CptS 570 MACHINE LEARNING HOMEWORK-1

Sharath Kumar Karnati

20th September 2024

---

1 Analytical Part (2 percent)

1. Answer the following questions with a yes or no along with proper justification.

a. Is the decision boundary of voted perceptron linear?

Answer:
The decision boundary of a voted perceptron isn't linear. This happens because each individual classifier within the voted perceptron has its own weight vector, meaning each one has its own decision boundary. As a result, the overall decision boundary of the voted perceptron is formed by the combination of these different, nonlinear boundaries.

b. Is the decision boundary of averaged perceptron linear?

Answer:
The decision boundary of an averaged perceptron is linear. This is because we calculate the average of all the weight vectors from the individual classifiers to get one single weight vector. As a result, the decision boundary for the averaged perceptron ends up being 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.

Answer:
Let $\mathbf{w}_{t+1}$ be the solution to the optimization problem:

$$\mathbf{w}_{t+1} = \min_{\mathbf{w}} \frac{1}{2}||\mathbf{w} - \mathbf{w}_t||^2$$

such that $y_t(\mathbf{w} \cdot \mathbf{x}_t) \geq M$.

The Lagrangian is given by:

$$\mathcal{L}(\mathbf{w}, \tau) = \frac{1}{2}||\mathbf{w} - \mathbf{w}_t||^2 + \tau(M - y_t(\mathbf{w} \cdot \mathbf{x}_t))$$

$$= \frac{1}{2}||\mathbf{w} - \mathbf{w}_t||^2 + \tau M - \tau y_t(\mathbf{w} \cdot \mathbf{x}_t)$$

Now, solving the dual optimization problem:

$$\max_{\tau \geq 0} \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \tau)$$

Taking the partial derivative of $\mathcal{L}$ with respect to $\mathbf{w}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \mathbf{w}_t - \tau y_t \mathbf{x}_t$$

Setting $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0$:

$$\mathbf{w} = \mathbf{w}_t + \tau y_t \mathbf{x}_t$$

Substituting $\mathbf{w} = \mathbf{w}_t + \tau y_t \mathbf{x}_t$ back into the Lagrangian:

$$\mathcal{L}(\tau) = \frac{1}{2}||(\mathbf{w}_t + \tau y_t \mathbf{x}_t - \mathbf{w}_t)||^2 + \tau(M - y_t(\mathbf{w}_t + \tau y_t \mathbf{x}_t) \cdot \mathbf{x}_t)$$

$$= \frac{1}{2}||\tau y_t \mathbf{x}_t||^2 + \tau(M - y_t \mathbf{x}_t \cdot \mathbf{w}_t - \tau(y_t \mathbf{x}_t)^2)$$

$$= -\frac{1}{2}||\tau y_t \mathbf{x}_t||^2 + \tau(M - y_t \mathbf{x}_t \cdot \mathbf{w}_t)$$

$$= -\frac{\tau^2}{2}||\mathbf{x}_t||^2 + \tau(M - y_t \mathbf{x}_t \cdot \mathbf{w}_t)$$

(Note: $y_t^2 = 1$)

The dual optimization problem is:

$$\min_{\mathbf{w}} -\frac{\tau^2}{2}||\mathbf{x}_t||^2 + \tau(M - y_t(\mathbf{x}_t \cdot \mathbf{w}_t))$$

Taking the derivative with respect to τ:

$$\frac{\partial \mathcal{L}}{\partial \tau} = -2\tau||\mathbf{x}_t||^2 + M - y_t(\mathbf{x}_t \cdot \mathbf{w}_t)$$

Setting $\frac{\partial \mathcal{L}}{\partial \tau} = 0$:

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

Thus, the final equation for the Passive-Aggressive weight update is:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau y_t \mathbf{x}_t$$

$$= \mathbf{w}_t + \frac{M - y_t(\mathbf{x}_t \cdot \mathbf{w}_t)}{||\mathbf{x}_t||^2} y_t \mathbf{x}_t$$

3.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.

