CSC 242 Project 4 Report

Group Members:

Shaotong Sun (ssun25@u.rochester.edu)

Junfei Liu (jliu137@u.rochester.edu)

Dingcheng Wang (dwang63@u.rochester.edu)

Compile Instructions:

To compile, enter the “src” folder and type the following command in the command line:

javac learn/lc/core/*.java

Run Instructions:

After finished compile steps, run the main method in Main class by typing the following command in the command line:

java learn/lc/core/Main <filename> <nsteps> <alpha> <instruction>

The first argument, filename, refers to the dataset used for training and testing. To ensure that the file can be found, please enter the absolute path.

The second argument, nsteps, refers to the number of weight updates. Generally speaking, the larger nsteps yields better results.

The third argument, alpha, refers to the alpha value used in training. A numerical value (0.0, 1.0] is accepted to set a constant alpha value. If value 0 is entered, the program will employ decaying learning rate schedule.

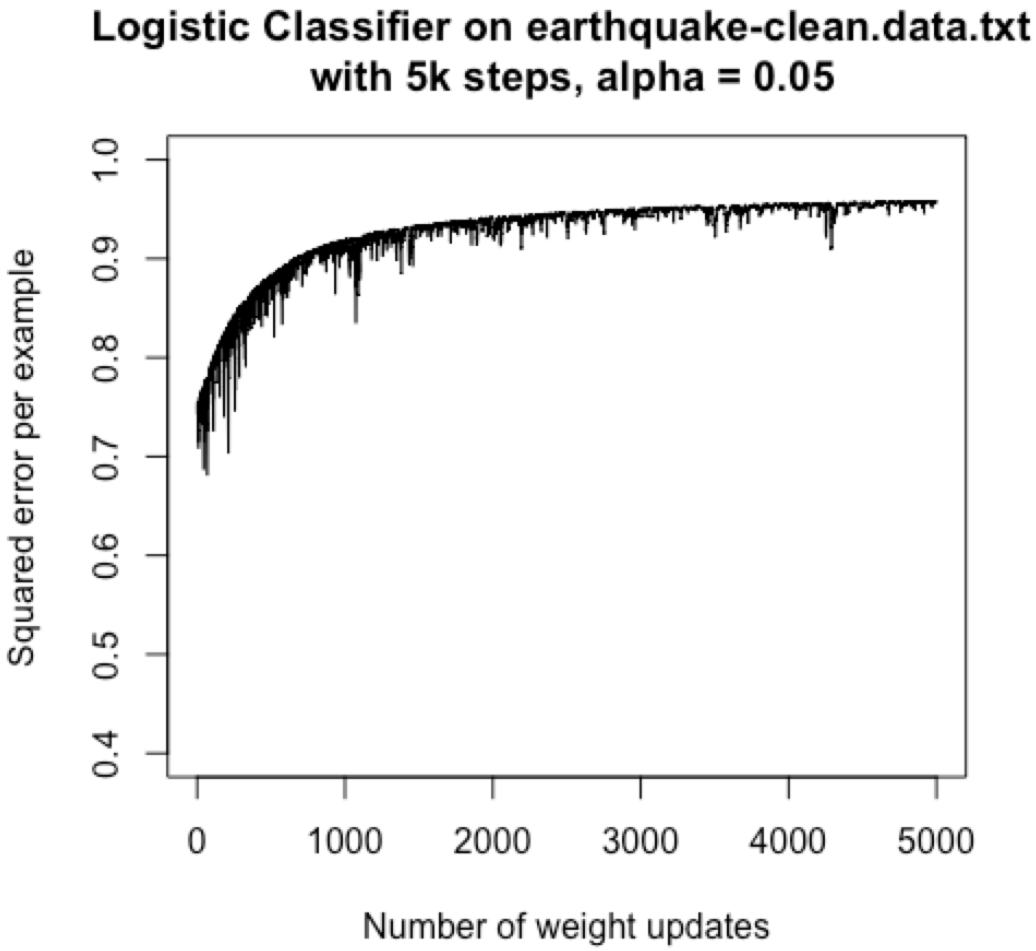
The fourth argument, instruction, refers to the type of linear classifier. String values of “Perceptron” or “Logistic” are accepted, which will set the type of classifier

correspondingly.

