A Report on

Basis Functions and Applications in Curve Fitting

Prepared as project report for

CS F320: Foundations of Data Science

Submitted to

Dr. Navneet Goyal, Professor

(Dept. of Computer Science and Information Systems)

On

April 7, 2021

Yash Gupta	2018A7PS0262P
Siddharth Kapoor	2018A7PS0232P

Table of Contents

Basis Functions

A **basis function** is an element of a particular basis for a function space. Every continuous function in the function space can be represented as a linear combination of basis functions, just as every vector in a vector space can be represented as a linear combination of basis vectors.

The major use of Basis functions in Data Science comes in the case of curve-fitting. Curve Fitting is a major problem in Data Science, in which we seek to find a curve which best fits the given set of data points. Instead of selecting any arbitrary function and testing its fit on the dataset, we streamline the process of finding the best-fit curve by taking basis-functions in use as building blocks.

Essentially, we select a few basic functions like polynomial, gaussian, sigmoidal to name a few and any curve can generally be formed by a linear combination of these functions. Hence what we are left with is just finding the parameters/constants which are multiplied to these functions to find the best fit curve.

The different kinds of Basis Functions are:

1. Polynomial
2. Sigmoidal
3. Gaussian
4. Fourier
5. Splines
6. Wavelets

For the analysis of the quality of fit of a particular basis function with respect to the actual dataset, we can take error calculation in consideration.

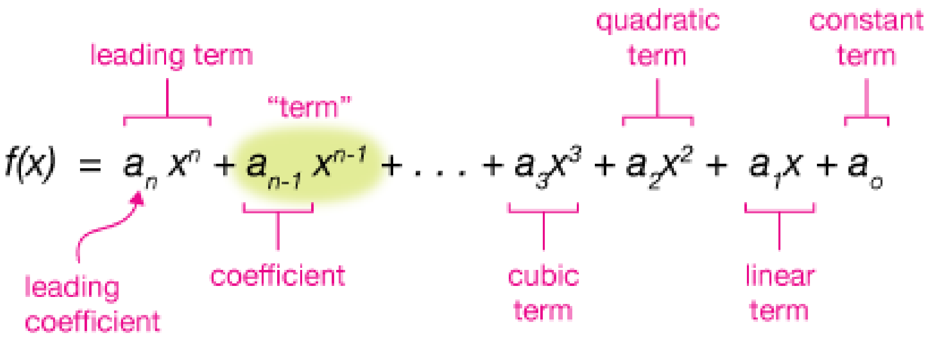
Root Mean Squared Error(RMSE) - It calculates the error between the predicted value and the actual value and sums it for all the data points. The formula for RMSE is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

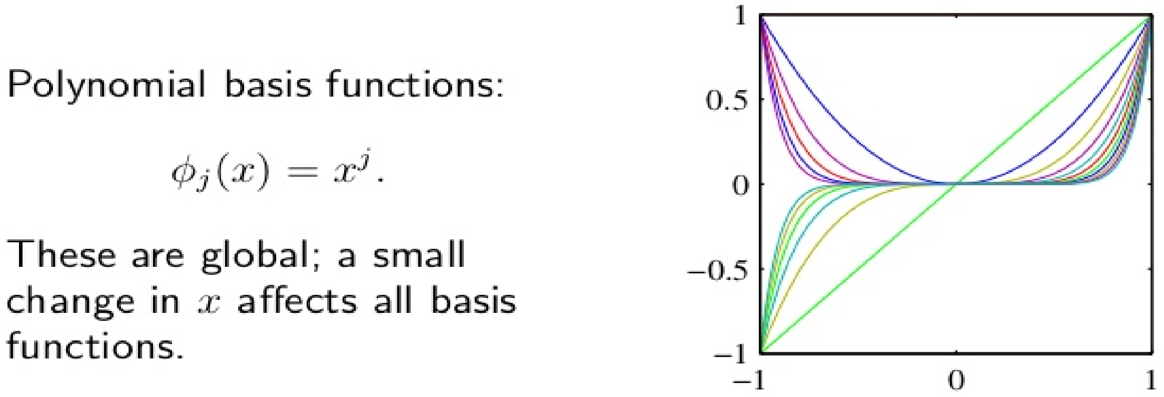
Essentially, the purpose of RMS Error calculation is to prevent underfitting. Underfitting is the condition in which the model gets oversimplified due to choosing a less complex basis function and hence does not fit the data well.

● Polynomial Basis

Polynomial basis uses a linear combination of well defined polynomial functions to generate the best fit curve. General equation of a polynomial function is given by:



We create basis functions by taking a monomial set : { 1, x, x^2, x^3……. x^n} and creating a linear combination of the polynomials present in the set to attain the best-fit curve.The coefficients are then determined using RMS error and standard deviation.

