homework*).zip.

   For example, *bz2058_hw2.zip*.

Please make sure your answers are clearly structured in the Jupyter Notebooks:

1. Copy the Template Notebooks provided on Brightspace and write your solutions there.
2. Label each question part clearly. Do not include written answers as code comments. The code used to obtain the answer for each question part should accompany the written answer.
3. All plots should include informative axis labels and legends. All codes should be accompanied by informative comments. All output of the code should be retained.
4. Math formulas can be typesetted in Markdown in the same way as LaTeX. A Markdown Guide is provided on Brightspace for reference.

For more homework-related policies, please refer to the syllabus.

## Problem 1 - *Algorithmic Performance Scaling* 25 points

OpenML (https://www.openml.org) has thousands of datasets for classification tasks. Select any sufficiently large (having greater than 50K instances) dataset from OpenML with multiple (greater than 2) output classes.

1. Summarize the attributes of the selected dataset: number of features, number of instances, number of classes, number of numerical features, and number of categorical features. Is the dataset balanced? Plot the distribution of the number of samples per class. **(5)**
2. For the selected dataset, select 80% of the data as the training set and the remaining 20% as the test set. Generate 10 different subsets of the training set by randomly subsampling $10\%, 20\%, \ldots, 100\%$ of the training set. Use each of these subsets to train two different classifiers: *Decision Tree* and *Gradient boosting* in sklearn. You will work with default hyperparameters for these classifiers in sklearn. When training a classifier also measure the wall clock time to train. After each training, evaluate the accuracy of trained models on the test set. Report model accuracy and training time for each of the 10 subsets of the training set for the two models in a table. **(8)**
3. Using the data collected in part 2, you will create *learning curve* for the two classifiers. A learning curve shows how the accuracy changes with increasing size of training data. You will create one chart with the horizontal axis being the percentage of the training set and the vertical axis being the accuracy on the test set. On this chart, you will plot two learning curves for both Decision Tree and Gradient Boosting. **(5)**

# Homework 2

4. Next, using the data collected in part 2, you will create a chart showing the training time of classifiers with increasing size of training data. So, for each classifier, you will have one plot showing the training time as a function of training data size. **(3)**

5. Study the scaling of training time and accuracy of classifiers with training data size using the two figures generated in parts 3 and 4 of this problem. Compare the performance of classifiers in terms of training time and accuracy and write 3 main observations. Which gives better accuracy? Which has a shorter training time? **(4)**

## Problem 2 - *Precision, Recall, ROC* 15 points

This question is based on a paper from ICML 2006 (reference below) that talks about the relationship between ROC and Precision-Recall (PR) curves and shows a one-to-one correspondence between them. You need to read the paper to answer the following questions.

1. Does true negative matter for both the ROC and PR curve? Argue why each point on the ROC curve corresponds to a unique point on the PR curve. **(5)**

2. Select one OpenML dataset with 2 output classes. Use two binary classifiers (*Adaboost* and *Logistic regression*) and create ROC and PR curves for each of them. You will have two figures: one containing two ROC and the other containing two PR curves. Show the point where an *all-positive classifier* lies in the ROC and PR curves. An *all-positive classifier* classifies all the samples as positive. **(10)**

*Reference paper:*

- Jesse Davis, Mark Goadrich, The Relationship Between Precision-Recall and ROC Curves, ICML 2006.

## Problem 3 - *Perceptron* 15 points

Consider a 2-dimensional data set in which all points with $x_1 > x_2$ belong to the positive class, and all points with $x_1 \leq x_2$ belong to the negative class. Therefore, the true separator of the two classes is a linear hyperplane (line) defined by $x_1 - x_2 = 0$. Now, create a training data set with 10 points randomly generated inside the unit square in the positive quadrant. Label each point depending on whether or not the first coordinate $x_1$ is greater than its second coordinate $x_2$. Now consider the following loss function for training pair $(\bar{X}, y)$ and weight vector $\bar{W}$:

$$L = \max\{0, a - y(\bar{W} \cdot \bar{X})\},$$

where the test instances are predicted as $\hat{y} = \text{sign}\{\bar{W} \cdot \bar{X}\}$. For this problem, $\bar{W} = [w_1, w_2]$, $\bar{X} = [x_1, x_2]$ and $\hat{y} = \text{sign}(w_1 x_1 + w_2 x_2)$. A value of $a = 0$ corresponds to the *perceptron criterion* and a value of $a = 1$ corresponds to *hinge-loss*.

1. You need to implement the perceptron algorithm without regularization, train it on the 10 points above, and test its accuracy on 5000 randomly generated points inside the unit square. Generate the test points using the same procedure as the training points. You need to have your own implementation of the perceptron algorithm, using the *perceptron criterion* loss function. **(6)**

2. Change the loss function from *perceptron criterion* to *hinge-loss* in your implementation for training and repeat the accuracy computation on the same test points above. Regularization is not used. **(5)**

3. In which case do you obtain better test accuracy and why? **(2)**

# Homework 2

4. In which case do you think that the classification of the same 5000 test instances will not change significantly by using a different set of 10 training points? **(2)**

*Reference:*

- Perceptron Algorithm in Python available at https://medium.com/hackernoon/implementing-the-perceptron-algorithm-from-scratch-in-python-48be2d07b1c0

## Problem 4 - *Linear Separability* 10 points

Consider a dataset with two features $x_1$ and $x_2$ in which the points $(-1,-1),(1,1),(-3,-3),(4,4)$ belong to one class and $(-1,1),(1,-1),(-5,2),(4,-8)$ belong to the other.

1. Is this dataset linearly separable? Can a linear classifier be trained using features $x_1$ and $x_2$ to classify this data set? You can plot the dataset points and argue. **(2)**
2. Can you define a new 1-dimensional representation $z$ in terms of $x_1$ and $x_2$ such that the dataset is linearly separable in terms of 1-dimensional representation corresponding to $z$? **(4)**
3. What does the separating hyperplane looks like? **(2)**
4. Explain the importance of nonlinear transformations in classification problems. **(2)**