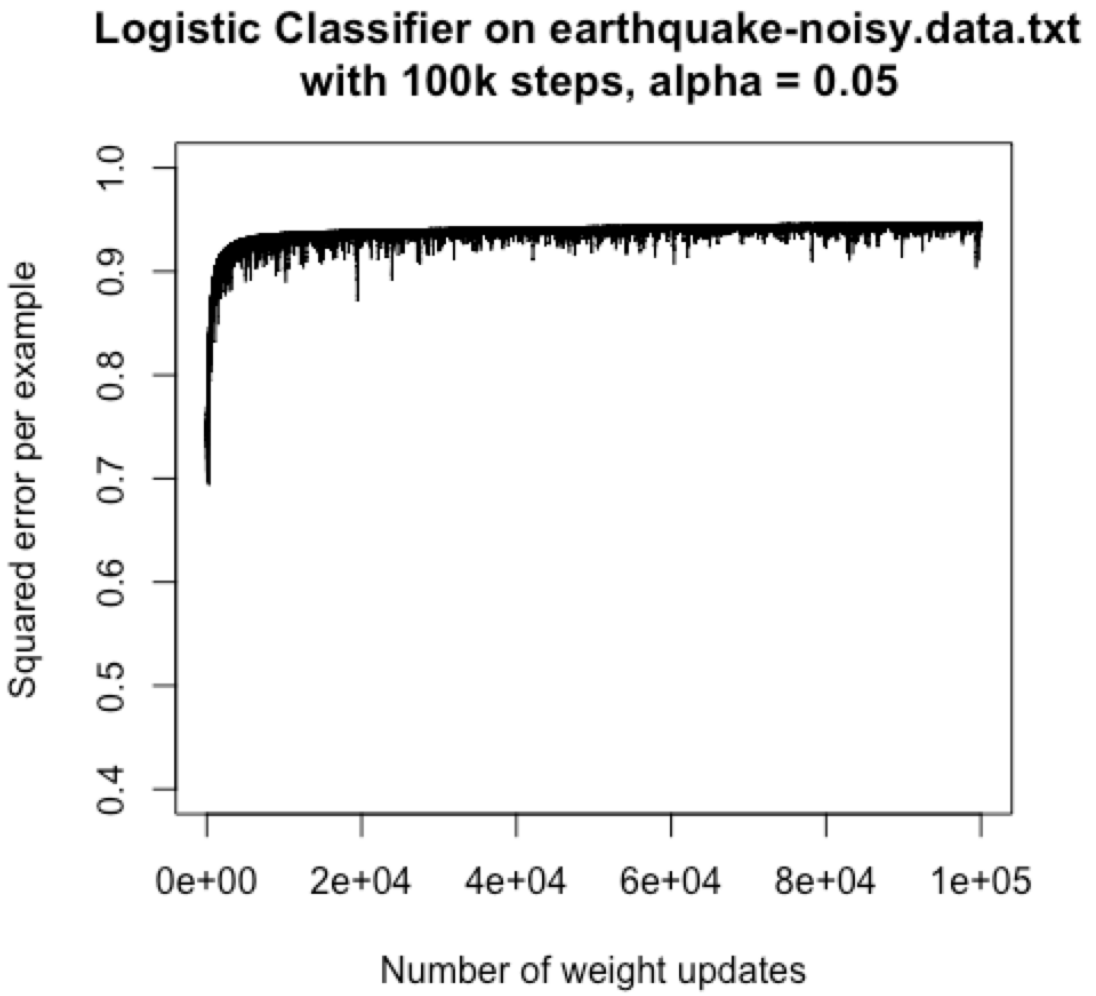
Results for Linear Classifier

Earthquake examples:

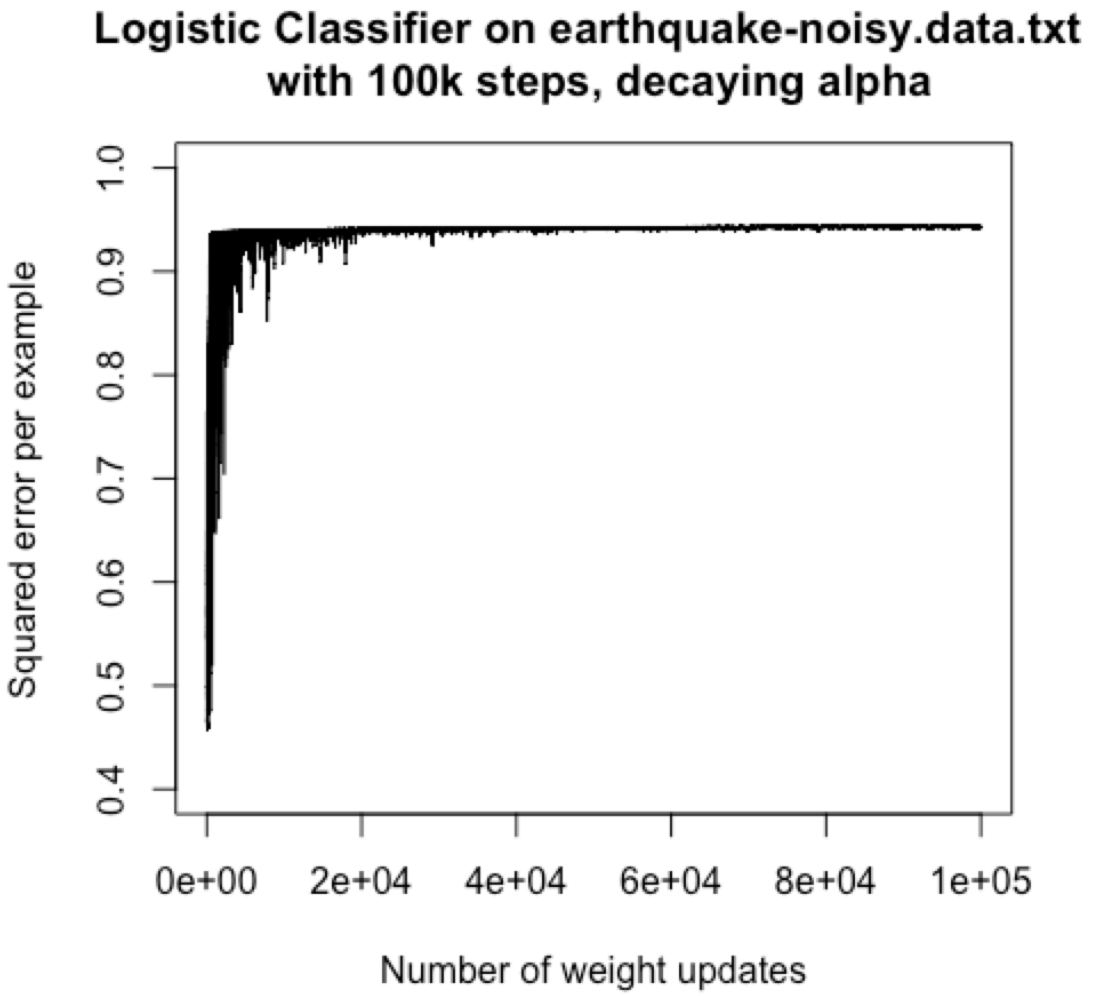
Logistic Classifiers: Figure a, b, and c.



a. In this scenario, the accuracy starts from about 0.68 and ends at about 0.96. Comparing to the learning process of perceptron classifier, the accuracy variation of logistic classifier is much smaller and shows a pattern that accuracy increases as the number of weight steps increases (which is similar to a logarithmic curve). the ending accuracy is approximately 0.95 after each run.

