Homework I

09/25/2019

Instructor: Janardhan Rao (Jana) Doppa

Student: Yu-Chieh Wang 11641327

1. Analytical Part

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?

   Answer:

   The decision boundary of voted perceptron is not linear, but averaged perceptron's is linear. According to the design of voted perceptron algorithm, the order of features will affect the result(final weight) of the training. However, averaged perceptron can solve this problem. Because of its design, the algorithm of averaged perceptron can get a better result which can make the rest results more stable.

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:

   The algorithm of PA has a learning rate which in perceptron is 1. According to the course PPT, we can get the equation which tries to achieve a margin equal to one as follow: $1 = y_t((w_t + \tau y_y x_t) \cdot x_t) = y_t(w_t \cdot x_t) + \tau \parallel x_t \parallel^2$

   Then, after solving the equation, we can get the learning rate $\tau = \frac{1 - y_t(w_t \cdot x_t)}{\parallel x_t \parallel^2}$

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 informa- tion? Please justify your answer.

Answer:

I will change the prediction method to $\hat{y}_t = sign(w \cdot x_t \cdot h_t)$. Because h_t follows by x_t , it should be related to the importance of each training material. Then, I am going to modify the method of updating weights as following: $w = w + \tau \cdot x_t \cdot h_t \cdot y_t$, so we can sure that the final weight we get is correct for prediction.

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 make some changes to the data before entering the perceptron algorithm. For example, we can multiply x_t and y_t in advance, so we will get a new symbol $\hat{x}_t$. Then, the function will look like this $\hat{y}_t = sign(w \cdot \hat{x}_t)$ which is as similar as the standard perceptron.

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.

Answer:

In this case, I will use averaged perceptron algorithm which I don't have to worry about the order of training data. On the other hand, in standard perceptron algorithm, the weight calculated by the the computer will be affected by the order of training data. For example, there will be a huge difference between training negative first and then training positive and training average scattered examples. Therefore, the standard one is more suitable for being used on evenly distributed data.

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.

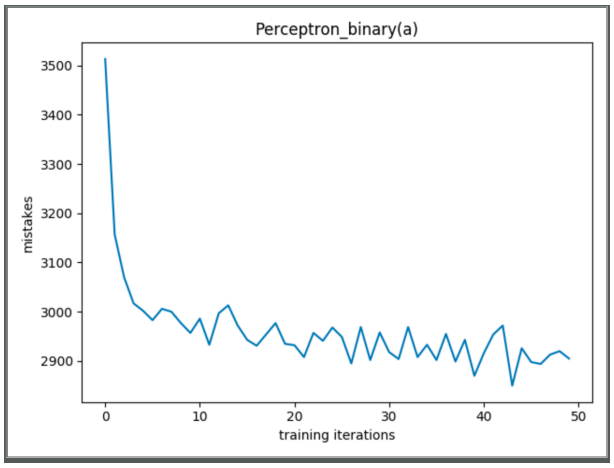
Answer:

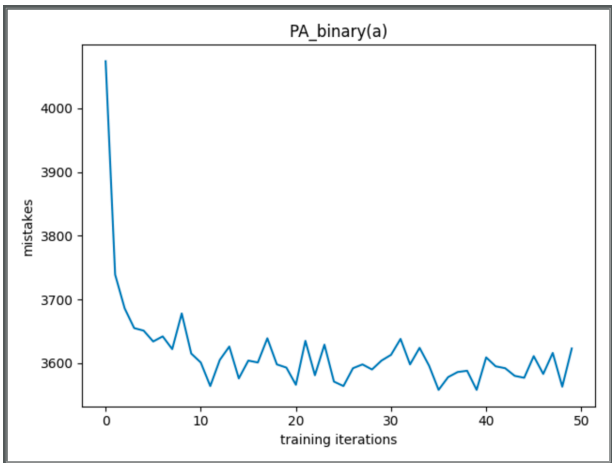
If I can only use the standard perceptron algorithm, I will break up all the examples and rearrange them. Moreover, I can rearrange the samples multiple times and train to get a more accurate weight.

2. **Programming and Empirical Analysis Part**

    1. Binary Classification

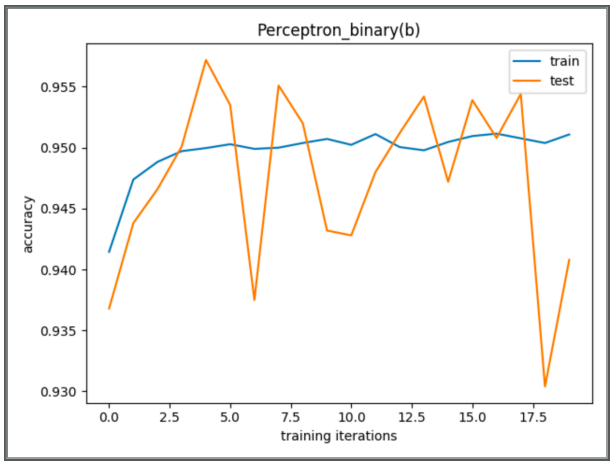
        A. Using Perceptron and PA algorithms to run training examples 50 times.

