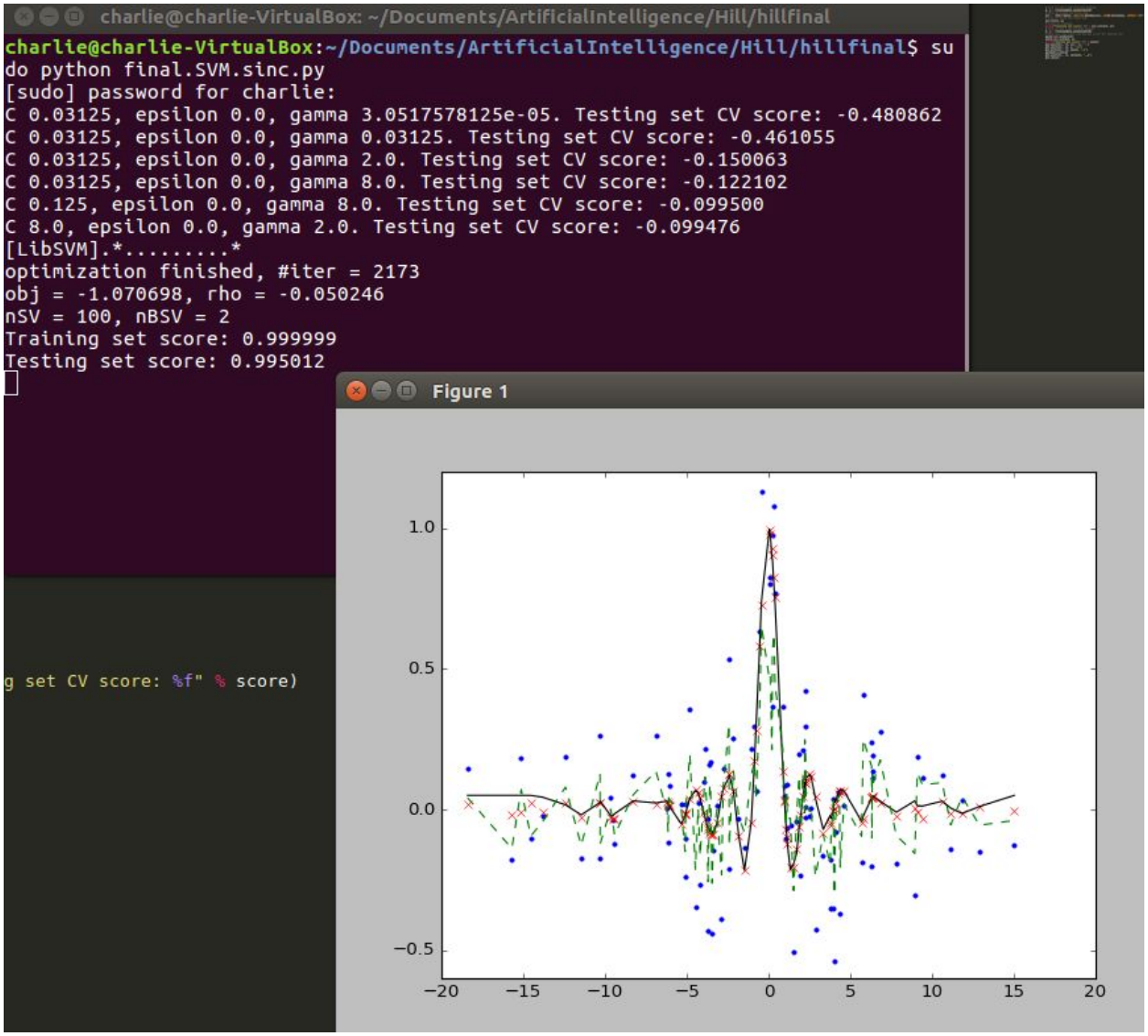
Charlie Hill
Professor Rivas
Artificial Intelligence
14 December 2016

Artificial intelligence Final

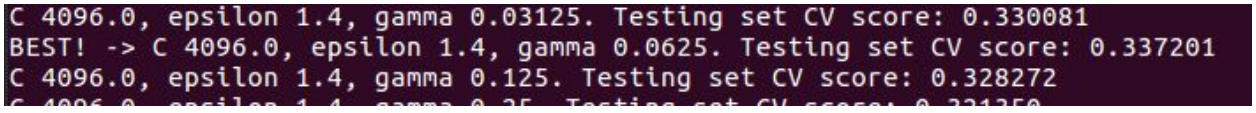
1. After running the program, I found that with each iteration the penalty, C, increases. Eventually once C gets large enough, that iteration is chosen. From that iteration, the best C, epsilon and gamma are taken and uses as hyper-parameters.

