**CPT\_S 534 HW1**

**Yang Zhang**

**11529139**

**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?**

The linearity of decision boundary of voted perceptron depends on what kind of problem it deals with. If it is linear classification problem, then decision boundary is linear otherwise not linear.

**b. Is the decision boundary of averaged perceptron linear?**

The linearity of decision boundary of averaged perceptron depends on what kind of problem it deals with. If it is linear classification problem, then decision boundary is linear otherwise not 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.**

Margin (respect to wt) = yt(wt • **x**t) = yt((wt+τyt**x**t) • **x**t) = yt(**wt** • **x**t) + τ‖**x**t‖2

Then, force margin to M, we got M = yt(**wt** • **x**t) + τ‖**x**t‖2, solve for τ,

plug above equation back to **,**

**3. (20 points) Consider the following setting. You are provided with n training examples: (x1, y1, h1), (x2, y2, h2), · · ·, (xn, yn, hn), where xi is the input example, yi is the class label (+1 or -1), and hi > 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.**

For standard perceptron, ,

We modify above to ,

In this way, the sample with higher **h** will change the weight factor bolder when the sample updates **w**

**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.**

By using standard perceptron, we can simply change the input set from format (**x2, y2, h2**) into format (**h2x2, y2**). Thus, we reduce the three-parameter input into standard two parameter input. And then,

**4. (20 points) Consider the following setting. You are provided with n training examples: (x1, y1), (x2, y2), · · ·, (xn, yn), where xi is the input example, and yi 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.**

To get the imbalanced data trained more balance, we can add emphasis weight on those imbalanced part, in other word, for this question, since we care more about the accuracy of positive examples, we can add more weight for perceptron updates of positive examples.

For standard perceptron, ,

We modify above to

In this way, the positive sample will change the weight factor bolder when it updates **w**. The reason we choose b = 9 for positive sample is that there is 10% of total data are positive samples, the ratio of positive over negative is 1/9.

**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.**

If apply standard perceptron, we can modify the data input to make positive sample more important. By doing this, we simply multiply every positive sample with an emphasis factor b.

For this question, b is 9 since the ratio of positive sample over negative sample is 1/9.

Thus,