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 decision boundary of voted perceptron is non-linear, because decision may be decided by serval weight vectors (they have the same survival count). In this scenario, the decision boundary is obviously non-linear.

b. Is the decision boundary of averaged perceptron linear?

The decision boundary of averaged perceptron is linear, because decision made by a single weight vector that is sum of product of vote counts with each weight vector. Therefore, the decision boundary is 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 w_t) = $y_t(w_t \bullet \mathbf{x}_t) = y_t((w_t+\tau y_t \mathbf{x}_t) \bullet \mathbf{x}_t) = y_t(\mathbf{w_t} \bullet \mathbf{x}_t) + \tau\|\mathbf{x}_t\|^2$

Then, force margin to M, we got $M = y_t(\mathbf{w_t} \bullet \mathbf{x}_t) + \tau\|\mathbf{x}_t\|^2$, solve for τ,

$$\tau = \frac{M - y_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t)}{\|\boldsymbol{x}_t\|^2}$$

plug above equation back to $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \tau y_t \boldsymbol{x}_t$,

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \frac{M y_t \boldsymbol{x}_t - \boldsymbol{x}_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t)}{\|\boldsymbol{x}_t\|^2}$$

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), · · ·, (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.

For standard perceptron, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \tau y_t \boldsymbol{x}_t, where\ \tau = \begin{cases} 1 & y_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t) \leq 0 \\ 0 & y_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t) > 0 \end{cases}$,

We modify above to $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \tau y_t \boldsymbol{x}_t, where\ \tau = \begin{cases} h_t & y_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t) \leq 0 \\ 0 & y_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t) > 0 \end{cases}$,

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 ($\mathbf{x_2}$, $\mathbf{y_2}$, $\mathbf{h_2}$) into format ($\mathbf{h_2x_2}$, $\mathbf{y_2}$). Thus, we reduce the three-parameter input into standard two parameter input. And then, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{X}_t, where\ \boldsymbol{X}_t = h_t \boldsymbol{x}_t$

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, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \tau y_t \boldsymbol{x}_t, where\ \tau = \begin{cases} 1 & y_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t) \leq 0 \\ 0 & y_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t) > 0 \end{cases}$,

We modify above to $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \tau y_t \boldsymbol{x}_t, where\ \tau = \begin{cases} b & y_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t) \leq 0 \\ 0 & y_t(\boldsymbol{w}_t \bullet \boldsymbol{x}_t) > 0 \end{cases}, b = \begin{cases} 9 & y_t = 1 \\ 1 & y_t = -1 \end{cases}$,

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.

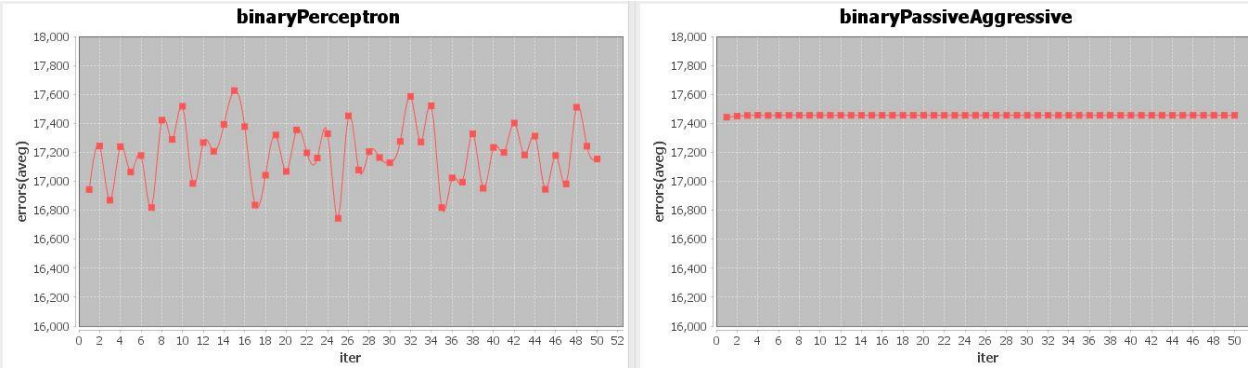
For this question, b is 9 since the ratio of positive sample over negative sample is 1/9.

