Pattern Recognition and Machine Learning Curve-Fitting (Linear Regression)

Pavan Gurudath

January 28, 2018

Abstract

Curve fitting is the process of constructing a continuous curve onto a set of data points, that would best fit these data points under certain constraints. This curve that fits the data points allows a new data point to be interpolated based on this curve with the least possible error. In statistics, simple linear regression is a method that allows us to study the relationship between two continuous (quantitative) variables. Applications in which the training data comprises of input vectors and its corresponding output target value, then the learning is said to be Supervised Learning. In this project, we solve the linear regression problem by two different approaches namely direct error minimisation and Bayesian approach. In both the approaches, we solve the problem with and without a regularisation parameter to obtain *error minimisation, error minimisation with the regularisation term, the Maximal likelihood estimator of the Bayesian approach and the Maximum A-Posteriori estimator of the Bayesian approach.*

Contents

1 Introduction

The concept of linear regression refers to the fitting of a statistical model to describe how a particular variable can be used to predict another variable (the outcome or dependent variable). A linear regression model consists of two parts: the equation for the best-fitting line through the data and an error term representing the variation around the linear trend. The error term is minimised as much as possible so as to obtain the *best fit*. There are four methods of regression that have been used:

- **Regression using error minimisation:** The goal to fit, say N data points, is achieved by reducing the sum of the squared errors between the actual output and the estimated output using a closed form solution.

- **Regression using error minimisation with regularisation:** Regularisation is a technique that is used to control the over fitting phenomenon. A penalty term is added in order to limit the coefficients from oscillating between large values.

- **Regression using Maximum Likelihood Estimator:** MLE is a method of estimating the parameters of a statistical model given observations, by finding the parameter values that maximise the likelihood of making the observations given the parameters.

- **Regression using Maximum a posteriori estimation:** MAP employs a prior distribution, that quantifies the additional information available through prior knowledge of a related event, over the quantity that is to be estimated. MAP estimation can therefore be seen as a regularization of ML estimation.

2 Approach

2.1 Data

This project is executed upon a set of 50 input data points, that is generated from the $generateData.m$ matlab file. This file generates fifty linearly spaced points between 1 and 4π . The output is a sinusoidal wave having angular frequency ω equal to 0.5. The target vector is obtained by passing the ouptut vector through a noise model which is Gaussian in nature having zero mean (μ) and variance (σ^2) equal to 0.09. The data points are as shown in Fig 1.

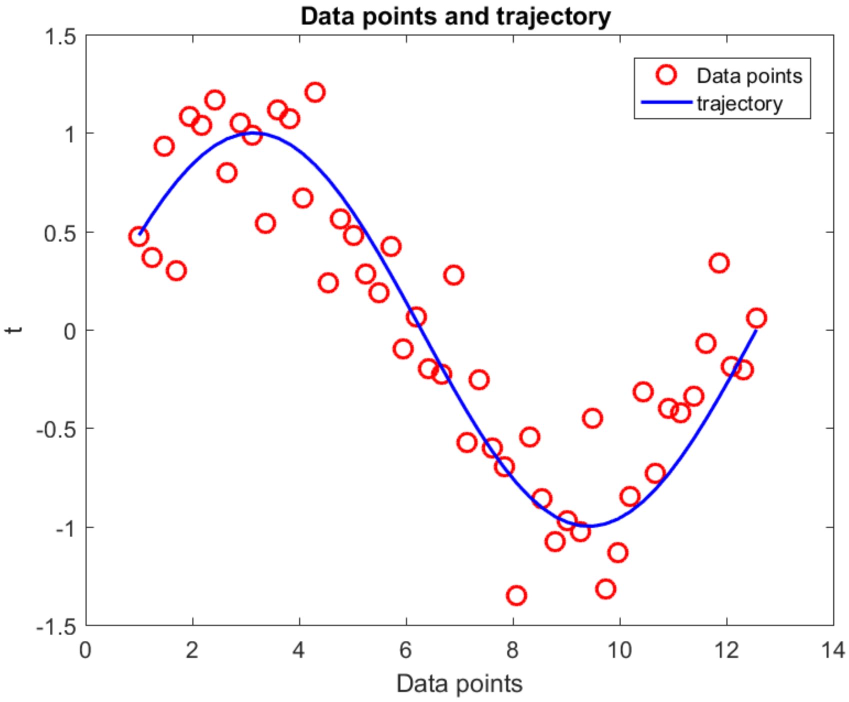


Figure 1: Data Points and 'Best Fit' trajectory

2.2 Methods

The main approach is to learn the weights such that the error function is minimized i.e., the average distances between the curve fit and the desired trajectory is minimal. In all the approaches, we have the input matrix, target vector and the weights (w) as follows:

$$\mathrm{X} = \begin{bmatrix} X_1^0 & X_1^1 & \cdots & X_1^M \\ X_2^0 & X_2^1 & \cdots & X_2^M \\ \vdots & \vdots & \ddots & \vdots \\ X_N^0 & X_N^1 & \cdots & X_N^M \end{bmatrix}, \mathrm{T} = \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_N \end{bmatrix}, \mathrm{W} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix}.$$

Using these parameters, we derive the conditions for each of the methods that are discussed in the following sections.

2.2.1 Regression with error minimization

This method is the direct approach of error minimization. The hypothesis function is an M^{th} order polynomial and it is linear in w. The equation 1 is an M^{th} order polynomial and is linear in w.

$$y(x, W) = w_0 + w_1x^1 + w_2x^2 + .. + w_Mx^M = \sum_{j=1}^{M} w_jx^j \tag{1}$$

Our error function can be computed as follows,

$$E(\mathbf{W}) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2 \tag{2}$$

Rewriting (1) in matrix form we get,

$$Y = XW \tag{3}$$

Since the main objective is to minimize the error function given in (2), we compute the derivative with respect to W and equate it to 0.

$$\begin{aligned}
&\frac{\partial E(\mathbf{W})}{\partial W} = 0\\
&\frac{\partial}{\partial W}\Big[(XW - T)^T(XW - T)\Big] = 0\\
\Rightarrow &\frac{\partial}{\partial W}\Big[(W^TX^T - T^T)(XW - T)\Big] = 0\\
\Rightarrow &\frac{\partial}{\partial W}\Big[(W^TX^TXW - W^TX^TT - T^TXW + T^TT\Big] = 0\\
\Rightarrow &\frac{\partial}{\partial W}\Big[(XW)^TXW - (T^TXW) - (T^TXW)^T + T^TT\Big] = 0
\end{aligned} \tag{4}$$

Using the following properties we simplify (4) to get,

$$\begin{aligned}
\frac{\partial x^Tx}{\partial x} &= 2x\\
\frac{\partial ax}{\partial a} &= x^T
\end{aligned} \tag{5}$$

$$\begin{aligned}
&\Rightarrow 2XWX^T - 2X^TT = 0\\
&\Rightarrow X^TXW = X^Tt
\end{aligned} \tag{6}$$

Thus, the solution of W is obtained that minimizes $E(W)$, and is given by equation 7.

$$W^* = (X^TX)^{-1}X^TT \tag{7}$$

2.2.2 Regression using error minimization with regularization

The hypothesis function in this method is similar to the one in Section 2.2.1, except that it contains a penalty term defined by λ. This term restricts the large oscillating values of the coefficients i.e. the weights and thereby it ensures the curve from over-fitting during training phase. Hence, the hypothesis function is as shown in the below equation.

$$E(W) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2 + \frac{\lambda}{2}\sum_{j=1}^{M}{w_j}^2 \tag{8}$$

$$E(W) = \frac{1}{2}\Big[(XW - T)^T(XW - T)\Big] + \frac{\lambda}{2}W^TW \tag{9}$$

In order to minimize our new modified error function (9), we compute the derivative of $E(W)$ in (9) with respect to W and set it to 0.

$$\begin{aligned}\frac{\partial E(W)}{\partial W} &= \frac{1}{2}\frac{\partial}{\partial W}\Big[(W^TX^T - T^T)(XW - T) + \lambda W^TW\Big] = 0\\ &\Rightarrow \frac{\partial}{\partial W}\Big[(W^TX^TXW - W^TX^TT - T^TXW + T^TT + \lambda W^TW\Big]\\ &\Rightarrow 2X^TXW - 2X^TT + 2\lambda W = 0\\ &\Rightarrow (X^TX + \lambda I)W = X^TT\end{aligned} \tag{10}$$

Thus, the solution of W is obtained that minimizes $E(W)$, and is given by equation 30.

$$W^* = (X^TX + \lambda I)^{-1}X^TT \tag{11}$$

Thus, we see that in using this approach, there is a dependency of $mathbfW^*$ on the data as well as the penalty factor λ. In this way, there is a control of the large oscillation of the values of the parameters.

2.2.3 Regression using Maximum Likelihood Estimation

Maximum likelihood estimation takes the Bayesian approach $i.e.$ it considers uncertainty in measuring the target values. It assumes that the target values are collected with a Gaussian noise model have mean μ and variance β^{-1}. The likelihood of the target value is a probability distribution given by equation 12.

$$P(t|x, w, \beta^{-1}) = N(t|y(x, w), \beta^{-1}) \tag{12}$$

It is assumed such that the mean of the Gaussian at each data point is the output of the hypothesis function, $y(x, w)$. Assuming that the points in the data are drawn independently, the joint likelihood function is given by equation 13.

$$P(T|X, W, \beta^{-1}) = \prod_{i=1}^{N} N(T|y(X, W), \beta^{-1}) = L, \tag{13}$$

The training data (X, T) is used to determine the parameters w and β by maximum likelihood. Equation 13 can be written as follows.

$$\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} exp\Big[\frac{-(x_i - \mu)^2}{(2\sigma^2)}\Big] \; where \; \sigma^2 = \beta^{-1} \tag{14}$$

Taking log on either sides, we have

$$\log(L) = \frac{N}{2}\log(\frac{1}{2\pi\sigma^2}) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{15}$$

$$\implies \log(L) = -\frac{N}{2}\log(2\pi) + \frac{N}{2}\log(\beta) - \frac{\beta}{2\pi}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{16}$$

To minimize the error, we maximize the log of likelihood by differentiating equation 16 with respect to μ,

$$\sum_{i=1}^{N} x_i - N\mu = 0 \implies \mu_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{17}$$

Differentiating 16 with respect to β, we get

$$\frac{1}{\beta} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{ML})^2 = \sigma_{ML}^2 \tag{18}$$

With the help of μ_{ML} and β, the target values can be predicted from unseen data using the likelihood probability distribution.