Answer: $(x_1, y_1, h_1), (x_2, y_2, h_2), \ldots, (x_n, y_n, h_n)$ represent additional or important weights given to the training examples. These weights provide extra information about each training example.

The regular perceptron's weight update is given by:

$$\mathbf{w}_t = \mathbf{w}_{t-1} + y_t \cdot \mathbf{x}_t$$

This weight update shifts the linear decision boundary closer to the misclassified example, placing it on the correct side of the label. We can incorporate the importance parameter by modifying the weight update as follows:

$$\mathbf{w}_t = \mathbf{w}_{t-1} + h_t \cdot y_t \cdot \mathbf{x}_t$$

The importance weight h_t is included in the weight and bias update of the standard perceptron. If the importance of an example is large, and the perceptron misclassifies it, there will be a large weight update, dramatically increasing the weight vector and, consequently, the decision boundary. On the other hand, if the importance of the example is low, even when the perceptron makes a mistake, the weight update will be small, and the decision boundary will change only slightly.

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.

Answer: We can use weighted sub-sampling, which means taking positive and negative instances and giving them equal weight relevance, resulting in balanced samples.

We can also apply scaling, where we increase the number of samples based on their importance.

Let the learning rate be 1.

- If no mistake occurs:

$$\mathbf{w}_{t+1} = \mathbf{w}_t$$

$$b_{t+1} = b_t$$

- If a mistake occurs:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y'_t \cdot \mathbf{x}_t$$

where $y'_t = y_t \cdot h_t$

$$b_{t+1} = b_t + y'_t$$

where $y'_t = y_t \cdot h_t$

4.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 percent of the examples are negative and 10 percent 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.

Answer: Let the learning rate be set to 1. If no mistake occurs, the weight and bias remain unchanged:

$$\mathbf{w}_{t+1} = \mathbf{w}_t$$

$$b_{t+1} = b_t$$

If a mistake occurs, the weight and bias are updated as follows:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y_t \cdot \mathbf{x}_t \cdot h_t$$

$$b_{t+1} = b_t + y_t \cdot h_t$$

Given that the data is imbalanced, with 90% negative and 10% positive examples, we can introduce a parameter h to account for this imbalance. Since we have fewer positive examples, we assign them a weight of 10, while negative examples are assigned a weight of 1. This makes the data more balanced during training.

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.

Answer:

We can apply sub-sampling, which involves taking only 10% of the negative examples. As a result, the perceptron will train on more balanced data, improving efficiency.

Alternatively, we can employ scaling, which involves increasing the number of positive examples to match the number of negative examples. This ensures that the positive and negative examples are equal in number during training.

5.1 Binary Classification: 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.

Answer:

