CS 460G: Homework #3 Write-up

By Steven Penava

1) **Files**
   - **a.** poly_grad.py: source code for the polynomial regression with gradient descent
   - **b.** data (folder)
     - **i.** synthetic-1.csv, synthetic-2.csv, synthetic-3.csv: test data
2) **Libraries Used**
   - **a.** matplotlib
   - **b.** numpy

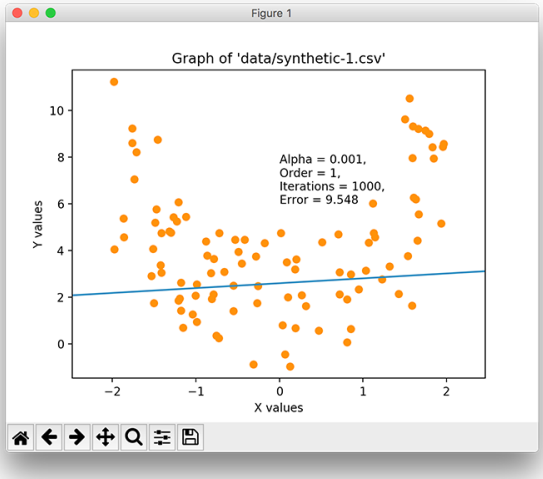
   Note: neither of these libraries are used to accomplish high-level calculations that cheat the work of the assignment.
3) **Instructions for Use**

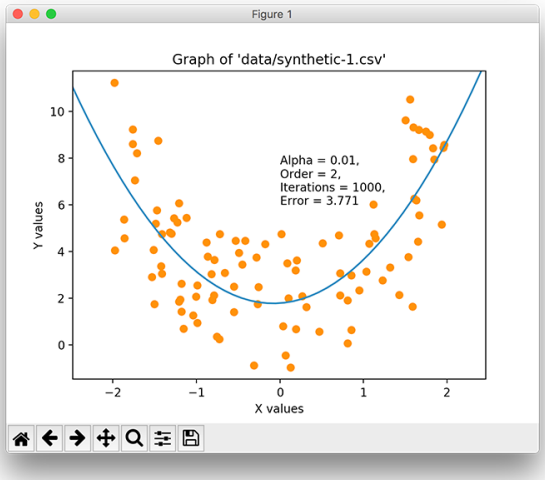
   I decided to make my program customizable according to the user:
   - Simply run the poly_grad.py file in the environment of your choice (just make sure the data folder containing the synthetic files is in the same directory as the script).
   - You will be prompted to pick a file (1, 2, or 3), an order (1, 2, 4, or 9), an alpha value (anything <= 0.01 is recommended), and an iteration count (1000 recommended, at the very least). **Note:** when running the program with an order of 9, use an alpha value of 0.00001 at the largest to prevent errors).
   - The program will run the gradient descent, and a graph should pop up pertaining to what you wanted! The error will print in the graph and in the console.
4) **Implementation Decisions**
   - I implemented batch gradient descent in my program. This means that the gradient for each weight is calculated by looping through the entire input data set.
   - I decided to use numpy arrays for my calculations since it is is perfect for programs like this.
   - I calculated the error using mean squared error, like the last homework assignment.
   - Here is my interpretation of the gradient of the formula given in the instructions in my code:
     - ```
       for k in range(len(weights)):
           gradients[k] += (1 / m) * (x ** k) * ( getHofTheta(weights, x) - y)
       ```
     - m is the length of the data set, and x and y are individual points.
     - "getHofTheta(weights, x)" is a function that dynamically builds the hypothesis function. It is a simple loop through the weights array and can be found on lines 18-22.
   - The 9th order polynomial with the 3rd data set is not the prettiest result. The more iterations you run, the better it will look! Use a very small alpha value, too.
       ```