2.2.4 Regression using Maximum A Posteriori Estimation

Maximum A Posteriori Estimation takes the Bayesian approach as well. However, unlike the MLE approach, it considers the likelihood as well as prior distribution. Let us consider a Gaussian distribution of the form

$$P_r(\mathbf{w}|\alpha) = N(\mathbf{w}|\mathbf{0}, \alpha^{\mathbf{-1}}\mathbf{I}) = (\alpha/2\pi)^{m+1/2}\exp -\alpha/2w^T w \tag{19}$$

where α is the precision of the distribution and $M+1$ is the total number of elements in the vector w for an M^{th} order polynomial. Using Bayes' theorem, the posterior distribution for w is proportional to the product of the prior distribution and the likelihood function

$$P(w|X,t,\alpha,\beta) \propto P(t|X,w,\beta)P(w|\alpha) \tag{20}$$

α is known as the hyper parameter, and β is the variance parameter. Here, the weights are also assumed to be distributed as a Gaussian with zero mean and variance α. Expanding the two Gaussians in 20, we get

$$P(\mathbf{w}|\mathbf{X},\mathbf{t},\alpha,\beta) = \prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma^2}} exp\Big[\frac{-(x_i-\mu)^2}{(2\sigma^2)}\Big]\ \Big[\frac{\alpha}{2\pi}^{(\frac{M+1}{2})}\Big] exp\Big[-\frac{\alpha}{2}w^T w\Big] \tag{21}$$

$$= \Big(\frac{1}{2\pi\sigma^2}\Big)^{\frac{N}{2}}\Big(\frac{\alpha}{2\pi}\Big)^{\frac{M+1}{2}} exp\Big(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2 - \frac{\alpha}{2}w^T w\Big) \tag{22}$$

It is more feasible to take log on either sides and maximize or minmize the negative log. Therefore, taking log on the either sides, we get

$$= \frac{N}{2}log(\frac{1}{2\pi}) + \frac{\mathbf{N}}{2}log(\beta) + \Big(\frac{M+1}{2}log(\alpha)\Big) - \Big(\frac{M+1}{2}\Big)log(2\pi) - \frac{\beta}{2}\sum_{i=1} N(x_i-\mu)^2 - \frac{\alpha}{2}w^T w \tag{23}$$

Differentiating equation 23 w.r.t β and equating it to zero, we get

$$\implies \frac{1}{\beta} = \frac{1}{N}\sum_{i=1} N(x_i-\mu)^2 \tag{24}$$

Similarly, differentiating equation 23 with respect to α and equating it to zero, we get

$$\frac{M+1}{2\alpha} - \frac{1}{2}w^T w = 0 \tag{25}$$

$$\implies \frac{1}{\alpha} = \frac{1}{M+1}w^T w \tag{26}$$

Differentiating equation 23 with respect to μ and equating it to zero, we get

$$-\frac{\beta}{2}2\sum_{i=1}^{N}(x_i - \mu)(-1) = 0 \tag{27}$$

$$\implies \mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{28}$$

Using equations 24, 26, 28 we get the maximum posterior error function,

$$E(\mathbf{w}) = \frac{\beta}{2}\Big(\sum_{i=1}^{N} y(x_n, w) - t_n\Big)^2 + \frac{\alpha}{2}w^T w \tag{29}$$

$$\begin{aligned} E(\mathbf{w}) =& \frac{\beta}{2}(XW - T)^T(XW - T) + \frac{\alpha}{2}w^T w & (30)\\ =& \frac{\beta}{2}(W^T X^T - T^T)(XW - T) + \frac{\alpha}{2}w^T w & (31)\\ =& \frac{\beta}{2}(W^T X^T XW - W^T X^T T - t^T XW + T^T) + \frac{\alpha}{2}w^T w & (32) \end{aligned}$$

Differentiating equation 32 with respect to W and equating it to 0, we get

$$\begin{aligned} &\implies \frac{\partial \mathbf{E}(\mathbf{w})}{\partial W} = 0 & (33)\\ &\implies \frac{\beta}{2}(2X^T XW - 2X^T T) + \alpha W = 0 & (34)\\ &\implies \frac{\beta}{2}(W^T X^T - T^T)(XW - T) + \frac{\alpha}{2}w^T w = 0 & (35)\\ &\implies \beta X^T XW - \beta X^T T + \alpha W = 0 & (36)\\ & & (37) \end{aligned}$$

Therefore, we have

$$\mathbf{W}^* = (\beta X^T X + \alpha I)^{-1}\beta X^T T \tag{38}$$

and it can be seen that there is a dependency upon both α and β. Therefore, by chosing an appropriate α it is possible to obtain a better estimate.

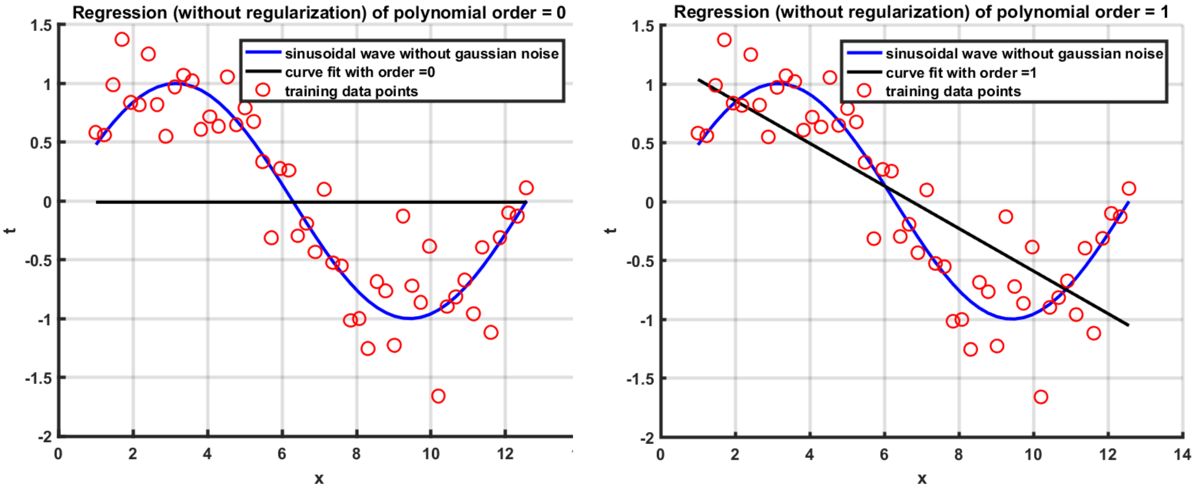
3 Results

This section details the results thus obtained after performing linear regression using the aforementioned methods. The plots shown below explains the performance of each of these methods and helps us analyse them. The four subsections contains the plots for the order of polynomials 0, 1, 3, 6, 9 and 20.

The extra credits have been organized as follows:
a) The plot for ln λ = -18 and -13 have been included in section 3.2 itself.
b) The plot for 200 data points have been included in section 3.1 and 3.3. However, the plot for the topics in section 3.2 and 3.4 have been executed in MATLAB and their images are saved in the working directory.
c) Table for a fixed degree of polynomial have been generated in section 3.5 for Error Minimization without regularization and with regularization.

3.1 Error Minimization without regularization

The proof for this method mentioned in Section 2.2.1 is validated by the plots as shown below. The curves for the order 0, 1, 3, 6, 9 and 20 have been plotted for a set of 50 data points in Fig 2, 3 and 4 respectively. The curves have been plotted for the same orders by increasing the number of data points from 50 to 200. This can be observed in Fig 5, 6 and 7 respectively.



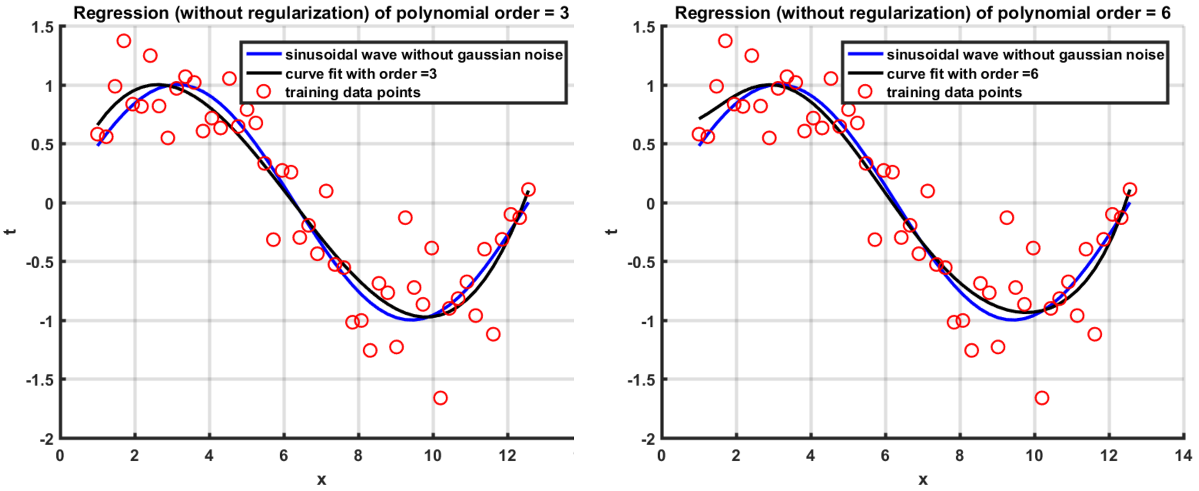
(a) order = 0

(b) order = 1

Figure 2: Curve fitting for $M = 0$ and $M = 1$ and 50 data points

It can be observed in Fig 2a and Fig 2b that the curve fitting with order 0 and 1 are poor. The difference between the curve fitting line(in yellow) and the trajectory(in blue) is large and hence the error has not been minimized completely. This observation is obvious since the set of data points is sinusoidal which is non-linear and using order as 0 and 1 is linear.