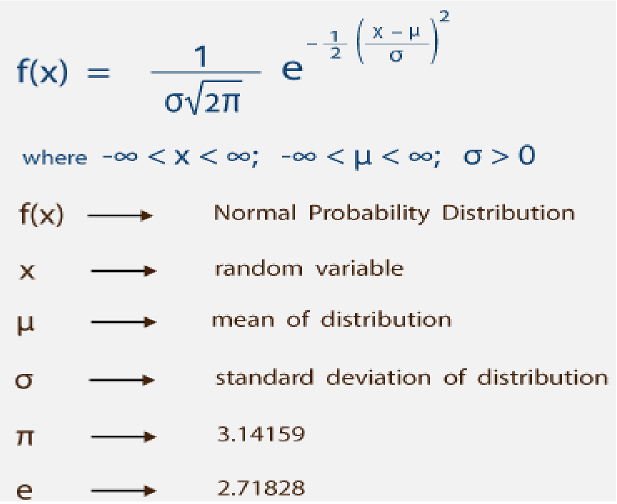

Polynomial basis functions:

$$\phi_j(x) = x^j.$$

These are global; a small change in x affects all basis functions.

• Gaussian Fitting

Gaussian Distribution, also known as Normal Distribution, is a continuous probability distribution curve. It is typically bell-shaped. The general form of its probability density function is:

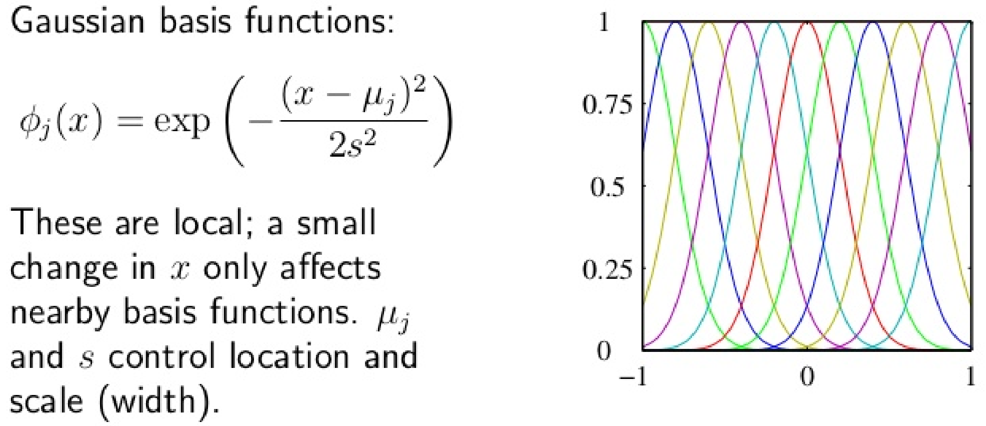


$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

where $-\infty < x < \infty$; $-\infty < \mu < \infty$; $\sigma > 0$

$f(x)$	→	Normal Probability Distribution
x	→	random variable
μ	→	mean of distribution
σ	→	standard deviation of distribution
π	→	3.14159
e	→	2.71828

Gaussian functions have the property that for a well defined basis set of gaussian functions, any new gaussian function can be represented as a linear combination of the functions present in the basis set.



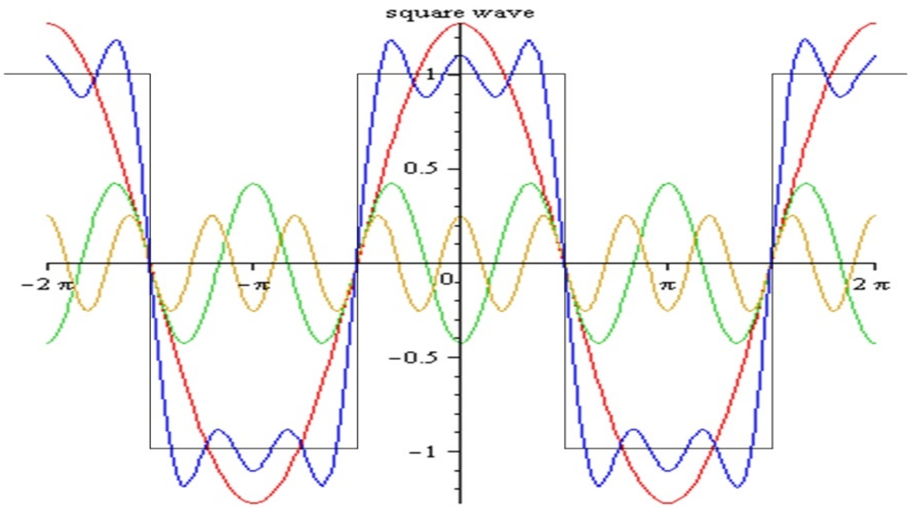
Gaussian basis functions:

$$\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$$

These are local; a small change in x only affects nearby basis functions. μ_j and s control location and scale (width).