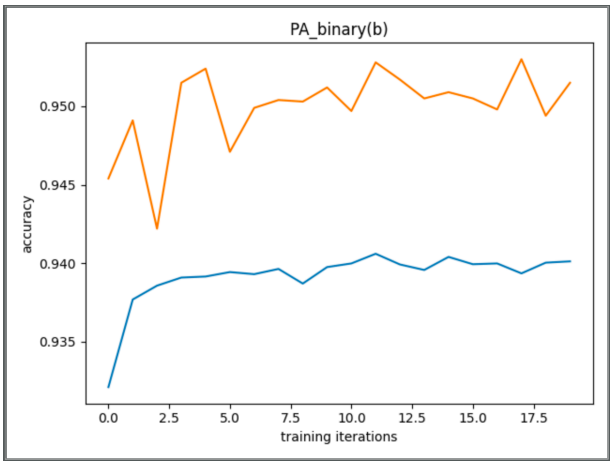




-To compare these two curves, both of them are going to be stable after training 10 times. However, PA has more mistakes than Perceptron.

B. Using Perceptron and PA algorithms to run training and testing examples 20 times.





-According to the two plots, we can see PA is more stable. Although its accuracy of training looks lower than Perceptron's, its testing accuracy is higher than its training accuracy. Therefore, the accuracy of PA is more convincing.

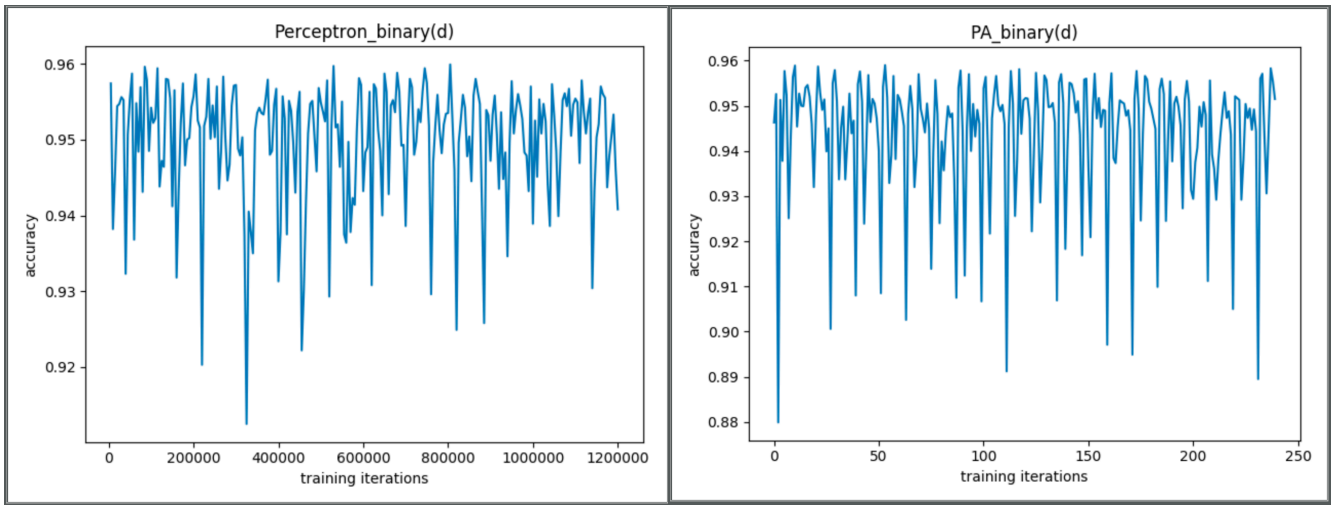
C. Compare the test accuracies of Plain Perceptron and Averaged Perceptron.

```
/usr/local/bin/python3.7 /Users/angel/PycharmProjects
Average accuracy:  0.9607
Plain: accuracy 0.9408

Process finished with exit code 0
```

-According to the result, the Averaged Perceptron has higher accuracy.