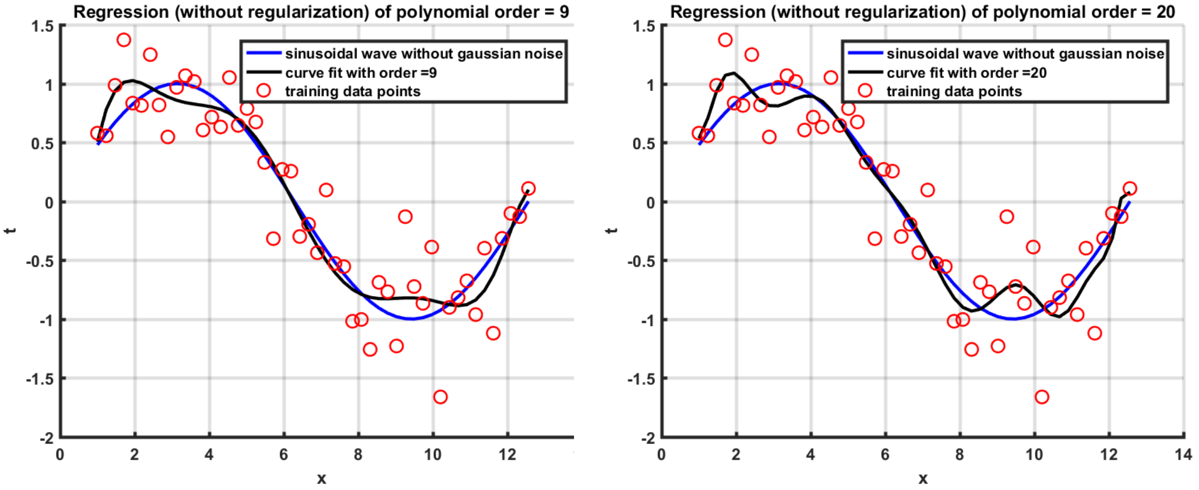
In Fig 3a and Fig 3b, as the order has increased by a considerate number of 3 and 6, it can be seen that the new curve fitting has improved and it fits the data points much better. Now as the order is further increased, it so happens that the curve fits the data points more accurately than ever as shown in Fig 4a and Fig 4b. However, this is not favoured since while the curve fits the training points better, it may not do so for the new data points that the model is to be tested upon and thereby the curve does not provide a good interpolation for new data points that the model has not been trained upon. This phenomenon of fitting the training data points exactly is known as over-fitting. It is to be ensured that while choosing parameters, the model should not result in overfitting the training data.



(a) order = 3

(b) order = 6

Figure 3: Curve fitting for $M = 3$ and $M = 6$ and 50 data points

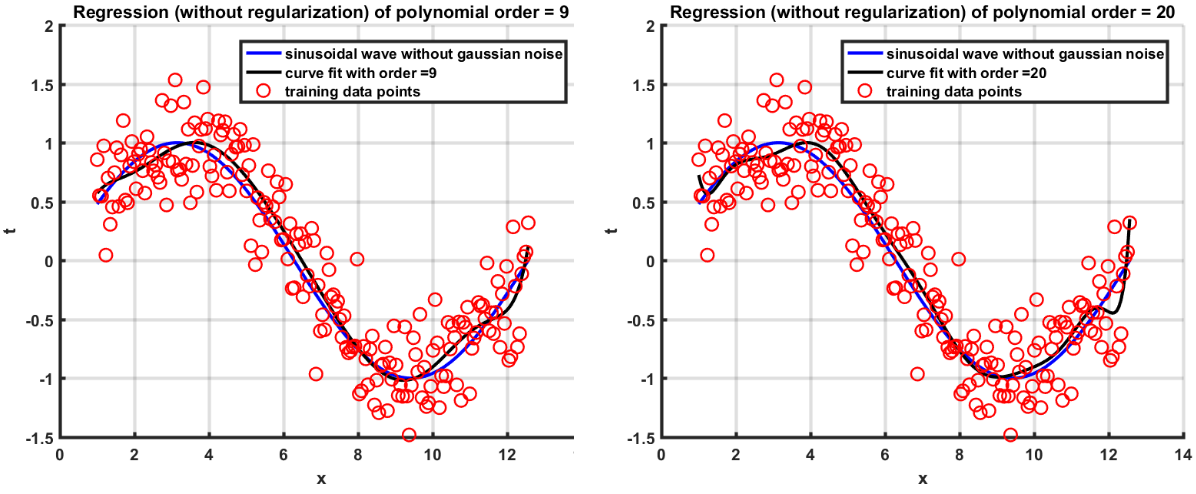


(a) order = 9

(b) order = 20

Figure 4: Curve fitting for $M = 9$ and $M = 20$ and 50 data points

However, if we increase the number of data points, then we can see from Fig 5, the curve fits decently as the order is comparatively small in comparison to the number of data points. This can be seen with the best fit curve(in black) is similar to the sinusoidal trajectory(in blue).

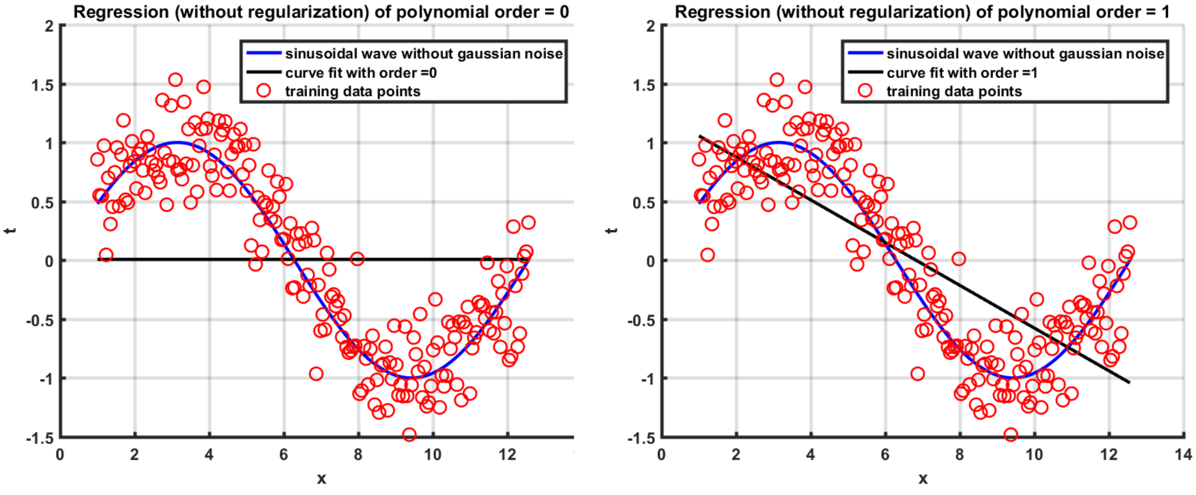


(a) order = 9

(b) order = 20

Figure 5: Curve fitting for $M = 9$ and $M = 20$ and 200 data points

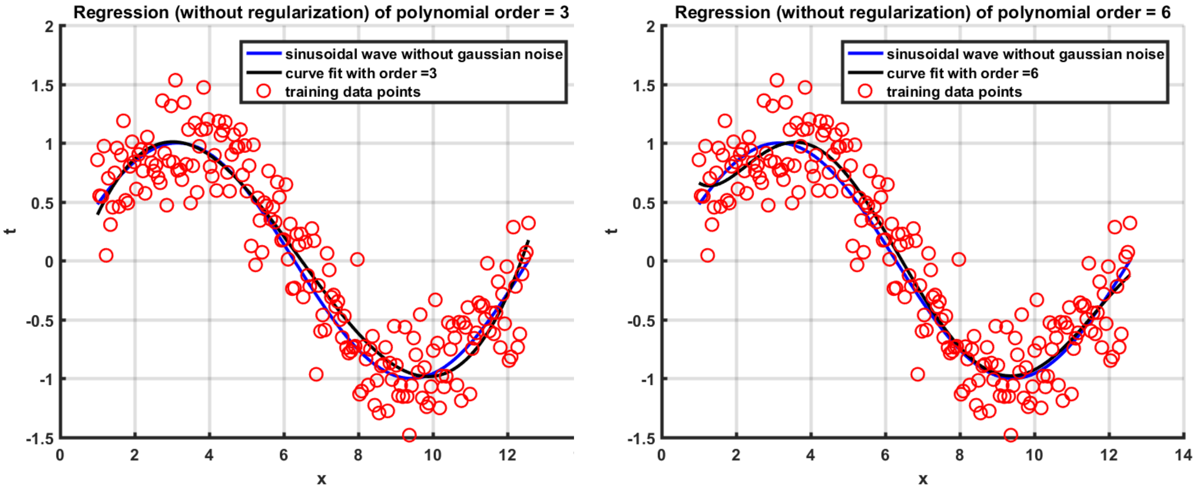
The plots for orders with M = 0, 1, 3 and 6 are shown in Fig 6 and 7



(a) order = 0

(b) order = 1

Figure 6: Curve fitting for $M = 0$ and $M = 1$ and 200 data points



(a) order = 3

(b) order = 6

Figure 7: Curve fitting for $M = 3$ and $M = 6$ and 200 data points

The plot of RMS error against order of the polynomial for 50 points is as shown in Fig 8.