● Fourier Analysis

A **Fourier series** is a periodic function composed of harmonically related sinusoids, combined by a weighted summation. With appropriate weights, one cycle (or *period*) of the summation can be made to approximate an arbitrary function in that interval. The process of deriving weights that describe a given function is a form of Fourier Analysis.



The figure shown on the left is an example of a Square Wave created through superimposition of multiple out of phase sinusoidal waves with varying weights. Mathematically, the square wave is a linear combination of the sine functions with different coefficients and different phases.

<u>Basis Functions</u> **: f(t) = {sin(ωx), cos(ωx), 1}**

$$P(t) = \frac{1}{2}a_0 + a_1 \cos \omega t + a_2 \cos 2\omega t + \ldots + b_1 \sin \omega t + b_2 \sin 2\omega t + \ldots$$
$$= \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos n\omega t + \sum_{n=1}^{\infty} b_n \sin n\omega t$$

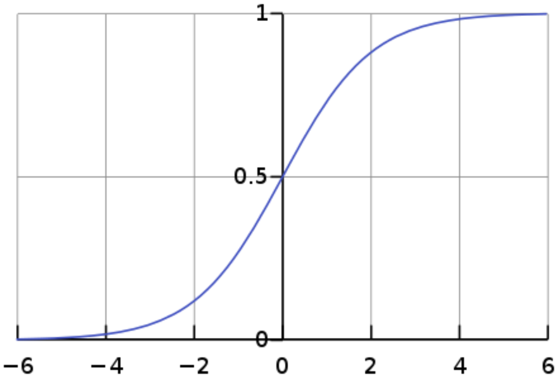
where,   n = degree of fourier series

$a_0, a_1, a_2....., b_1, b_2, b_3.....$ are learnable parameters of linear combination.

● Sigmoidal Basis

A **sigmoid function** is a mathematical function having a characteristic "S"-shaped curve or **sigmoid curve**. A common example of a sigmoid function is the logistic shown in the first figure and defined by the formula:

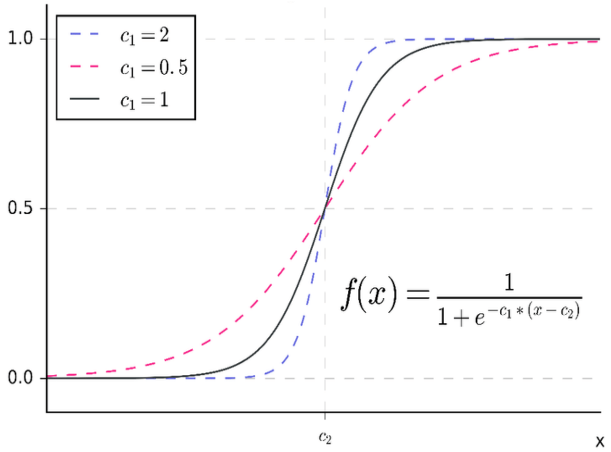
$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} = 1 - S(-x).$$



Here the range of S(x) $\in \{0, 1\} \forall\ x \in R$

We can however alter the shape and range of the function by introducing basis functions and taking varied parameters to obtain different function curves.

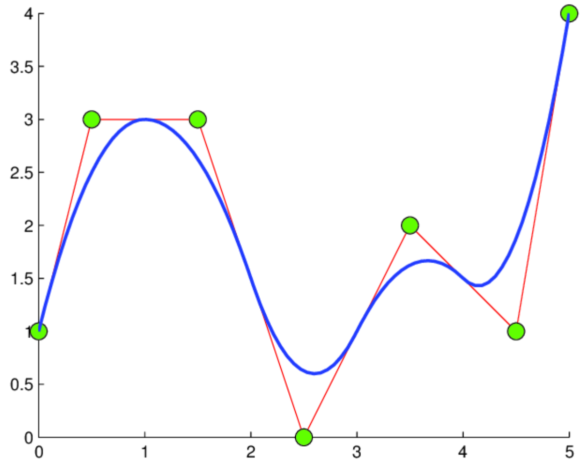
An example of the same is shown below:



In the figure shown, we take a sigmoidal function with two parameters, $c_1$and c_2, and change the function's shape according to our requirement by varying the values of the parameters

- ## Splines

A piecewise polynomial function that can have a locally very simple form, yet at the same time be globally flexible and smooth. Splines are very useful for modeling arbitrary functions.



