CSC242_Project4_Learning Report

Part 1. Group member:

Name: Tianbo Liu
NetID: tliu36
Email: tliu36@u.rochester.edu

Name: Hanrui Liu
NetID: hliu60
Email: hliu60@u.rochester.edu

Name: Xubin Lou
NetID: xlou5
Email: xlou5@u.rochester.edu

Part 2. How to build and compile our project:

1. We have implemented the linear classifier in this project; and for the extra credit (NN), we have completed the code in "learn.nn.core".
2. The java project for this assignment is in the "CSC242_Project4_Learning" folder.
3. Add all the files you going to use to test our algorithm into "learn.lc.examples" package. The original three files (earthquake-clean.data.txt, earthquake-noisy.data.txt, and house-votes-84.data.num.txt) are already built in our project, but if you want to test other files, you should add that file into "learn.lc.examples" package and follow the command line instruction below.
4. Open the terminal, enter the "src" directory, and use the command line listed below to run our program.

Command line instruction (examples will be listed in part 3):

For Perceptron Classifier:

1. Enter the directory /src
2. javac learn/lc/examples/PerceptronClassifierTest.java
3. java learn.lc.examples.PerceptronClassifierTest [testing file name] [nsteps] [alpha, 0 for decay with time] [output file name]

For Logistic Classifier:

1. Enter the directory /src
2. javac learn/lc/examples/LogisticClassifierTest.java
3. java learn.lc.examples.LogisticClassifierTest [testing file name] [nsteps] [alpha, 0 for decay with time] [output file name]

5. The "output.csv" will be stored in the src directory.

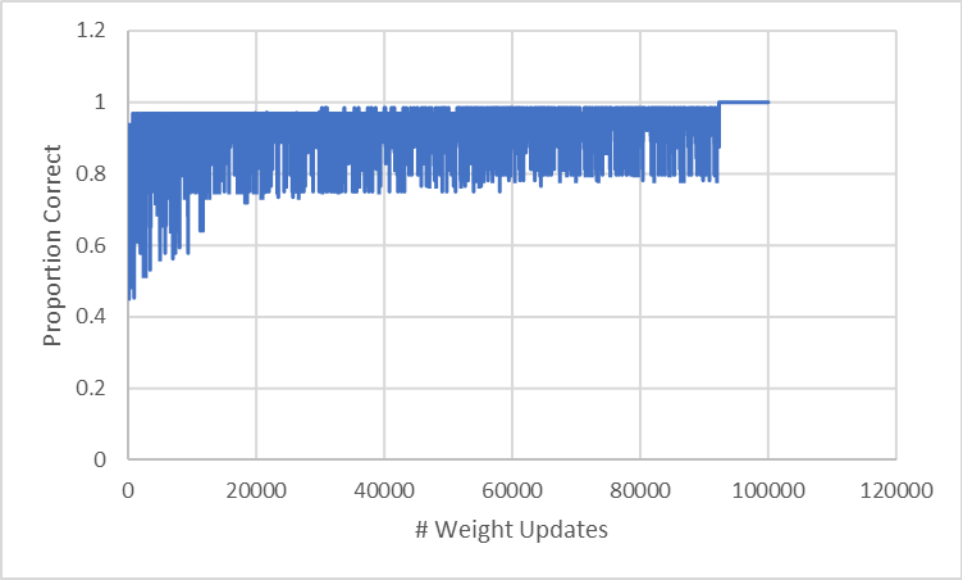
Part 3 Linear Classifiers Training report：

This section includes 12 graphs: first 6 using perceptron learning rule and rest using logistic regression. Our program will automatically generate the "output.csv", so we just use the Excel to draw these graphs.

Though we cannot promise that our model will achieve 100% correct, we have tried different parameter combinations many times, and the parameters in each command are most likely to yield the most optimal result, which is very close to 1.

