**CS446: Machine Learning** **Spring 2017**

# Problem Set 3

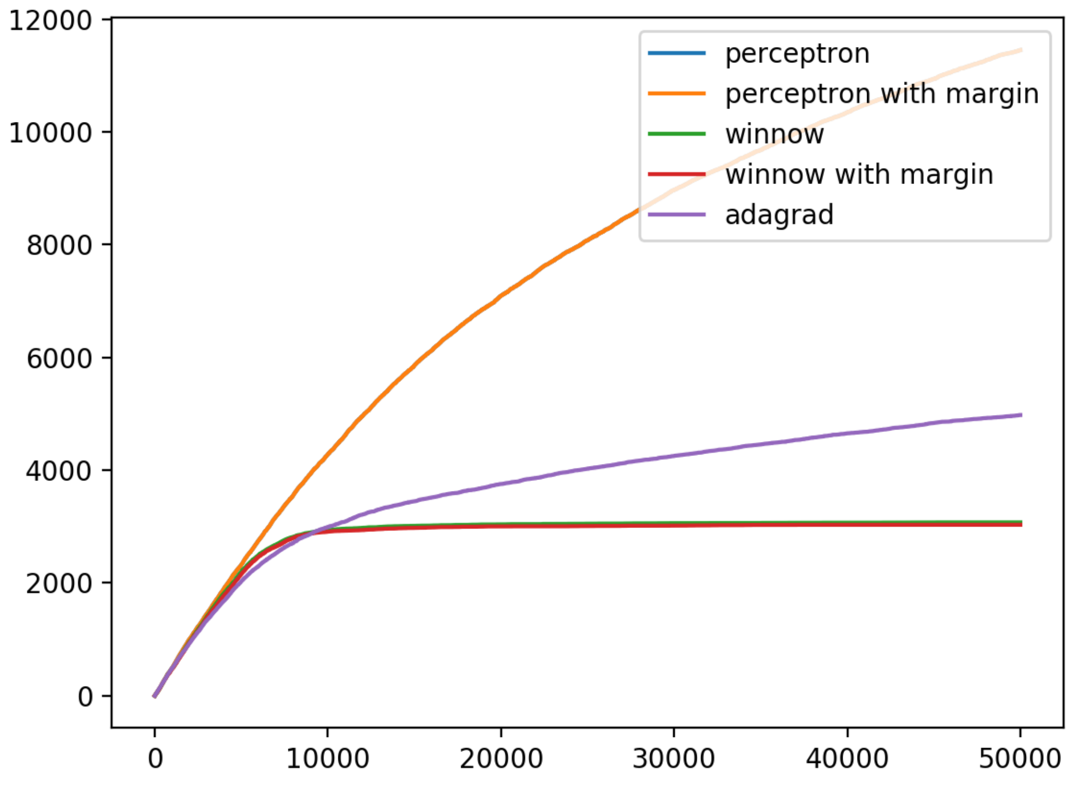
Zhiyu Zhu (zzhu24) *Handed In: February 28, 2017*

