CS446: Machine Learning	**Fall 2016**
Problem Set 3 Solution	
Handed Out: October 11, 2016	*Due: N/A*

Experiment 1: Number of examples vs Number of mistakes

Algorithm	Parameters	Dataset n=500	Dataset n=1000
Perceptron	NA	NA	NA
Perceptron w/margin	η	0.005	0.005
Winnow	α	1.1	1.1
Winnow w/margin	α, γ	1.1,2	1.1,2
AdaGrad	η	0.25	0.25

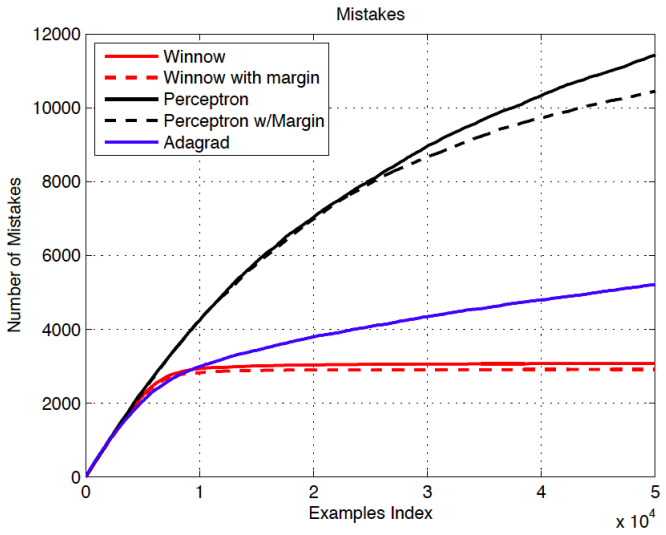

Figure 1: n=500 Mistake Plot

The plots of the cumulative mistakes are given Figure and Figure , with n = 500; 1000. You can see that for both Perceptrons the number of mistakes goes up linearly with n, while for the Winnows the growth seems logarithmic as a function of n. AdaGrad seems closer to Winnow, but does not scale as well.

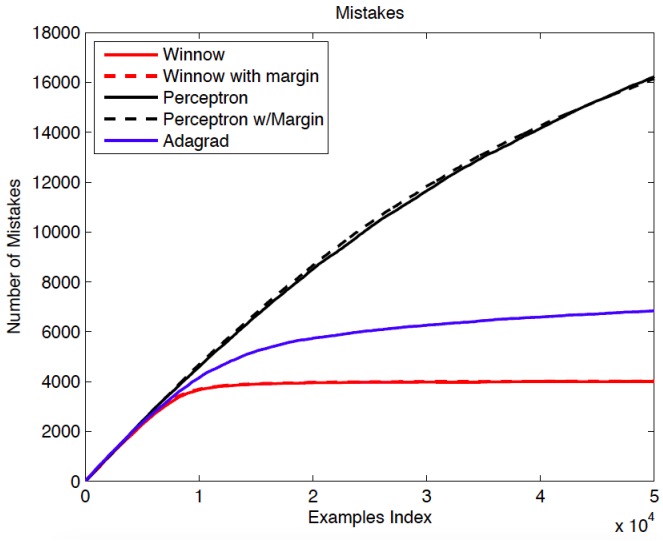

Figure 2: n=1000 Mistake Plot

Experiment 2: Learning Curves of Online Learning Algorithms

Algorithm	Parameters	n=40	n=80	n=120	n=160	n=200
Perceptron	NA	NA	NA	NA	NA	NA
Perceptron w/margin	η	1.5	0.03	0.25	0.25	0.25
Winnow	α	1.1	1.1	1.1	1.1	1.1
Winnow w/margin	α, γ	1.1	1.1	1.1	1.1	1.1
AdaGrad	η	1.5	0.25	1.5	1.5	1.5

Comments: You are expected to find that Winnow and Winnow with margin always make the fewest errors before convergence. The Winnow with margin should be a little bit better than Winnow, but the two are pretty close. It's OK if in your plot the difference is not discernible. Perceptron with margin should be better than the original perceptron, but they can also be close. AdaGrad is more sensitive to the data generation; while it can be better or worse than the two perceptrons, it is always worse than the Winnows. Also note that the number of mistakes should always go up as n increases since our problem is harder as n is larger. However, you might see some variability due to the randomness of the data generation, especially for AdaGrad. You will get the credit if the difference between Winnow, Perceptron and AdaGrad is reasonable.