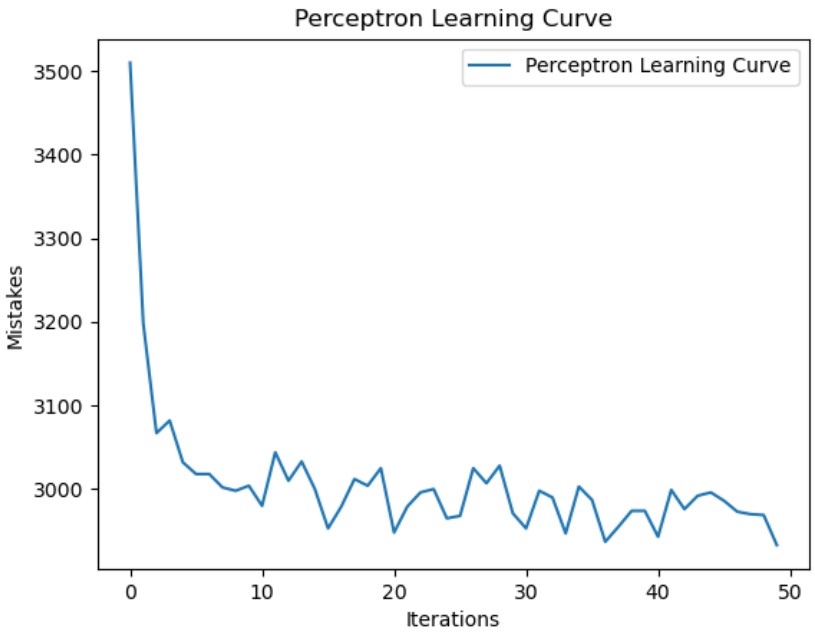

Figure 1: Perceptron Learning Curve.

- The learning curve for the Perceptron model shows that the number of mistakes starts at a maximum of 3,500 and goes down to a minimum of 2,900. With each iteration, the number of errors keeps decreasing, which means the model is getting more accurate and confident in its predictions as it learns from the data..

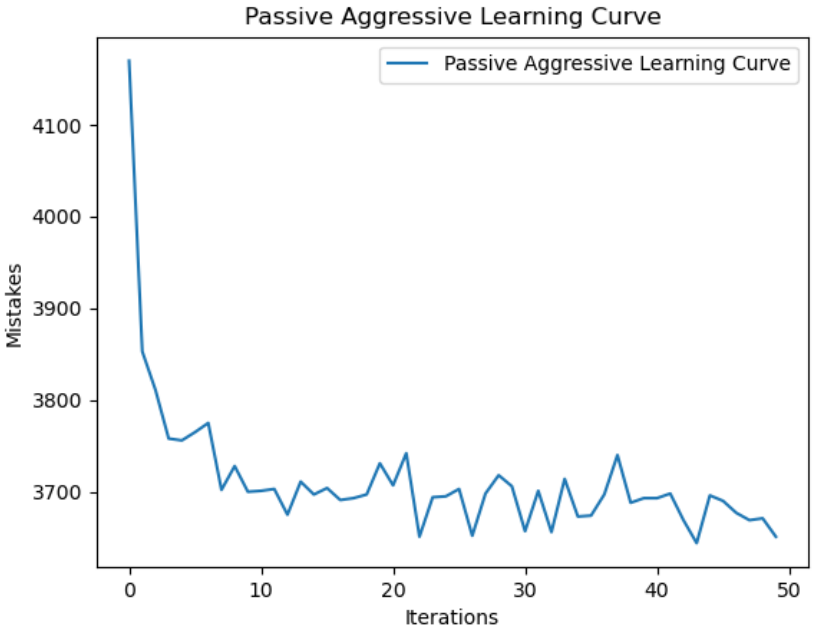

Figure 2: Passive Aggressive Learning Curve.

- In the Passive Aggressive model, the learning curve shows that the number of mistakes starts at 4,300 and gradually decreases to 3,500. As the training progresses through each iteration, the model makes fewer errors, indicating it's learning and improving its accuracy over time.