D. Run training examples 20 times to get the Perceptron and PA's general learning curves(run 5000 times test 1 time).

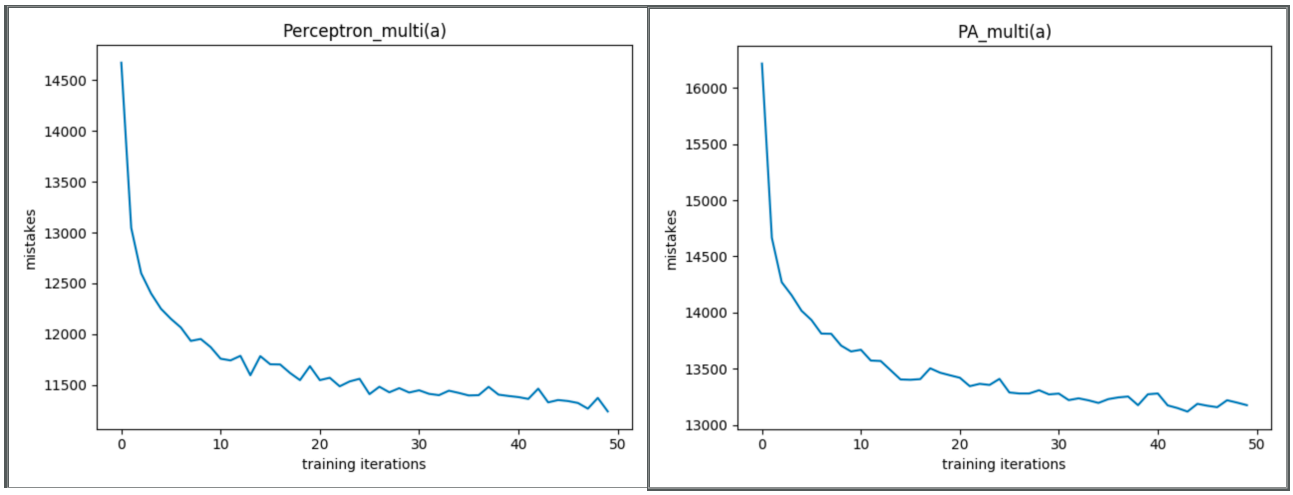


-It totally counts 240 times of testing accuracy.

-According to the two plots, the range of PA's result is bigger, but both of PA and Perceptron's accuracies are high.

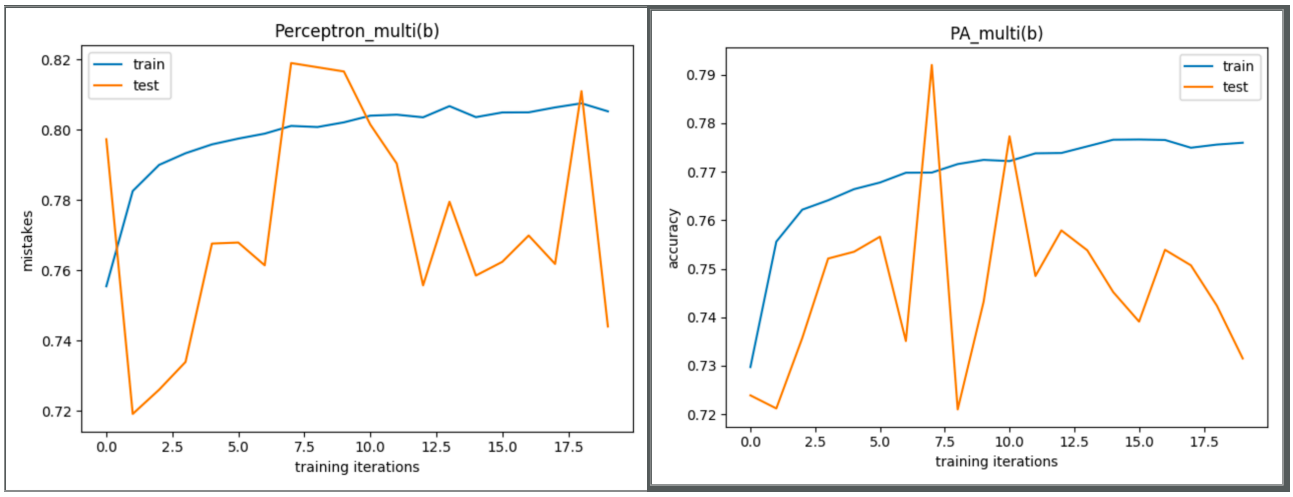
2. Multi-Class Classification

A. Using Perceptron and PA algorithms to run training examples 50 times.



-According to the two plots, the situation is as same as binary part. PA' mistakes are higher than Perceptron's.