Experiment 3: Use online learning algorithms as batch learning algorithms

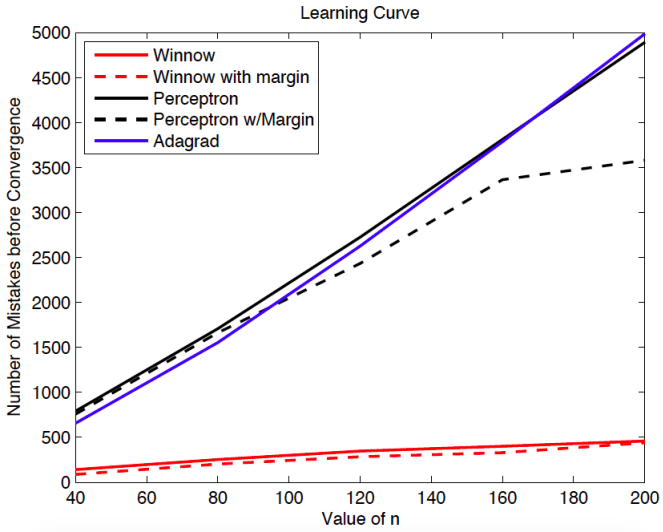

Figure 3: Batch Learning

Algorithm	Par. m=100	Acc. m=100	Par. m=500	Acc. m=500	Par. m=1000	Acc. m=1
Perceptron	NA	≈ 90	NA	≈ 80	NA	≈ 75
Perceptron w.m.	0.03	≈ 95	0.25	≈ 85	1.5	≈ 75
Winnow	1.1	≈ 95	1.1	≈ 75	1.1	≈ 75
Winnow w.m.	α=1.1 γ=0.3	≈ 90	1.1,0.006	≈ 85	1.1,0.004	≈ 70
AdaGrad	0.25	≈ 95	1.5	≈ 70	1.5	≈ 75

Comments: A few observations: (1) The best parameters are not so sensitive to changes in m, but the accuracy is sensitive to changes in m, and generally goes down with the increase in m. (2) The fact that we train on noisy data has signicant impact on the performance of all algorithms. (3) Notice that in this case, as m grows, the target function becomes dense; almost all the features become important; this explains the fact that, unlike the previous experiments, here Perceptron is doing better than Winnow.

Experiment 4. Bonus Question

The plot for the configuration $l = 10, m = 20, n = 40$ is given in Figure 4. Your plot may be different from the one given in the solution but you should see similar trends that are discussed below.

Comments: Even though the Hinge loss over the dataset reduces with each training round, the misclassification error may not show a corresponding decrease. This is because the loss function (Hinge Loss) is a convex-upper bound on the 0-1 loss.