A quadratic (p = 2) B-spline curve

Splines have interesting properties. We define splines as piecewise continuous polynomial functions. Hence the control points where the polynomial changes is of significance. The continuity of the curves at the control point is of three types:

1. C^0 Continuity: meaning that the two segments match values at the join.
2. C^1Continuity: meaning that they match slopes at the join.
3. C^3Continuity: meaning that they match curvatures at the join.

Any spline function of a given degree can be expressed as a linear combination of B-splines of that degree.

Defining Basis Splines:

Let U be a set of $m + 1$ non-decreasing numbers, $u_0 <= u_2 <= u_3 <= ... <= u_m$. The u_i's are called *knots*, the set U the *knot vector*, and the half-open interval $[u_i, u_{i+1})$ the *i-th knot span*.

The i-th B-spline basis function of degree p, written as $N_{i,p}(u)$, is defined recursively as follows:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Implementation Details

The general formula for Linear Regression is

$$W = \left(\Phi^T \cdot \Phi\right)^{-1} \cdot \Phi^T \cdot Y$$

where W is the Coefficient Matrix, Φ is the Matrix formed by evaluating the Basis Functions at the values of X-coordinates present in the dataset, and Y is the Matrix with the Y-coordinates of the dataset. Here it is assumed that the errors in the coordinates in the dataset are Normally distributed with a mean of 0.

$$\Phi = \begin{bmatrix} \phi_0(x_1) & \cdots & \phi_m(x_1) \\ \vdots & \iddots & \vdots \\ \phi_0(x_n) & \cdots & \phi_m(x_n) \end{bmatrix}$$

$$W = \left[w_0\, w_1 \ldots\, w_m\right] \qquad Y = \left[y_1\, y_2 \ldots\, y_n\right]$$

$$F(x) = w_0\phi_0(x) + w_1\phi_1(x) + \ldots + w_m\phi_m(x)$$

$\phi_i(x)$ is the i-th Basis Function evaluated at x.

The matrices are stored as 2D arrays in the program and the inverse of a matrix is calculated using the Row-Transformation method using an Augmented Matrix for efficiency.

The coefficients are calculated using the given formula and stored in the coefficient matrix. In our implementation, we have assumed the data to be 2-dimensional although this formula can be applied to any d-dimensional data.

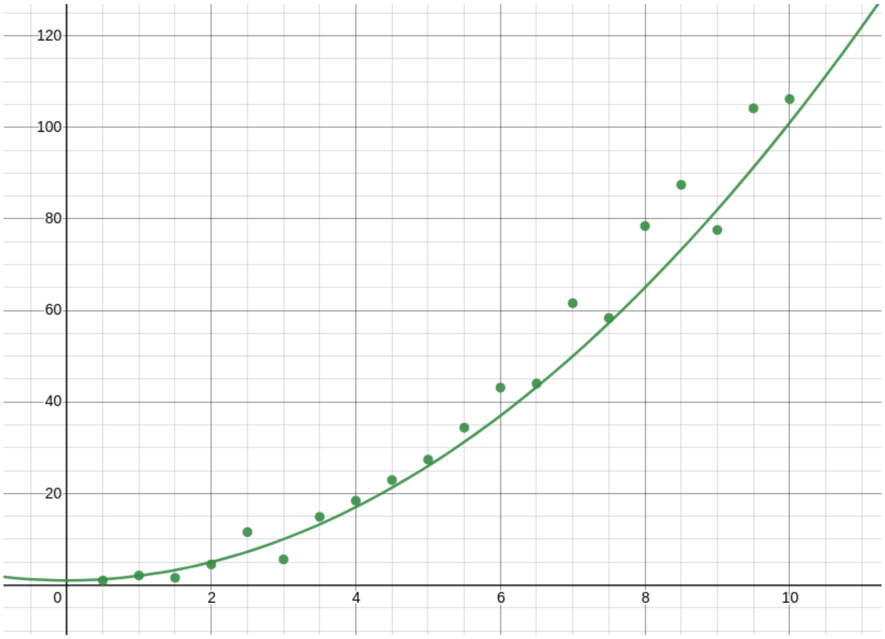
Finally, using the calculated coefficients, the Root Mean Square Error is calculated by plugging the X-coordinates into the obtained function F(x) by the following formula

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(y_i - F\left(x_i\right)\right)^2}{N}}$$

The program first starts with only 1 basis function and with each iteration increases the number of basis functions till finally they are equal to the number of data points in our dataset.

