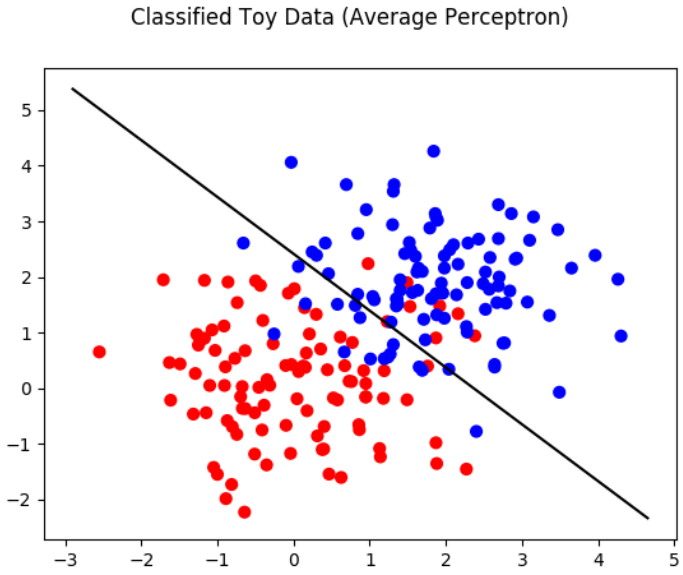
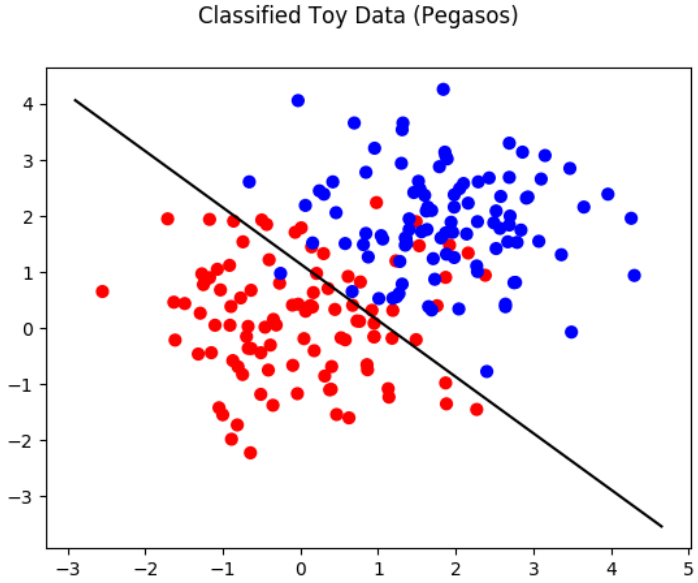
Milo Knowles
6.036
March 3, 2017

Project 1 Writeup

1.7. **The perceptron and average perceptron have different decision boundaries because the average perceptron takes into account all previous theta vectors, whereas the normal perceptron only returns the final theta. The Pegasos uses a different update step, and is stochastic, so its theta must be different from that of either perceptron algorithm.**





2.9b. **Below are the training and validation accuracies for each algorithm with T=10 and L=0.01:**

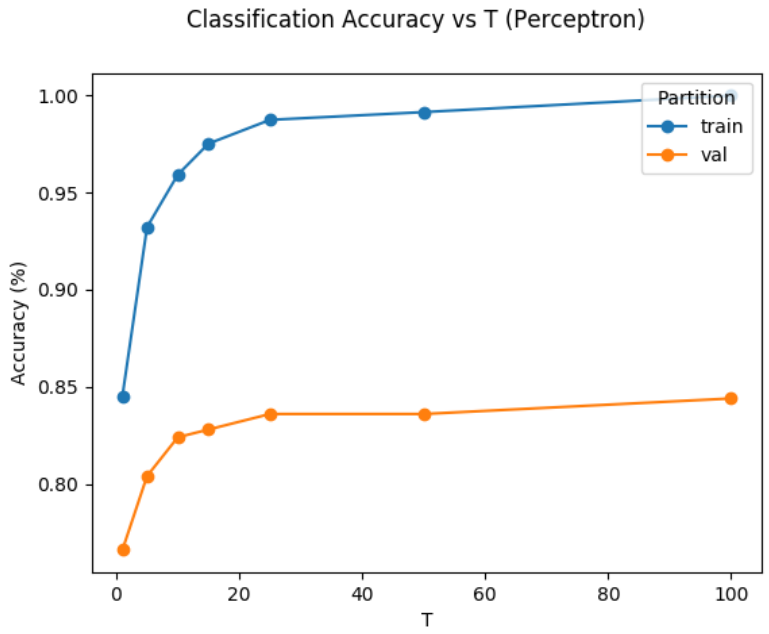
```
[mknowles (master *) project1 $ python3 main.py
Training accuracy for perceptron:   0.9593
Validation accuracy for perceptron: 0.8240
Training accuracy for average perceptron:   0.9760
Validation accuracy for average perceptron: 0.8420
Training accuracy for Pegasos:                      0.8752
Validation accuracy for Pegasos:                    0.8320
```