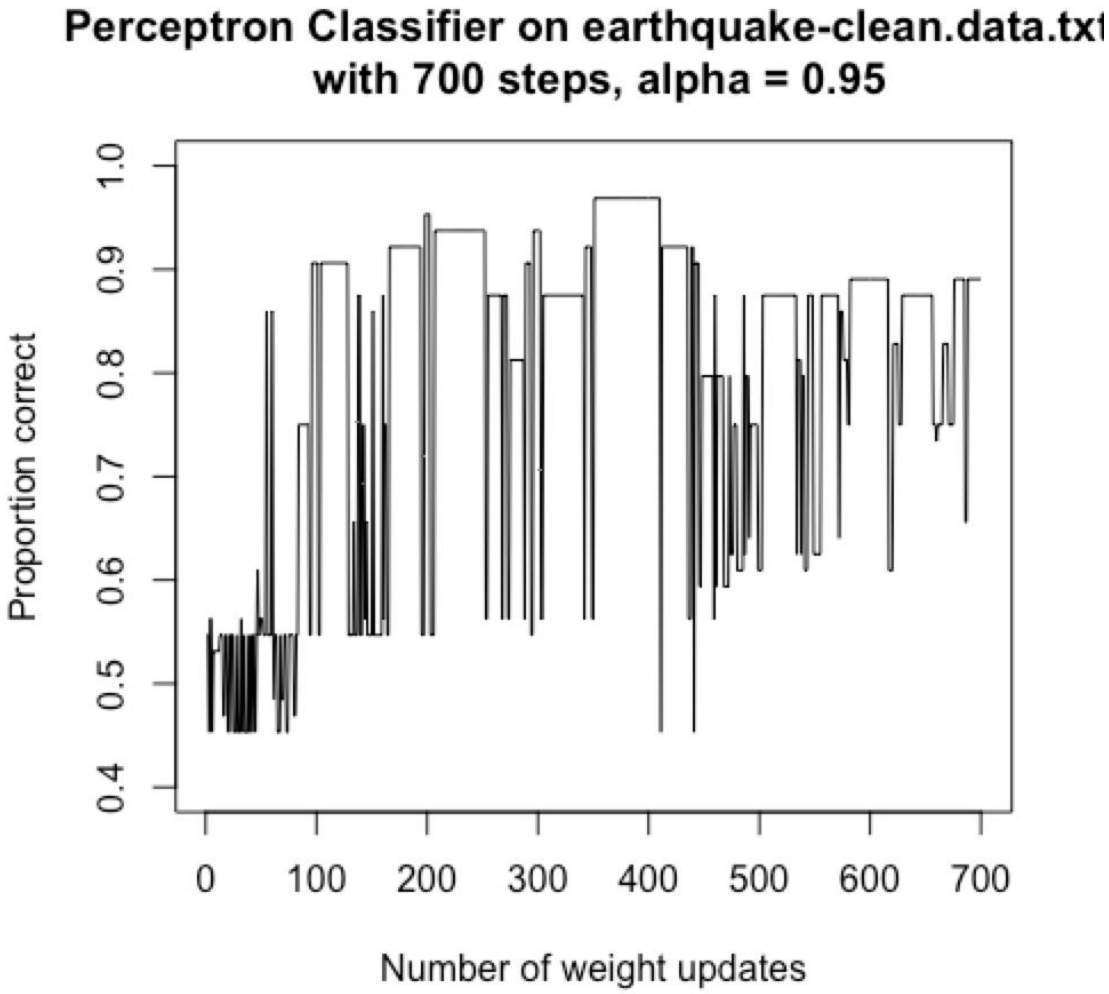


b. In this scenario, the accuracy starts from about 0.7 and ends at about 0.95. Comparing to the learning process on "earthquake-clean" (a), the accuracy variation of "earthquake noisy" is larger when the number of steps is small but reduces when it increases. Similar to scenario a, it is also noticeable to see a positive correlation with the shape of a logarithmic curve. With 100K steps of updates in total, the accuracy of this logistic classifier is stable around the value of 0.93 after each run.

