Pattern Recognition and Machine Learning (EE552) Project 1

Savinay Nagendra (sxn265)

January 28, 2018

Abstract

Supervised learning problems are formulated in applications where the training data comprises of input vectors along with their corresponding target vectors. If the desired vector consists of continuous variables, then the process is called Regression. Regression algorithms are generally used to predict a target value t from an observed data x. In this report, four methods of linear regression have been explored in the context of curve fitting, on a synthetically generated data set. The first two methods deal with **regression with and without regularization**, with the assumption of no uncertainty over the value of target variable. The latter methods, namely, **Maximum Likelihood Estimation** and **Maximum a Priori** Estimation take the Bayesian approach. They consider uncertainty over the value of target variable using probability distributions. The methods have been compared based on their performance efficiency, i.e., their ability to fit the given data accurately with least constraints.

Contents

1 Introduction

Regression is a supervised machine learning algorithm, used to predict new target values from unseen data. The ability of a regression algorithm to predict data with high accuracy is called generalization[1]. In training phase, the coefficients (weights) of a hypothesis function are learnt using training data. This is achieved by minimizing the error between the desired (target) values and the output of the hypothesis function. Performance of different regression algorithms are validated based on their degree of generalization[2]. Training data has been generated synthetically so that the precise process that generated the data is known for comparison against any learned model. Four methods of linear regression have been used to approach a curve fitting problem:
1) **Regression using error minimization**:
The hypothesis function is trained to minimize error between the desired trajectory and the actual output using a closed form solution.
2) **Regression using error minimization with regularization**:
A penalty term is added to the error function to inhibit the coefficients of the hypothesis function from reaching large values. This method is also called ridge regression.
3) **Regression using Maximum Likelihood Estimation (MLE)**:
Unlike the first two methods which assumed no uncertainty over target values, MLE assumes a Gaussian likelihood over the target values, given the data. The parameters of the Gaussian are thus estimated my maximizing the likelihood or conversely, minimizing the error function.
4) **Regression using Maximum a Priori Estimation (MAP)**:
MAP is analogous to MLE with regularization. A prior probability of coefficients of the hypothesis function is assumed to be a zero mean Gaussian with some variance. The distribution of weights given the data, also called the posterior distribution is calculated by using the prior and the likelihood.

All the four methods have been compared and validated based of their performance, and their ability to avoid consequences of curse of dimensionality such as the phenomenon of over-fitting.

2 Approach

2.1 Data

Linearly spaced data points have been generated between 1 and 4π. The data vector x is then passed through a sinusoidal function $sin 0.5x$ to generate the actual desired values. A random Gaussian noise with zero mean and standard deviation of 0.3 is added to each desired value to generate a target vector t. The goal of linear regression algorithms is to fit the actual curve before adding a random Gaussian Noise using the training data (x, t). 1 shows the training data and the desired curve, which has to be fit.