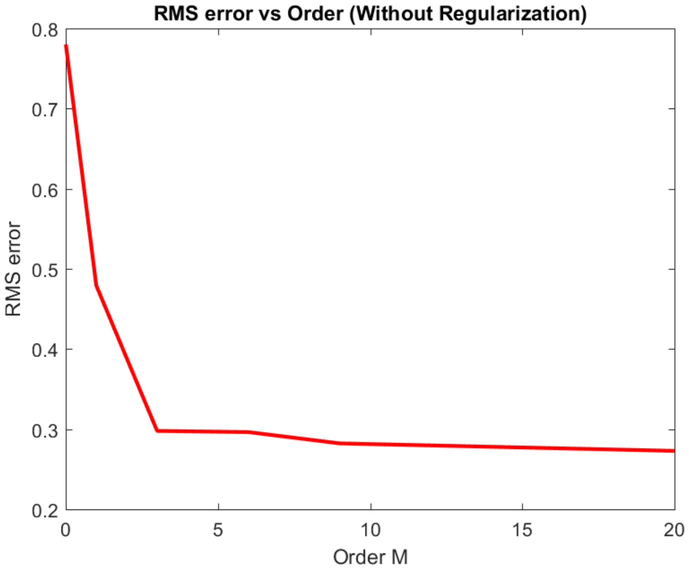
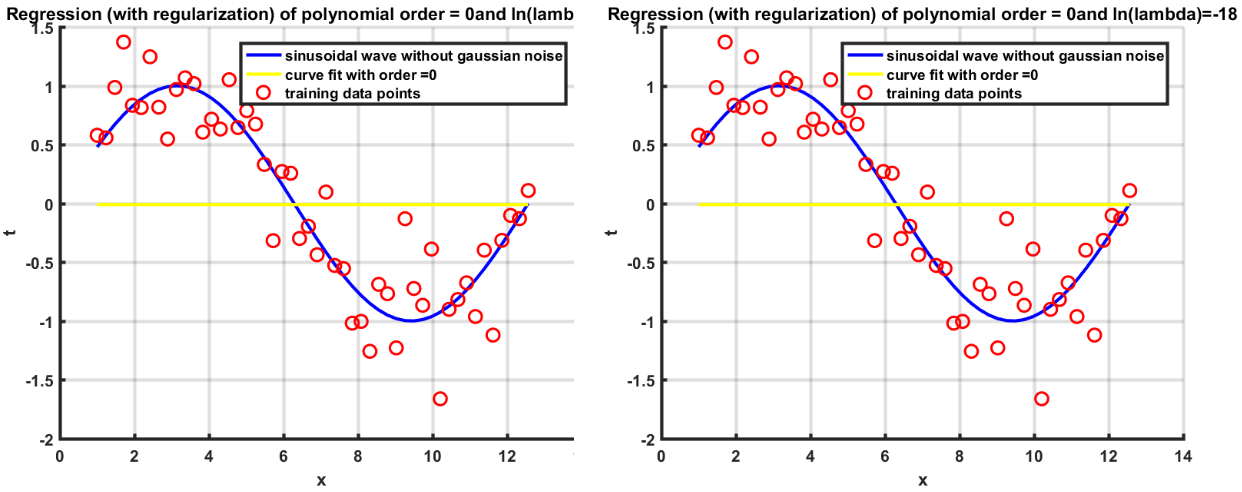


Figure 8: Graph of root-mean-square error vs order of polynomial

This graph is true only for the case of training phase.The decease in error for increase in order of polynomial is due to over-fitting. However, on an unknown test set, this regression model would not perform well.

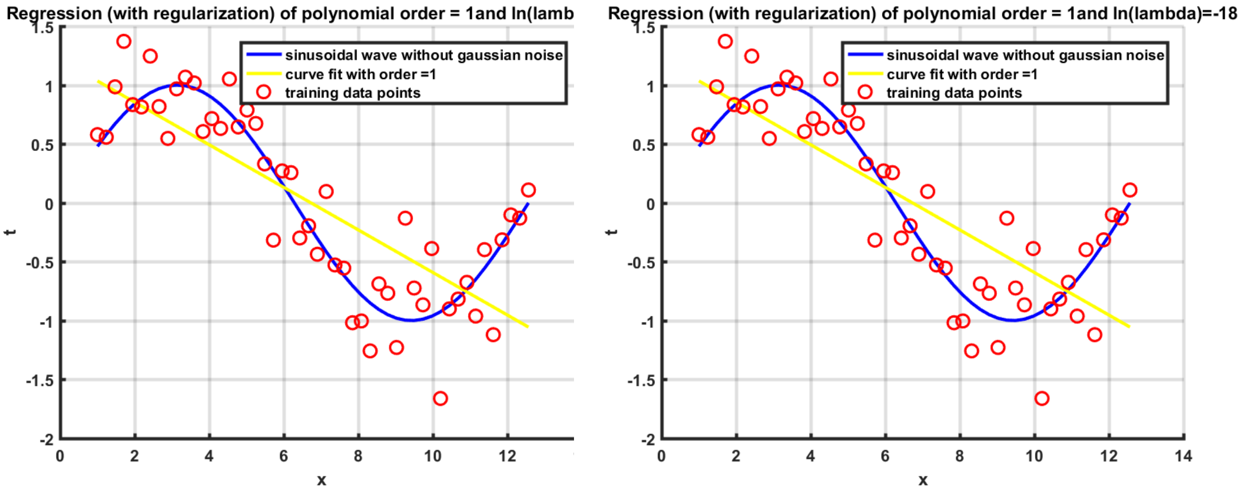
3.2 Error Minimization with regularization

The proof for this method mentioned in Section 2.2.2 is validated by the plots as shown below. The curves for the order 0, 1, 3, 6, 9 and 20 have been executed for a set of 50 data points for $ln\lambda$ equal to -18, -15 and -13. However, for the case of clarity, the figures have been plotted for the order 0, 1, 3, 6 and 9 for $ln\lambda$ equal to -18 and -13. The figures are plotted in Fig 9, 10, 11, 12, 13 and 14 respectively with respect to the aforementioned orders of the polynomial. A comparative plot can be observed for $ln\lambda$ equal to -18 and -13 in each of these figures.



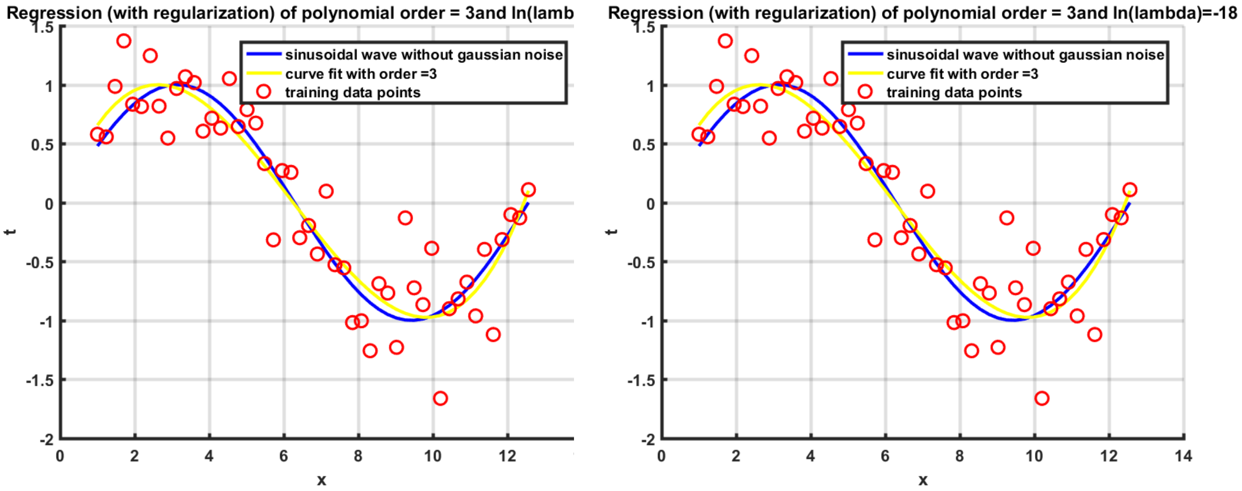
(a) $\ln\,\lambda = -13$ (b) $\ln\,\lambda = -18$

Figure 9: Curve fitting for $M = 0$ and 50 data points



(a) $\ln\,\lambda = -13$ (b) $\ln\,\lambda = -18$

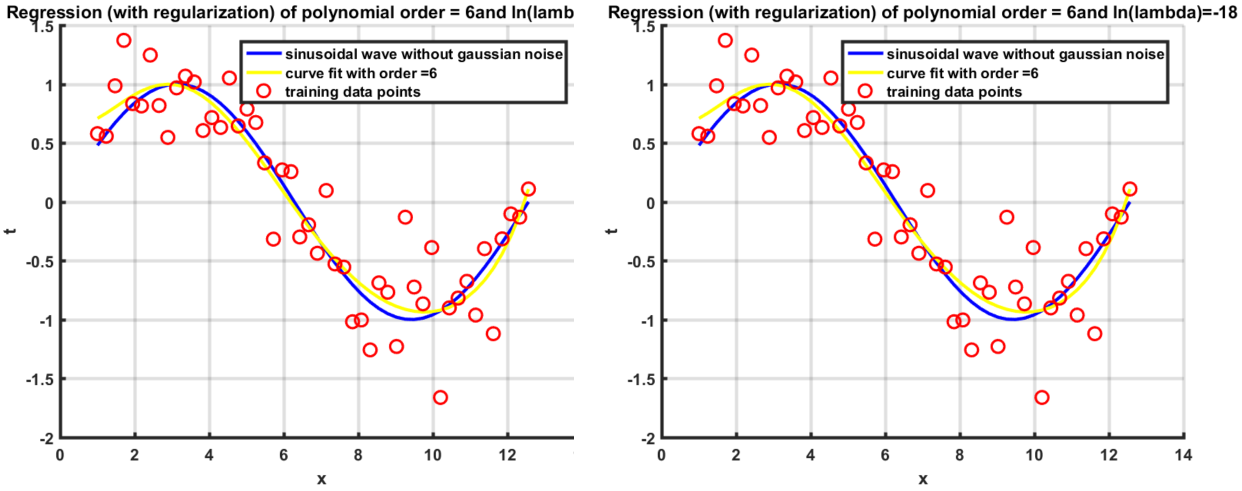
Figure 10: Curve fitting for $M = 1$ and 50 data points