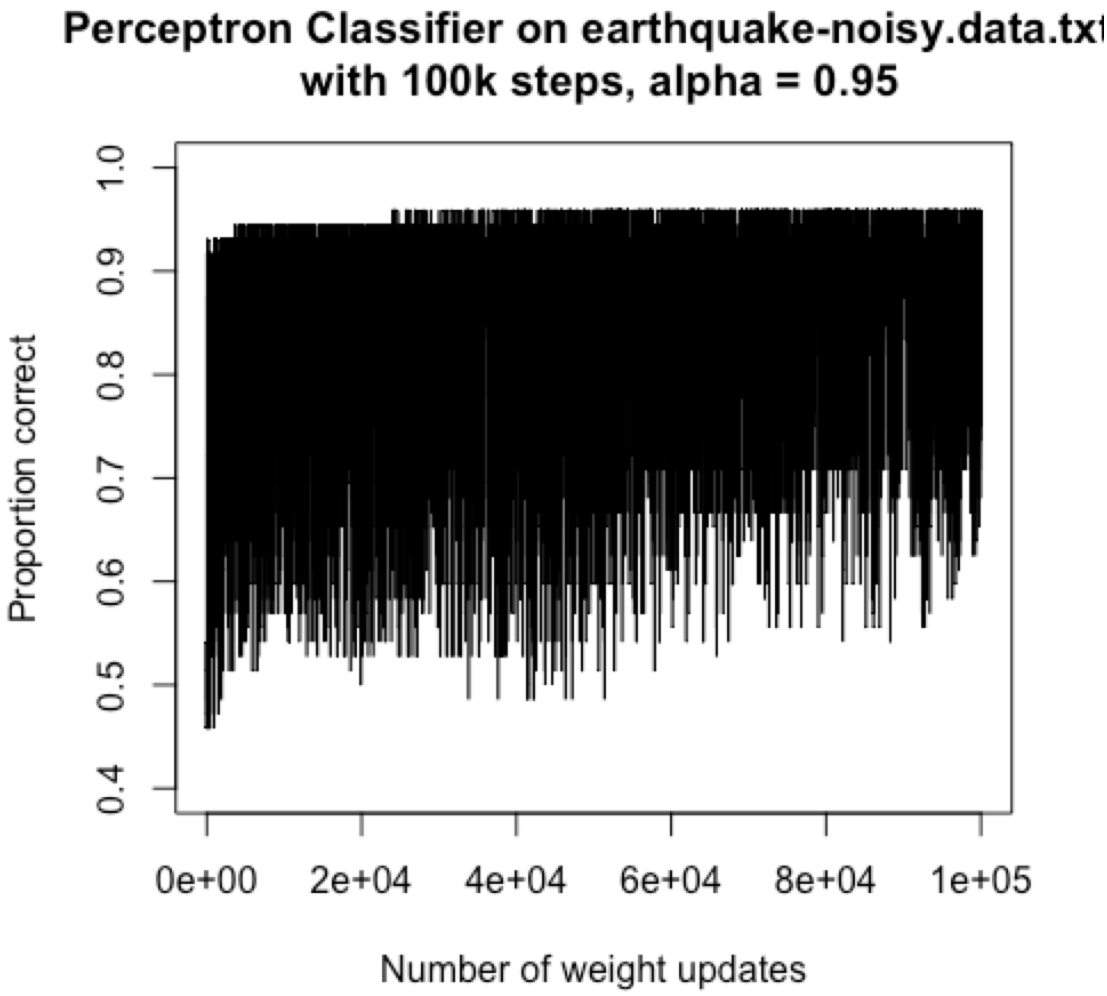


c. In this scenario, the accuracy starts from about 0.5 and ends at about 0.94. Comparing to the learning process of the logistic classifier with constant alpha, the accuracy variation of logistic classifier with decaying alpha is larger when the number of steps is small but reduces quickly when the number of steps rises. The accuracy increases quickly and stays at 0.94 around 20K steps. In general, we can see that with the same weight updates (100K), the accuracy of logistic classifier with decaying alpha has less variation than the one using constant alpha (b) after each run.

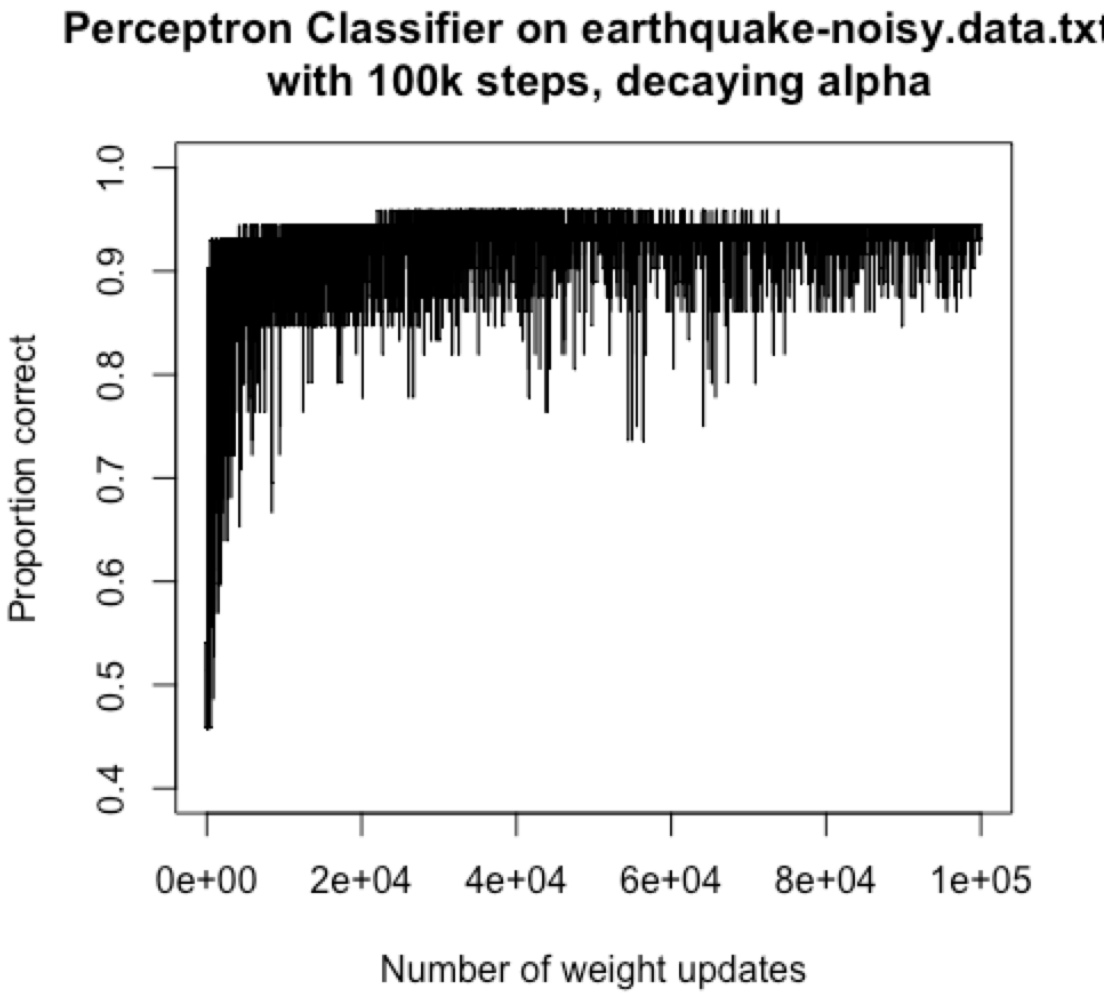
Perceptron Classifiers: figure d, e, and f.



d. In this scenario, the proportional correctness starts from about 0.45 and ends at about 0.87. During the weight updating process, there are large variations: for example, proportional correctness drops from 0.97 (step 412) to 0.45 (step 414) and reaches 0.92(step 417). Therefore, it is hard to find a pattern between number of weight updates and proportional correctness and produce a stable solution due to the limit of small step size. With 700 weight updates in total, the proportional correctness after the entire learning process has varied a lot after each run.