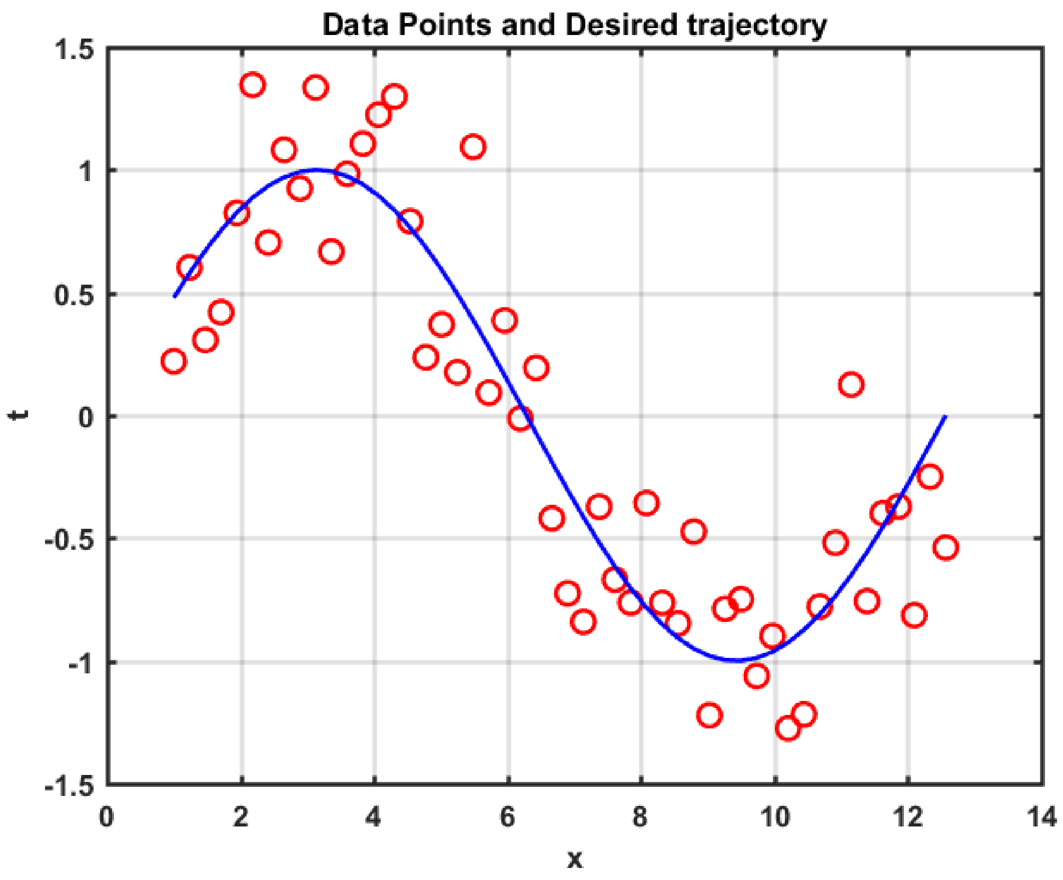


Figure 1: Data Points and Desired curve

2.2 Methods

In this section, Closed form solutions have been derived for all the four methods of linear regression and working of each algorithm has been explained.

2.2.1 Regression using error minimization

The hypothesis function is trained to learn the weights so that the error function is minimized, i.e., the average distances between the curve fit and every point on the desired trajectory is minimized. Even though the hypothesis function is a M^{th} order polynomial, it is linear in w. The hypothesis function is chosen to be:

$$y(x, W) = w_0 + w_1 x^1 + w_2 x^2 + .. + w_M x^M = \sum_{j=1}^{M} w_j x^j \tag{1}$$

Error function is a quadratic function:

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

(1) in matrix form:

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

where,

$$\begin{aligned}
W &= [w_0, w_1, ..., w_M]^T && \in \mathbb{R}^{(M+1)\times 1} \\
t &= [t_0, t_1, ..., t_N]^T && \in \mathbb{R}^{N\times 1} \\
X &= \begin{bmatrix} 1 & x_1^1 & \cdots & x_1^M \\ 1 & x_2^1 & \cdots & x_2^M \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N^1 & \cdots & x_N^M \end{bmatrix} && \in \mathbb{R}^{N\times(M+1}
\end{aligned}$$

To minimize the error function in (2), we compute $\frac{\partial E(W)}{\partial W} = 0$

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

To simplify (4) we use the following axioms:

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

Using the above two axioms to simplify 4,

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

Hence, W^* that minimizes $E(W)$ is:

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

2.2.2 Regression using error minimization with regularization

The hypothesis function in this method is similar to the previous method, with an extra penalty term[1]. This term inhibits the weights of the hypothesis function to reach larger values, and thereby, ensures the curve from over-fitting during training phase. The penalty term in this method takes the form of sum of squares of coefficients with a gain term λ, which, in the vectorized form, can be written as $\lambda W^T W$. Hence, the hypothesis function becomes:

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

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

To minimize the modified error function (9), Therefore we compute $\frac{\partial E(W)}{\partial W} = 0$

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

Hence, W^* that minimizes $E(W)$ is:

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

We can observe that, unlike 8, W^* in the above equation is dependent on both the data as well as the regularization term λ . Hence, this term aids in the control of the optimum weights of the hypothesis function, so that over-fitting can be avoided, i.e., with the same number of data points, a higher order polynomial hypothesis function can be constructed without having to deal with curse of dimensionality.