The dataset is a set of 20 X,Y coordinates obtained by adding random, normally distributed errors in to the values of the function ***F(x) = x²* + *1***

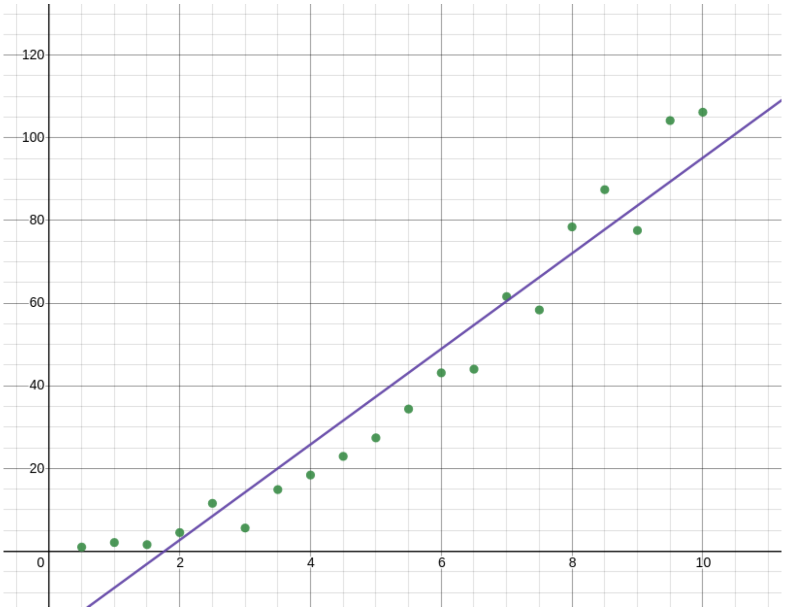
Original Plot



Polynomial Regression Implementation

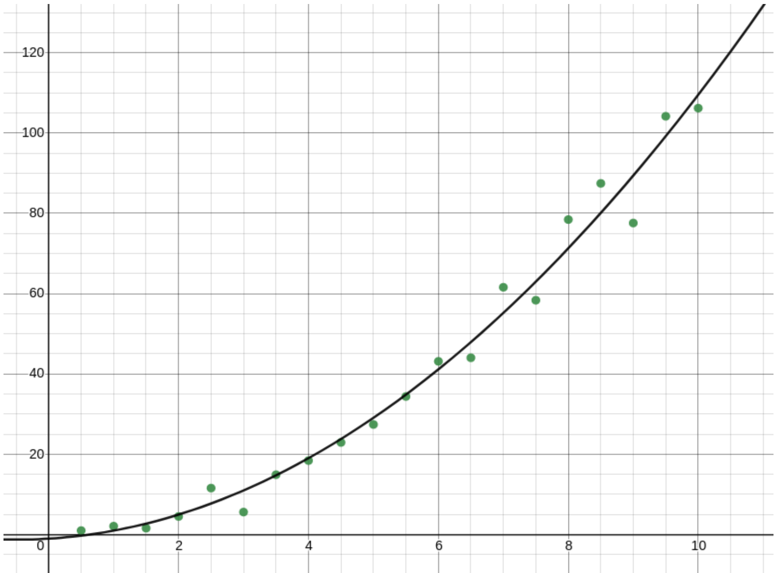
In Polynomial Linear Regression, the basis functions are of the form ***1, x, x^2, … , x^m***, with m<=20. First we start with just ***1*** as the basis function and with every iteration, we add a new monomial as a basis function. For lower degree polynomials, the curve obtained underfits, but with higher degree polynomials tend to overfit. A reasonably good fit was obtained at degree 2.