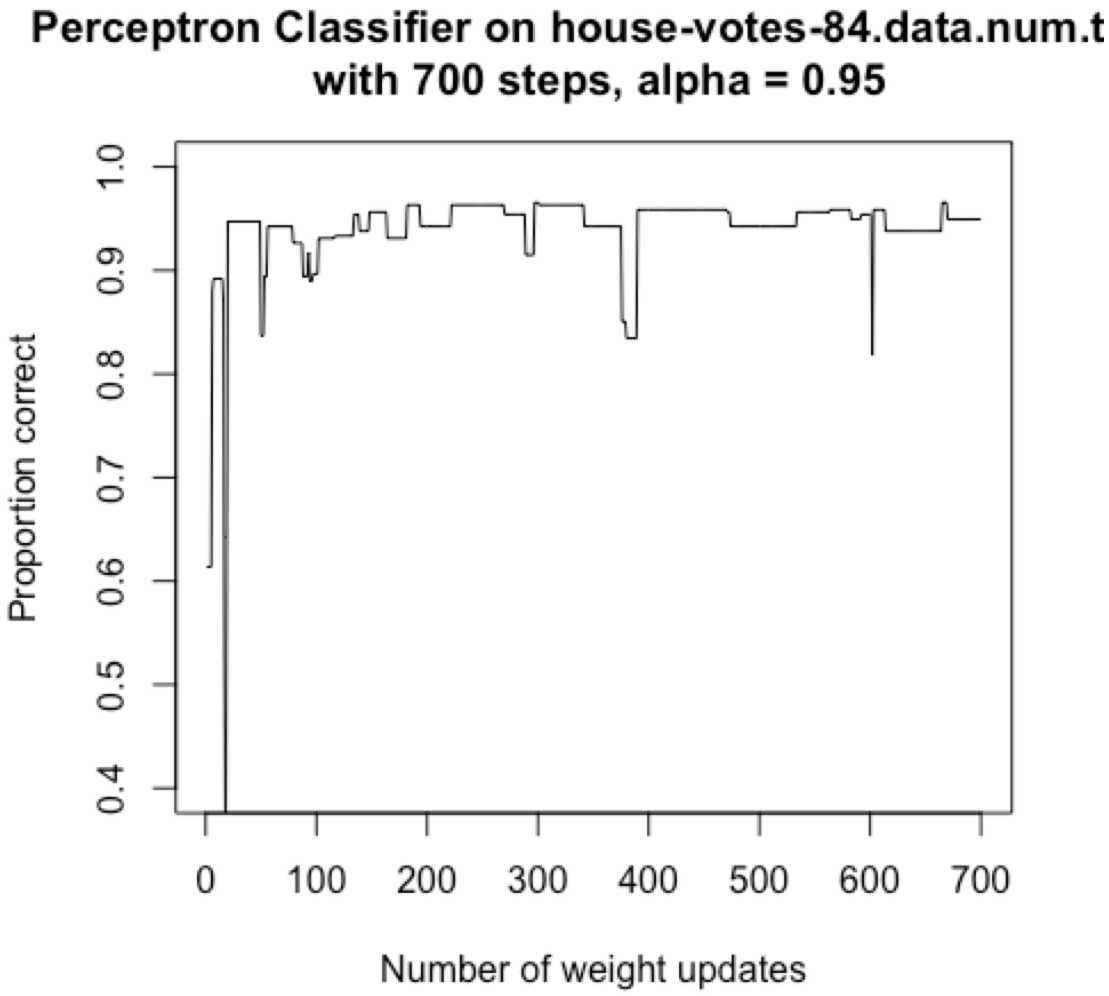
e. In this scenario, the proportional correctness starts from about 0.45 and ends at about 0.94. Similar to the last figure, the variation of proportional correctness is still large (basically in the 0.45-0.95 range) and hard to recognize a pattern in the general picture. After applying 100K weight updates in total, the final proportional correctness is stabler than the last scenario with 700 weight updates. the final proportional correctness stays in the 0.92~0.95 range for each run.



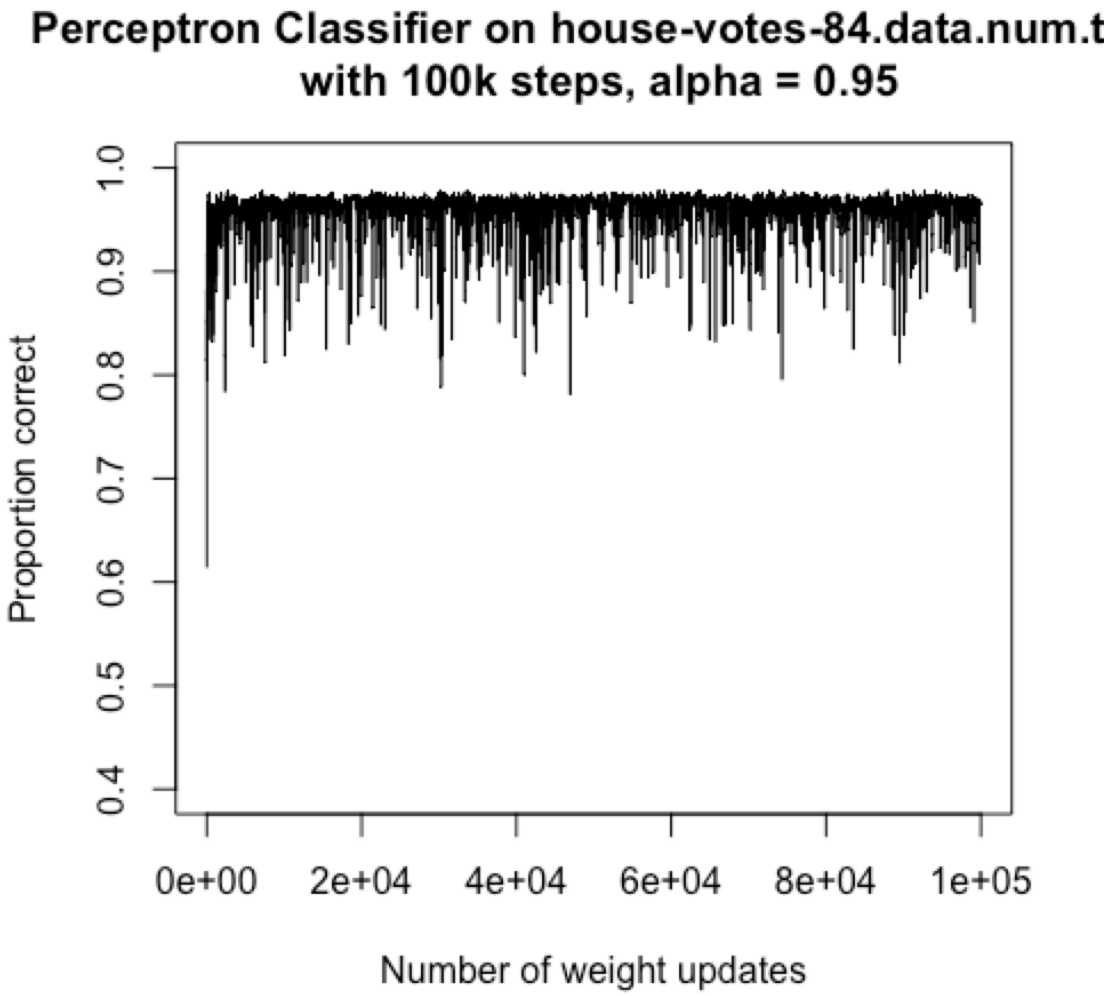
f. In this graph, the proportional correctness starts from about 0.46 and rapidly goes up and stay around 0.95 until the end of training. The variation of the proportional correctness is much smaller than the previous graph after 10K steps and a positive correlation between the number of weight updates and proportion correctness is recognized. the ending proportional correctness stays in the 0.9~0.95 range.