(a) $\ln\lambda = -13$

(b) $\ln\lambda = -18$

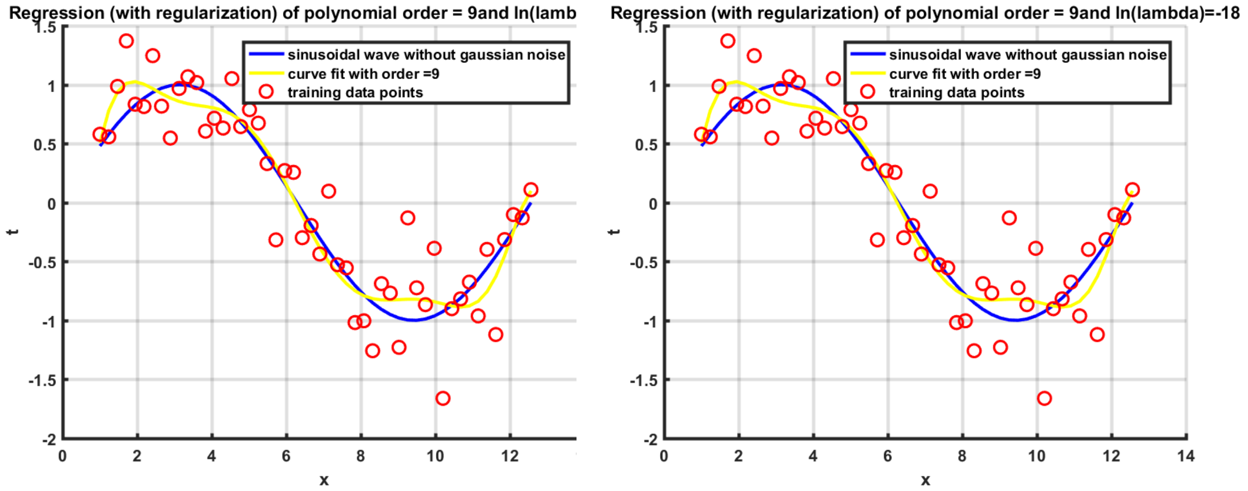
Figure 11: Curve fitting for $M = 3$ and 50 data points



(a) $\ln\lambda = -13$

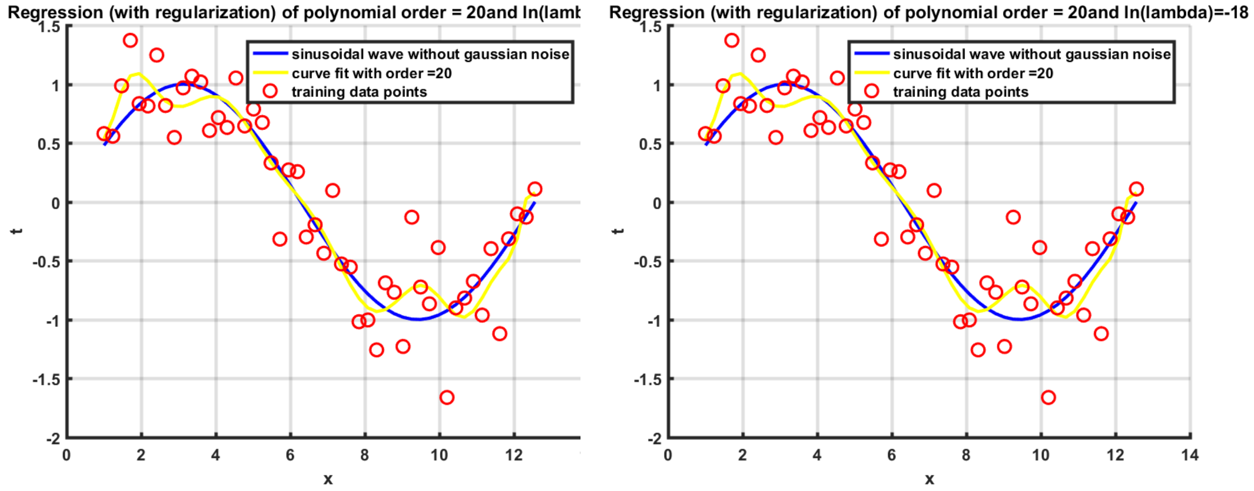
(b) $\ln\lambda = -18$

Figure 12: Curve fitting for $M = 6$ and 50 data points



(a) ln $\lambda = -13$ (b) ln $\lambda = -18$

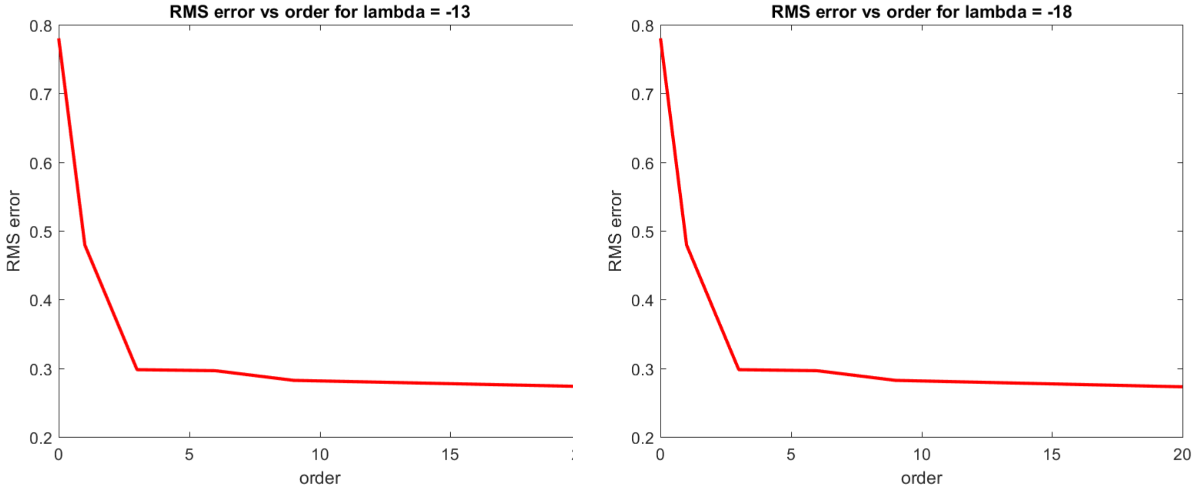
Figure 13: Curve fitting for $M = 9$ and 50 data points



(a) ln $\lambda = -13$ (b) ln $\lambda = -18$

Figure 14: Curve fitting for $M = 20$ and 50 data points

There isn't much difference between $ln\lambda$ equal to -18 and -13 but as $ln\lambda$ is changed from 0 to more negative values, there is a considerable amount of difference. But if the figures 14 and 4b were to be compared, then it can be seen that there is a significant control of the shape of the graph. Therefore, this method of using a regularization parameter helps in controlling the phenomenon of over-fitting. The plot of root-mean-square error against order of the polynomial has been plotted in Fig 15a and 15b for their $ln\lambda$.



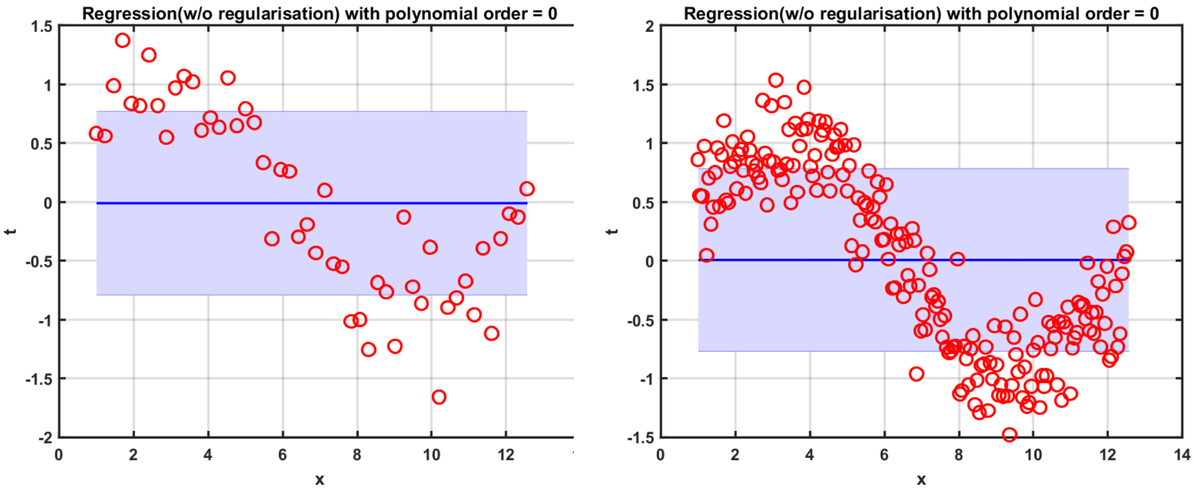
(a) $\ln\,\lambda = -13$

(b) $\ln\,\lambda = -18$

Figure 15: Graph of root-mean-square error vs order of polynomial

3.3 Regression using Maximum Likelihood Estimation

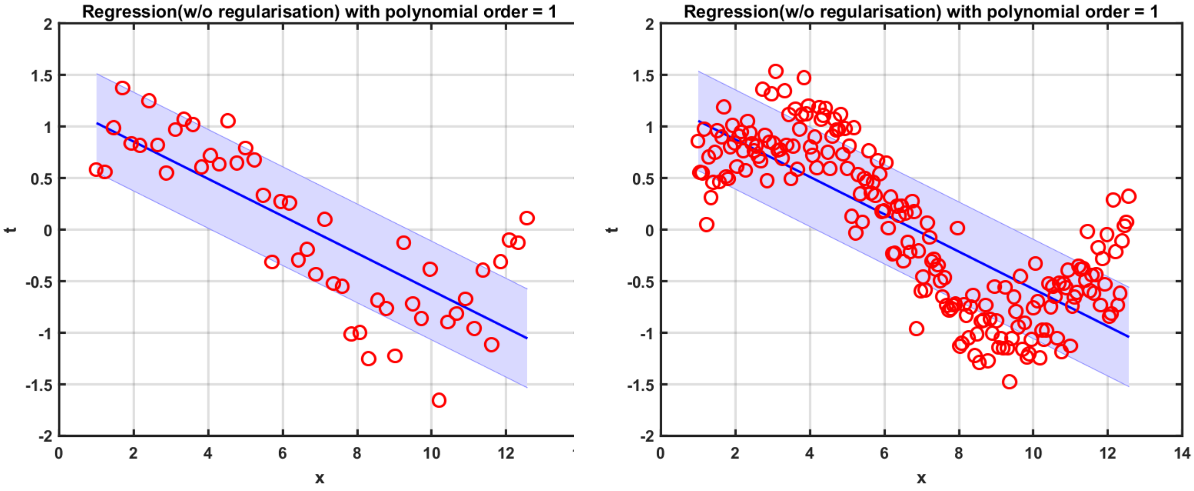
The proof for this method mentioned in Section 2.2.3 is validated by the plots as shown below. The curves for the order 0, 1, 3, 6, 9 and 20 have been executed for a set of 50 and 200 data points. The figures are shown in Fig 9, 10, 11, 12, 13 and 14 respectively with respect to the aforementioned orders of the polynomial. A comparative plot can be observed for 50 and 200 data points in each of these figures.