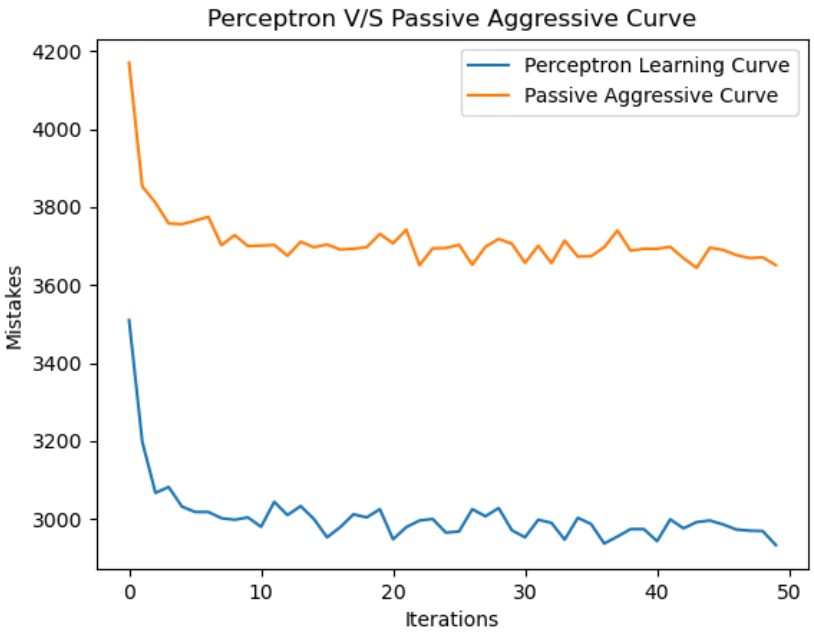

Figure 3: Passive Aggressive Vs Perceptron Learning Curve.

- When we compared both the algorithms, we can observe that the Perceptron makes less mistakes than compared to the passive aggressive algorithm.

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.

Answer:

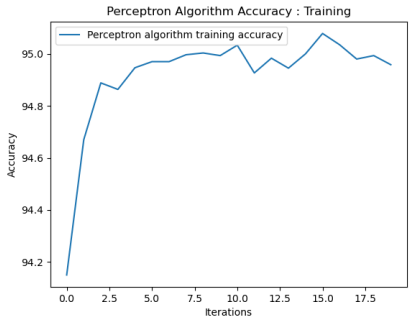

Figure 4: Perceptron Accuracy:for training data

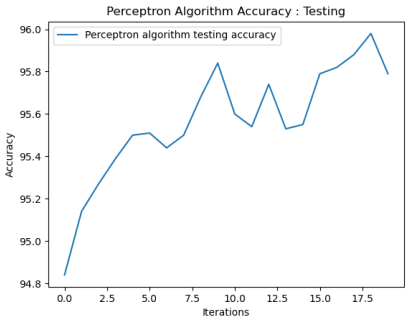

Figure 5: Perceptron Accuracy:for testing data

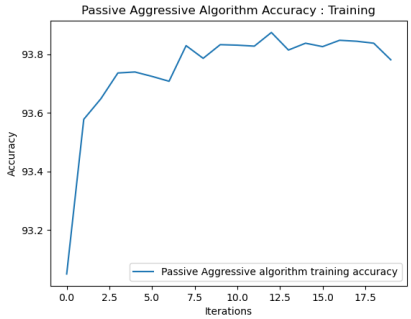

Figure 6: PA Accuracy:for training data

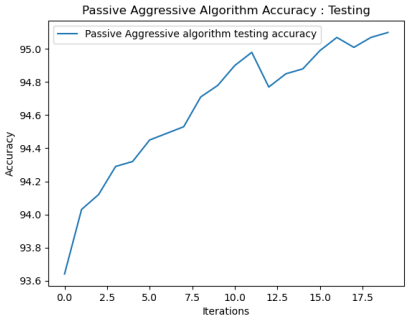

Figure 7: PA Accuracy:for testing data

- The accuracy curves in Figures 4 through 7 display the performance of both the Perceptron and Passive Aggressive (PA) algorithms on training and testing datasets. From these graphs, it's clear that for the training data, the Perceptron consistently achieves higher accuracy compared to Passive Aggressive during each repetition. However, when we look at the development and test datasets, the Passive Aggressive algorithm shows steady improvement in accuracy with each iteration. On the other hand, the Perceptron initially improves, but as the number of iterations increases, its accuracy spikes and then significantly drops.