Thus,

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{X}_t, where\ \boldsymbol{X}_t = \begin{cases} 9\boldsymbol{x}_t & y_t = 1 \\ \boldsymbol{x}_t & y_t = -1 \end{cases}$$

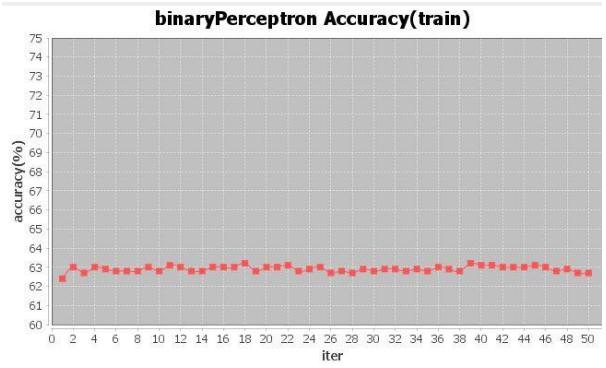
5.1

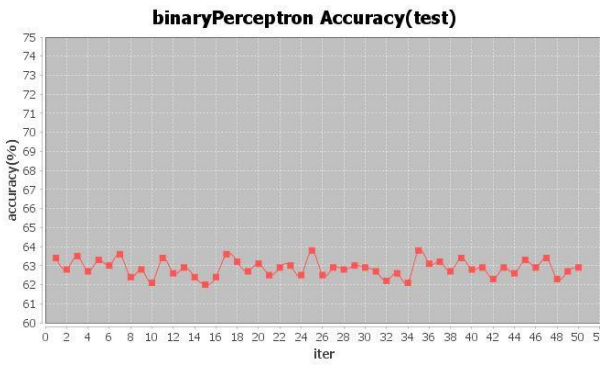
a)

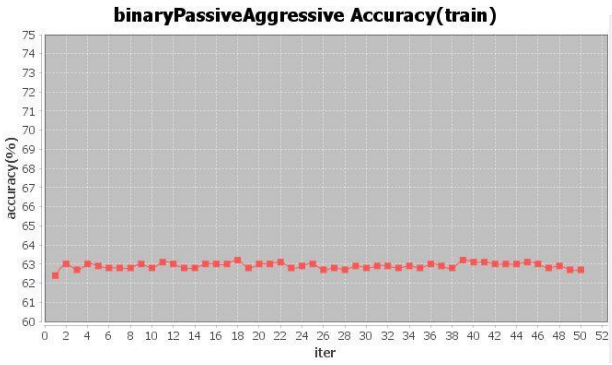


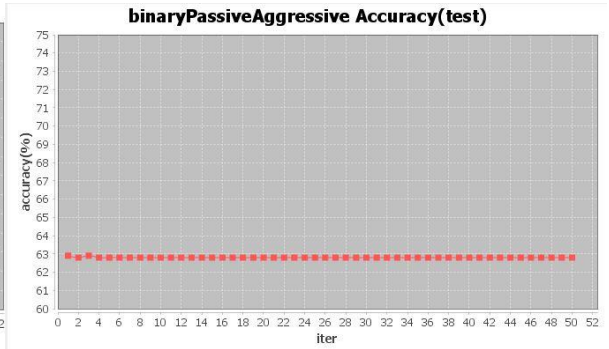
According to graph above, the learning curve of perceptron is fluctuant all the time along with iteration times increased, while the curve of Passive-Aggressive is stable. This is because that the standard perceptron algorithm doesn't guarantee decision bound margin, so that the decision boundary made by its weight vector can be closed to the threshold points.

b)