2.2.3 Regression using Maximum Likelihood Estimation

[1] Maximum likelihood estimation takes the Bayesian approach. It considers uncertainty over the target values. The likelihood of a target value is a probability distribution given by:

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

So, it is assumed that the target values are distributed as a Gaussian with mean μ and variance β^{-1}. 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 (i.i.d), the joint likelihood function is given by:

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

Training data (X, T) is used to determine the parameters w and β by maximum likelihood: 13 can be written as:

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

Taking log of 14,

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

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

Maximizing log of likelihood,
Differentiating 16 w.r.t 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 w.r.t β,

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

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

2.2.4 Regression using Maximum A Priori Estimation

[2]Maximum A Priori Estimation takes the Baysian approach as well. But, unlike MLE, MAP considers both the likelihood as well as a prior distribution. The likelihood distribution is considered to be a Gaussian with mean parameter μ, which is equal to the output of the hypothesis function, $y(x, w)$ at each data point and the variance parameter β. It is assumed that the weights of the hypothesis function have a prior distribution. The weights are also considered to be distributed as a Gaussian with zero mean and variance parameter α. Here, α is called the hyper parameter. This is shown in the equation below:

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

where, M is the order of the hypothesis function. The goal of MAP is to learn the weights of the hypothesis function. Hence, we need to calculate the posterior probability of the weights given the training data α and β. This is calculated using Bayes Theorem.

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

Expanding the two Gaussians and taking product,

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

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

The optimum Weight vector is obtained by maximizing the posterior, or conversely, minimizing the error function. For convenience, it is feasible to maximize the log, or minimize the negative log of the posterior. Taking log of the posterior and simplifying,

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

Differentiating w.r.t β, we get

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

Similarly, differentiating w.r.t α, 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 w.r.t μ, 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 28,26, 24, the maximum posterior error function is:

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

Vectorizing the above equation,

$$E(\mathbf{w}) = \frac{\beta}{2}(XW-T)^T(XW-T) + \frac{\alpha}{2}w^T w \tag{30}$$

$$= \frac{\beta}{2}(W^TX^T - T^T)(XW-T) + \frac{\alpha}{2}w^T w \tag{31}$$

$$= \frac{\beta}{2}(W^TX^TXW - W^TX^TT - t^TXW + T^T) + \frac{\alpha}{2}w^T w \tag{32}$$

To minimize the error function, we calculate

$$\frac{\partial \mathbf{E(w)}}{\partial W} = 0 \tag{33}$$

$$\implies \frac{\beta}{2}(2X^TXW - 2X^TT) + \alpha W = 0 \tag{34}$$

$$\implies \frac{\beta}{2}(W^TX^T - T^T)(XW - T) + \frac{\alpha}{2}w^Tw = 0 \tag{35}$$

$$\implies \beta X^TXW - \beta X^TT + \alpha W = 0 \tag{36}$$

$$\tag{37}$$

Therefore, the optimum weight vector W^* which minimizes the above equation is:

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

We can observe that W^* in MAP depends on both β and α. Hence, unlike MLE, MAP estimation not only depends on the data, but also the prior hyper-parameter. This means that we have better control over the estimation when the available training data is sparse. Hence, by choosing a suitable hyper-parameter α, we can get a better estimate of W^*, which results in better prediction.

3 Results

This section has the results corresponding to each method of linear regression, which will help us validate their performance. The results are organized as follows: For the results in the first four sub-sections: 1) Number of data points used is 50. However, since the target data values have an additional random Gaussian Noise, the data set used for training changes in each run of the program. 2) The methods have been tested for these 5 orders: $M = [0136915]$

The last section contains results corresponding to the extra questions.

3.1 Regression using Error Minimization

We can observe that in 2, the performance of curve fitting is very poor. This is because of the fact that a linear hypothesis function cannot fit a non-linear curve. As the order increases as in 3, we can observe that the curve fit is better. In fact, for M = 3, the fitted curve is very similar in visualization, to the desired polynomial. Visually, M = 3 performs better than M = 6, where the fitted curve is starting to deviate from the desired curve. So, for M = 3 to M = 6, the inference is that the hypothesis function has learnt the optimum weights to fit unseen data as well with good accuracy. In 4, it is very evident that the fitted curve is oscillating around the desired response. Here, the fitted curve is trying to go through as many data points as possible to reduce the training error. This phenomenon is termed as **over-fitting**, which is a consequence of curse of dimensionality. The accepted thumb rule is that the number of data points in the training data set should be at least 5 times the order chosen for the hypothesis function, which in this case, is not true. Hence, even if the training error decreases, the resulting polynomial is unable to fit the given data accurately. It can be observed that,in figure 5, the training error monotonically decreases as order increases. But, the test error decreases till M = 6 and starts increasing after this point. Hence, this figure is an evidence of over-fitting.