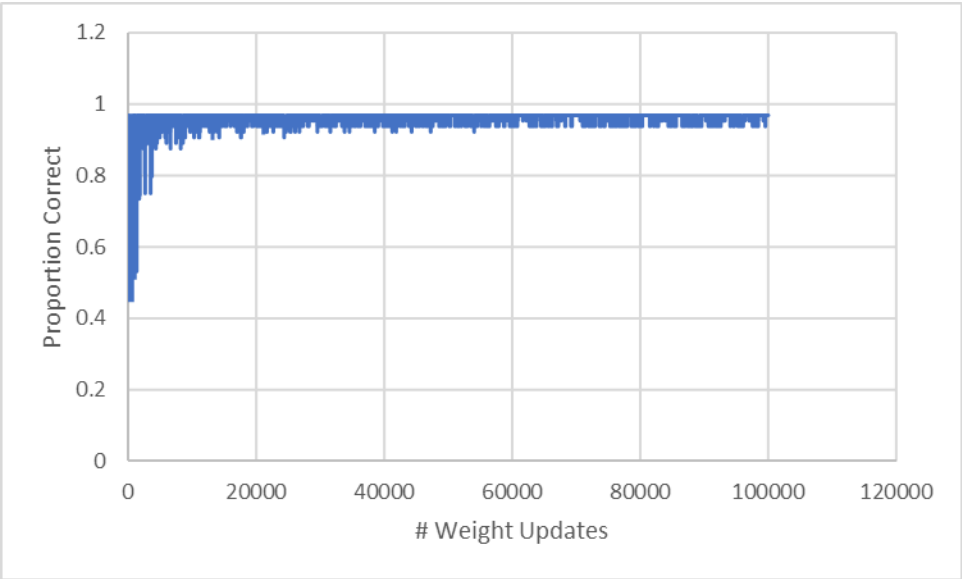
3.1 Using **perceptron learning rule** to test **earthquake-clean** dataset with **fixed alpha** -> Converge to 1.

Command line example: java learn.lc.examples.PerceptronClassifierTest earthquake-clean.data.txt 100000 0.95 P_earthquake_clean_alpha_report



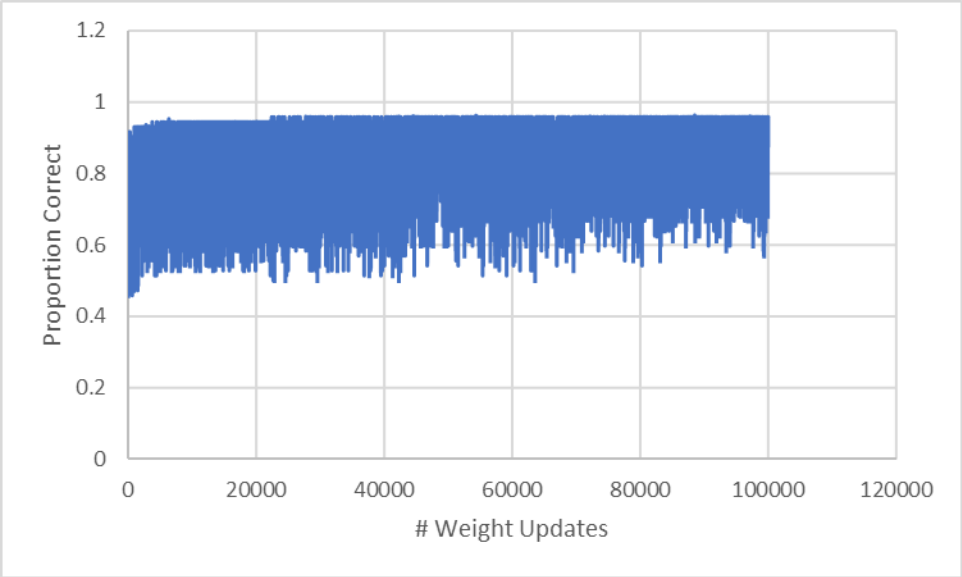
3.2 Using **perceptron learning rule** to test **earthquake-clean** dataset with **decaying** -> The convergence is not pretty, but it can achieve 0.96875 in the end.

Command line example: java learn.lc.examples.PerceptronClassifierTest earthquake-clean.data.txt 100000 0 P_earthquake_clean_decay_report



