CS446: Machine Learning **Spring 2017**

Problem Set 3

Bangqi Wang *Handed In: February 28, 2017*

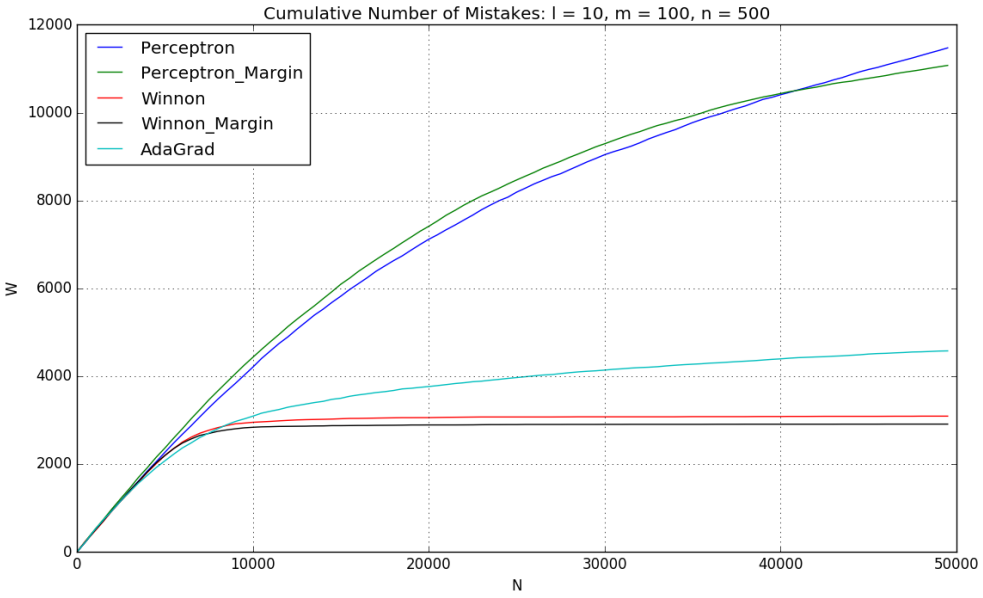
1. Answer to problem 1

   a. Run the tuning procedure and record the optimal parameters.

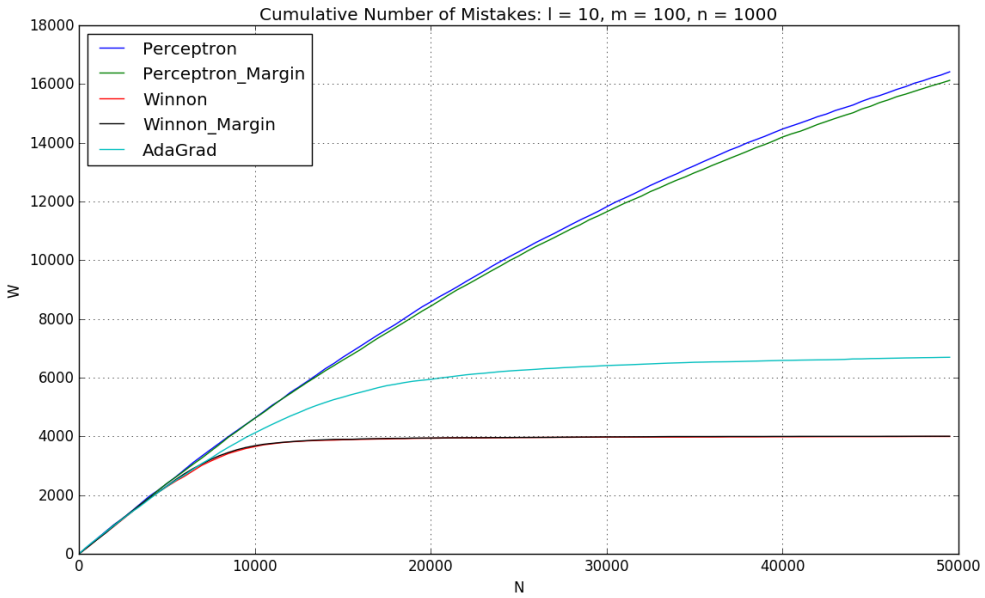
Algorithm	Parameters	Dataset n=500	Dataset n=1000
Perceptron	-	-	-
Perceptron w/margin	lr	0.005	0.005
Winnow	alpha	1.1	1.1
Winnow w/margin	alpha, lr	1.1, 2.0	1.1, 0.3
AdaGrad	lr	0.25	0.25

   .

   b. Plot the cumulative number of mistakes made.