1. **Number of examples versus number of mistakes**

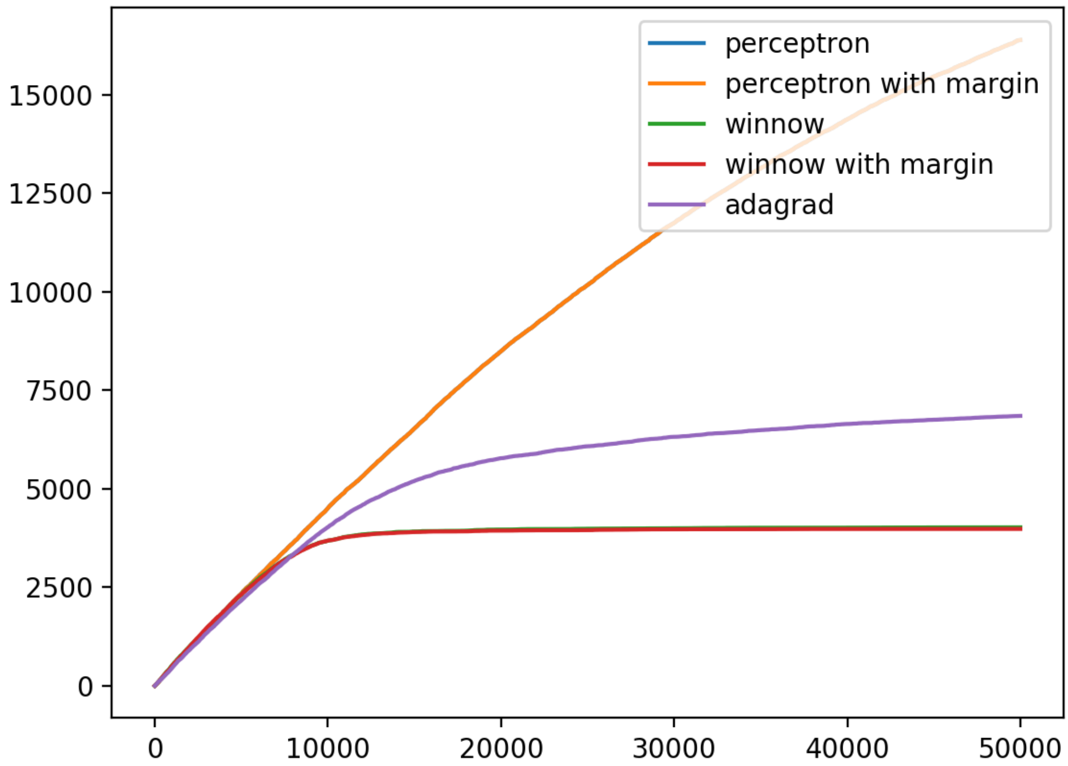
| `Algorithm` | `Parameter` | `n = 500` | `n = 1000` |
|---|---|---|---|
| *Perceptron* | Learning Rate | 1 | 1 |
| *Perceptronwithmargin* | Learning Rate | 1.5 | 0.25 |
| *Winnow* | Promotion/Demotion | 1.1 | 1.1 |
| *Winnowwithmargin* | Promotion/Demotion and Margin | $\alpha = 1.1\gamma = 0.04$ | $\alpha = 1.1\gamma = 0.3$ |
| *AdaGrad* | Learning Rate | 1.5 | 0.25 |

Table 1: Result Evaluation

n = 500:




n = 10000:




As we can see from the plot, the difference between Perceptron, winnow and AdaDrad are pretty obvious. But the plot of Perceptron is the same as the Perceptron with margin and the plot of Winnow is the same as the Winnow with margin.

(a) Analysis:
For n = 500, before the sample size reaches 500, there is no difference between the correctness of all algorithm. As the sample size grows, however, winnow much improve the performance and the mistakes number almost stop growing until sample size grows to 50000. The AdaGrad also has a good performance that the number of mistakes is a litter larger than that of Winnow. The Perceptron, however, as the training sample size grow, the percentage of mistakes generating does not decrease a lot.

For n = 10000, before the sample size reaches 1000, there is no difference between the correctness of all algorithm. The trend and differences between performance of algorithm is almost the same as that of n=500. As the sample size grows, however, winnow much improve the performance and the mistakes number almost stop growing until sample size grows to 50000. The AdaGrad also has a good performance that the number of mistakes is a litter larger than that of Winnow. The Perceptron, however, as the training sample size grow, the percentage of mistakes generating does not decrease a lot. Another diference is that the performance is worse than n = 500 but report a more precise learning rate, alpha and margin value.

(b) Explanation:
Since the target function is sparse, the multiplicative algorithm will have much better performance than the addictive algorithm. That's why we can see from the plot that the number of mistakes of winnow almost stop growing as the sample space grows. Another probe that multiplicative algorithm is better than addictive algorithm on sparse sample space is that when the n value grows form 500 to 1000, perceptron and adagrad makes more mistake than the first test because of the sparse grow. However, the number of winnow algorithm still remains the same without making more mistakes.