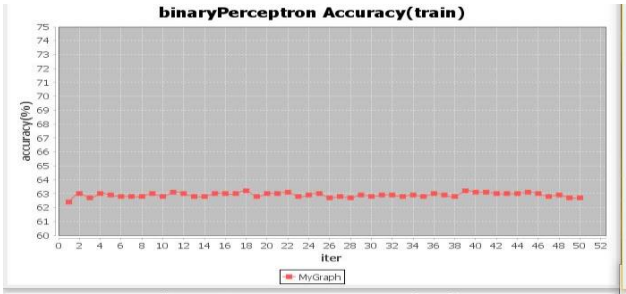


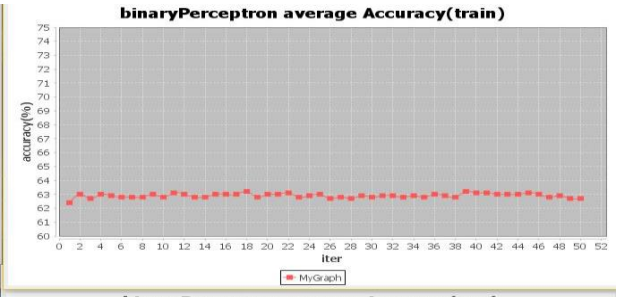


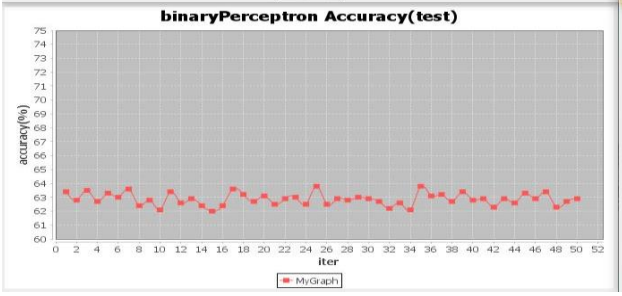


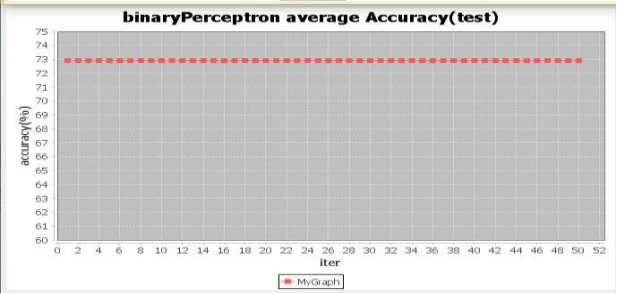
Comparing the 4 accuracy curves, I found that the accuracy of passive aggressive training is similar with the accuracy of standard perceptron training and the accuracy of standard perceptron testing, while the accuracy of passive aggressive testing is significantly more stable than accuracy of standard perceptron testing.

c)