   In this plot, **perceptron** made most mistakes; **perceptron with margin** made a little less after long runing; **winnon with margin** made least mistakes; **winnon** made a little more; **adagrad** made mistakes in between.



In this plot, the overall tendency is almost the same, except that the number is larger. The number of mistakes made by **perceptron** increases from around 12000 to around 16000. The number of mistakes made by **adagrad** increases from around 5000 to around 7000. The number of mistakes made by **winnon** increases from around 3000 to around 4000. The amounts of increase are different.

c. According to the plots above, we can verify that **perceptron** made $O(n)$ mistakes, and **winnon** made $O(mlogn)$ mistakes. Since n is much larger than m, **winnon** made less mistakes. **adagrad** is a modified version of **perceptron**, which encourages the model to exploit infrequent features, made less mistakes when l is far less than m.

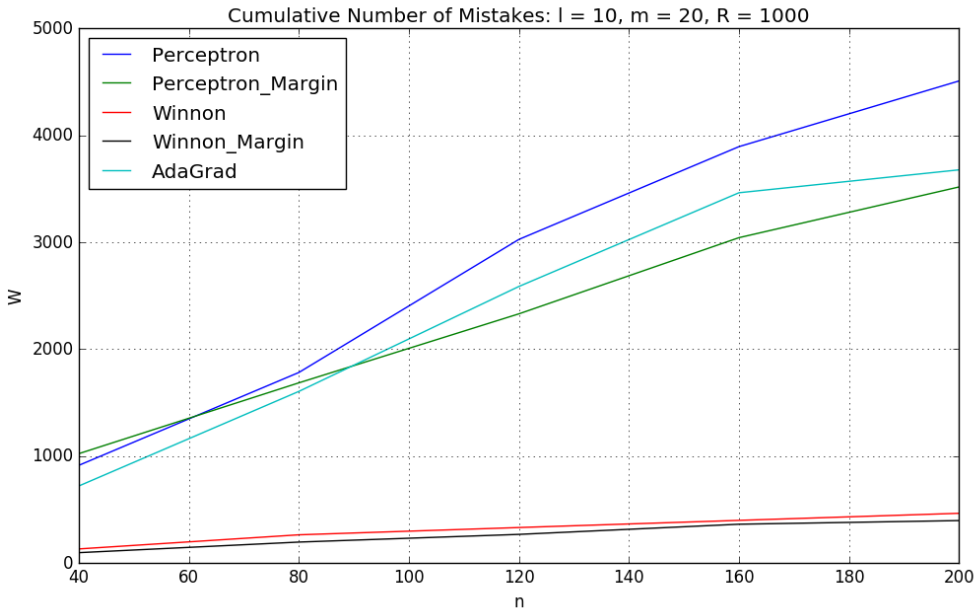
2. Answer to problem 2

a. Record the chosen parameters.

Algorithm	Params.	n=40	n=80	n=120	n=160	n=200
Perceptron	-	-	-	-	-	-
Perceptron w/margin	lr	0.25	0.03	0.03	0.03	0.03
Winnow	α	1.1	1.1	1.1	1.1	1.1
Winnow w/margin	α, γ	1.1, 2.0	1.1, 2.0	1.1, 2.0	1.1, 2.0	1.1, 2.0
AdaGrad	lr	1.5	1.5	1.5	1.5	1.5

.

b. Plot the cumulative number of mistakes made.



c. In the plot above, **perceptron** has worst performance; **perceptron with margin** and **adagrad** have a little better performance; **winnow** and **winnon with margin** have much better performance, and **winnon with margin** is little better than **winnon**. The data with large n are sparser and therefore need more rounds to achieve high accuracy.