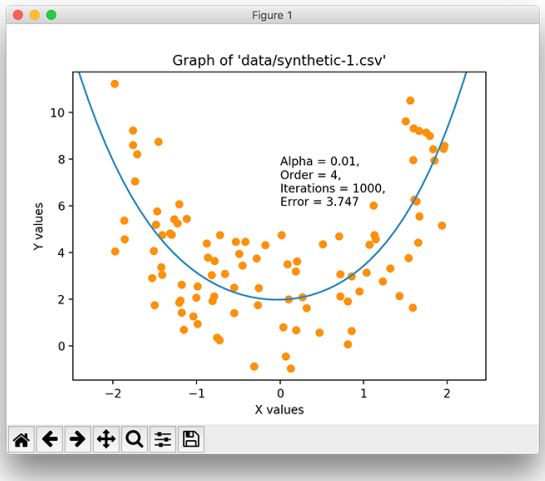
5) Results

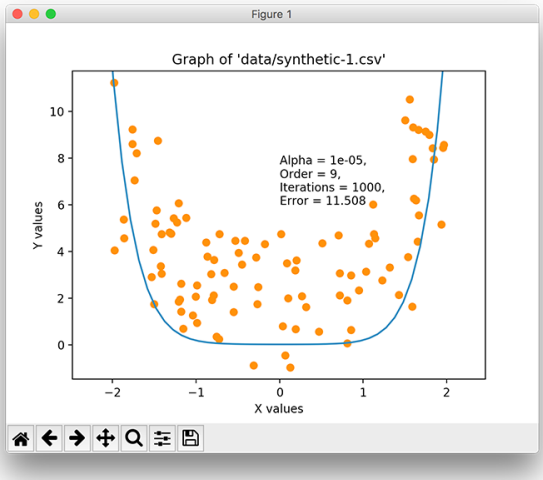
Here are my graphs from the different synthetic data sets on order 1, 2, 4, and 9. **Errors, orders, and alpha values are reported on the graphs.**

