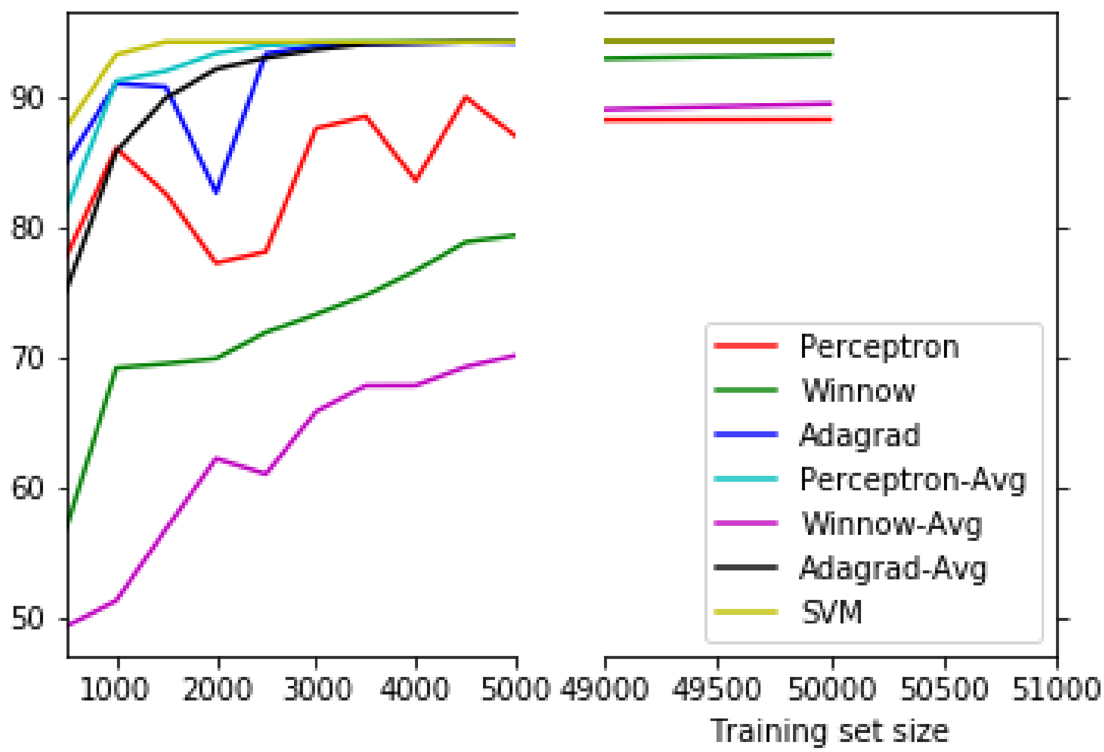
# CIS 419/519: Homework 2

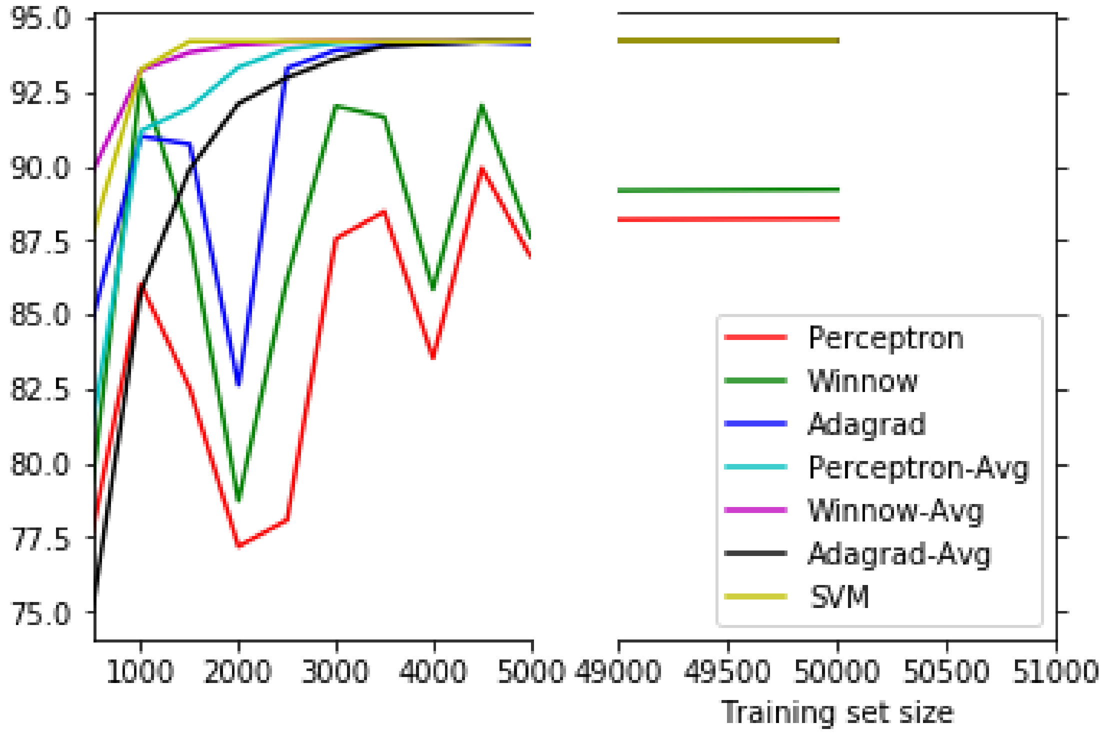
{Shruti Sinha}

Although the solutions are my own, I consulted with the following people while working on this homework: {Hemanth Kothapali, Ji Xiayan, Andres}