3. Answer to problem 3

a. Tuned parameters and resulting accuracy.

Algorithm	m=100		m=500		m=1000	
	acc.	params.	acc.	params.	acc.	params.
Perceptron	82.03%	-	62.08%	-	75.57%	-
Perceptron w/margin	99.83%	lr: 0.001	97.96%	lr: 0.005	82.41%	lr: 0.25
Winnow	97.72%	α: 1.1	89.09%	α: 1.01	71.43%	α: 1.1
Winnow w/margin	96.03%	α: 1.1, γ: 0.04	92.41%	α: 1.1, γ: 0.001	80.89%	α: 1.005, γ: 2.0
AdaGrad	99.96%	lr: 0.03	99.31%	lr: 1.5	83.59%	lr: 0.25

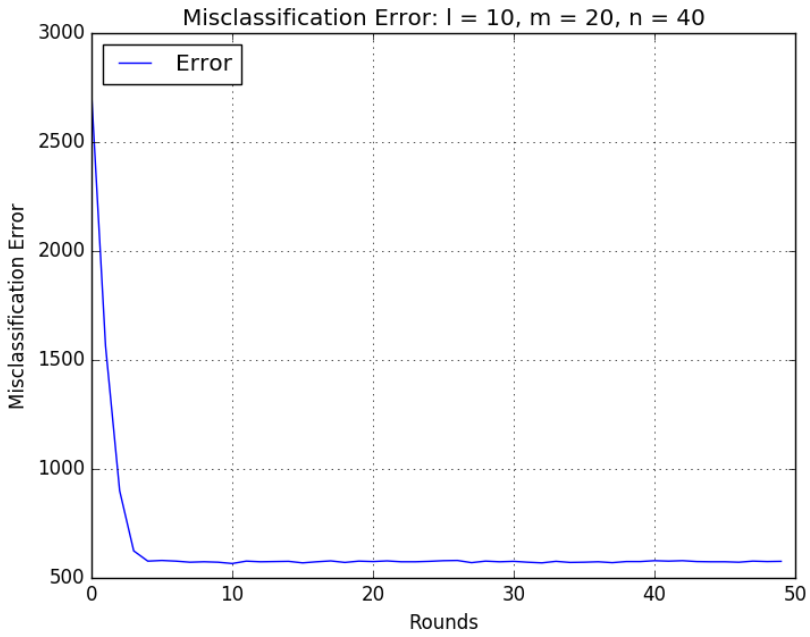
.

b. From table above, we know that algorithms get lower accuracy in dataset with larger m. **adagrad** has highest accuracy; **perceptron with margin** has second highest accuracy; **winnon with margin** has third highest accuracy; **winnon** is forth, and **perceptron** is the last one. The algorithms with margin can achieve higher accuracy than original algorithms. According to plots in question 1 and 2,

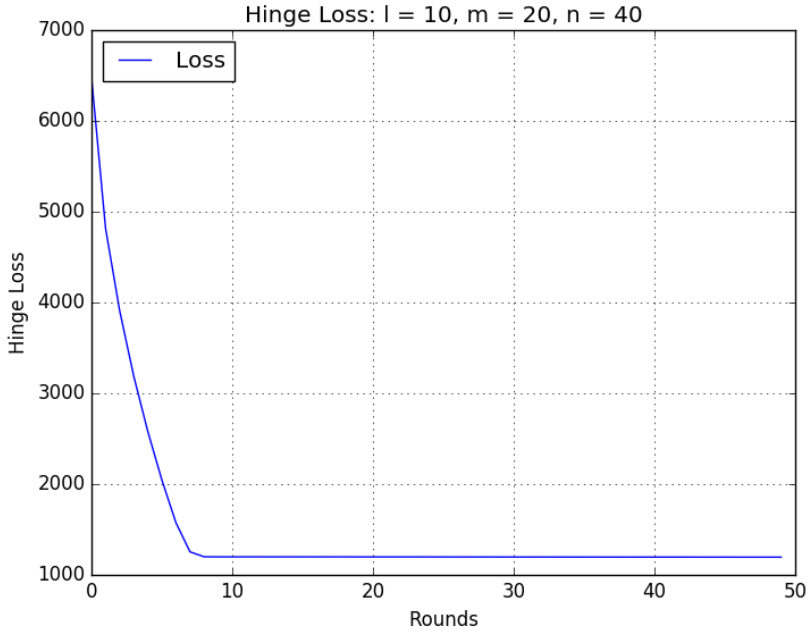
winnon made far less mistakes but **adagrad** has higher accuracy.