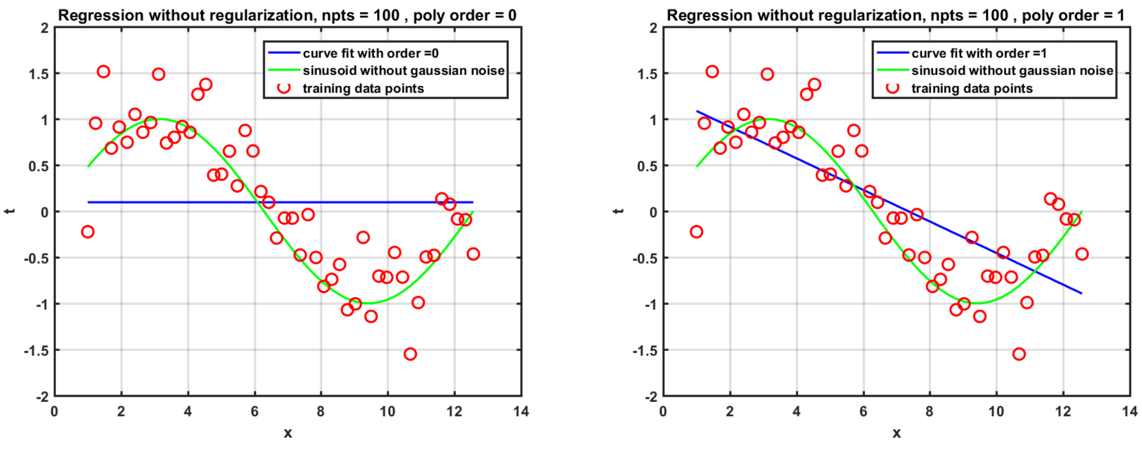


Figure 2: Plot of polynomials having orders M = [0 1] respectively shown as blue curves, fitted to the data set shown in 1

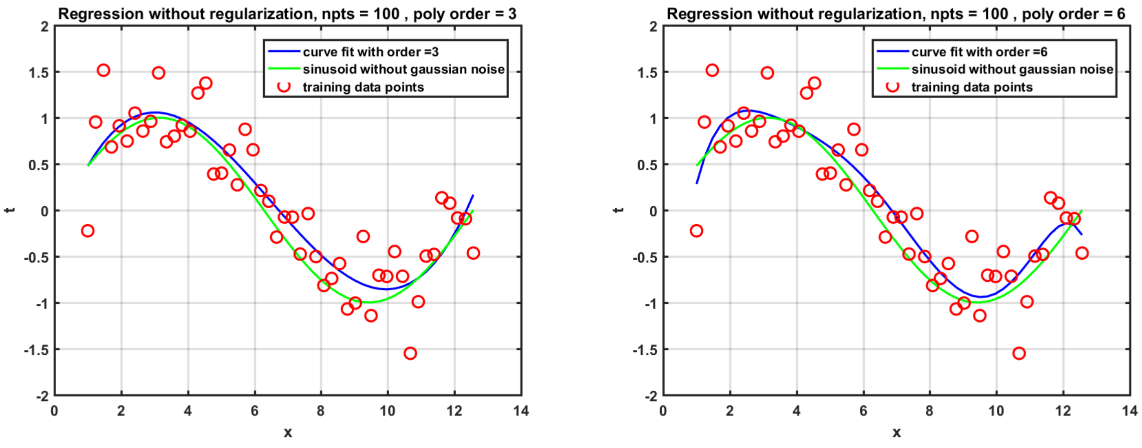


Figure 3: Plot of polynomials having orders M = [3 6] respectively shown as blue curves, fitted to the data set shown in 1