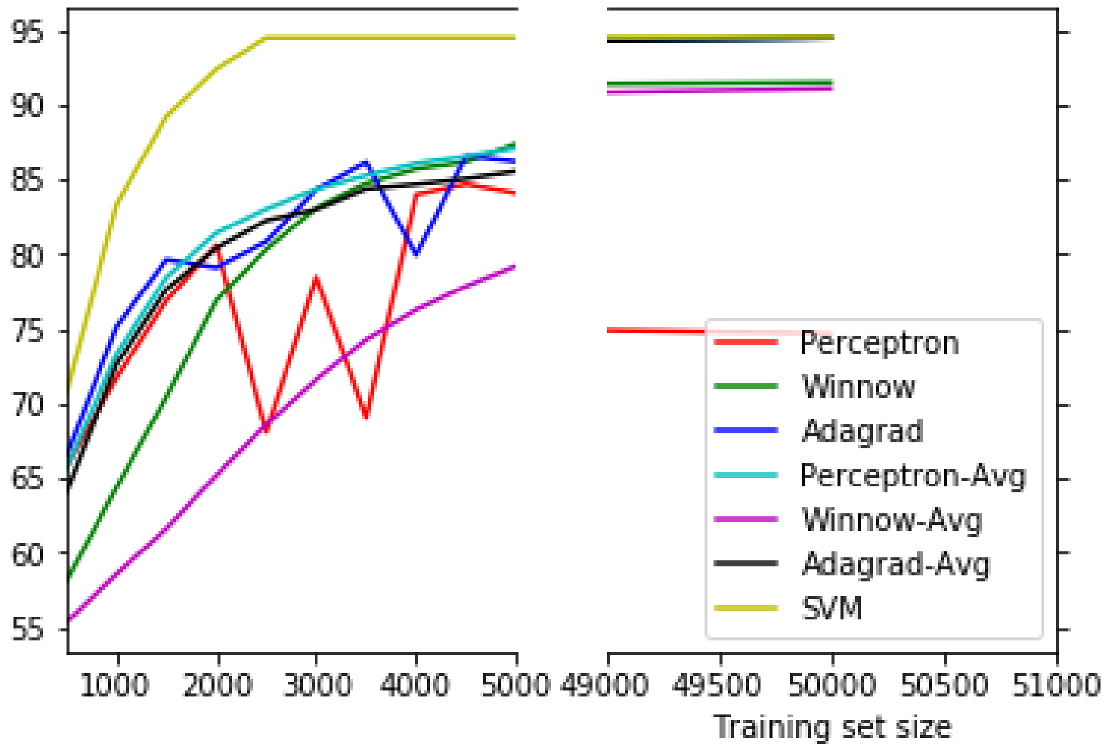
1. **Learning curves**

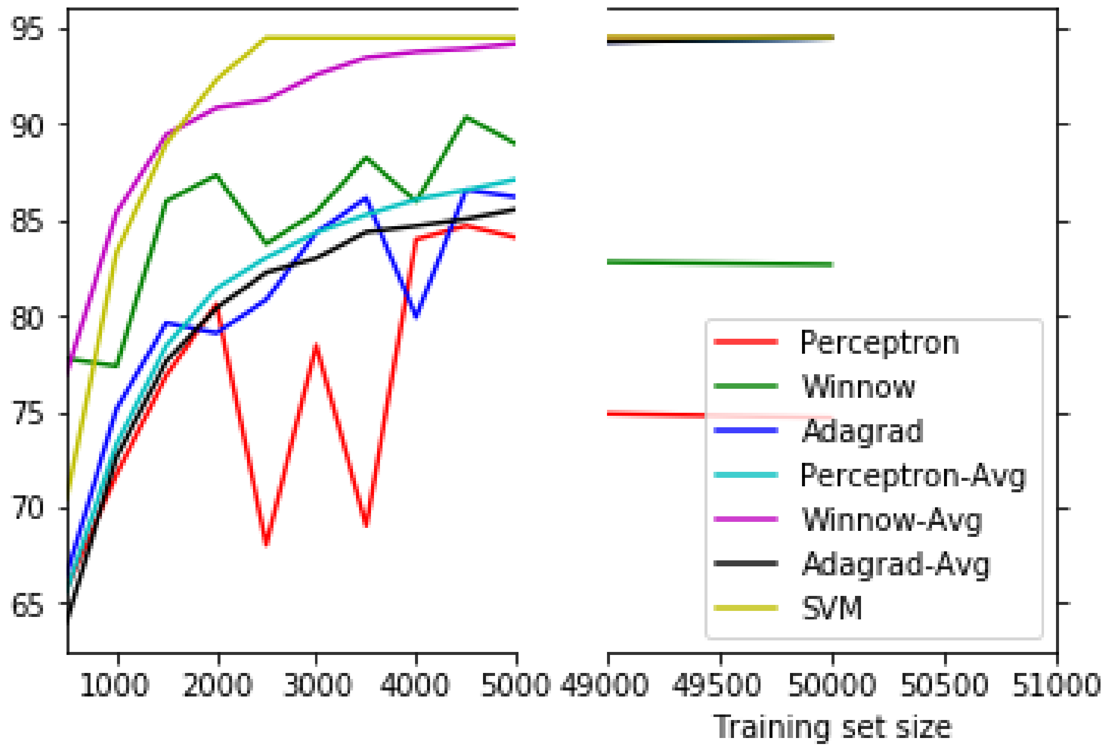
   **Dense data(Fig1 alpha = 1.0005, Fig2 alpha=1.1)**
   **Sparse Data(Fig3 alpha = 1.005, Fig4 alpha = 1.1)**
   Refer discussion Point 6





95.0
92.5
90.0
87.5
85.0
82.5
80.0
77.5
75.0
1000
2000
3000
4000
5000
49000
49500
50000
50500
51000
Training set size
Perceptron
Winnow
Adagrad
Perceptron-Avg
Winnow-Avg
Adagrad-Avg
SVM


95
90
85
80
75
70
65
60
55
1000
2000
3000
4000
5000
49000
49500
50000
50500
51000
Training set size
Perceptron
Winnow
Adagrad
Perceptron-Avg
Winnow-Avg
Adagrad-Avg
SVM




2. **Accuracy of sparse synthetic data**
   We choose alpha = 1.005 and eta = 1.5

| Algorithm | Train accuracy | Dev accuracy |
|---|---|---|
| Perceptron | 74.512 | 74.69 |
| Winnow | 91.998 | 91.51 |
| Adagrad | 94.59 | 94.49 |
| Perceptron averaged | 94.592 | 94.49 |
| Winnow averaged | 91.39 | 91.11 |
| Adagrad averaged | 94.576 | 94.48 |
| SVM | 94.592 | 94.49 |

3. **Accuracy of dense synthetic data**
