Linear Classifiers Report

Build Instructions:

Go to directory: cd

Compile java files: javac \*.java

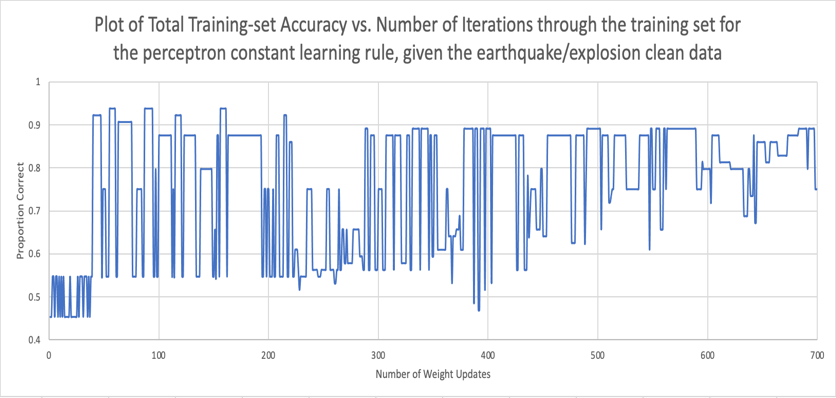
Execute main file: java classifierTester

Graphical user interface, chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceChart

Description automatically generated with low confidenceChart

Description automatically generatedPerceptron Classifiers:

**Short summary:** These graphs visualize the data of the clean and noisy earthquake/explosion data as well as the clean and noisy house-votes data utilizing the perceptron classifiers. Perceptron classifiers are essentially linear classifiers with hard thresholds, meaning there is a strict cutoff of one classification to the other. When experimenting with the constant learning rate, we utilized a learning rate of .95. Given this constant learning rate, you can see that for both the clean and noisy data in both datasets, it’s unpredictable and jumps a lot but “converges” to a value (the pattern is not as noticeable). When there is a decaying learning rate, you can see that for both datasets, the convergence is a lot smoother (the pattern is much more visible) and there isn’t as much jumping around.

Application, table

Description automatically generatedA picture containing chart

Description automatically generatedApplication, table, Excel

Description automatically generatedTable

Description automatically generatedLogistic Classifiers:

**Graphical user interface, application, table, Excel

Description automatically generatedGraphical user interface, application, table, Excel

Description automatically generatedShort Summary:** These graphs visualize the data of the clean and noisy earthquake/explosion data as well as the clean and noisy house-votes data utilizing the Logistic classifiers. Logistic classifiers are essentially linear classifiers with soft thresholds, meaning there isn’t a strict cutoff of one classification compared to the other since it utilizes the sigmoid function which is both continuous and differentiable from range 0 to 1. When experimenting with the constant learning rate, we utilized a learning rate of .95. Given this constant learning rate, you can see that for both the clean and noisy data in both datasets, it’s unpredictable and jumps a lot but “converges” to a value. When there is a decaying learning rate, you can see that for both datasets, the convergence is a lot smoother and there isn’t as much jumping around.

Comparing Perceptron Classifiers and Logistic Classifiers:

When comparing perceptron classifiers to logistic classifiers, you can see the trends for the logistic classifiers more clearly. This happens because the logistic classifiers utilize a soft threshold and allows for some variance while the perceptron classifiers utilize a hard threshold which doesn’t allow for variance (only 0 or 1). Because of the enhanced visualizations for trends, most companies within the industry utilize logistic classifiers to make decisions based on data.

Comparing Constant Learning Rate and Decaying Learning Rate:  
When comparing the constant learning rate and the decaying learning rate, you can see that the decaying learning rate graphs are much smoother than the constant learning rate graphs. This happens because the decaying learning rate takes smaller step sizes for each update to reach the global minimum while the constant learning rate takes the same step size for each update to reach the global minimum (which it might overstep). Because of the smoother visualizations for trends, most companies within the industry utilize a decaying learning rate when using linear classifiers.