(a) 50 data points

(b) 200 data points

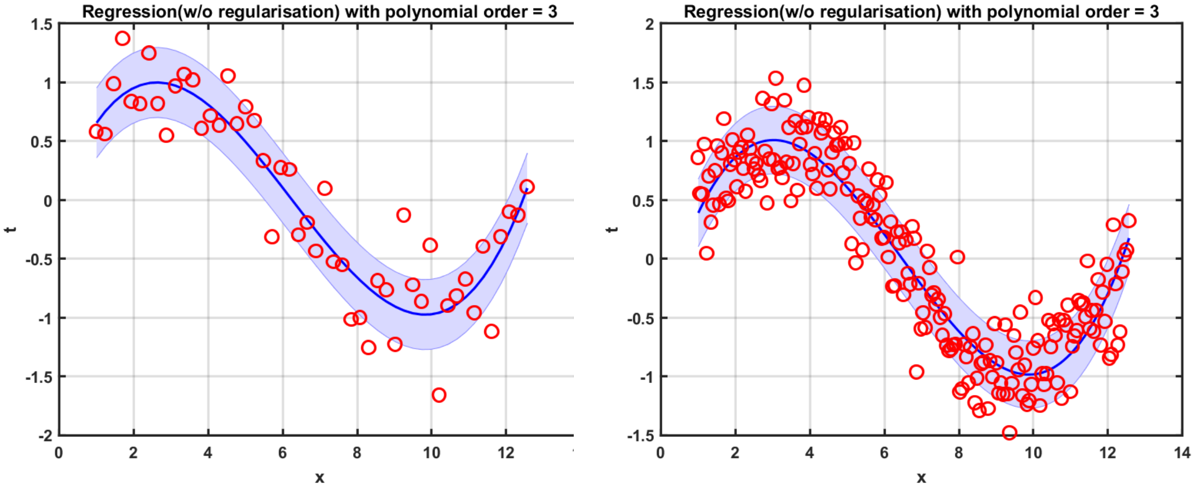
Figure 16: Curve fitting for order of polynomial = 0



(a) 50 data points

(b) 200 data points

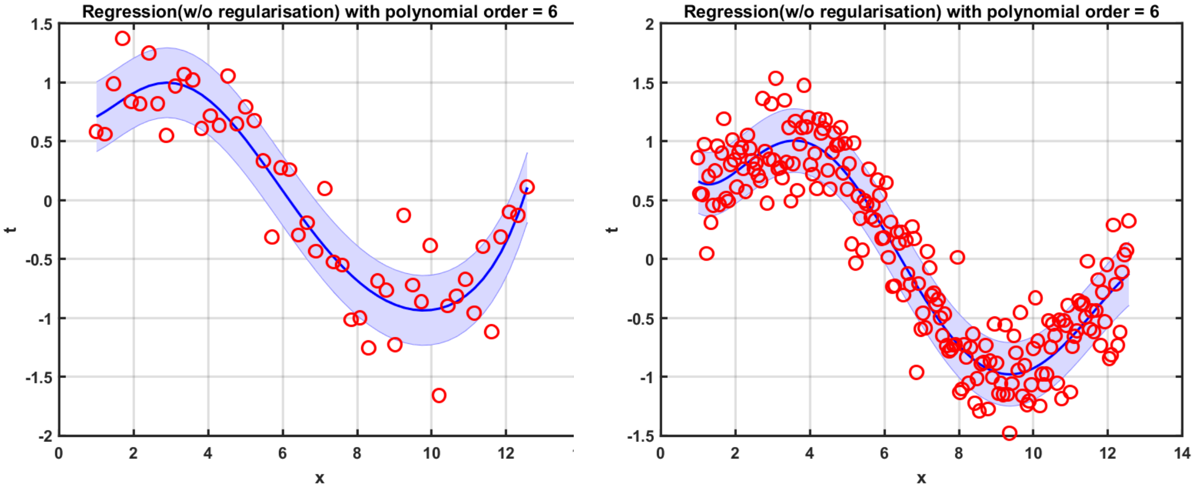
Figure 17: Curve fitting for order of polynomial = 1



(a) 50 data points

(b) 200 data points

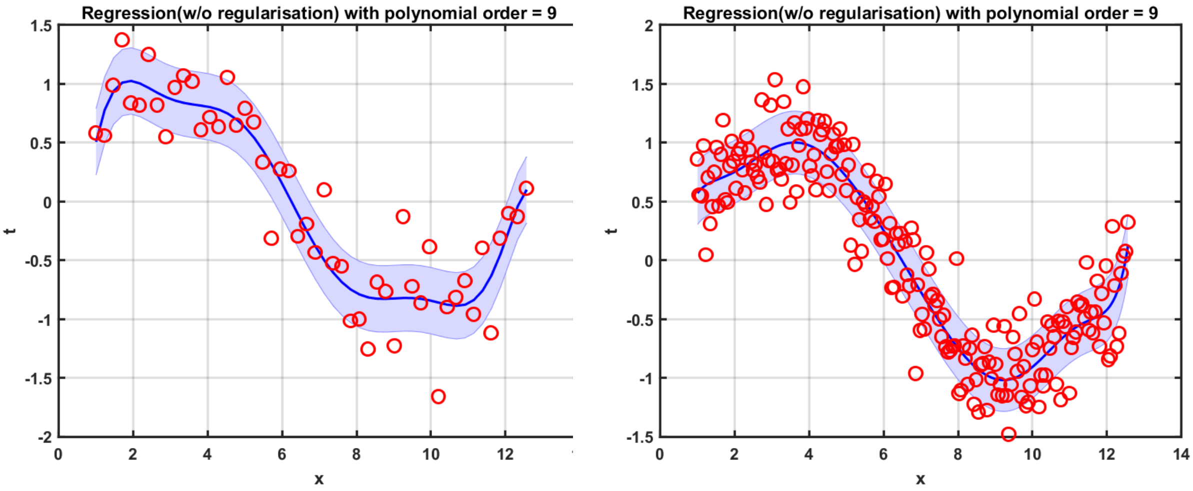
Figure 18: Curve fitting for order of polynomial = 3



(a) 50 data points

(b) 200 data points

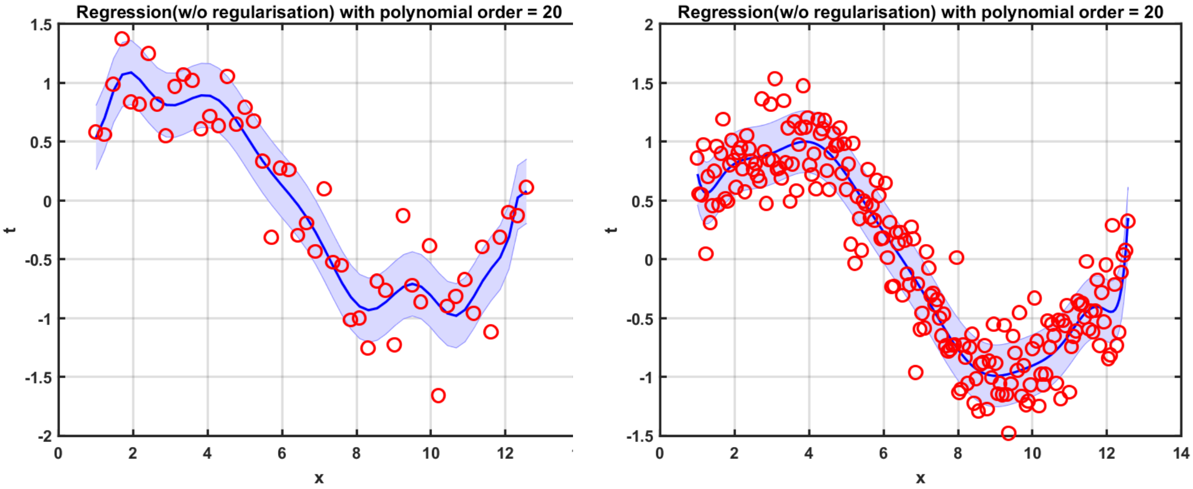
Figure 19: Curve fitting for order of polynomial = 6



(a) 50 data points

(b) 200 data points

Figure 20: Curve fitting for order of polynomial = 9



(a) 50 data points

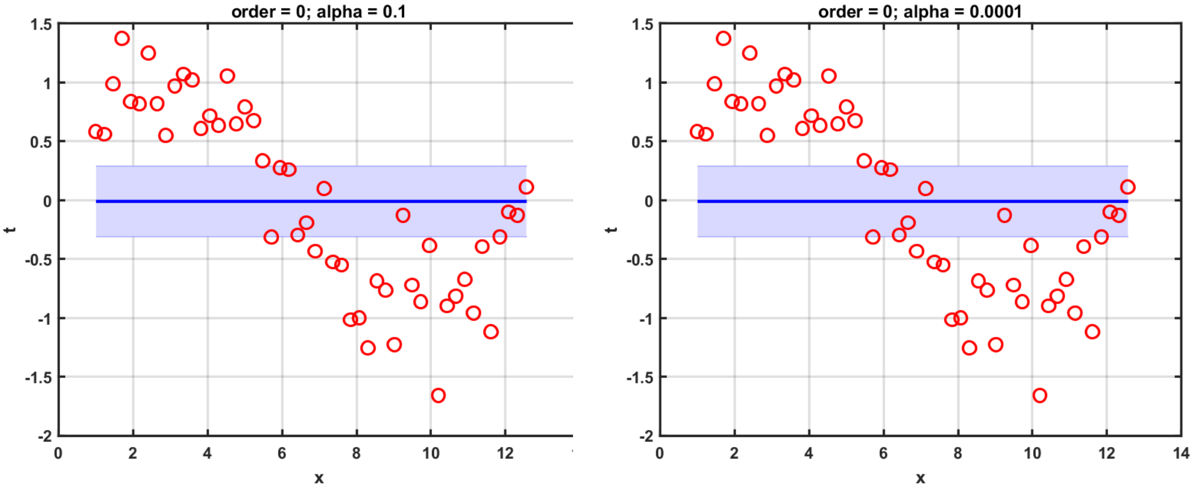
(b) 200 data points

Figure 21: Curve fitting for order of polynomial = 20

It can be observed that the plots obtained here are similar in comparison to the plots in section 3.1. However the root mean square errors have been plotted only for a training data and have not considered a test data set. If that were to be considered, then there would be a significant efficiency that can be seen in the method of MLE in comparison to the error minimization using the direct approach without the regularization parameter.

