Ta’Quawn Watts

Plotting, Salting, and Smoothing using MatLab

For Project 2 of CSCI-3327 Probability and Applied Statistics course at Stockton University, my task was to use create a MatLab script that could create various graphs. The first graph was the graph of a function. The second graph was the graph with “salted” data. The third graph was the graph with “smoothed” data. Salting, or adding noise to, data means to add randomness to the data. Salting data is common with machine learning and neural networks as it benefits neural network and models performance with variations in data. Smoothing data means to reduce the randomness from a data set by removing outliers. Using algorithms, smoothing data can be used to find and predict trends within a data set.

A significant part of this task was learning the syntax of MatLab. Prior to this, I had no experience with MatLab or its syntax. I followed a tutorial on using MatLab created by MathWorks, the creators of MathLab. The tutorial, titled “Create 2-D Line Plot”, went through you can use MatLab to create simple line plots. The first line plot was the value of the sine function from 0 to shown below:



The second line plot was a line plot of the values of sine and cosine function from 0 to . It demonstrated MatLab’s ability to plot multiple line plots in the same figure window aad also changing line colors, line styles, and adding markings as shown below:



MatLab also has the ability to change the appearance of the plot by either omitting line styles or adding markers as show below:





Throughout the tutorial, I became more experienced with the syntax of MatLab. I noticed that MatLab was very similar to other programming languages like Python and Java, making it easier to understand its uses. Using the experience from this tutorial and other documentation, I was able to create a MatLab script that produced a function graph, salted graph, and smoothed graph.

Creating the three graphs was very simple compared to programs using Apache Commons Math Library and JFreeCharts or just Java programming. First, the MatLab script titled “plotter.m” created the function. I decided to use the function . I originally chose this graph because it was a parabola that I was curious on its look when salted and smoothed. Salting the data was as simple as adding a randomize number to the y value. Smoothing was even simpler as MatLab already has a function named “smoothdata” that did all of the work for you. All I had to was just call the function. Afterwards, I created a figure that contained three line plots, graphing the original data, salted data, and smoothed data.



The original function graph was just a curve. However, after salting the data, the curve begins to have jagged unpredictable lines. Smoothing these lines made the jagged lines smooth again with a slight difference from the original function.

Using MatLab to plot, salt, and smooth data seems to be simplest as the only challenge is learning the syntax and how exactly it works. Salting is simple with just adding a random number and smoothing is even simpler with its own function. After that, it is just plotting which is way simpler in MatLab versus using JFreeCharts.