   We choose alpha = 1.0005 and eta = 1.5

| Algorithm | Train accuracy | Dev accuracy |
|---|---|---|
| Perceptron | 88.79 | 88.23 |
| Winnow | 93.522 | 93.23 |
| Adagrad | 94.588 | 94.19 |
| Perceptron averaged | 94.588 | 94.19 |
| Winnow averaged | 89.604 | 89.41 |
| Adagrad averaged | 94.588 | 94.19 |
| SVM | 94.588 | 94.19 |

4. **Accuracy of dense synthetic data without noise**

| Algorithm | Train accuracy | Dev accuracy |
|---|---|---|
| Perceptron | 88.79 | 93.47 |
| Winnow | 93.522 | 98.86 |
| Adagrad | 94.588 | 100 |
| Perceptron averaged | 94.588 | 100 |
| Winnow averaged | 89.604 | 94.15 |
| Adagrad averaged | 94.588 | 100 |
| SVM | 94.588 | 100 |

5. **F1 of real data**

| Algorithm | Train F1 CONLL | Dev F1 CONLL | Dev F1 Enron |
|---|---|---|---|
| Perceptron averaged | 96.97 | 81.25 | 23 |
| SVM | 98.90 | 82.78 | 24.06 |

6. **Discussion**
   We test our algorithms over different target functions:Synthetic Dense data set, Synthetic Sparse Dataset and Real data set Conll and Enron.