House-votes examples:

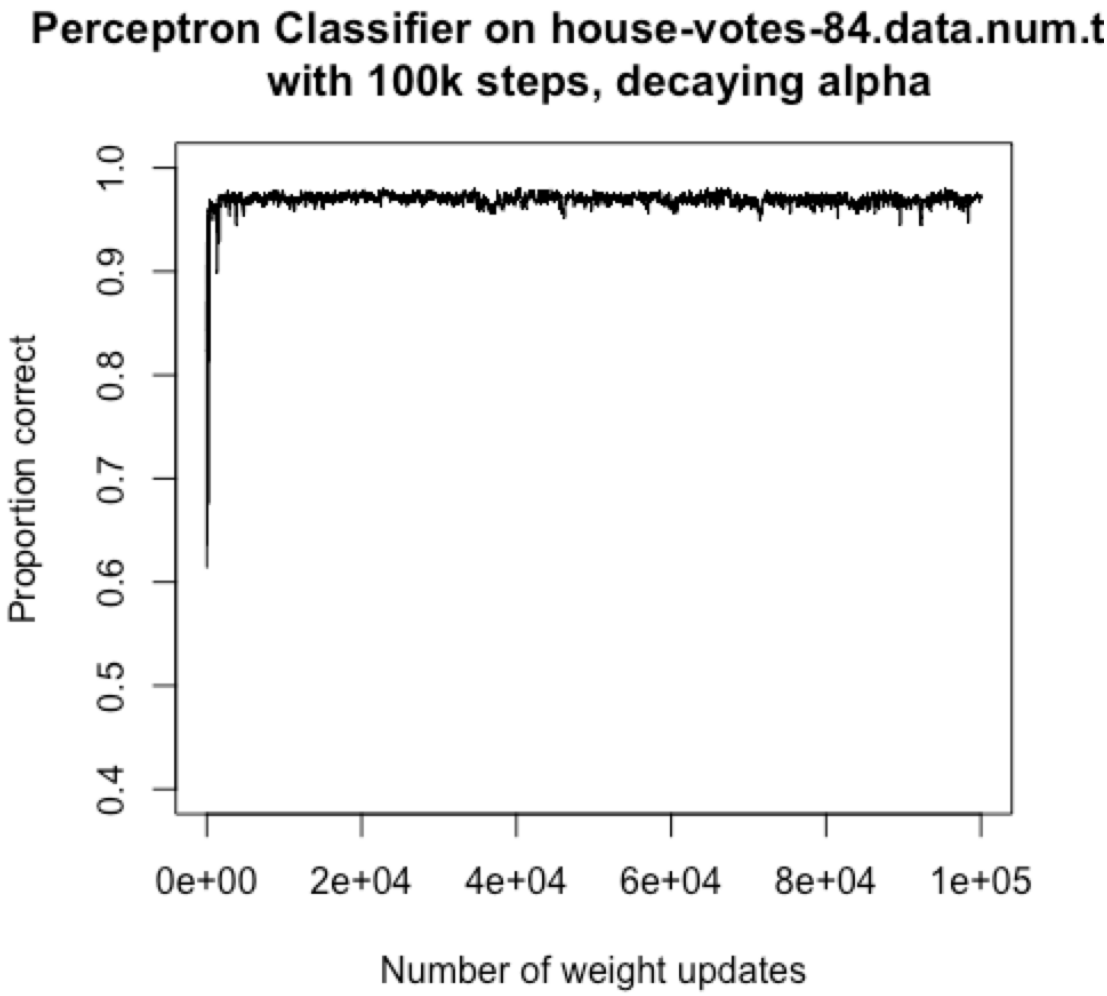
Perceptron Classifiers: figure j, k, and l.



j. In this graph, the proportional correctness starts from 0.62 around and ends at around 0.94. Base on the graph above, there is not obvious relationship between number of weight updates and proportional correctness. The variation of the proportional correctness is large. The final proportional correctness is around 0.95 and less stable.