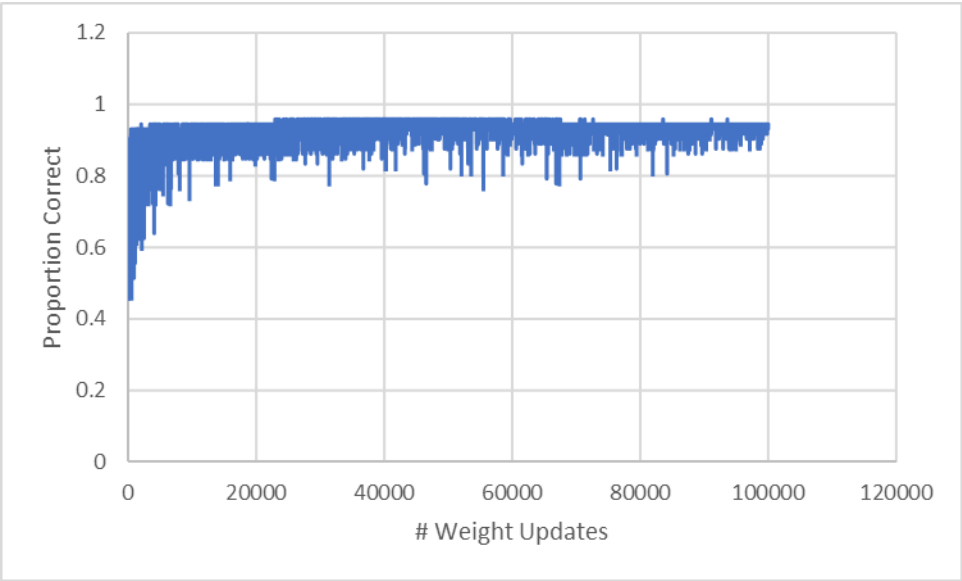
3.3 Using **perceptron learning rule** to test **earthquake-noisy** dataset with **fixed alpha** -> Fluctuate a lot, but it can achieve 0.95833 in the end.

Command line example: java learn.lc.examples.PerceptronClassifierTest earthquake-noisy.data.txt 100000 0.3 P_earthquake_noisy_alpha_report



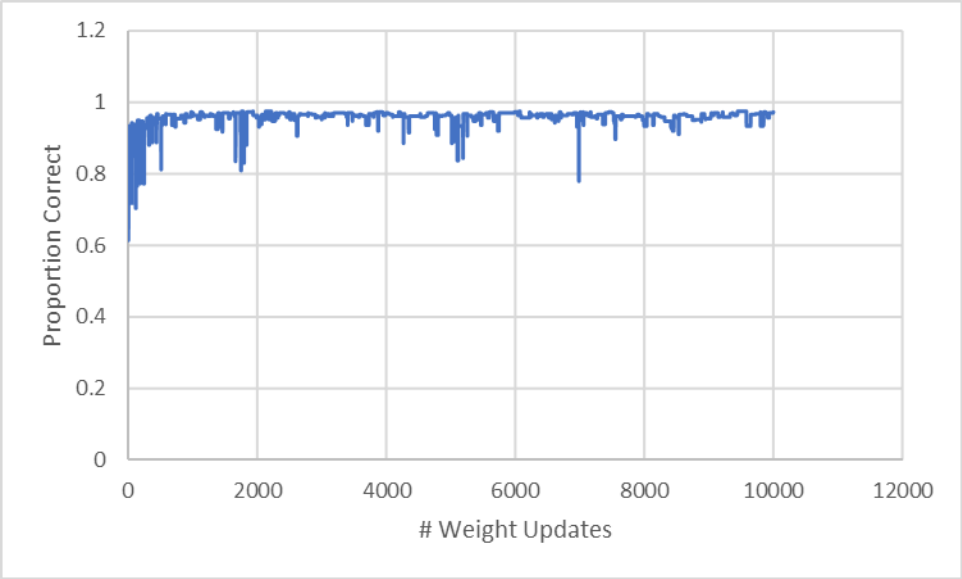
3.4 Using **perceptron learning rule** to test **earthquake-noisy** dataset with **decaying** -> The convergence is not pretty, but it can achieve 0.94444 in the end.