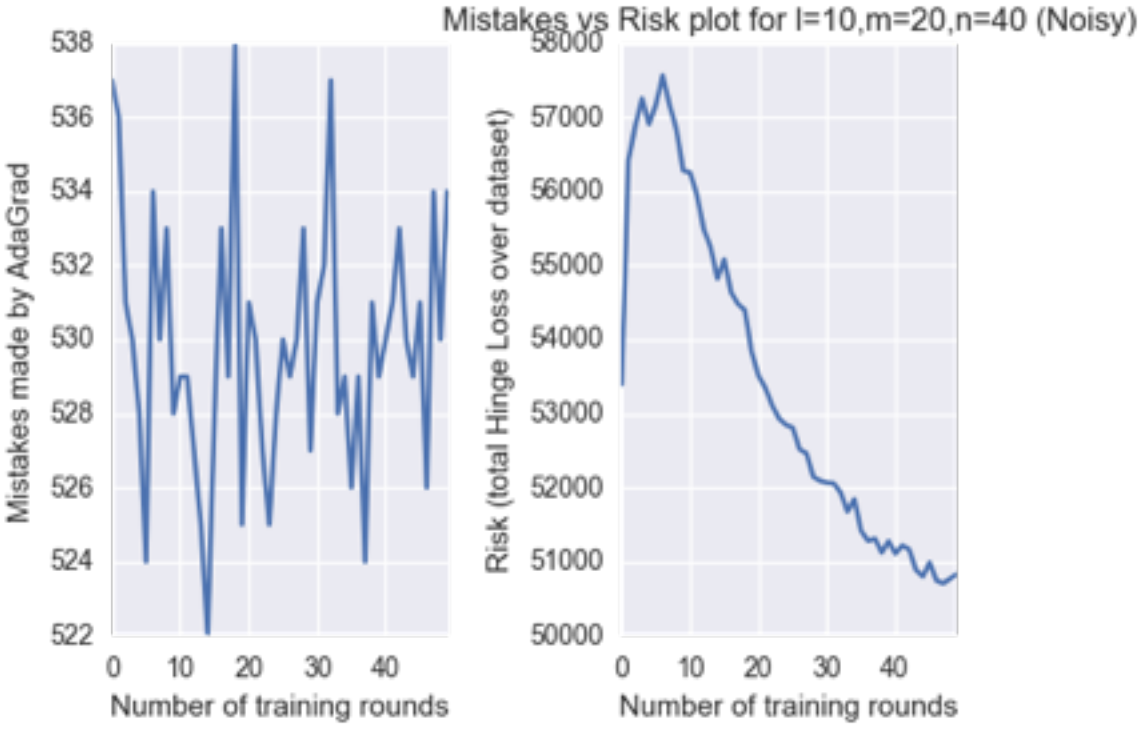

Figure 4: Bonus Problem: l = 10, m = 20, n = 40

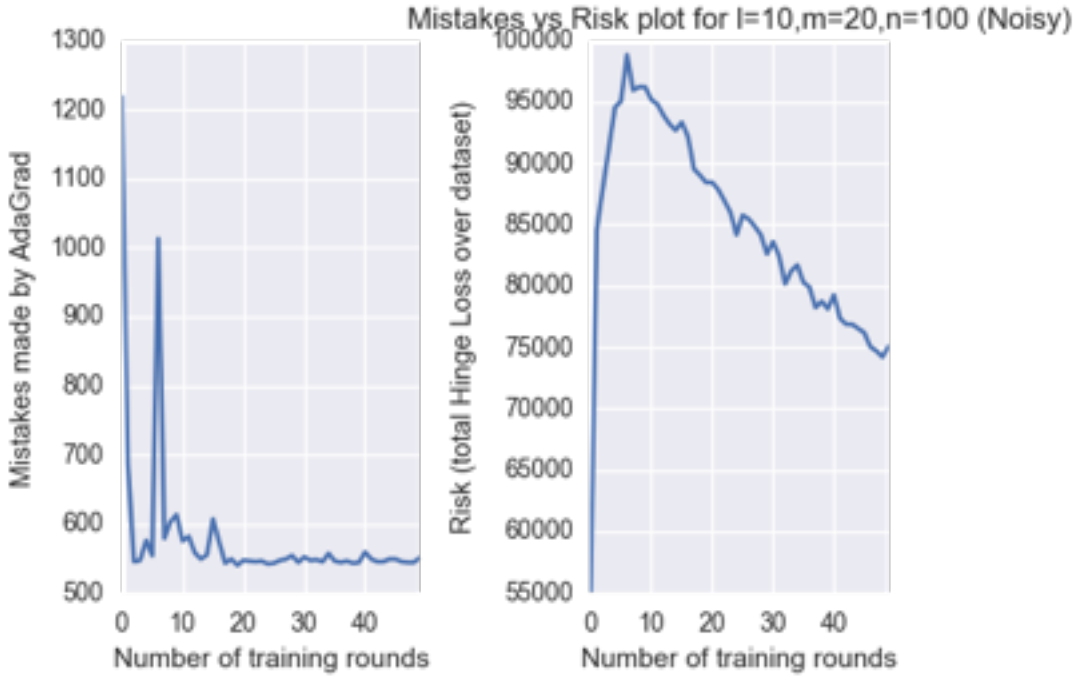

Figure 5: Bonus Problem: l = 10, m = 20, n = 100