Synthetic 1



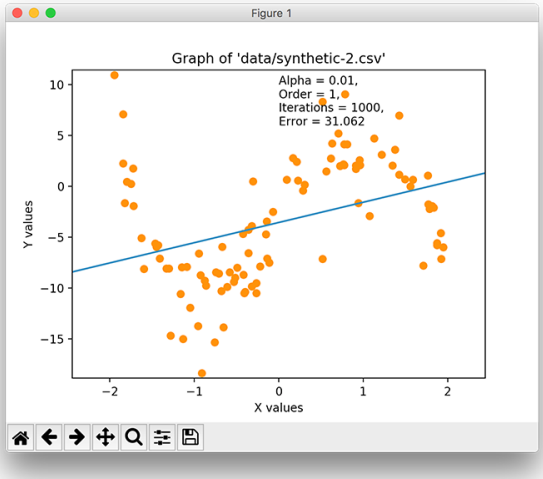


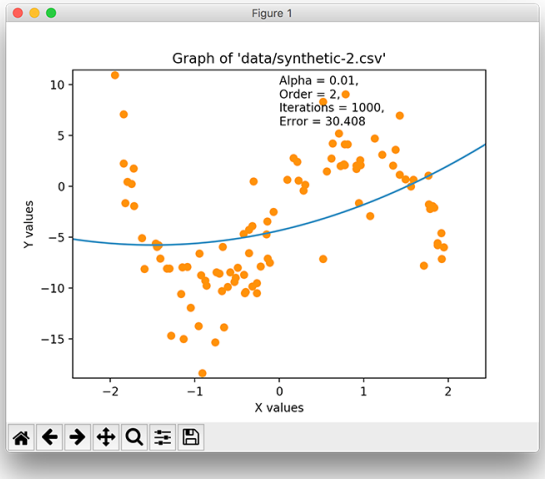


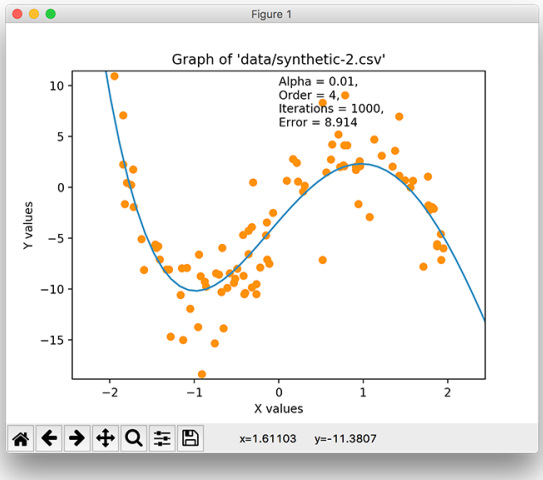


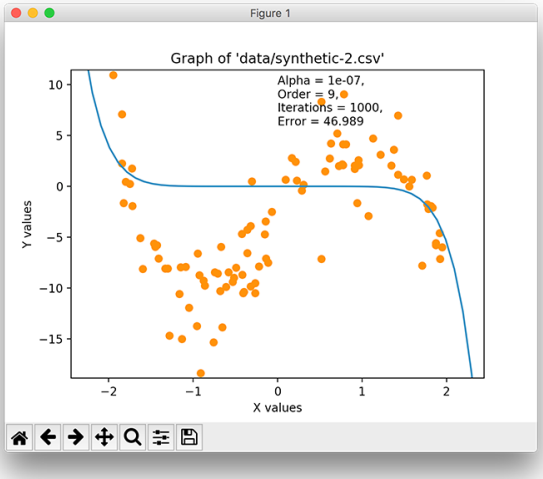
Synthetic 2 and 3 next pages…

Synthetic 2









Synthetic 3 next page…

Synthetic 3

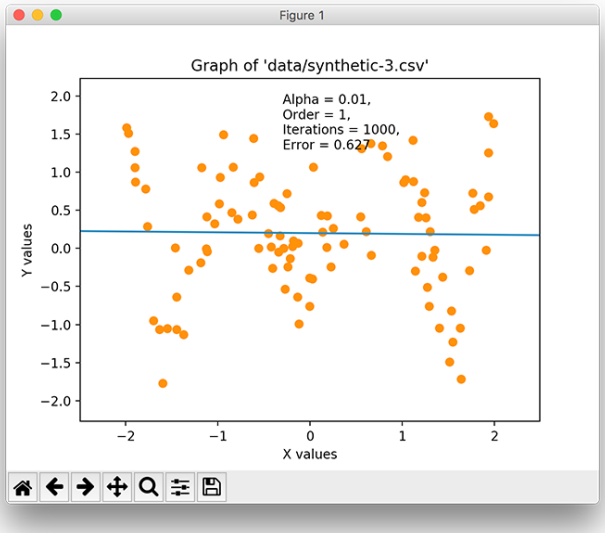


Figure 1
Graph of 'data/synthetic-3.csv'
Alpha = 0.01,
Order = 1,
Iterations = 1000,
Error = 0.627
Y values
X values

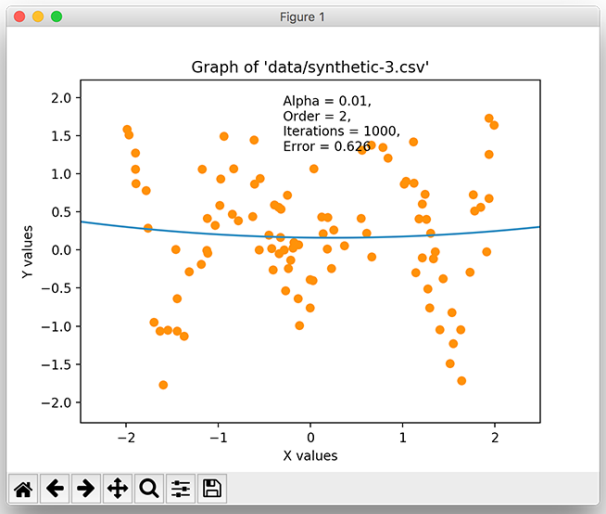


Figure 1
Graph of 'data/synthetic-3.csv'
Alpha = 0.01,
Order = 2,
Iterations = 1000,
Error = 0.626
Y values
X values

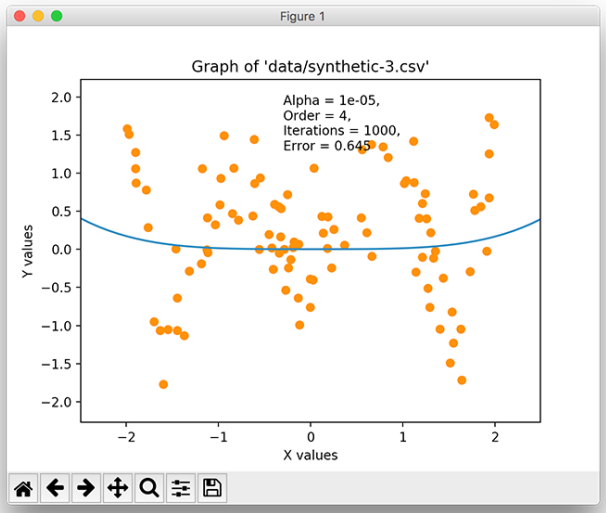


Figure 1
Graph of 'data/synthetic-3.csv'
Alpha = 1e-05,
Order = 4,
Iterations = 1000,
Error = 0.645
Y values
X values

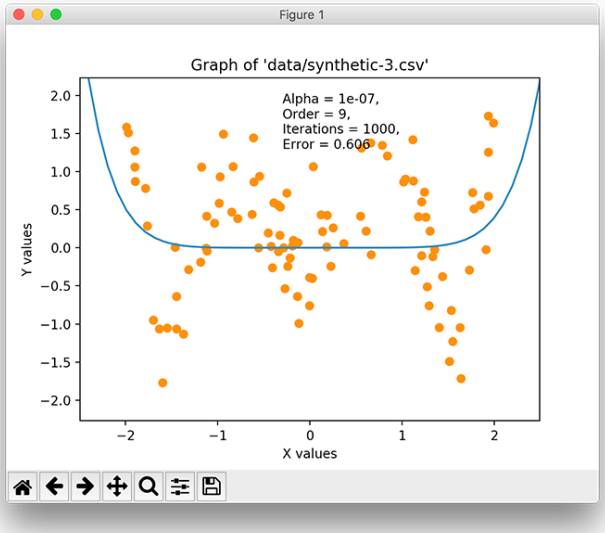


Figure 1
Graph of 'data/synthetic-3.csv'
Alpha = 1e-07,
Order = 9,
Iterations = 1000,
Error = 0.606
Y values
X values