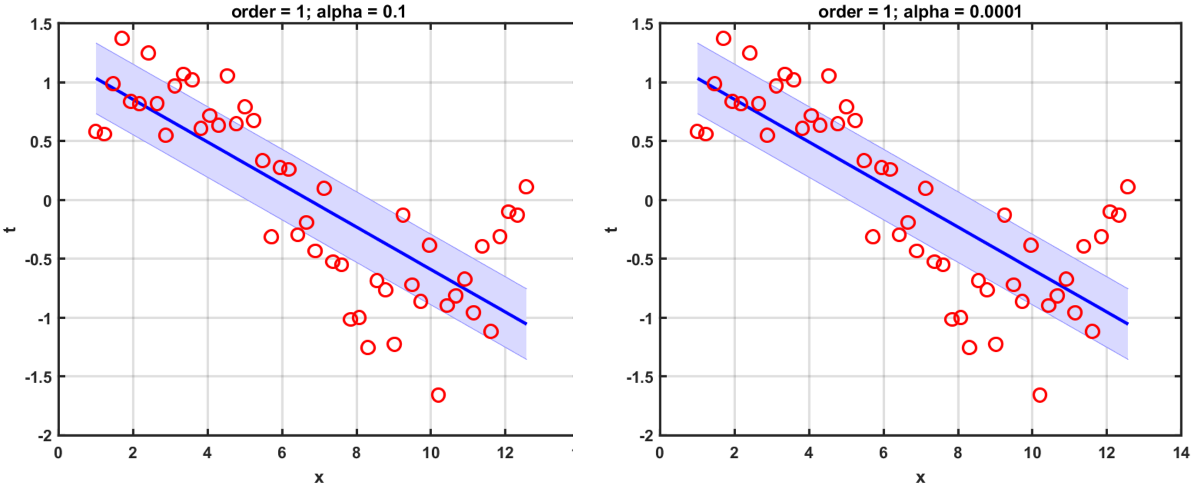
3.4 Regression using Maximum A Posteriori Estimation

The proof for this method mentioned in Section 2.2.4 is validated by the plots as shown below. The curves for the order 0, 1, 3, 6, 9 and 20 have been executed for a set of 50 data points. Values of α equal to $5 \times 10^{-3}, 10^{-4}$ and 10^{-1} have been used and executed in the MATLAB program. For convenience, only α equal to 10^{-4} and 10^{-1} have been shown in Fig 22, 23, 24, 25, 26 and 27 corrosponding to the order of polynomial 0, 1, 3, 6, 9 and 20 respectively.



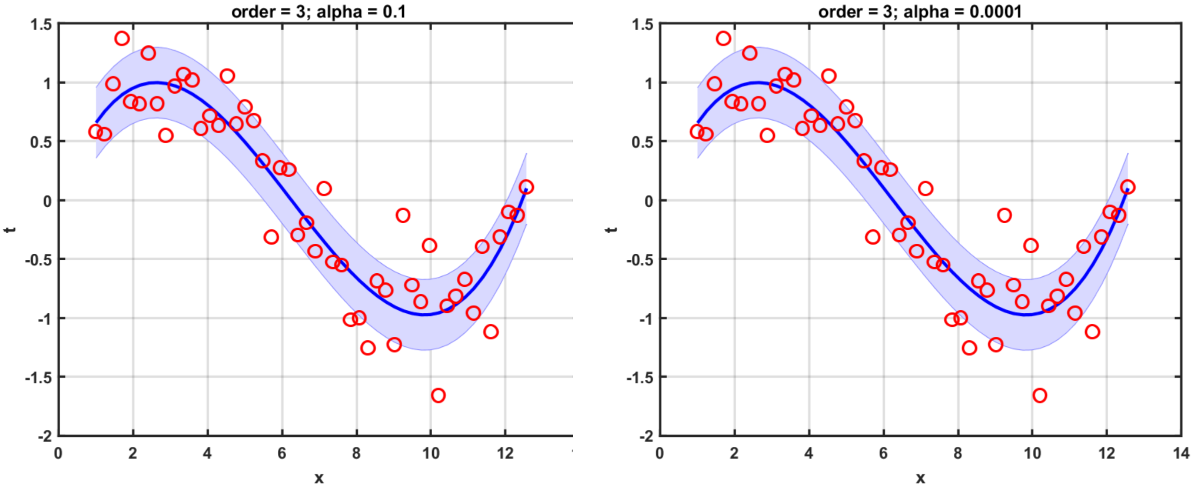
(a) $\alpha = 10^{-1}$ (b) $\alpha = 10^{-4}$

Figure 22: Curve fitting for order of polynomial = 0



(a) $\alpha = 10^{-1}$ (b) $\alpha = 10^{-4}$

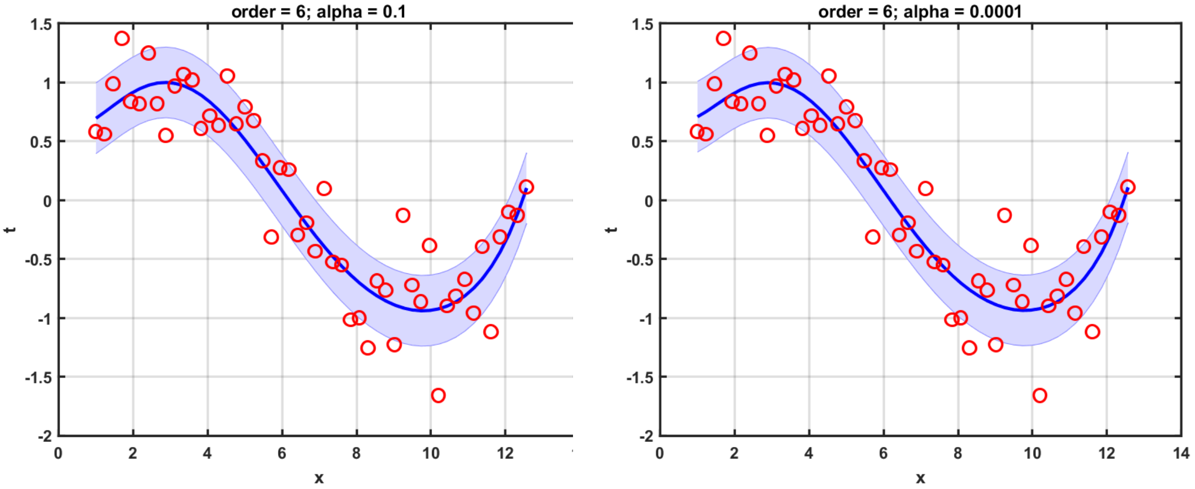
Figure 23: Curve fitting for order of polynomial = 1



(a) $\alpha = 10^{-1}$

(b) $\alpha = 10^{-4}$

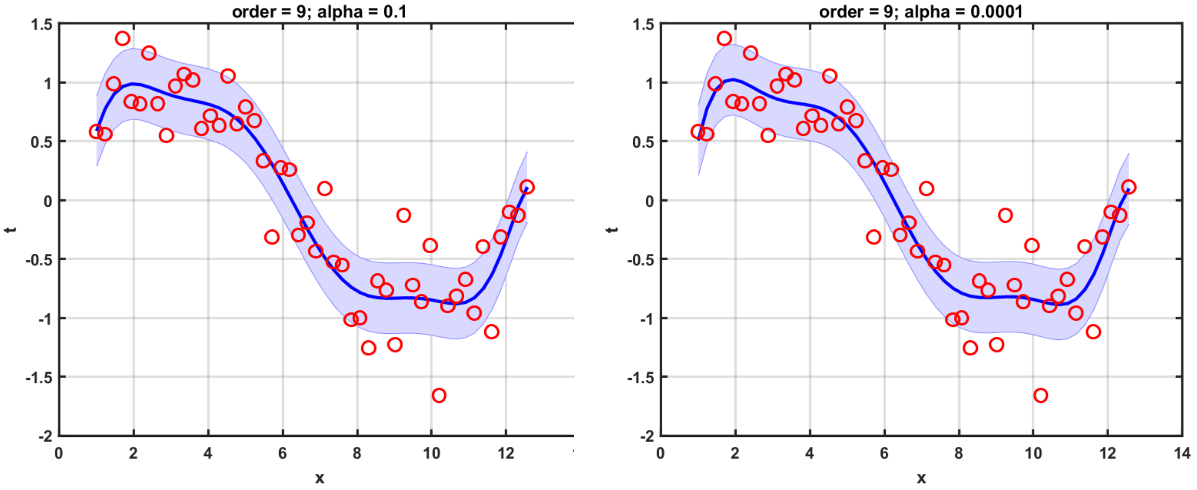
Figure 24: Curve fitting for order of polynomial = 3