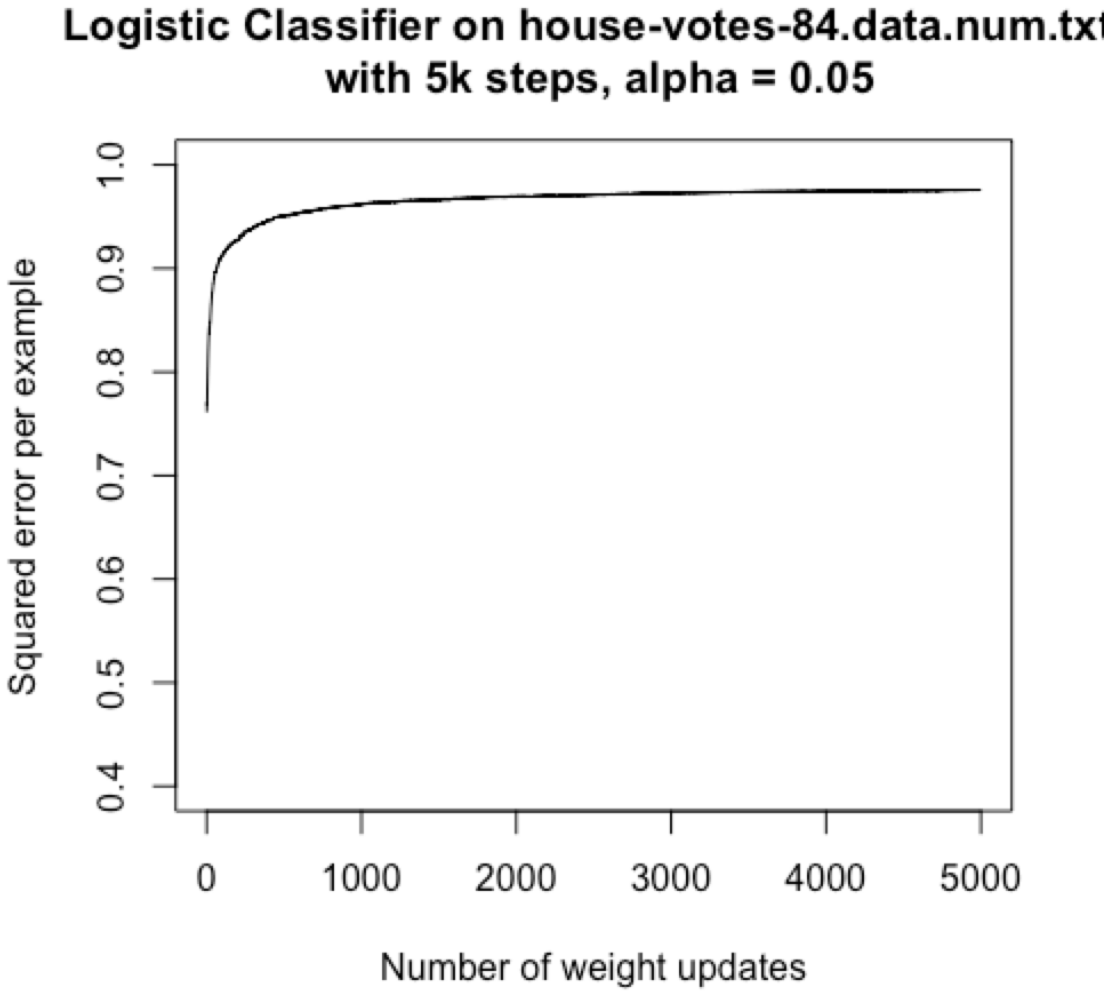


k. In this graph, the proportional correctness starts from 0.61 around and ends at around 0.97. In general, The variation of the proportional correctness is considerable and no pattern can be recognized between the two properties. The proportional correctness after the entire learning process is not so stable: the final proportional correctness stays about 0.95 and usually higher.

