**CptS 570 Machine Learning, Fall 2019**
**Homework #1**
Due Date: Sep 24

NOTE 1: Please use a word processing software (e.g., Microsoft word or Latex) to write your answers. The rationale is that it is sometimes hard to read and understand the hand-written answers. Thanks for your understanding.

NOTE 2: Please ensure that all the graphs are appropriately labeled (x-axis, y-axis, and each curve). The caption or heading of each graph should be informative and self-contained.

# 1 Analytical Part (2 percent grade)

This part will be graded as a PASS or FAIL.

1. Answer the following questions with a yes or no along with proper justification.
   a. Is the decision boundary of voted perceptron linear?
   b. Is the decision boundary of averaged perceptron linear?

2. In the class, we saw the Passive-Aggressive (PA) update that tries to achieve a margin equal to *one* after each update. Derive the PA weight update for achieving margin $M$.

3. Consider the following setting. You are provided with n training examples: $(x_1, y_1, h_1), \cdots, (x_n, y_n, h_n)$, where $x_i$ is the input example, $y_i$ is the class label (+1 or -1), and $h_i > 0$ is the importance weight of the example. The teacher gave you some additional information by specifying the importance of each training example.
   a. How will you modify the perceptron algorithm to be able to leverage this extra information? Please justify your answer.
   b. How can you solve this learning problem using the standard perceptron algorithm? Please justify your answer. I'm looking for a reduction based solution.

4. Consider the following setting. You are provided with n training examples: $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$, where $x_i$ is the input example, and $y_i$ is the class label (+1 or -1). However, the training data is highly imbalanced (say 90% of the examples are negative and 10% of the examples are positive) and we care more about the accuracy of positive examples.
   a. How will you modify the perceptron algorithm to solve this learning problem? Please justify your answer.
   b. How can you solve this learning problem using the standard perceptron algorithm? Please justify your answer. I'm looking for a reduction based solution.

# 2 Programming and Empirical Analysis Part (6 percent grade)

1. Programming and empirical analysis question.
   Implement a binary classifier with both perceptron and passive-aggressive (PA) weight update as shown below.

**Algorithm 1** Online Binary-Classifier Learning Algorithm

**Input**: $\mathcal{D}$ = Training examples, $T$ = maximum number of training iterations
**Output**: $w$, the final weight vector

Initialize the weights $w = 0$
**for** each training iteration $itr \in \{1, 2, \cdots, T\}$ **do**
  **for** each training example $(x_t, y_t) \in \mathcal{D}$ **do**
    $\hat{y}_t = sign(w \cdot x_t)$ *// predict using the current weights*
    **if** mistake **then**
      $w = w + \tau \cdot y_t \cdot x_t$ *// update the weights*
    **end if**
  **end for**
**end for**
**return** final weight vector $w$

For standard perceptron, you will use $\tau = 1$, and for Passive-Aggressive (PA) algorithm, you will compute the learning rate $\tau$ as follows.

$$\tau = \frac{1 - y_t \cdot (w \cdot x_t)}{\|x_t\|^2} \tag{1}$$

Implement a multi-class online learning algorithm with both perceptron and passive-aggressive (PA) weight update as shown below. Employ the single weight vector represenation (representation-II as discussed in the class). This representation is defined as follows. Each training example is of the form $(x_t, y_t)$, where $x_t \in \Re^d$ is the input and $y_t \in \{1, 2, \cdots, k\}$ is the class (output) label. In this representation, you will have a single weight-vector $w \in \Re^{k \cdot d}$ and the augmented feature function $F(x_t, y) \in \Re^{k \cdot d}$ will have $k$ blocks of size $d$ and it will have zeroes everywhere except for the $y^{th}$ block, which will have $x_t$ in it.

**Algorithm 2** Online Multi-Class Classifier Learning Algorithm

**Input**: $\mathcal{D}$ = Training examples, $k$ = number of classes, $T$ = maximum number of training iterations
**Output**: $w$, the final weight vector

Initialize the weights $w = 0$
**for** each training iteration $itr \in \{1, 2, \cdots, T\}$ **do**
  **for** each training example $(x_t, y_t) \in \mathcal{D}$ **do**
    $\hat{y}_t = \arg max_{y \in \{1,2,\cdots,k\}}\ w \cdot F(x_t, y)$ *// predict using the current weights*
    **if** mistake **then**
      $w = w + \tau \cdot (F(x_t, y_t) - F(x_t, \hat{y}_t))$ *// update the weights*
    **end if**
  **end for**
**end for**
**return** final weight vector $w$

For standard perceptron, you will use $\tau = 1$, and for Passive-Aggressive (PA) algorithm, you will compute the learning rate $\tau$ as follows.

$$\tau = \frac{1 - (w \cdot F(x_t, y_t) - w \cdot F(x_t, \hat{y}_t))}{\|F(x_t, y_t) - F(x_t, \hat{y}_t)\|^2} \tag{2}$$

You will use the Fashin MNIST data (https://github.com/zalandoresearch/fashion-mnist). There is a fixed training and testing set.