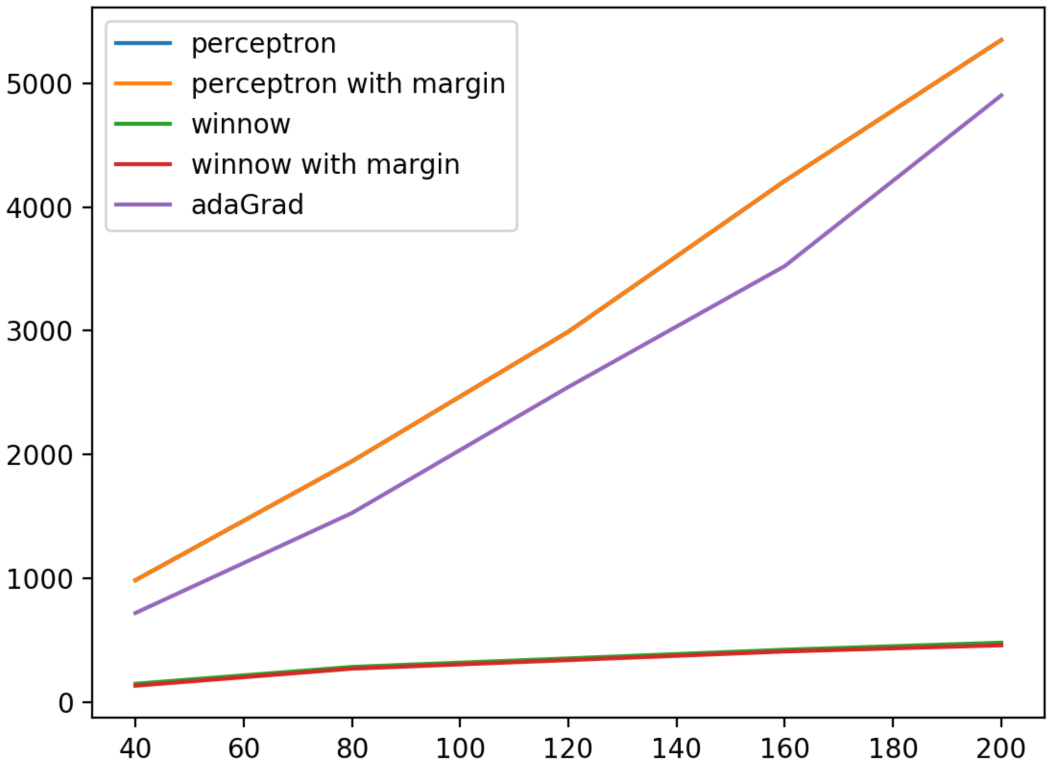
For perceptron, the margin makes it's mistakes to increase more slowly when the sparse is larger. However, we can hardly see the margin effect on the mistakes generating because it is less effective than the change of data sparsing.

For AdaGrad, since it updates its weight at a very fast speed and high frequency, the generating of mistakes is much slower than the perceptron when the sample space grows larger.

2. **Learning Curves of online learning algorithms**

| `Algorithm` | `Parameter` | `n = 40` | `n = 80` | `n = 120` | `n = 160` | `n = 200` |
|---|---|---|---|---|---|---|
| Perceptron | $\eta$ | 1 | 1 | 1 | 1 | 1 |
| Perceptron Margin | $\eta$ | 1.5 | 1.5 | 0.25 | 0.25 | 0.25 |
| Winnow | $\alpha$ | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 |
| Winnow Margin | $\alpha and \gamma$ | 1.1, 0.04 | 1.1, 0.04 | 1.1, 0.3 | 1.1, 0.3 | 1.1, 0.3 |
| AdaGrad | $\eta$ | 1.5 | 0.25 | 1.5 | 1.5 | 1.5 |

Table 2: Result Evaluation




(a) Analysis:
From the plot we can see that the mistake number of winnow is the most stable, it generates very litte mistakes and does note increases as the sparsing grows larger.

For Perceptron and AdaGrad, however, as the sparsing grows larger, the percentage of mistakes also increases. In addition, the performance of perceptron is little bit worse than taht of AdaGrad.

(b) Explanation:
In this training data, we still did not added in the noise. So since winnow is much better at dealing with sparsing, it's accuracy is the highest and keeps making least mistakes. Then the AdaGrad, using updating weight, makes less mistakes than the perceptron as the sparsing grows