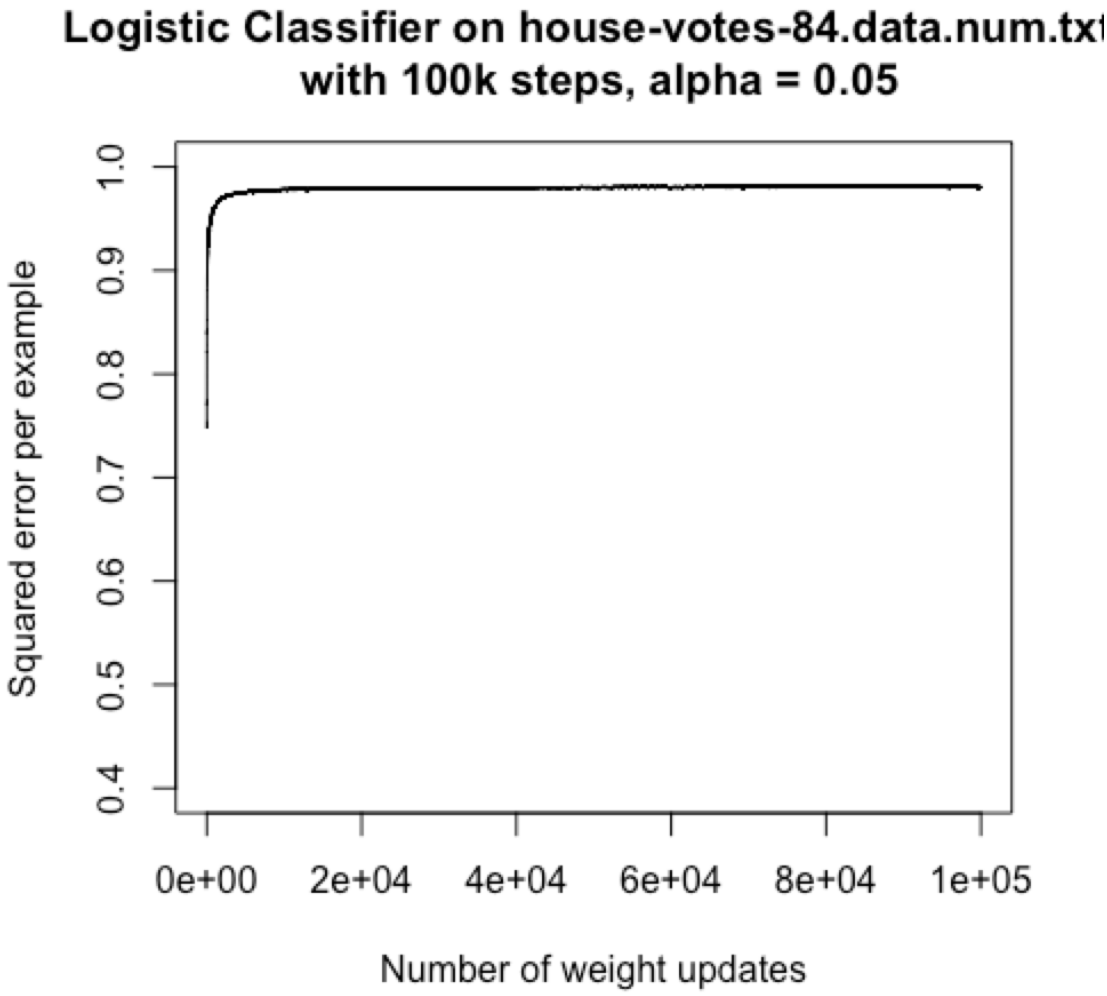


l. In this graph, the proportional correctness starts from 0.62 around and ends at around 0.96. Base on the graph above, there is not obvious relationship between number of weight updates and proportional correctness. However, the variation of the proportional correctness is significantly smaller than figure j. For the final proportional correctness is around 0.95.

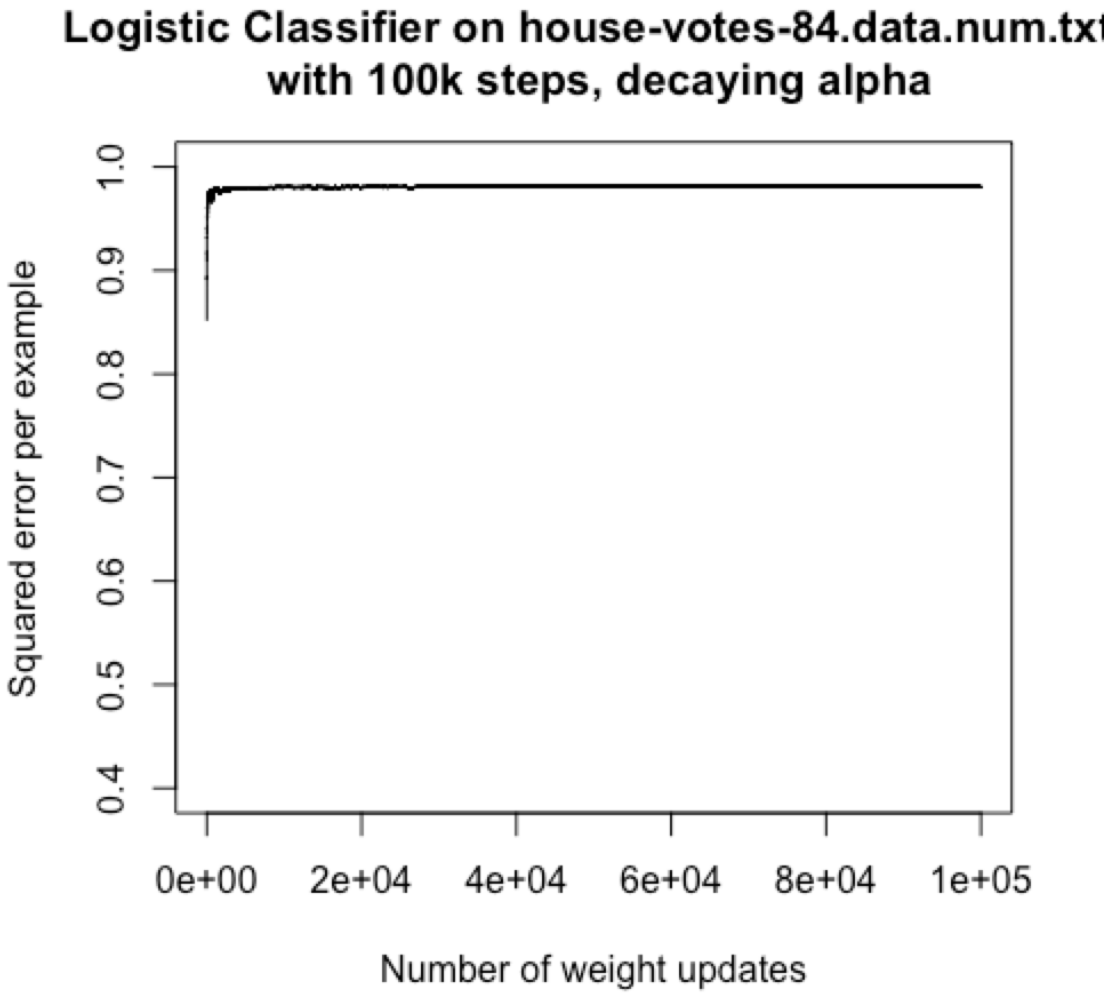
Logistic Classifiers: Figure g, h, and i.



g. In this scenario, the accuracy starts from about 0.76 and ends at about 0.98. Comparing to the learning process of perceptron classifier, the accuracy variation of logistic classifier is much smaller. As the number of weight updates increases, accuracy increases rapidly and stays at 0.98 with small variation and forms similar as a logarithmic curve. After 5K weight updates, the ending accuracy is approximately 0.98 after each run.



h. In this scenario, the accuracy starts from about 0.74 and ends at about 0.98. Comparing to the learning process of perceptron classifier, the accuracy variation of logistic classifier is much smaller. As the number of weight updates increases, accuracy increases rapidly and stays at 0.98 with small variation and forms similar as a logarithmic curve. After 100K weight updates, the ending accuracy is approximately 0.98 after each run.



i. In this scenario, the learning process is similar to the last graph and the accuracy starts from about 0.85 and ends at about 0.98. Comparing to the learning process of the logistic classifier with constant alpha, the accuracy of logistic classifier with decaying alpha rises much quicker. As the number of weight updates increases, accuracy increases with quickly and stays at 0.98 with very small variation.