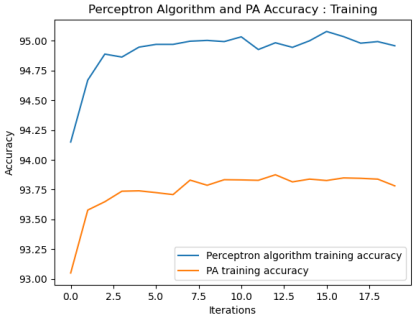

Figure 8: PA and Perceptron Accuracy :for training data

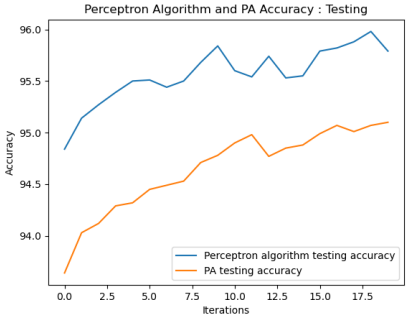

Figure 9: PA and Perceptron Accuracy:for testing data

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

Answer:

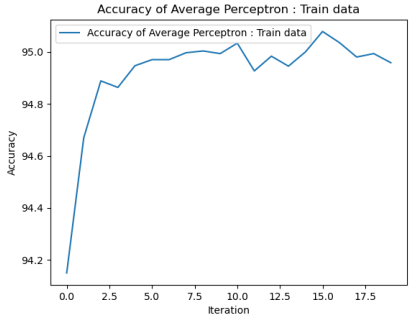

Figure 10: Averaged Perceptron Accuracy :for training data

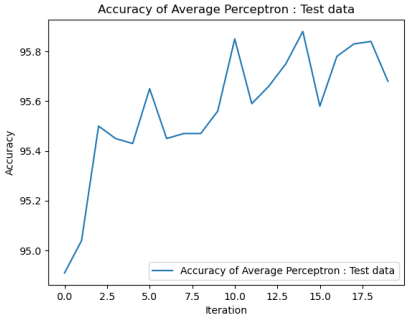

Figure 11: Averaged Perceptron Accuracy:for testing data

- The accuracy curves in Figures 4,5,10,11 illustrate the performance of the Perceptron and Averaged Perceptron for both the training and test datasets. By analyzing these graphs, we can see that the training accuracy curves for both the Perceptron and Averaged Perceptron are quite similar. However, the testing accuracy of the normal Perceptron is noticeably better compared to the averaged Perceptron. Based on this observation, we can conclude that the normal Perceptron performs better overall, especially on unseen data, making it a stronger model than the averaged Perceptron.

d. Compute the general learning curve (vary the number of training examples starting from 100 in the increments of 100) 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.

Answer:

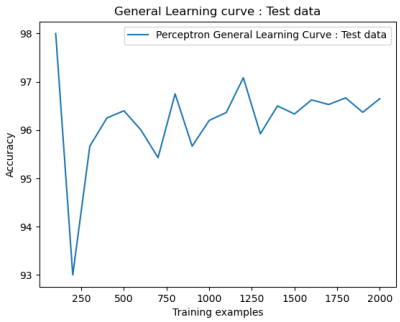

Figure 12: General Learning Curve for Perceptron

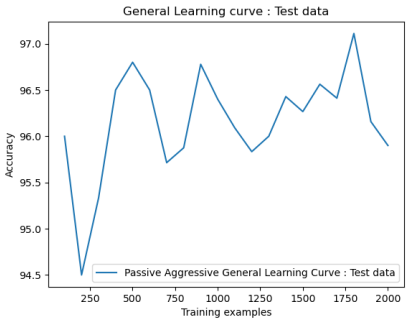

Figure 13: General Learning Curve for passive aggressive

- The Perceptron's testing accuracy starts off lower but improves as it learns over successive iterations, eventually surpassing its earlier performance. In contrast, the Passive Aggressive model shows a slight initial decline in accuracy, but it then learns rapidly through the iterations. However, this aggressive learning results in a decrease in accuracy towards the end of the process, suggesting that it may be overfitting or struggling to generalize effectively.

5.2 Multi-Class Classification Learn a multi-class classifier to map images to the corre- sponding 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.