4. Answer to bonus problem

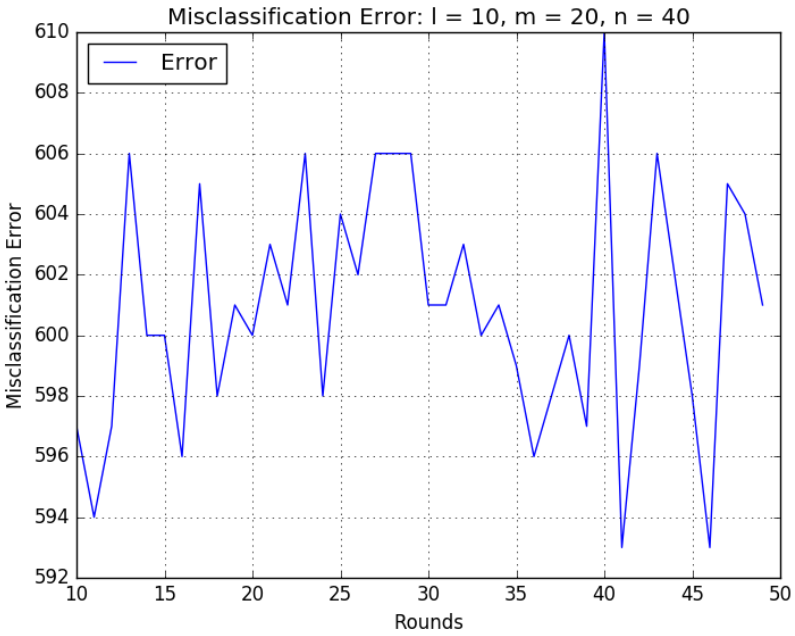
   a. Plot the misclassification error.



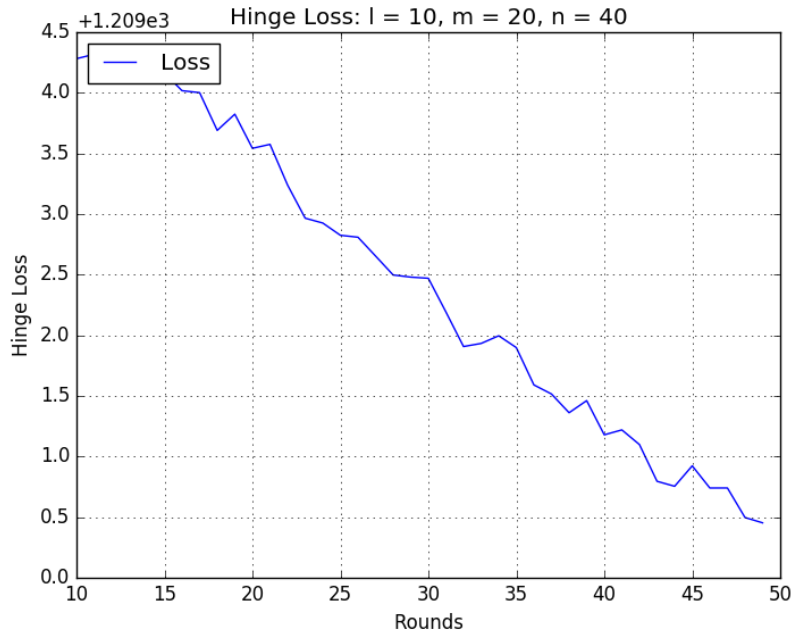
   b. Plot the hinge loss.



c. Plot the misclassification error after first 10 rounds.



d. Plot the hinge loss after first 10 rounds.



e. From plots above, we know that **error** drops much faster than **loss** does. However, after first several rounds, **error** starts to vibrate around a certain range, but **loss** keeps decreasing.