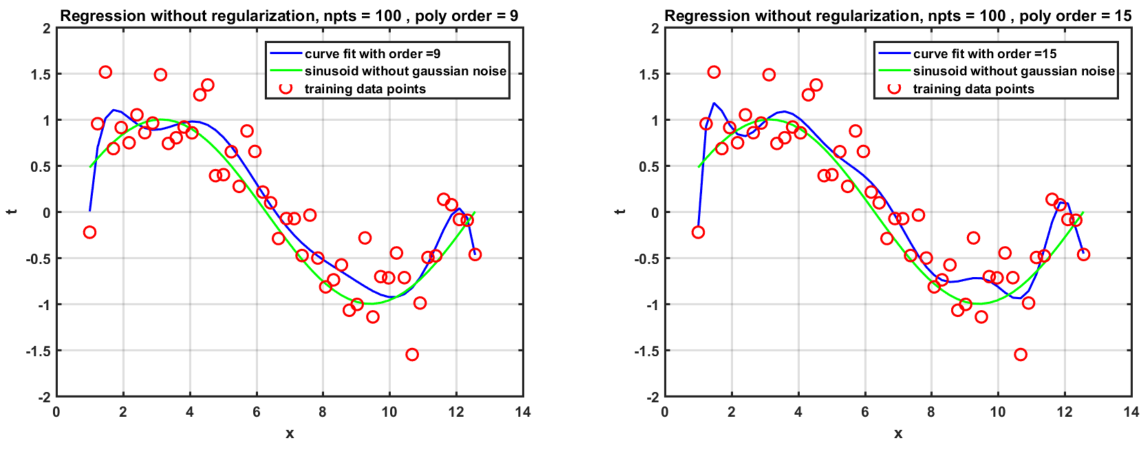


Figure 4: Plot of polynomials having orders M = [9 15] respectively shown as blue curves, fitted to the data set shown in 1

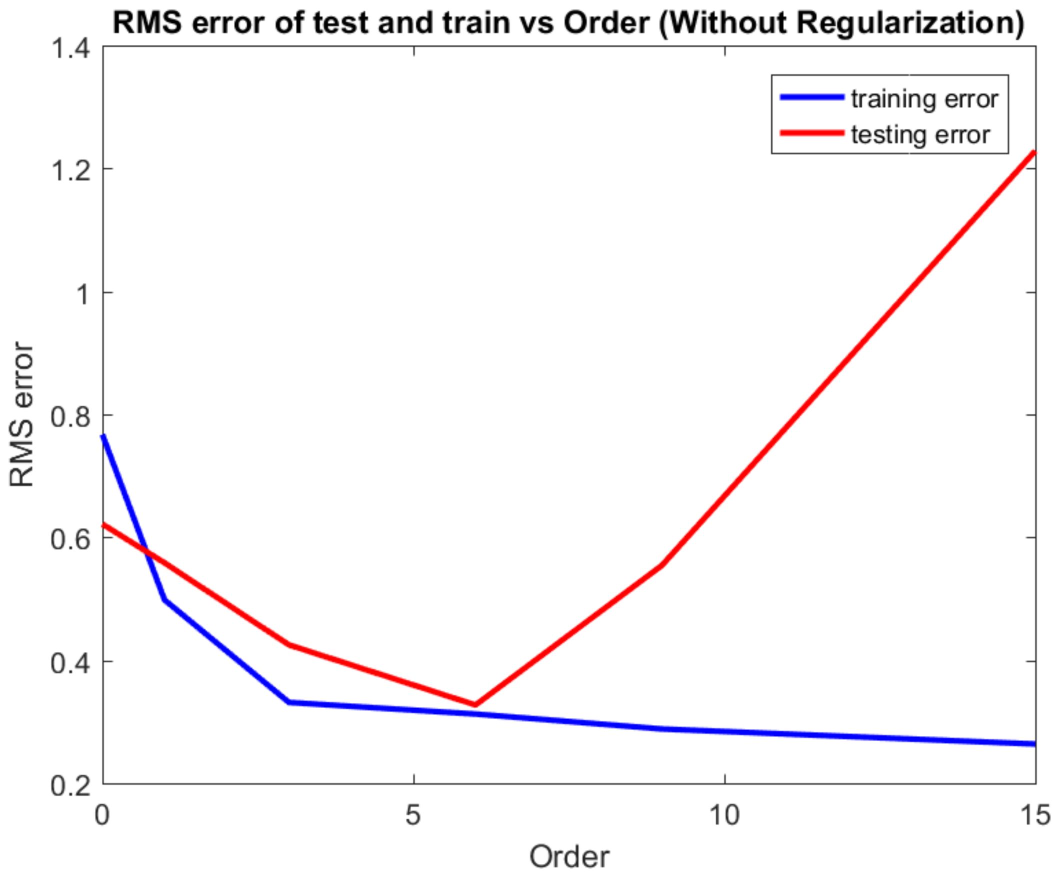


Figure 5: Plot of root-mean-square error, evaluated on the training set and independent test set for various values of order M.

3.2 Error Minimization with regularization

With a quadratic regularization term added to the error function as in 8, the closed form solution for the optimal weight vector is no longer dependent only on data. So, we have a better control over the polynomial curve-fitting in this case. The reason for oscillations in the fitted curves in 3 and 4 was that the weights for higher order had a huge variance factor. i.e., the weights had an oscillating behavior. But, the regularization term λ helps in avoiding these oscillations. So, the problem of over-fitting can be avoided, even when a polynomial of higher order is chosen on sparse data.
Consider figures 6,7 and 8. These show the plots of fitted polynomials for orders M = 9 and M = 15, with different regularization parameters of magnitudes 0.01, 0.1 and 1. Comparing these figures with 4, it is very evident that the oscillation of fitted curves around the desired curve has been significantly reduced. This also means that the weights of optimum weight vector have a reduced variance factor. Hence, the constraints over the chosen orders for the given curve fitting problem is less due to regularization. Hence, with a regularization factor, higher order polynomials can be chosen even when the available training data is less.
Choosing the parameter λ is crucial. But unfortunately, there are no specific rules to choose the value of this parameter and it has to be chosen by the method of trial and error.