   A few important observations:
   1)Parameter Tuning:
   Alpha - For each alpha value, we train Winnow algorithm on synthetic train dense dataset. As accuracies over the development data are evaluated for each parameter value, it is observed that alpha = 1.0005 gives the highest accuracy of 93.23. The same alpha gives the best training accuracy too. Thus, we choose alpha = 1.0005 for Winnow averaged for training on dense data.

   For sparse synthetic dataset, alpha = 1.005 gives the best accuracy of 91.51. Thus, alpha = 1.005 is chosen as the alpha for algorithm Winnow averaged too.

**Table1:Accuracy on dense data for different alpha**

| Alpha | Accuracy |
|---|---|
| 1.1 | 89.21 |
| 1.01 | 92.42 |
| 1.005 | 92.47 |
| 1.0005 | 93.23 |
| 1.0001 | 86.42 |

**Table2:Accuracy on sparse data for different alpha**

| Alpha | Accuracy |
|---|---|
| 1.1 | 82.66 |
| 1.01 | 91.23 |
| 1.005 | 91.51 |
| 1.0005 | 89.1 |
| 1.0001 | 67.78 |

Eta- For Adagrad, it is observed that eta = 1.5 gives the best accuracy for synthetic train dense as well as sparse dataset. It was observed that with increasing eta, the accuracy increased. With eta=0.001 giving the poorest accuracy of 50.73 on dense development data and eta=1.5 giving accuracy of 94.19.

**Table3:Accuracy on dense data for different eta**

| Eta | Accuracy |
|---|---|
| 1.5 | 94.19 |
| 0.25 | 94.14 |
| 0.03 | 74.39 |
| 0.005 | 65.84 |
| 0.001 | 50.73 |

**Table4:Accuracy on sparse data for different eta**

| Eta | Accuracy |
|---|---|
| 1.5 | 94.49 |
| 0.25 | 91.74 |
| 0.03 | 84.63 |
| 0.005 | 82.52 |
| 0.001 | 50.29 |

2)From the accuracy tables it was observed that perceptron had the poorest accuracy across sparse and dense dataset. Adagrad performed consistenly well with the best accuracy among the three non averaged version of perceptron, winnow and adagrad. Due to its adaptive learning rate, which assigns a higher learning rate to stable features and lower ones to those that frequently change, adagrad gives high accuracies.

3)SVM gave the best performance among all algorithms on dense data as well as sparse data.

4)Referring to the accuracy tables, we observe that the algorithms are indeed robust to the noise in the training set. Despite the presence of some noise in the training set, development accuracy ranging in 88-94 is observed, thus indicating that noise doesn't prevent the algorithm from learning the correct hypothesis.

When comparing performance between development and development with no noise dataset, Adagrad, Perceptron Averaged, Adagrad-Averaged and SVM gave accuracy of 100 for no noise set. This indicates that without noise, algorithms perform really well and thus are robust to noise present in training set.

5)Perceptron Average and SVM performance are compared over 4 different datasets. We consider here the case of transfer learning.
On observation we see, that with the same training set-news train, f1 score for news development is 82.78 while when tested over a different data set email development, a very low f1 score of 24.06 is observed with averaged perceptron algorithm. This presents the problem associated with transfer learning.

6)Two learning curves are shown for dense and sparse dataset one with alpha = 1.0005 for dense and alpha = 1.005 for sparse, which gives me highest accuracy on the entire 50k dataset and alpha = 1.1 for both dense and sparse which supports the theoretical nature of winnow algorithm.

Winnow uses multiplicative updates to get the final weight vector instead of additive updates used for Perceptron and Adagrad. Winnow being multiplicative learns faster and should give higher accuracies when compared with Perceptron. This is reflected in fig 2 and fig 4(alpha = 1.1)

However, alpha =1.0005 and alpha = 1.005 despite giving highest accuracy for entire dataset for dense and sparse respectively, indicate a smaller learning rate and thus winnow learns slower, resulting in poor performance when compared with perceptron as shown in fig1 and fig3. The table representing accuracies for each alpha are shown in Table1/Table2 for dense and sparse respectively.

7)It is also observed from the learning curves that the averaged counterparts of every algorithm gave a smoother increasing curve with increase in sample size, thus indicating that they produce a globally optimal solution on unseen data.

8)The average run time for sparse dataset took longer than dense dataset, due to the sparsity in the number of active features in sparse data. Also the run time for real dataset was longer than synthetic data due to the presence of large no. of examples.