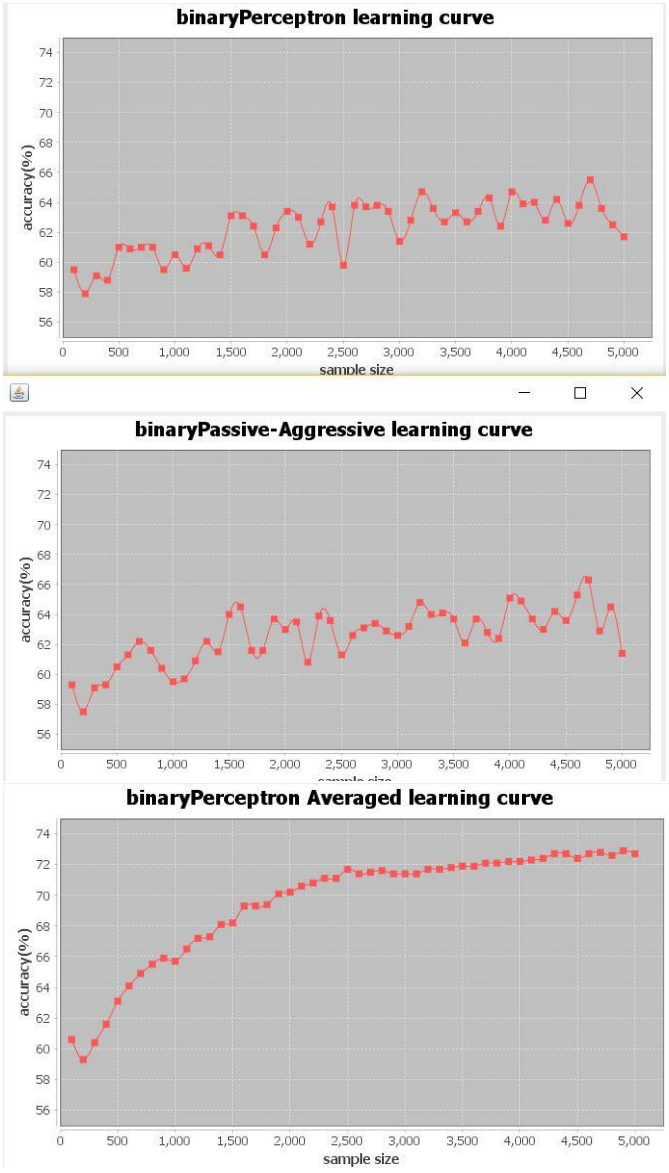






Comparing the graphs above, I found that the accuracy of perceptron averaged training is same as the accuracy of standard perceptron training, this because for the training part, both algorithms use the same weight vector to learn. While the accuracy of perceptron averaged testing is significantly higher than accuracy of standard perceptron testing and is very stable.

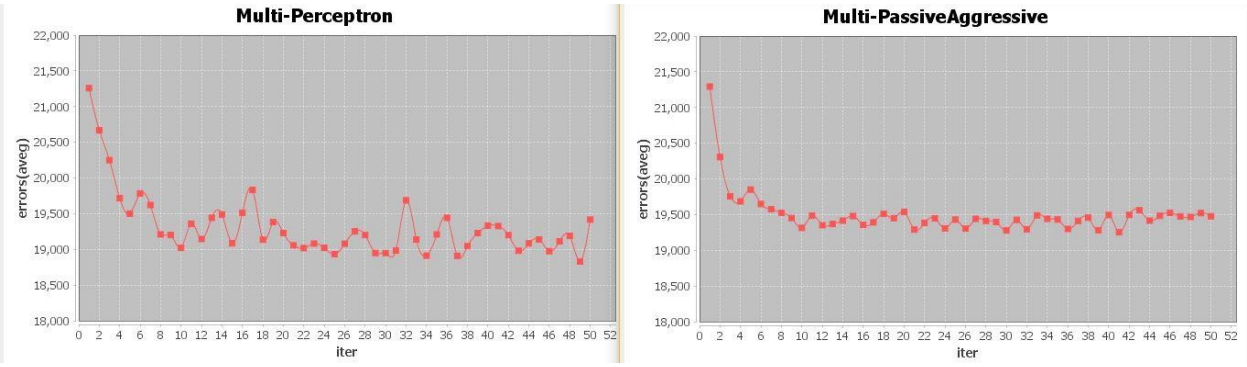
d)



Comparing the graphs above, in general, all the three algorithms increase their accuracy with increased training data size, while the perceptron averaged increased its accuracy more and more stable than other two algorithms.