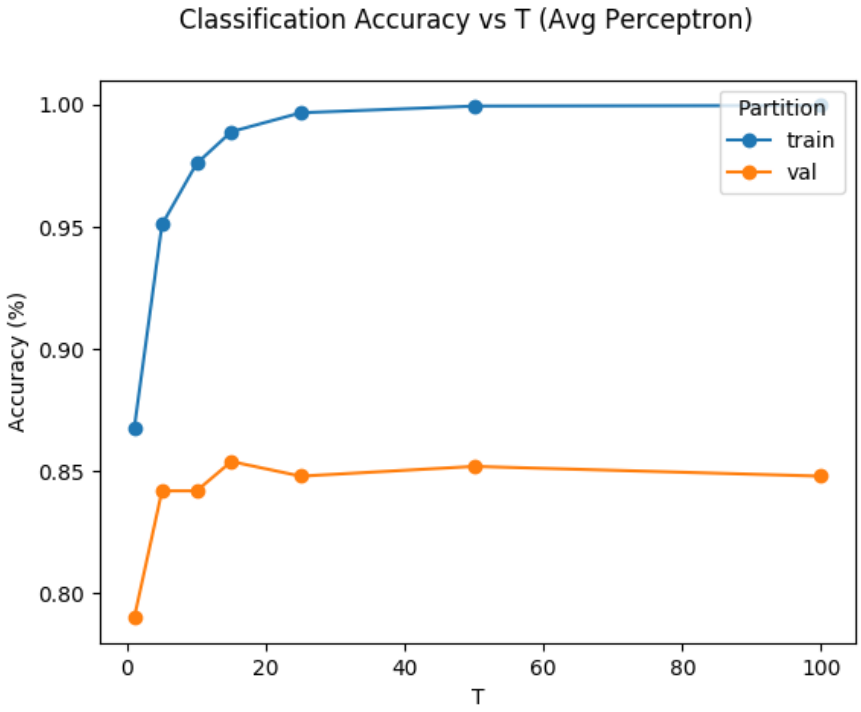
2.10.

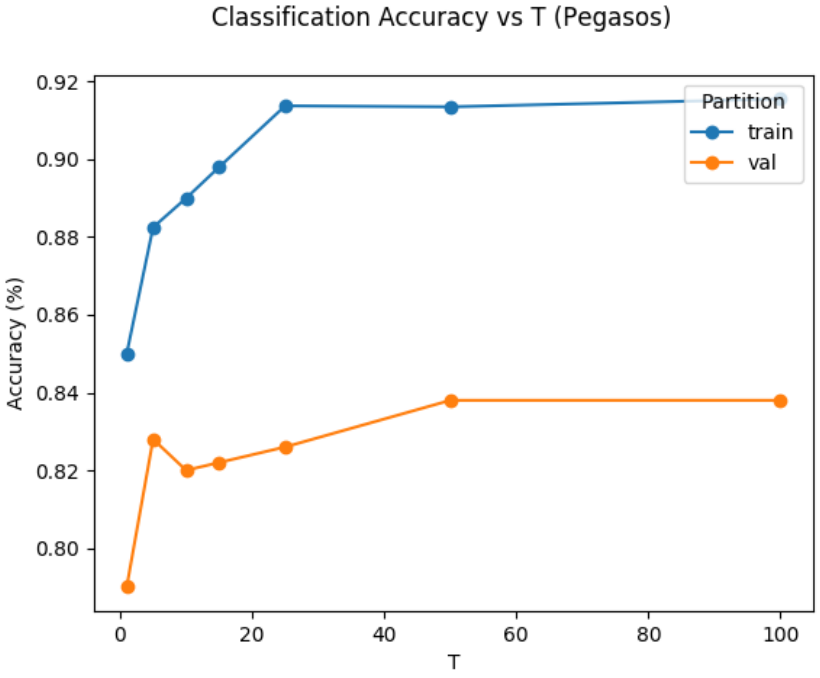
(a) The training and validation accuracies behave slightly differently. For all three algorithms, accuracy continues to improve on the training data. This is because the algorithm continues to fit theta to the training data, so accuracy asymptotically improves. However, on the validation data, performance begins to suffer for higher T for the average perceptron and pegasos algorithm. This is because we are overfitting to the training data, so the algorithm's generalization worsens. For the Pegasos algorithm, any increase in L causes the algorithm to perform poorly on both the training and validation sets.

