CptS 570 Machine Learning, Fall 2016
Homework #1
Due Date: Sep 20

NOTE 1: Please use a word processing software (e.g., Microsoft word or Latex) to write your answers and submit a printed copy to me at the begining of the class on Sep 20. 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.

NOTE 3: Please send a copy of your source code and instructions on how to run your code to be able to verify the reported results via email (mislam1@eecs.wsu.edu)

1. **(5 points)** 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. **(5 points)** 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. **(20 points)** Consider the following setting. You are provided with n training examples: $(x_1, y_1, h_1), (x_2, y_2, h_2), \cdots, (x_n, y_n, h_n)$, 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. **(20 points)** 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.

5. **(50 points)** 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 τ 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 τ 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 are provided with the OCR handwriting data. There are 10 different training and testing sets (named as folds 0 to 9).

The format of the file is as follows. Each line is one classification example: the 128 binary bits after *im* correspond to the input features (binary image of the handwritten character) and the character letter at the end correspond to the output label. We have 26 classes.

You will use ONLY training data for training and testing data for evaluation. Repeat this for all the 10 different folds and compute the averaged results.

5.1 Binary Classification Learn a binary classifier to classify *vowels* (a, e, i, o, u) and *consonants* (non-vowels).

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 50 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 100 in the increments of 100) for 50 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 Learn a multi-class classifier to map binary handwritten character images to the corresponding character letter (a-z).

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 50 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 100 in the increments of 100) for 50 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.