Command line example: java learn.lc.examples.PerceptronClassifierTest earthquake-noisy.data.txt 100000 0 P_earthquake_noisy_decay_report



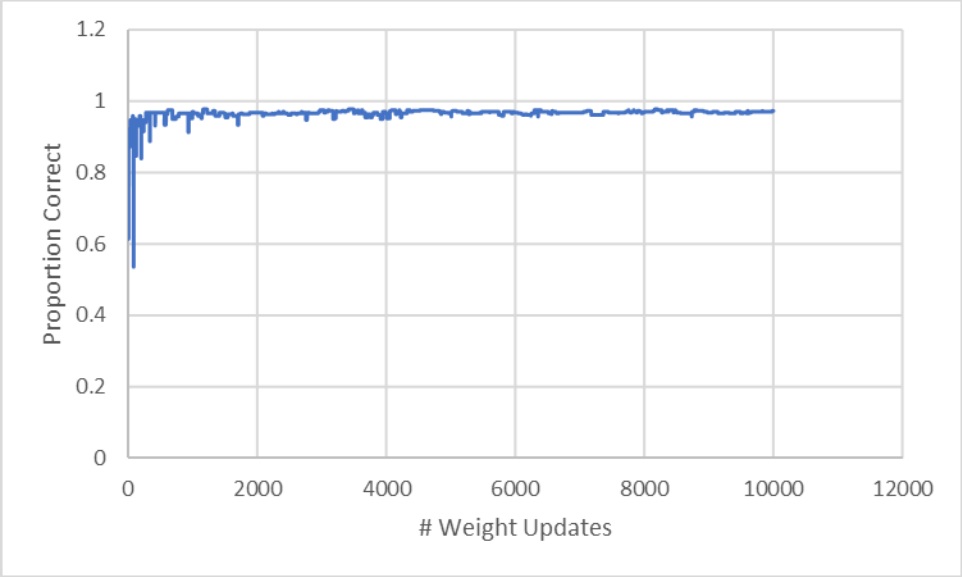
3.5 Using **perceptron learning rule** to test **house-votes-84** dataset with **fixed alpha** -> Fluctuate a little bit, but converge to 0.972 in the end.

Command line example: java learn.lc.examples.PerceptronClassifierTest house-votes-84.data.num.txt 10000 0.3 P_HouseVotes_alpha_report



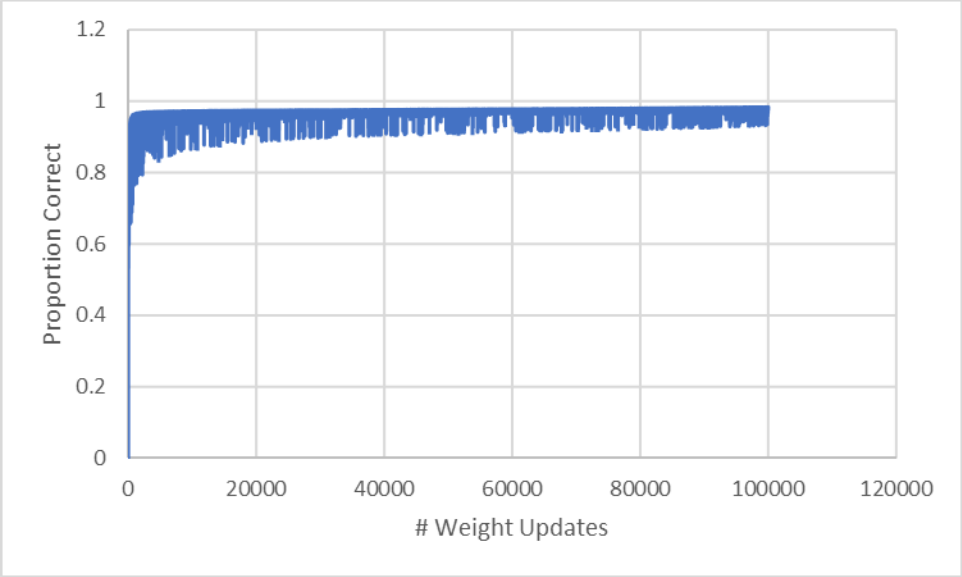
3.6 Using **perceptron learning rule** to test **house-votes-84** dataset with **decaying** -> Converge to 0.975.