Polynomial Regression Curve 1



RMS Error	8.73969
Equation	-20.4305 + 11.5581x
Fit	Underfit

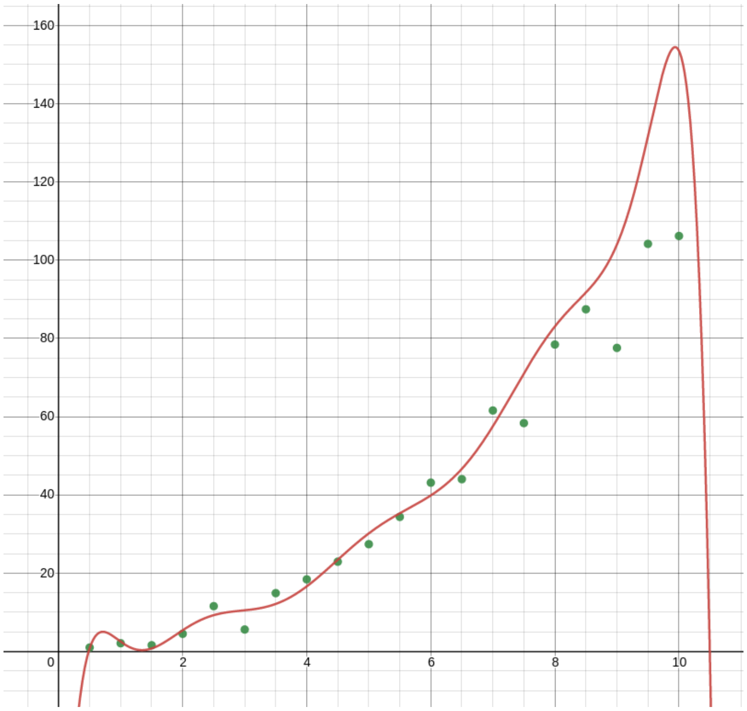
Polynomial Regression Curve 2



RMS Error	4.55143
Equation	-1.0399 + 0.981416x + 1.0073x^2

Fit	Good

Polynomial Regression Curve 3

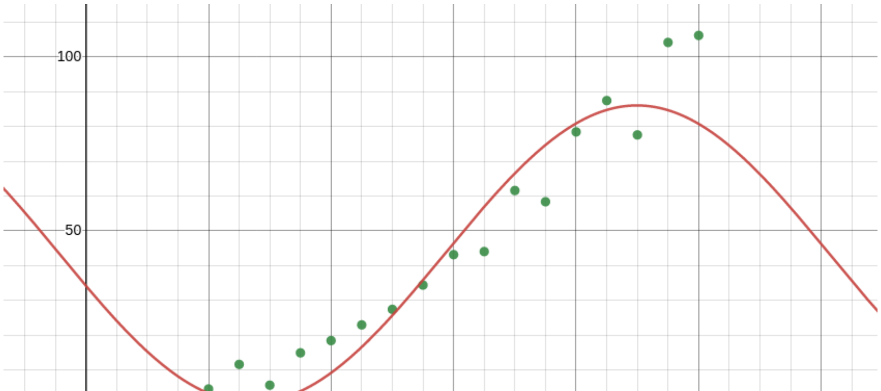


RMS Error	3.8951
Equation	-111.4 + 521.581x + -912.279x^2 + 817.087x^3 - 424.471x^4 + 136.598x^5 - 28.0296x^6 + 3.66818x^7 - 0.296017x^8 + 0.0134119x^9 + -0.00026083x^{10}
Fit	Overfit

Fourier Regression Implementation

The basis functions are of the form ***1, sin(kwx), cos(kwx)***. In our implementation, we assumed ***w=0.5***, and varied ***k*** from 1 to n/2. The first iteration started with just ***1*** as the basis function, but subsequent iterations included the sine and cosine terms. For a low number of sine and cosine basis, the curve underfits, but as the number of these terms increases, the general accuracy of the curve increases.

