Symbols, Patterns, and Signals - Coursework Report

Ruitang Chen (rc17320), 12th March 2020

1. Introduction

Pattern Recognition is one of the most exciting fields among the study of information technology and computer science these days. This field concentrates on how computer simulates or implements human learning behaviours in order to acquire new knowledge or skills, and reorganize the knowledge with existing structure to continuously improve the performance.

The primary idea of supervised learning is to teach computers how to achieve self-learning. The purpose of this assignment is to improve our ability to predict models based on unknown data and optimize the performance of computers via collecting feedbacks.

2. Least Squares Method

<2.1> Data

In this coursework, I was required to determine the model of the 11 given files, each consists several line segments made up by 20 different points.

By collecting a certain numbers of known samples and determining the characteristics of these samples, data-based pattern recognition enables computers to use samples as training machines and classify them after training, instead of relying on people's knowledge on the research object to establish a classification system. [1]

<2.2> Least Squares Equation

The term regression refers to constructing the most likely straight line to fit the data, in order to predict an accurate output value based on existing data. The selected parameter determines the accuracy of the obtained straight line with respect to the training set. The difference between the predicted value of the model and the actual value in the training set is the modelling error.

My goal is to choose the model which can produce a modelling parameter with the smallest sum of squared errors and this method is known as Least Squares Method. In order to calculate the reconstruction error I used the following formula:

$$R(a,\ b)\ =\ \sum_i (y_i - (a + bx_i))^2 \qquad (1)$$

For more complex parametric forms, such as polynomial generalisation or unknown model, matrix is the better choice. Because vectors and matrices provide efficient ways to organize large amounts of data, especially when dealing with huge training datasets.

$$y_{i = a_1 + a_2x + a_3x^2 + \ldots + a_px^{p-1}} \qquad (2) \text{ the expression of y}$$

$sum\ squared\ error\ =\ \sum(y\ - \hat{y})^2$ (3) formula of sum squared error

$a_{LS}\ = (X^T X)^{-1} X^T y$ (4) formula to solve least square in matrix form

Express this in code as:

```
A = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(yi)
sumerror = np.sum((yhat - y ) ** 2)
```

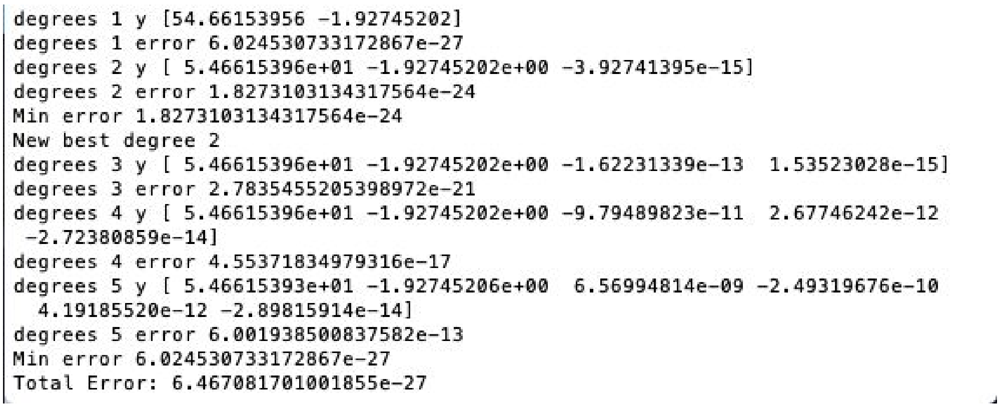
Matrix dimension is very important here, especially I made some mistakes during the test, noting the size of multiplied matrices.

3. Result Analysis

<3.1> choosing equation and error discuss

In order to determine the correct function type of each line segment, we need to compare the 'sum of square error' under linear, polynomial or unknown functions. The smallest result we get is function we need.

As shown in Figure 1. is the data in `basic_2.csv' under different functions and different degrees(up to degree 5).



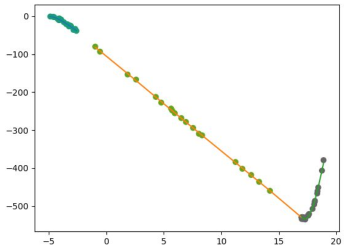
```
degrees 1 y [54.66153956 -1.92745202]
degrees 1 error 6.024530733172867e-27
degrees 2 y [ 5.46615396e+01 -1.92745202e+00 -3.92741395e-15]
degrees 2 error 1.8273103134317564e-24
Min error 1.8273103134317564e-24
New best degree 2
degrees 3 y [ 5.46615396e+01 -1.92745202e+00 -1.62231339e-13  1.53523028e-15]
degrees 3 error 2.7835455205398972e-21
degrees 4 y [ 5.46615396e+01 -1.92745202e+00 -9.79489823e-11  2.67746242e-12
 -2.72380859e-14]
degrees 4 error 4.553718349793l6e-17
degrees 5 y [ 5.46615393e+01 -1.92745206e+00  6.56994814e-09 -2.49319676e-10
  4.19185520e-12 -2.89815914e-14]
degrees 5 error 6.001938500837582e-13
Min error 6.024530733172867e-27
Total Error: 6.467081701001855e-27
```

Figure 1. Example of training_data/basic_2.csv(rc17320's results)

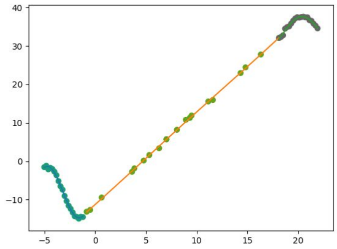
During programming, I divided the situation for different types of functions and polynomial degrees, calculated the results obtained under each function and compared them, in order to get the minimum one.