Command line example: java learn.lc.examples.PerceptronClassifierTest house-votes-84.data.num.txt 10000 0 P_HouseVotes_decay_report



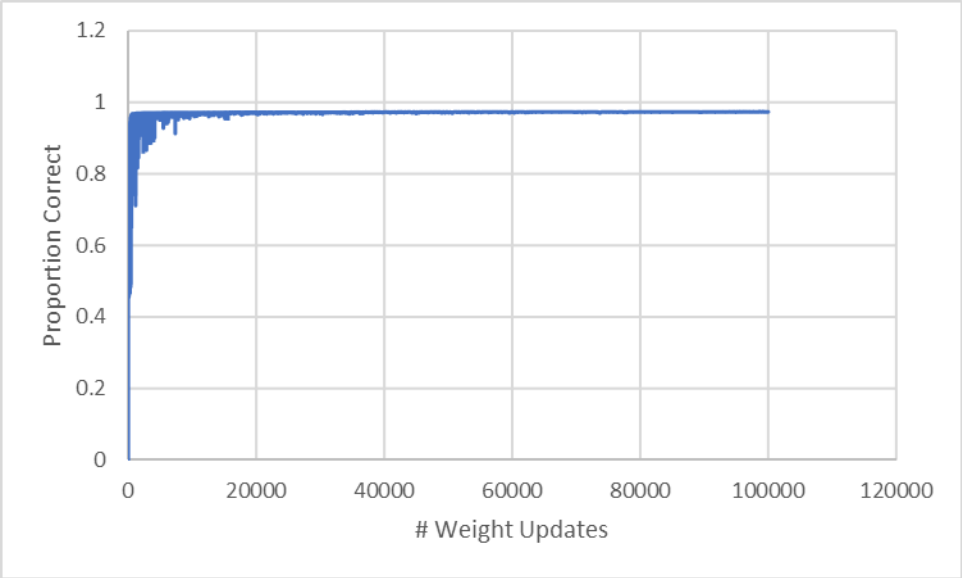
3.7 Using **logistic regression** to test **earthquake-clean** dataset with **fixed alpha** -> The convergence is not pretty, but it can achieve 0.98316 in the end.

Command line example: java learn.lc.examples.LogisticClassifierTest earthquake-clean.data.txt 100000 0.3 L_earthquake_clean_alpha_report



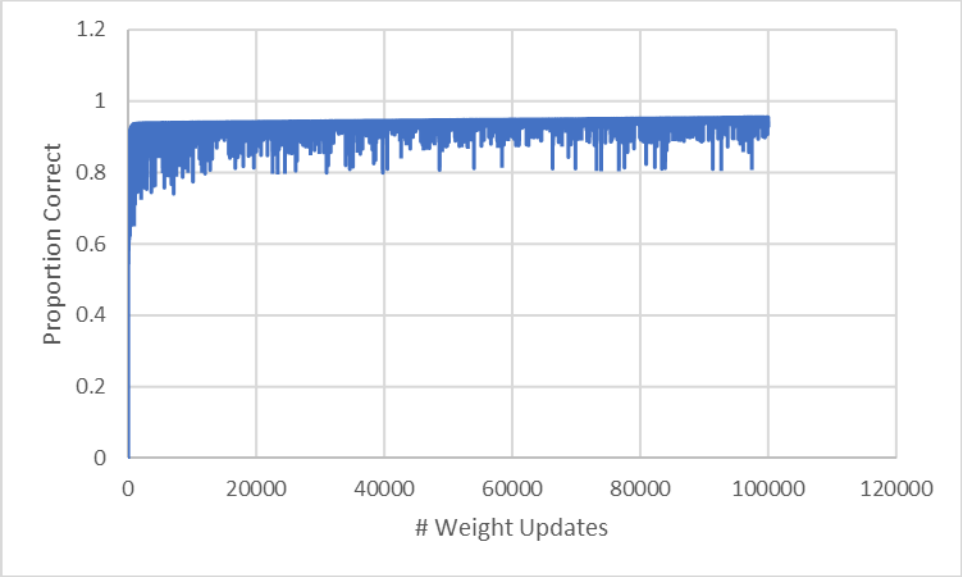
3.8 Using **logistic regression** to test **earthquake-clean** dataset with **decaying** -> Converge to 0.973.

Command line example: java learn.lc.examples.LogisticClassifierTest earthquake-clean.data.txt 100000 0 L_earthquake_clean_decay_report