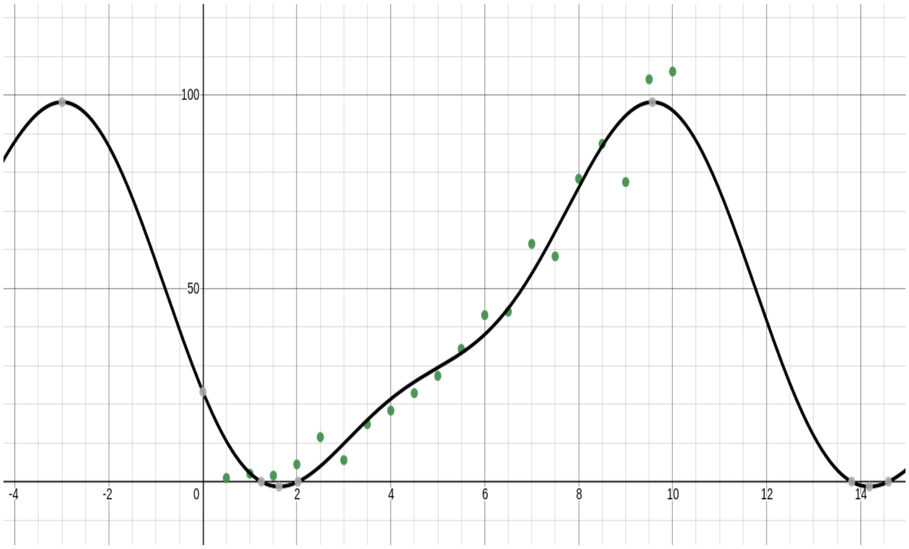
Fourier Regression Curve 1



RMS Error	11.6708
Equation	43.2909 -

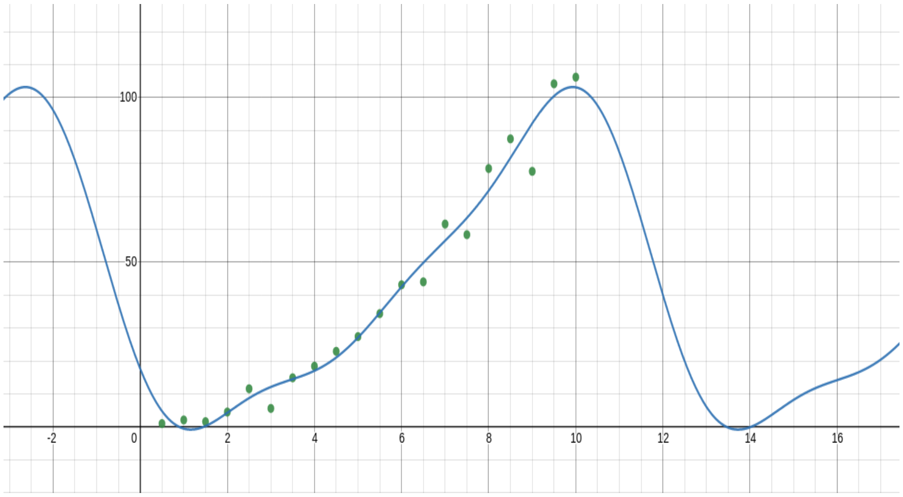
	41.7622sin(x/2) - 9.14914cos(x/2)
Fit	Underfit

Fourier Regression Curve 2



RMS Error	6.25436
Equation	43.6481 - 43.2042sin(x/2) - 7.83284sin(x) - 9.19388cos(x/2) - 11.1678cos(x)
Fit	Good

Fourier Regression Curve 3



RMS Error	4.89973
Equation	44.1706 - 44.3948sin(x/2) - 9.29267sin(x) - 1.07743sin(3x/2) - 8.73093cos(x/2) - 11.9791cos(x) - 5.82388cos(3x/2)
Fit	Good

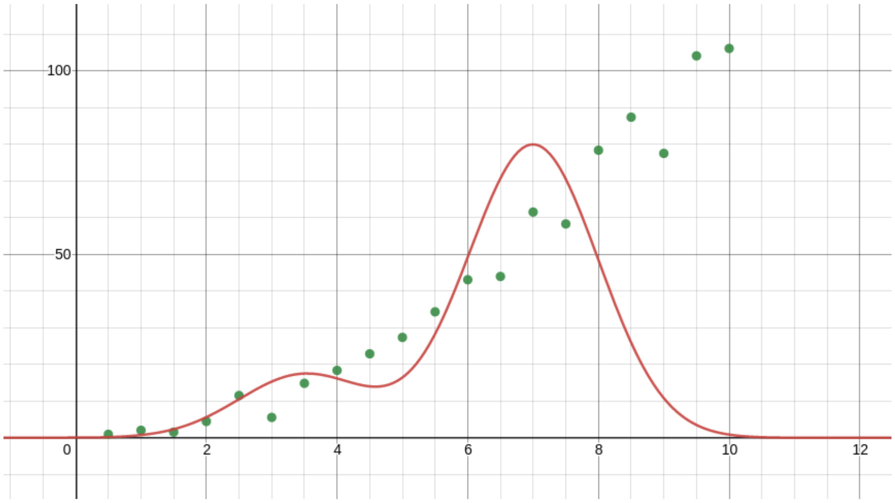
Gaussian Regression Implementation

In Gaussian Linear Regression, the basis functions are of the form

$$e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

with the mean being the point around which the bell curve is centered and the variance determining the spread of the bell curve. In our implementation, the variance was assumed to be equal to 1. Initially only 1 gaussian is set as the basis function, which is centered at the mean of all the X-coordinates in the dataset. With each iteration, a new bell curve is added to the set of basis functions, each distributed at equal intervals. When the number of bell curves is low, the curve underfits. But as the number of gaussians is increased, the curves tend to be a good fit.