Each example is a 28x28 grayscale image, associated with a label from 10 classes: 0 T-shirt/top, 1 Trouser, 2 Pullover, 3 Dress, 4 Coat, 5 Sandal, 6 Shirt, 7 Sneaker, 8 Bag, 9 Ankle boot.

You will use ONLY training data for training and testing data for evaluation.

**5.1 Binary Classification (40 points)** Learn a binary classifier to classify *even labels* (0, 2, 4, 6, 8) and *odd labels* (1, 3, 5, 7, 9).

a. Compute the online learning curve for both Perceptron and PA algorithm by plotting the number of training iterations (1 to 50) on the x-axis and the number of mistakes on the y-axis. Compare the two curves and list your observations.

b. Compute the accuracy of both Perceptron and PA algorithm on the training data and testing data for 20 training iterations. So you will have two accuracy curves for Perceptron and another two accuracy curves for PA algorithm. Compare the four curves and list your observations.

c. Repeat experiment (b) with averaged perceptron. Compare the test accuracies of plain perceptron and averaged perceptron. What did you observe?

d. Compute the general learning curve (vary the number of training examples starting from 5000 in the increments of 5000) for 20 training iterations. Plot the number of training examples on x-axis and the testing accuracy on the y-axis. List your observations from this curve.

**5.2 Multi-Class Classification (40 points)** Learn a multi-class classifier to map images to the corresponding fashion label.

a. Compute the online learning curve for both Perceptron and PA algorithm by plotting the number of training iterations (1 to 50) on the x-axis and the number of mistakes on the y-axis. Compare the two curves and list your observations.

b. Compute the accuracy of both Perceptron and PA algorithm on the training data and testing data for 20 training iterations. So you will have two accuracy curves for Perceptron and another two accuracy curves for PA algorithm. Compare the four curves and list your observations.

c. Repeat experiment (b) with averaged perceptron. Compare the test accuracies of plain perceptron and averaged perceptron. What did you observe?

d. Compute the general learning curve (vary the number of training examples starting from 5000 in the increments of 5000) for 20 training iterations. Plot the number of training examples on x-axis and the testing accuracy on the y-axis. List your observations from this curve.

## 3 Instructions for Submission

Please follow the below instructions. If you do not follow them, your homework will not be graded. We will provide a dropbox folder link for the homework submission.

**PDF submission.** One PDF file with both answers for analytical part (Part I) and empirical analysis questions with results/analysis (Part-II). Please label x-axis, y-axis, and name of the graphs appropriately. Please name this file as WWSUID-LASTNAME.pdf (e.g., 111222-Fern.pdf).

**Code submission.** You will submit one zip file for your code as per the instructions below. If your script and/or code does not execute, we will try to give some partial credit by looking at the overall code contents.

- Mention the programming language and version (e.g., Python 2.5) that you used.
- Submit one folder with name WSUID-LASTNAME.zip (e.g., 111222-Fern.zip) and include a README file.
- Include a script to run the code and it should be referred in the README file. Please make sure that your script runs on a standard linux machine.
- Don't submit the data folder. Assume there is a folder "data" with all the files.
- Output of your programs should be well-formatted in order to answer the empirical analysis questions.
- Structure your code meaningfully and add comments to make it readable.

If you have collaborated or discussed the homework with some student, please provide this information with all the relevant details. If we find that the code of two different students has traces of plagiarism, both students will be penalized and we will report the academic dishonesty case to graduate school (see https://communitystandards.wsu.edu/policies-and-reporting/academic-integrity-policy/). The bottom line is please DO NOT even think of going this route. It is very unpleasant to deal with these things for both faculty, TA, and students involved.

## 4 Grading Rubric

Each question in the students work will be assigned a letter grade of either A,B,C,D, or F by the Instructor and TAs. This five-point (discrete) scale is described as follows:

- **A) Exemplary (=100%)**.
  Solution presented solves the problem stated correctly and meets all requirements of the problem.
  Solution is clearly presented.
  Assumptions made are reasonable and are explicitly stated in the solution.
  Solution represents an elegant and effective way to solve the problem and is not overly complicated than is necessary.

- **B) Capable (=75%)**.
  Solution is mostly correct, satisfying most of the above criteria under the exemplary category, but contains some minor pitfalls, errors/flaws or limitations.

- **C) Needs Improvement (=50%)**.
  Solution demonstrates a viable approach toward solving the problem but contains some major pitfalls, errors/flaws or limitations.

- **D) Unsatisfactory (=25%)**
  Critical elements of the solution are missing or significantly flawed.
  Solution does not demonstrate sufficient understanding of the problem and/or any reasonable directions to solve the problem.

- **F) Not attempted (=0%)**
  No solution provided.

The points on a given homework question will be equal to the percentage assigned (given by the letter grades shown above) multiplied by the maximum number of possible points worth for that question. For example, if a question is worth 6 points and the answer is awarded a $B$ grade, then that implies 4.5 points out of 6.