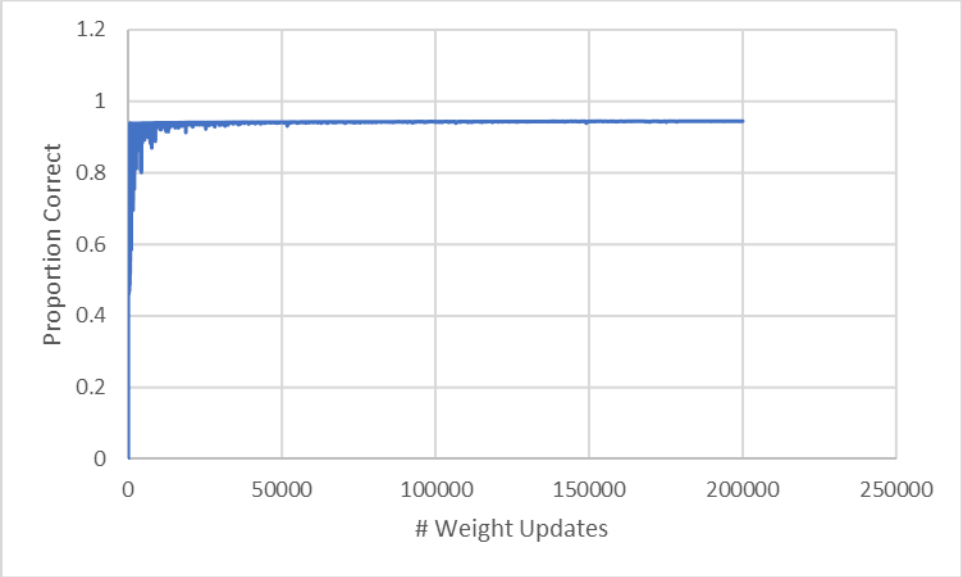
3.9 Using **logistic regression** to test **earthquake-noisy** dataset with **fixed alpha** -> The convergence fluctuates, but it can achieve 0.955 in the end.

Command line example: java learn.lc.examples.LogisticClassifierTest earthquake-noisy.data.txt 100000 0.3 L_earthquake_noisy_alpha_report



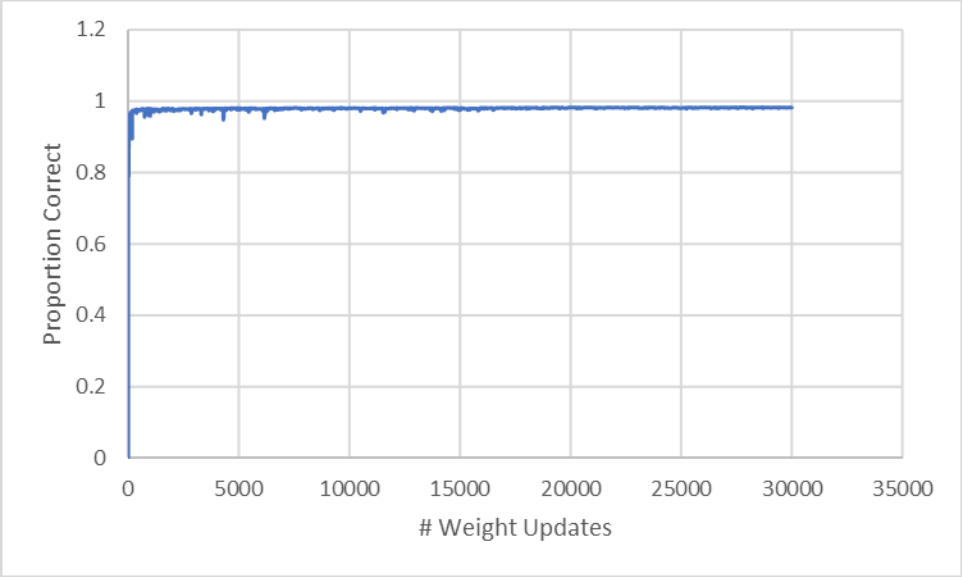
3.10 Using **logistic regression** to test **earthquake-noisy** dataset with **decaying** -> Converge to 0.94444.

Command line example: java learn.lc.examples.LogisticClassifierTest earthquake-noisy.data.txt 200000 0 L_earthquake_noisy_decay_report



3.11 Using **logistic regression** to test **house-votes-84** dataset with **fixed alpha** -> Converge to 0.982.

Command line example: java learn.lc.examples.LogisticClassifierTest house-votes-84.data.num.txt 30000 0.95 L_HouseVotes_alpha_report



3.12 Using **logistic regression** to test **house-votes-84** dataset with **decaying** -> Converge to 0.98.

Command line example: java learn.lc.examples.LogisticClassifierTest house-votes-84.data.num.txt 30000 0 L_HouseVotes_decay_report