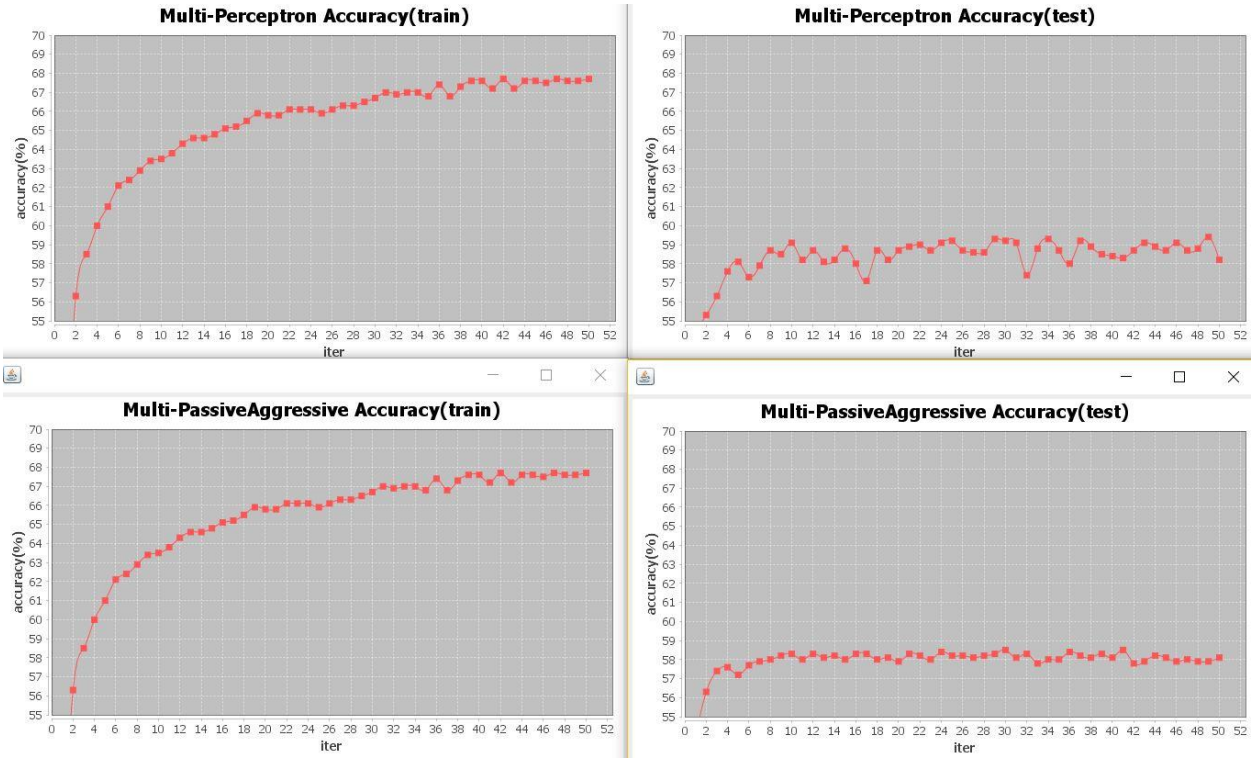
5.2

a)



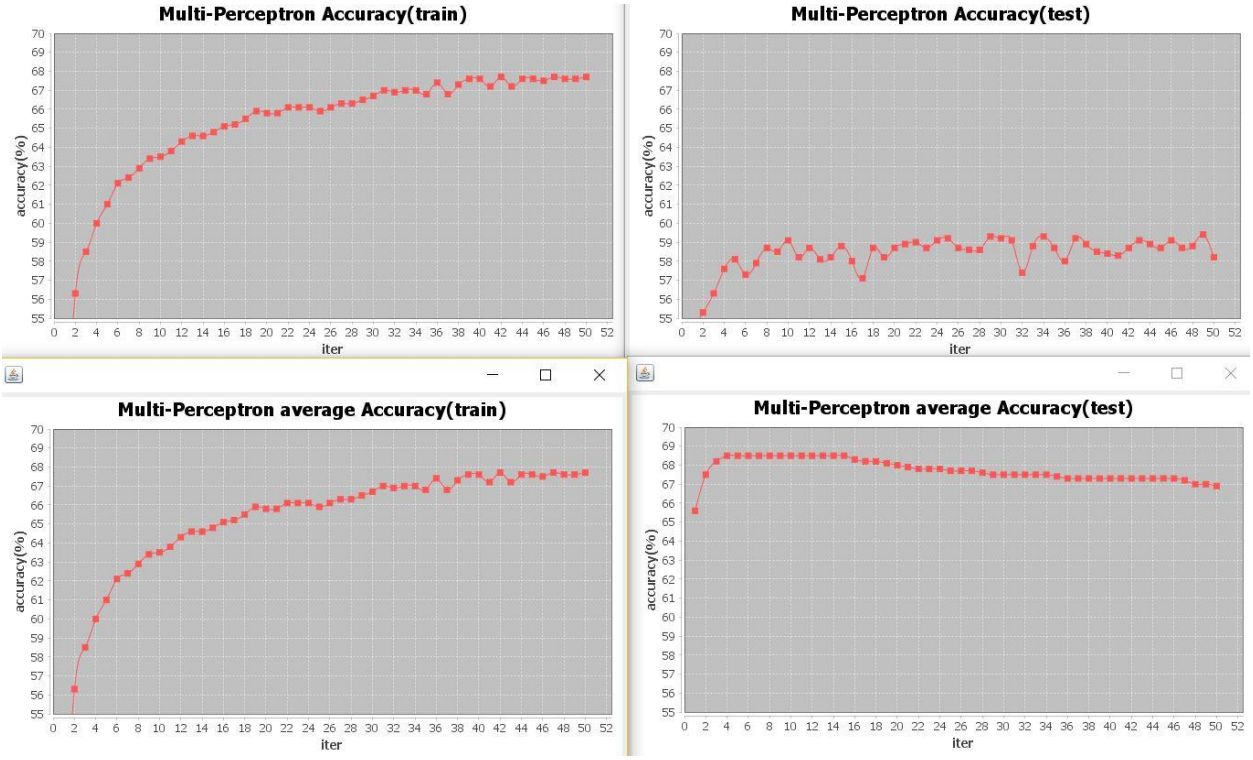
According to graph above, in general, both algorithms reduced the errors with increased iteration times. The learning curve of perceptron is fluctuant all the time, while the curve of Passive-Aggressive is stable. This is because that the standard perceptron algorithm doesn't guarantee decision bound margin, so that the decision boundary made by its weight vector can be closed to the threshold points.