Answer:

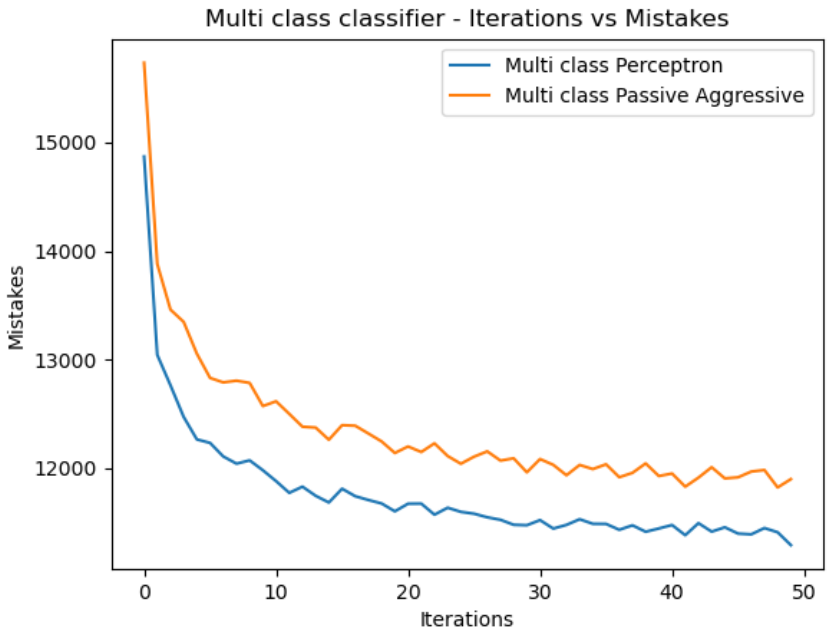

Figure 14: PA vs Perceptron Learning curves.

- The Perceptron model demonstrates superior performance compared to the Passive Aggressive model, consistently exhibiting fewer errors. This pattern holds true across both binary and multiclass classifiers, as the same notation is applied in both cases. Therefore, we can infer that this trend of the Perceptron outperforming the Passive Aggressive model may extend to other categorization models based on these algorithms.

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.

Answer:

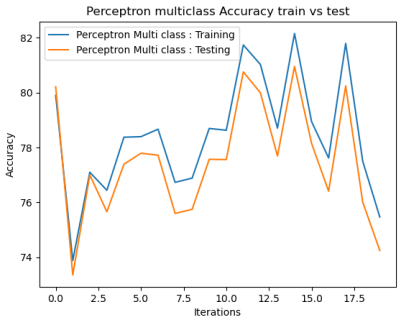

Figure 15: Perceptron Multiclass Accuracy : Training vs Testing

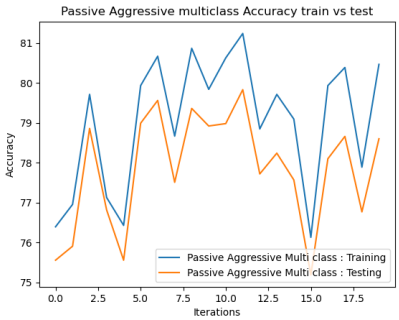

Figure 16: PA Multiclass Accuracy : Training vs Testing

- The Perceptron starts with higher accuracy compared to the Passive Aggressive model but eventually falls below it. Additionally, the gap between the training and testing accuracy for the Perceptron is significantly smaller than that for the Passive Aggressive model. In multiclass classifiers, the test accuracy is lower than the training accuracy, which contrasts with the behavior observed in binary classifiers. Overall, while the Perceptron maintains higher general training and testing accuracy than Passive Aggressive, it experiences a decline in accuracy during the final few iterations.