<3.2> figure plotting and line fitting

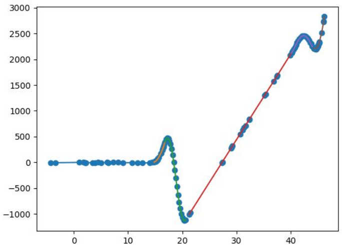
After choosing the suitable equation for each test file, I draw a reconstructed line to fit the target points. Visualisation plays a vital enabling role in our ability to understand large and complex data, which can be in two or more dimensions.[2]



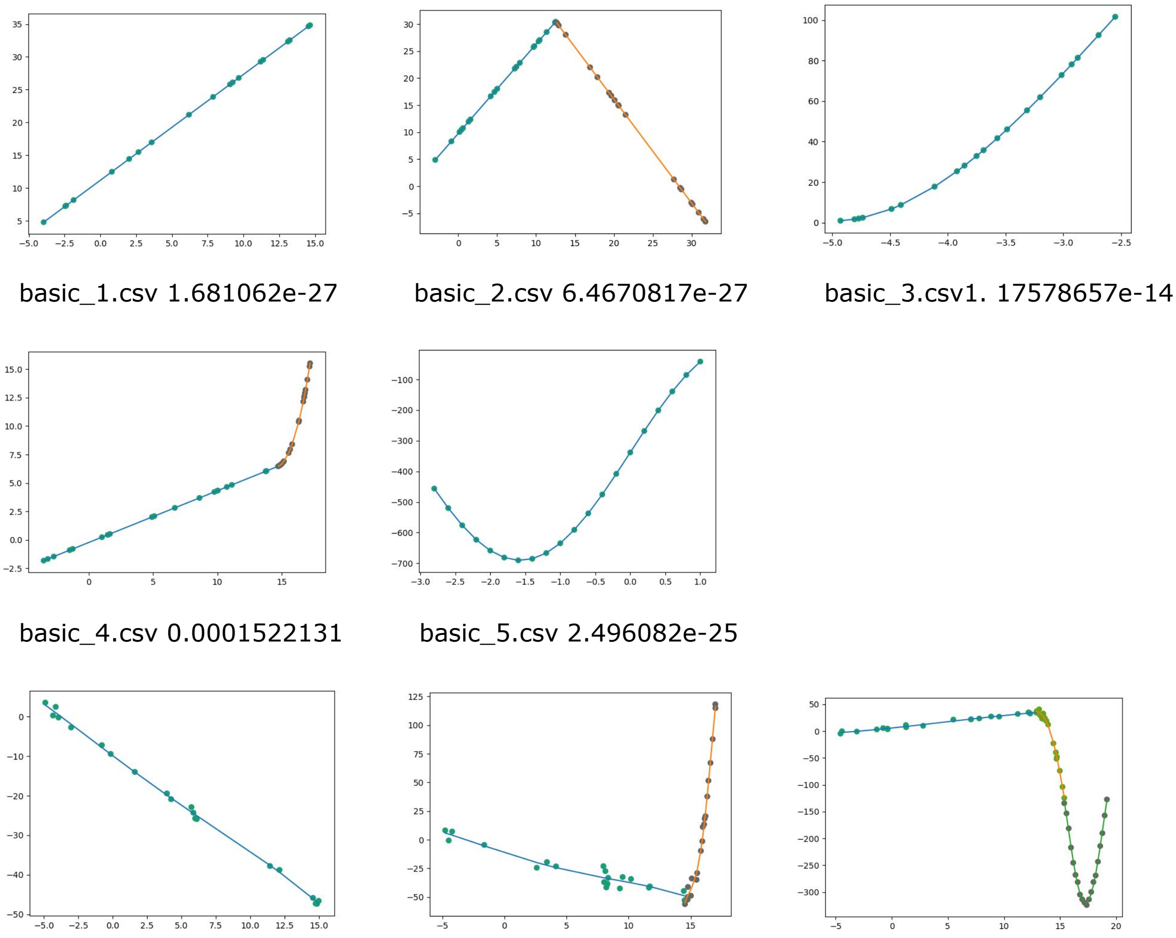
adv_1.csv 381.16247373



adv_2.csv 3.39715050



adv_3.csv 829.809036451



basic_1.csv 1.681062e-27 basic_2.csv 6.4670817e-27 basic_3.csv1. 17578657e-14

basic_4.csv 0.0001522131 basic_5.csv 2.496082e-25

noise_1.csv 10.537383349 noise_2.csv 727.2004104 noise_3.csv 440.868759

Figure 2: Plotting graph for provided training_data file (rc17320's graph)

The result graphs for the provided training data are shown in Figure 2. above. This gives an overview of how well the function I get fits the provided data sets.

Reference:

1. Pattern Classification -- Richard O. Duda, David G. Stork, Peter E.Hart https://www.academia.edu/33126492/Pattern_Classification_by_Richard_O._Duda_David_G._Stork_Peter_E.Hart

2. Visualization handbook -- Hansen and Johnson(2004) http://www.nlpinfocentre.com/downloads/feb2015/Charles%20D.%20Hansen%20,%20Chris%20R.%20Johnson%20-%20The%20Visualization%20Handbook%20.pdf