(a) $\alpha = 10^{-1}$

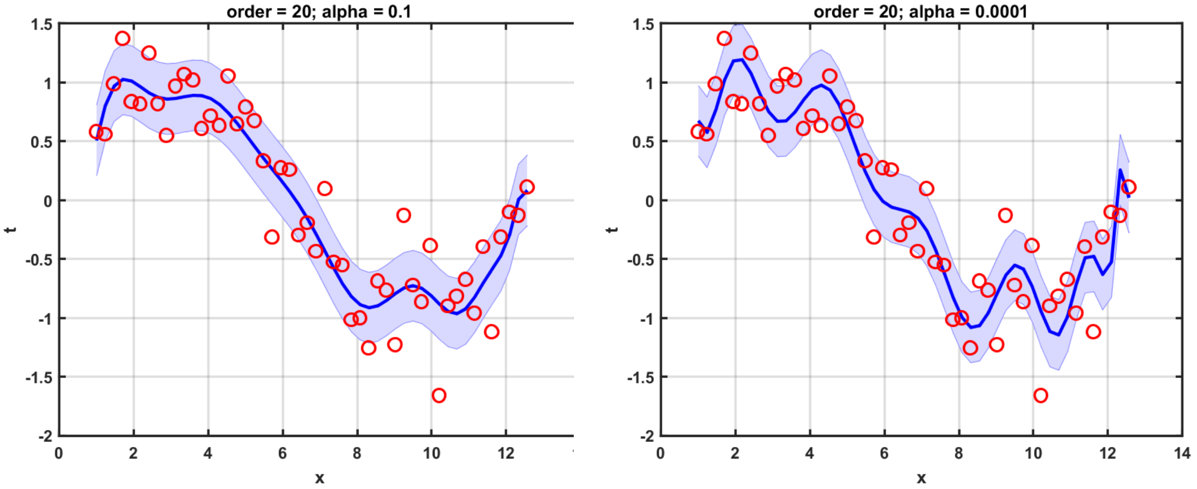
(b) $\alpha = 10^{-4}$

Figure 25: Curve fitting for order of polynomial = 6



(a) $\alpha = 10^{-1}$ (b) $\alpha = 10^{-4}$

Figure 26: Curve fitting for order of polynomial = 9



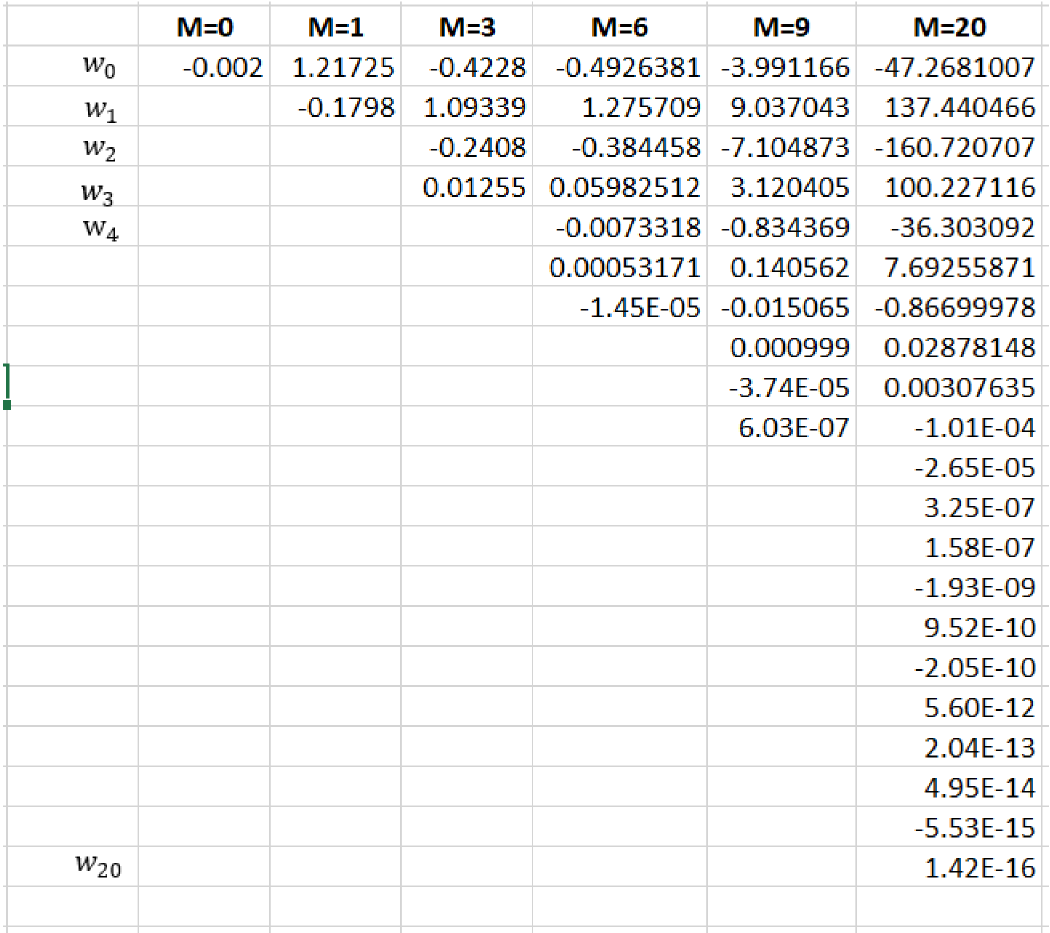
(a) $\alpha = 10^{-1}$ (b) $\alpha = 10^{-4}$

Figure 27: Curve fitting for order of polynomial = 20

It can be seen that this method is considered to be the best amongst the aforementioned methods. For an order of polynomial 20, it can be seen that the figure 27 in comparison to 21a, 14 and 4b seems to be the best fit for a collection of 50 dataset points as the curve fits efficiently without any indication of over fitting the dataset.

3.5 Extras

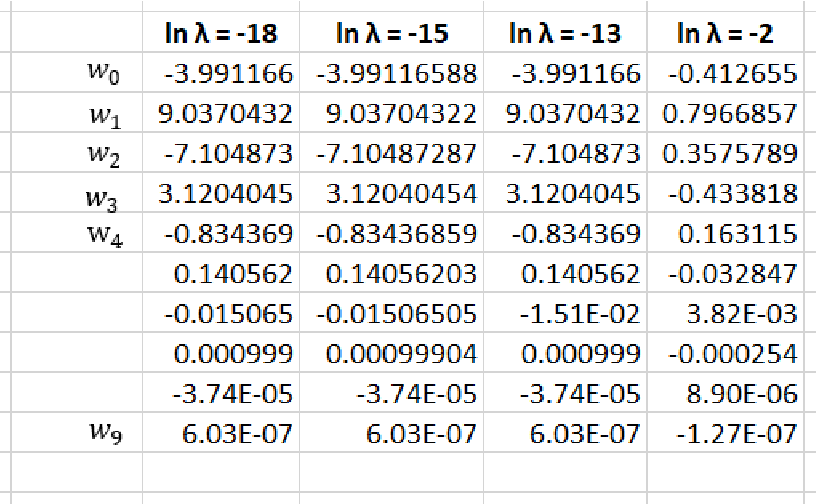
a) The table for a varying degree of polynomial has been generated for Error minmization without regularization and is as shown in fig 28.



	M=0	M=1	M=3	M=6	M=9	M=20
w_0	-0.002	1.21725	-0.4228	-0.4926381	-3.991166	-47.2681007
w_1		-0.1798	1.09339	1.275709	9.037043	137.440466
w_2			-0.2408	-0.384458	-7.104873	-160.720707
w_3			0.01255	0.05982512	3.120405	100.227116
w_4				-0.0073318	-0.834369	-36.303092
				0.00053171	0.140562	7.69255871
				-1.45E-05	-0.015065	-0.86699978
					0.000999	0.02878148
					-3.74E-05	0.00307635
					6.03E-07	-1.01E-04
						-2.65E-05
						3.25E-07
						1.58E-07
						-1.93E-09
						9.52E-10
						-2.05E-10
						5.60E-12
						2.04E-13
						4.95E-14
						-5.53E-15
w_{20}						1.42E-16

Figure 28: Table of the coefficients W^* for polynomials of various order

The table for a fixed degree of polynomial but with different values of the regularization parameter λ and is as shown in fig 29



	ln λ = -18	ln λ = -15	ln λ = -13	ln λ = -2
w_0	-3.991166	-3.99116588	-3.991166	-0.412655
w_1	9.0370432	9.03704322	9.0370432	0.7966857
w_2	-7.104873	-7.10487287	-7.104873	0.3575789
w_3	3.1204045	3.12040454	3.1204045	-0.433818
w_4	-0.834369	-0.83436859	-0.834369	0.163115
	0.140562	0.14056203	0.140562	-0.032847
	-0.015065	-0.01506505	-1.51E-02	3.82E-03
	0.000999	0.00099904	0.000999	-0.000254
	-3.74E-05	-3.74E-05	-3.74E-05	8.90E-06
w_9	6.03E-07	6.03E-07	6.03E-07	-1.27E-07

Figure 29: Table of the coefficients W^* for M = 9 polynomials with various values for the regularization parameter λ

4 Conclusion

Using the results obtained by performing this project, certain conclusions can be made. By introducing a regularization parameter λ, the phenomenon of over-fitting the data by a learning model can be avoided. Using this parameter, there is a better control over the entire model. The variance factor of the weights increases with the order of the polynomial as observed in figure 28. It is possible to overcome the over fitting phenomenon by increasing the number of training data points.

Another interesting observation that can be done is that the Maximum A Posteriori estimation is similar to the Maximal likelihood estimation with a regularization factor of $\frac{\alpha}{\beta}$. Finally, amongst all the methods tried in this project, it can be concluded that the MAximum Posteriori Estimation performs with the highest efficiency in minimizing the error thereby having the least mean square error. It is so, since this approach is not solely dependent upon the data but also the priori.

References

[1] Bishop, Christopher M. *Pattern Recognition and Machine Learning.* 2006.

[2] Prince, Simon JD *Computer vision: models, learning, and inference.* 2012