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

Answer:

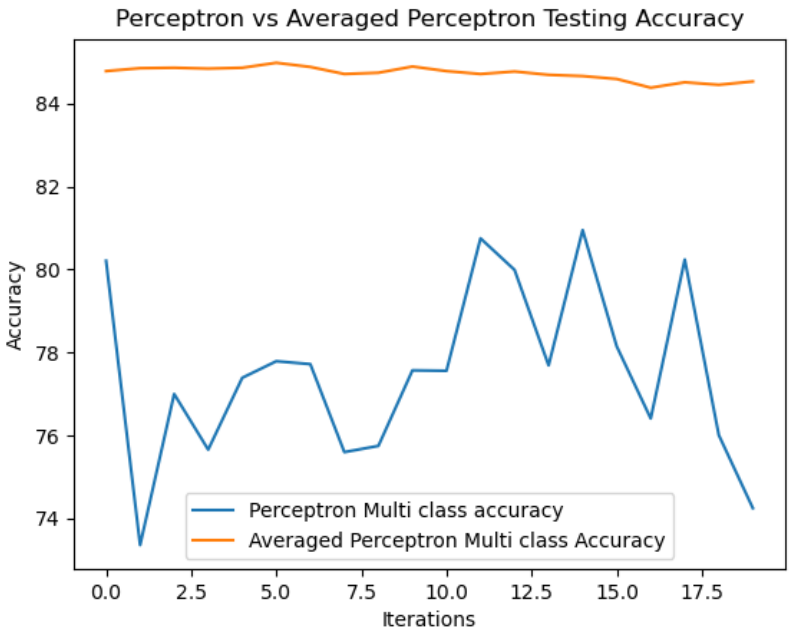

Figure 17: Averaged Perceptron vs Perceptron .

- The Perceptron's test accuracy initially drops but then improves, reaching a range of 74 to 80 percent. In contrast, the Averaged Perceptron's test accuracy consistently stays above 84.5 percent. This indicates that the Averaged Perceptron demonstrates higher test accuracy compared to the standard Perceptron, reinforcing its effectiveness as a model.

d. Compute the general learning curve (vary the number of training examples starting from 100 in the increments of 100) 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.

Answer:

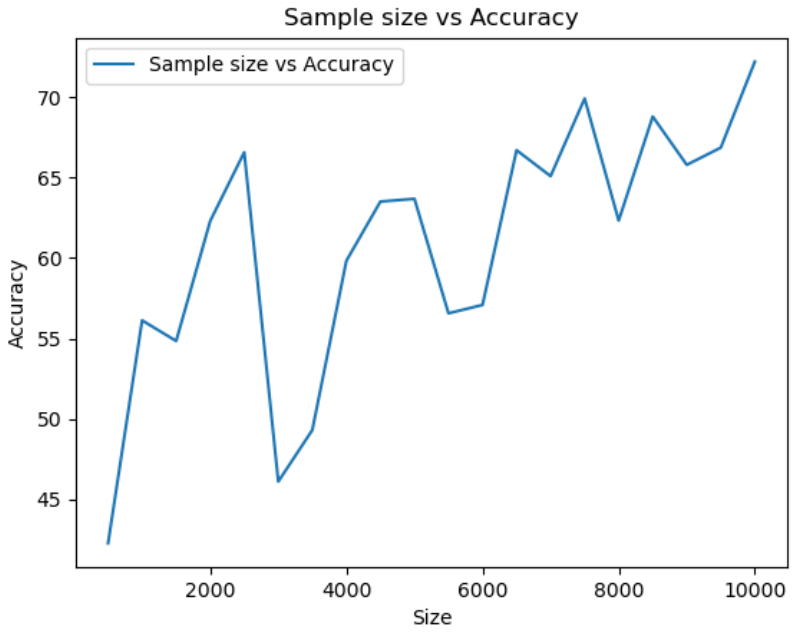

Figure 18: General Learning Curve for Multiclass Classifier

- When the sample size is small, we can observe fluctuations in accuracy, as the model struggles to learn effectively from the limited data. In this case, accuracy ranges from 77 to 85 percent. However, as the number of iterations increases, the graph becomes more stable, and we can see that the accuracy levels off around 80 percent. This stabilization indicates that the model is better able to generalize and perform consistently with more iterations.