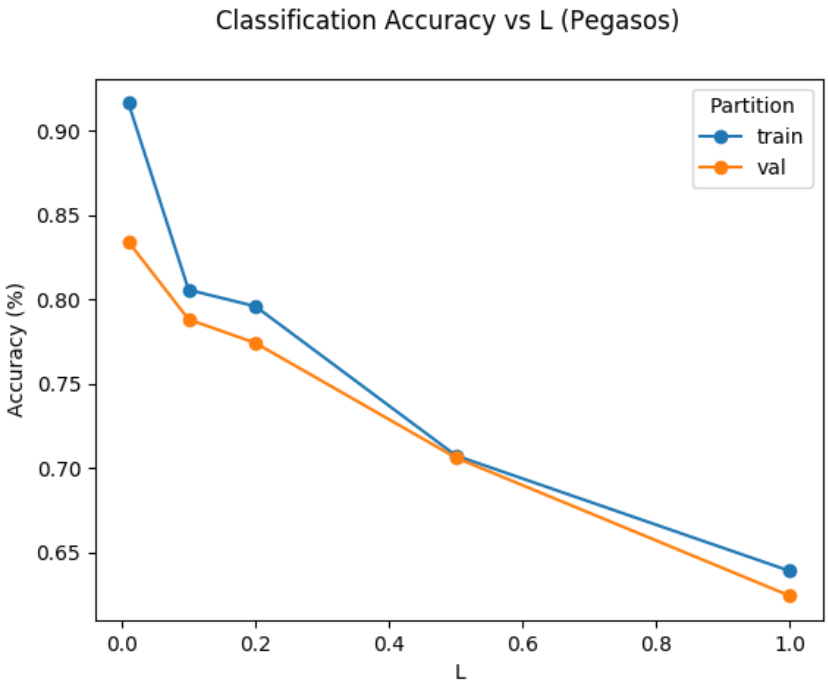
(b) The Average Perceptron performed the best out of the three.

(c) Best parameters:
Perceptron: T=100
Average Perceptron: T=14
Pegasos: T=50, L=0.01




Classification Accuracy vs T (Avg Perceptron)
Partition
train
val
Accuracy (%)
1.00
0.95
0.90
0.85
0.80
0
20
40
60
80
100
T


Classification Accuracy vs T (Pegasos)
Partition
train
val
Accuracy (%)
0.92
0.90
0.88
0.86
0.84
0.82
0.80
0
20
40
60
80
100
T



2.11.a. **Using T=14 for the average perceptron, test accuracy was 79.6%.**

2.11.b. **Most Explanatory Word Features:**
['delicious', 'amazing', 'wonderful', 'glad', 'pleased', 'plan', 'canned', 'helped', 'excellent', 'run']

3.12. **Methods Used to Improve Performance**

A. Removing Stopwords from the Feature Vectors

First, I chose to remove stopwords from the feature vectors, since these words should not provide much information about a review, and only enlarge the feature space. I compared the performance of the average perceptron with the original training features to the performance of the average perceptron trained with feature vectors with the stopwords removed.

Results of A:
Length of Unmodified Dictionary: 12688 words
Length of Dictionary With Stopwords Removed: 12561 words

Test set accuracies:
Average Perceptron (Unmodified): 79.6 %
Average Perceptron (Stopwords Removed): 79.2%

B. Adding Word Frequencies to the Feature Vector

Rather than simply use binary features that indicate whether or not a word is present in the text, I tried counting up the number of instances of that word in the text and using this sum as the value in the feature vector. To control the number of feature modifications I was testing at a given time, I tested this new modification without stopwords removed from the feature space. Below is a comparison of the performance of the average perceptron with the original binary features, and the performance of the average perceptron with word frequency features.

Results of B:
Test set accuracies:
Average Perceptron (Unmodified): 79.6 %
Average Perceptron (Stopwords Removed): 79.0%

C. Using both modifications

Notice that for both feature modifications, the performance of the average perceptron decreased slightly compared to the original. However, when both modifications were use in conjunction, the performance increased. Below, I included the performance results on the test set when stopwords were removed and word frequencies were used as features.

Final Results:
Test set accuracies:
Average Perceptron (Unmodified): 79.6 %
Average Perceptron (Stopwords Removed): 81.6%