Gaussian Regression Curve 1

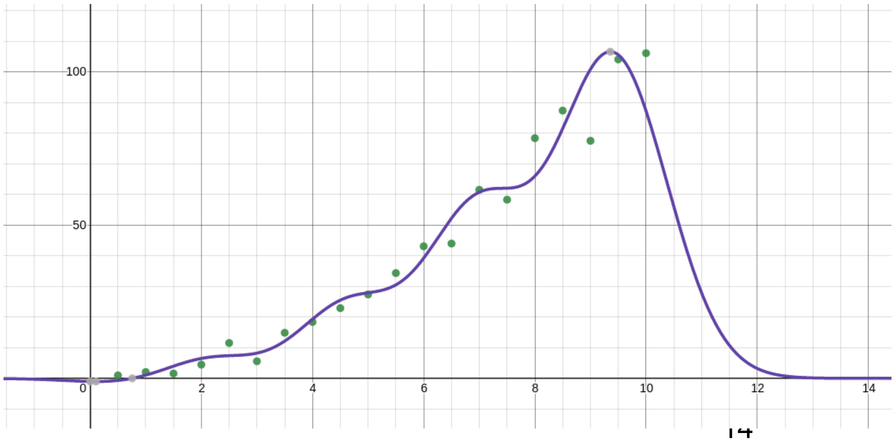


RMS Error	39.9607

Equation	17.3608exp(-(x-3.5)^2/2)
	+ 79.9717exp(-(x-7)^2/2)

Fit	Underfit

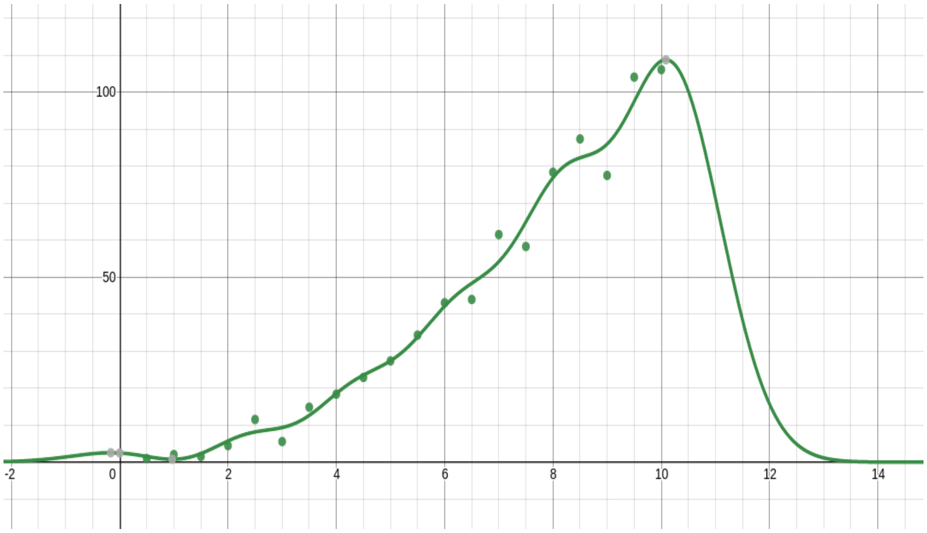
Gaussian Regression Curve 2



RMS Error	4.07566

Equation	$15.2095exp(-(x-0.875)^2/2) - 42.8338exp(-(x-1.75)^2/2) + 82.8373exp(-(x-2.625)^2/2) - 114.872exp(-(x-3.5)^2/2) + 163.202exp(-(x-4.375)^2/2) - 179.547exp(-(x-5.25)^2/2) + 233.581exp(-(x-6.125)^2/2) - 219.068exp(-(x-7)^2/2) + 262.351exp(-(x-7.875)^2/2) - 175.193exp(-(x-8.75)^2/2) + 173.128exp(-(x-9.625)^2/2)$
Fit	Good

Gaussian Regression Curve 3



RMS Error	3.88921
Equation	$9.14949exp(-(x-0.7)^2/2) - 13.2087exp(-(x-1.4)^2/2) - 5.31987exp(-(x-2.1)^2/2) + 52.3361exp(-(x-2.8)^2/2) - 85.9123exp(-(x-3.5)^2/2) + 97.9242exp(-(x-4.2)^2/2) - 46.5247exp(-(x-4.9)^2/2) + 7.22181exp(-(x-5.6)^2/2) + 43.3054exp(-(x-6.3)^2/2) + 22.7608exp(-(x-7)^2/2) - 97.0185exp(-(x-7.7)^2/2) + 259.776exp(-(x-8.4)^2/2) - 250.074exp(-(x-9.1)^2/2) + 213.914exp(-(x-9.8)^2/2)$
Fit	Good Fit

RESULT

The above analysis shows that given a dataset, multiple basis functions can prove to be a good fit. The criteria to always keep in mind is the number of dataset points, number of parameters, degree of the curve and the error generated. As we can observe, polynomial basis set seems to do relatively well in comparison to the other kinds of basis functions, which could possibly be due to the original function being (x^2+1) polynomial in nature. However the best fit in each of the three cases is quite close to each other which re-affirms the fact that for a given set of dataset basis functions are pretty capable in finding a good fitting curve within an acceptable error range.