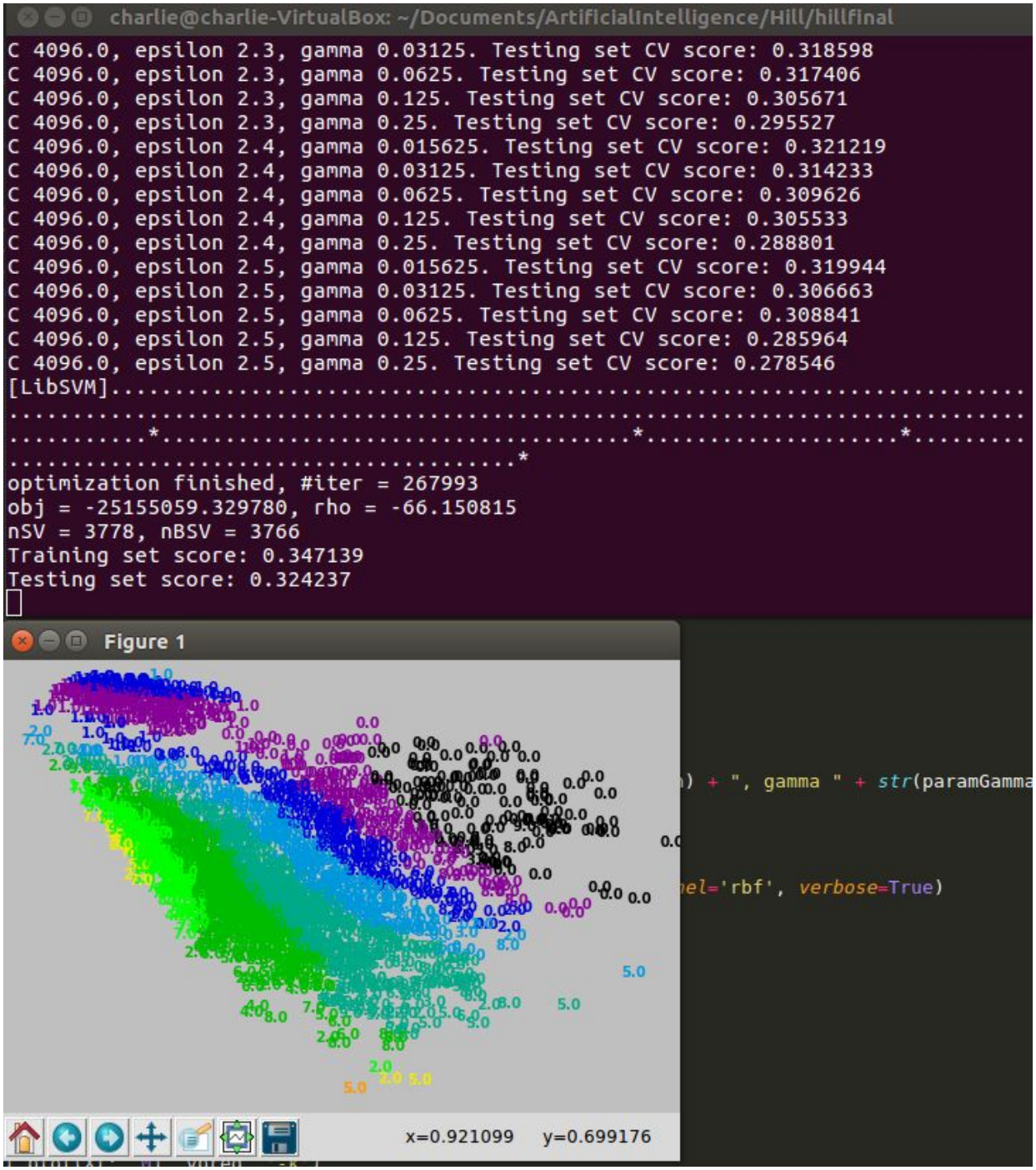
   The hyper-parameters are found through the first part of the code. Running the support vector regression on some test data allows to grab the best C, epsilon and gamma. By finding the best C, epsilon and gamma we are able to run the support vector regression again with them as hyper parameters. So rather than the support vector regressor starting off with no specific parameters we start it with the hyper parameters. This gives the support vector regressor a head start. C is the penalty factor and it ends up being a great penalty. Making the penalty so large enables less errors. Therefore as our penalty increases our error decreases. C is our penalty and *e* is our error. Therefore as C increases, *e* decreases. For this data though epsilon remains zero. The number of datapoints greatly affects how fast this program runs. The program runs within 5 minutes with just 100 data points. However when I up the dataset to 500 or 1000, I am waiting hours for the program to finish. It is also noteworthy that as our our penalty increases our testing CV score increases as well. This is also probably related to the fact that our error becomes low as our penalty increases.

Below is a screencap of the program run with 100 samples. I was not able to reach the plot stage with 1000 samples because my terminal crashes before the data can be plotted.



2. After 267,993 iterations I was able to find the hyper-parameters. C was 4096. Epsilon was 1.4. Gamma was 0.0625. The testing set CV score was 0.33721.





This is what the final plot looked liked. Each number is assigned a color and then the data is grouped by proximity to its respective color. For example all the numbers in black are guessed as zero. The data is not easily separated as depicted by the graph. There is much overlap between

the numbers as handwritten numbers can sometimes looks like one another. If you ask me the plot looks like a rainbow.

The first part of the code found the hyper-parameters. Running the support vector regression on some test data allows to grab the best C, epsilon and gamma. By finding the best C, epsilon and gamma we are able to run the support vector regression again with them as hyper parameters. So rather than the support vector regressor starting off with no specific parameters we start it with the hyper parameters. This gives the support vector regressor a head start. C is the penalty factor and it ends up being a great penalty. Making the penalty so large enables less errors. Therefore as our penalty increases our overall error decreases. C is our penalty and *e* is our error. Therefore as C increases, *e* tends to decrease. When testing the hyper-parameters the code tested with different penalties and as the tests went on, C increased. C began as less than zero but ended up at 4096. As C increases the epsilon is reset to its default value and then tested with increasing error values. The best values are then saved once they are found based on the testing CV score. Finally once the best values are saved they are used in the last test and that generates the plot.