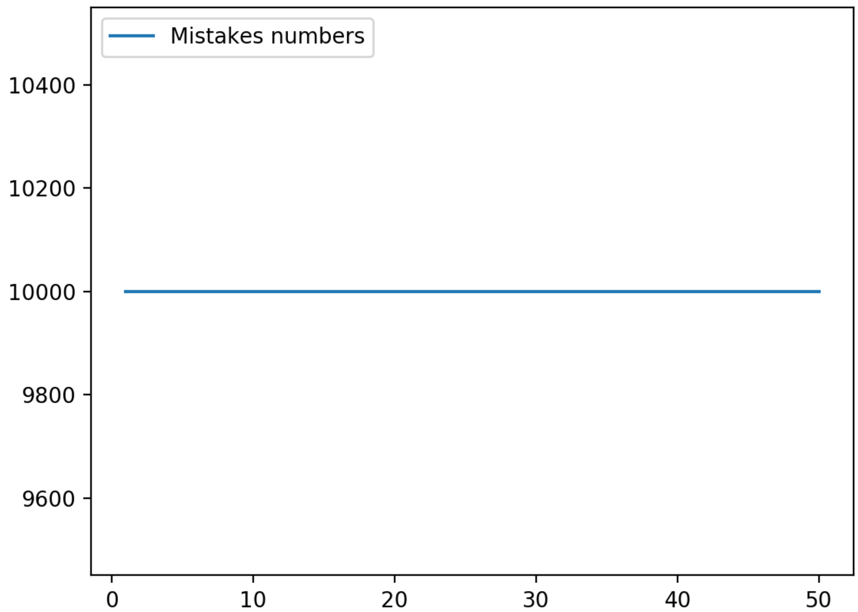
3. **Use online learning algorithms as batch learning algorithms**

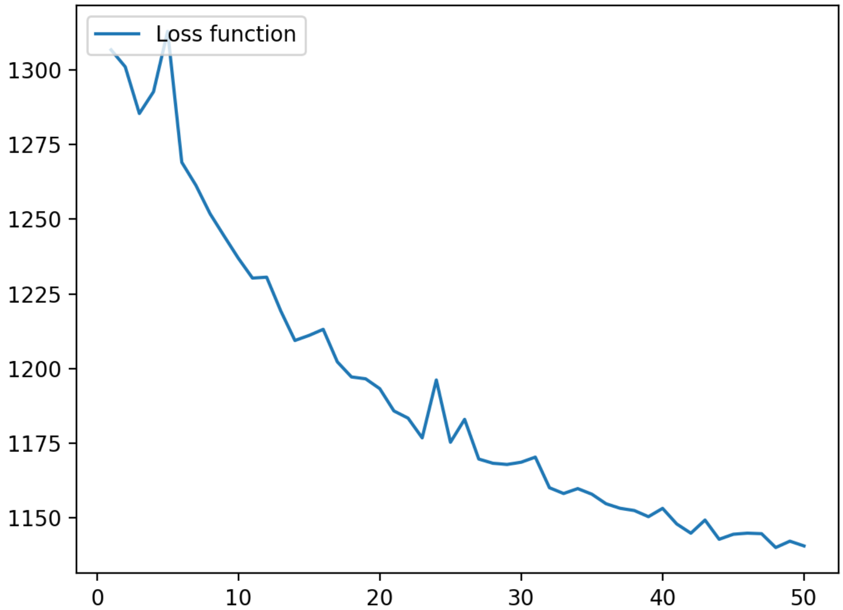
| Algorithm | m=100 | | m=500 | | m=1000 | |
|---|---|---|---|---|---|---|
| | acc. | params. | acc. | params. | acc. | params. |
| Perceptron | 0.8 | 1 | 0.72 | 1 | 0.70 | 1 |
| Perceptron w/margin | 0.81 | 1.5 | 0.80 | 0.25 | 0.73 | 0.25 |
| Winnow | 0.89 | 1.1 | 0.89 | 1.1 | 0.78 | 1.1 |
| Winnow w/margin | 0.90 | 1.1,0.04 | 0.87 | 1.1,0.03 | 0.83 | 1.1,0.03 |
| AdaGrad | 0.87 | 0.25 | 0.87 | 0.25 | 0.84 | 0.25 |

(a) Analysis:
As we can see from the result, the result of winnow and AdaGrad are very closed. They all have much better performance than that of the Perceptron. As m increases, accuracy of each algorithm decreases.

(b) Explanation:
In this part of evaluation, we added noises into the training data. With the dimension of features increases, the influence of noise increases and therefore decreases the accuracy. Since winnow is good at sparsing and AdaGrad updates weights vector in a large range and high frequency, their accuracies are much higher than that of Perceptron.

4. **Bonus**






As the time of training wents up, we can better get rid of the influence of mupltiple features dimension and the affect of noise in the training data.