B. Using Perceptron and PA algorithms to run training and testing examples 20 times.



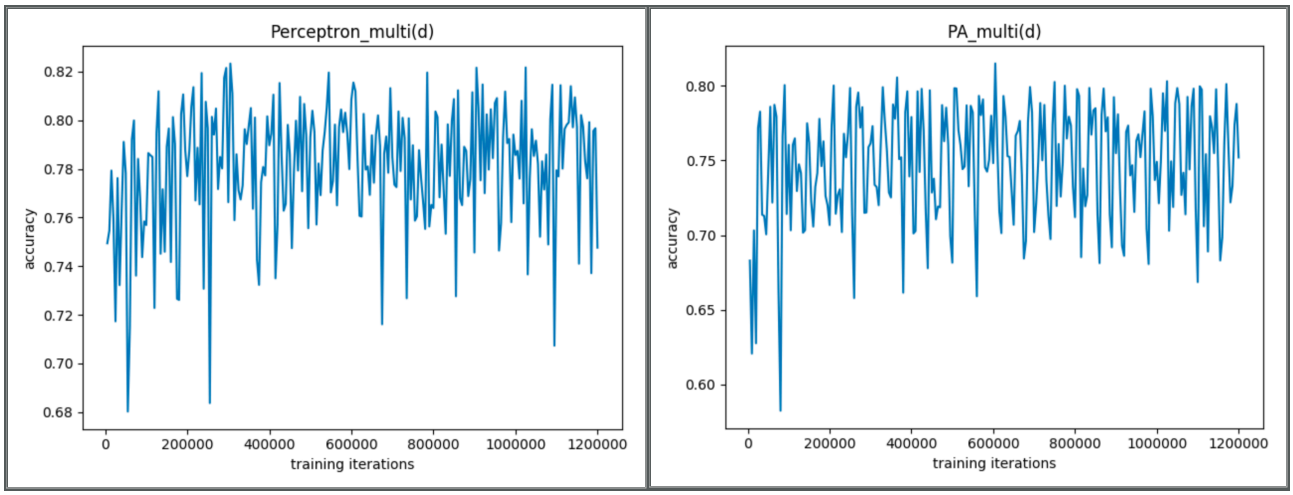
-This result is different from the binary part. Both of Perceptron and PA's testing accuracies are lower than training accuracies. It is worth noting that both of their training curves are similar, but their testing curves are very different.

C. Compare the test accuracies of Plain Perceptron and Averaged Perceptron.

```
Average accuracy:  0.7357
Plain: accuracy 0.744
```

-This result is different from the binary part which Averaged Perceptron has higher accuracy. Moreover, the results of Average and Plain accuracy are really close.

D. Run training examples 20 times to get the Perceptron and PA's general learning curves(run 5000 times test 1 time).



-According to the two plots, we can find that after training with a certain number of examples by PA method, the accuracy of the prediction results will be as small as a certain range.

SUMMARY

According to the article, “A Few Useful Things to Know About Machine Learning[1]” authored by P. Domingos, the author highlights several key points of machine learning for people who are learning it.

First of all, learning equals to representation plus evaluation and optimization. We choose a programming language to represent classifiers at the beginning. Then, we evaluate these classifiers and pick one of them which has highest score of prediction. Second, generalization is important. Because the examples we have are a small part of the real world, we cannot use all examples for training but leave some of them for testing, or the training result may has low accuracy. Third, data alone is not enough. In addition to the data we have, we still need to improve our professional skills and academic knowledge so that we can make our machine learning become stronger. Fourth, overfitting is related to multi-test, but we can solve this problem by cross-validation. Fifth, intuition fails in high dimensions. The increase in dimensions makes the operation more complicated and more difficult to debug. Sixth, Theoretical guarantees are not what they seem. The most common machine theory is how many examples of machine learning to train to reach a certain degree of precision. However, we must accept that this is just a theory, in fact the results may be different. Seventh, feature engineering is the key. Algorithms tailored to different goals can help machine learning results achieve the highest level of precision. Eighth, more data beats a cleverer algorithm because collecting data is easier than build a hard algorithm. Also, computers will spent a lot of time to learn hard algorithm. Relatively, collecting spends less time. Ninth, we should learn more models because they have their own place where they can come in handy. Tenth, simple learning algorithm is not equal to high accuracy. There is no free lunch. It is impossible to achieve everything in the simplest way. Next, the algorithm which can be represent does not mean it can be learned. For example, standard dedication tree cannot learn examples with too much leaves. Finally,

although we try to find the relationships between training examples, the prediction results may not have any relationships between each other.

Reference

[1] P. Domingos, “A Few Useful Things to Know About Machine Learning,” in COMMUNICATIONS OF THE ACM, vol. 55, no. 10, pp. 78-87, Oct. 2012. doi: 10.1145/2347736.2347755