b)



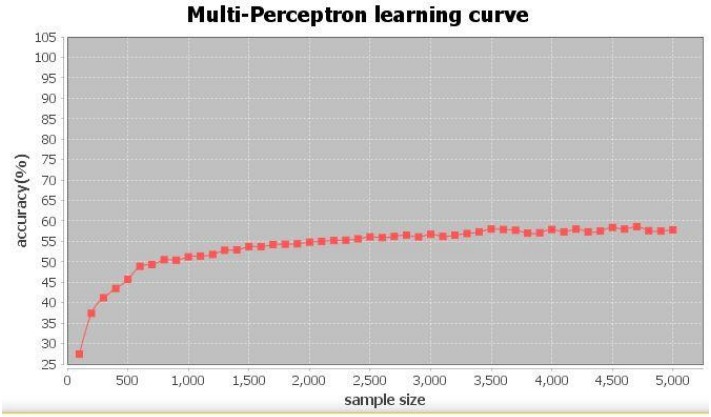
Comparing the 4 accuracy curves, in general, both algorithms increased the accuracy with increased iteration times. I found that the accuracy of passive aggressive training is similar with the accuracy of standard perceptron training, while the accuracy of passive aggressive testing is significantly more stable than accuracy of standard perceptron testing.

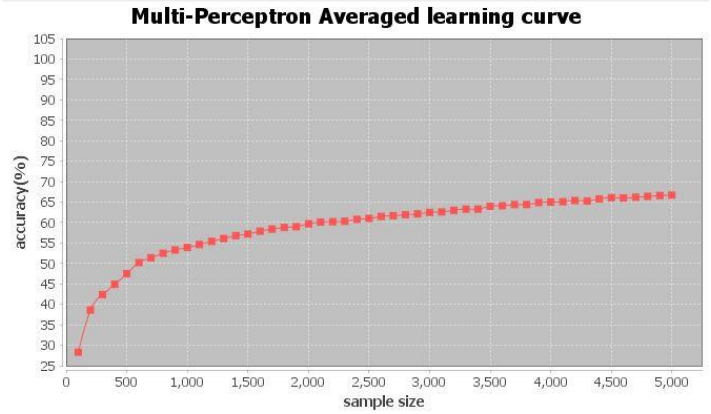
c)

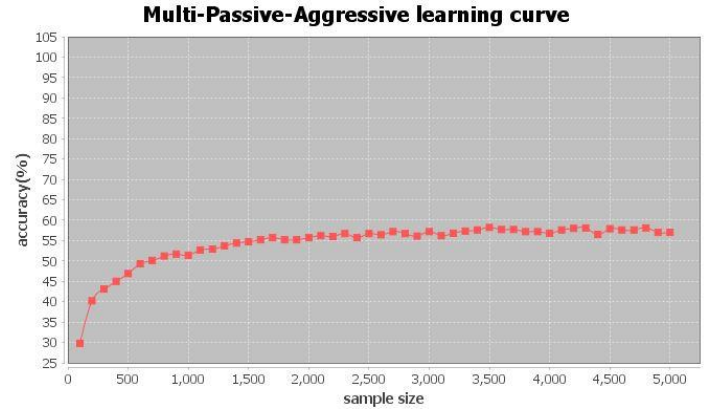


Comparing the graphs above, I found that the accuracy of perceptron averaged training is same as the accuracy of standard perceptron training, this because for the training part, both algorithms use the same weight vector to learn. While the accuracy of perceptron averaged testing is significantly higher than accuracy of standard perceptron testing and is more stable.

d)







Comparing the graphs above, in general, all the three algorithms increase their accuracy with increased training data size, while the perceptron averaged has smoother curve and higher accuracy.