Figure 9 shows the RMS error plots for training and independent test data sets. It can be observed that the error in test set has significantly decreased when compared to 5. It can also be observed that the RMS error is least for higher orders for $\lambda = 1$.

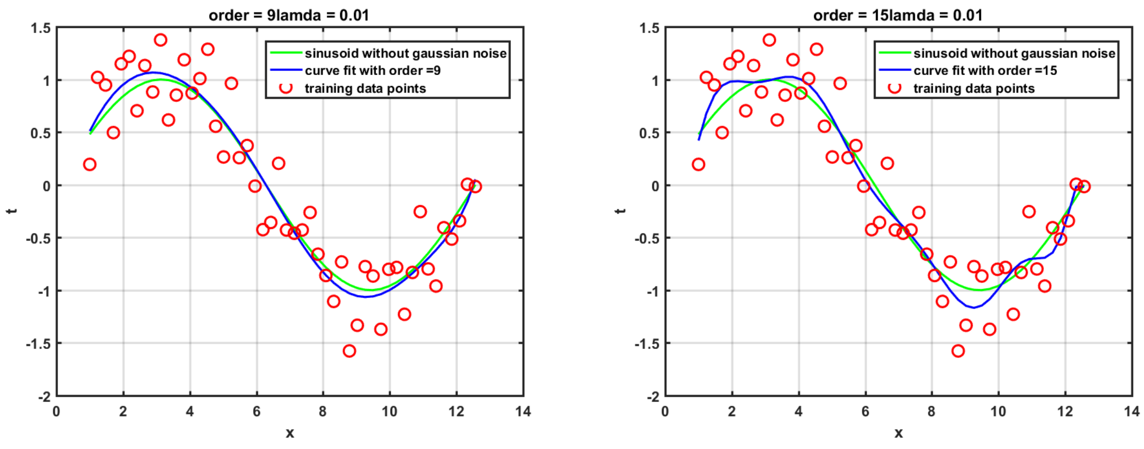


Figure 6: Plot of polynomials having orders M = [9 15] respectively shown as blue curves, fitted to the data set shown in 1 with a regularization term 0.01

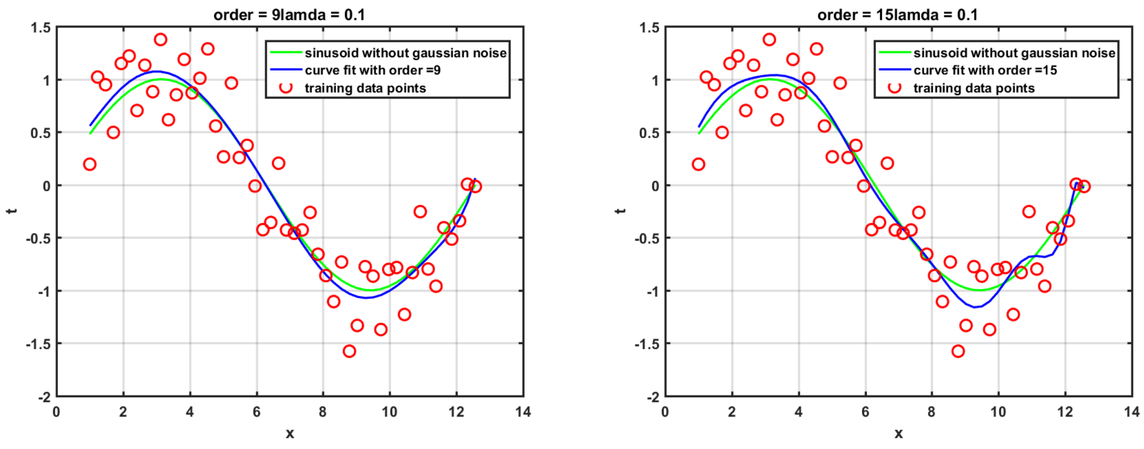


Figure 7: Plot of polynomials having orders M = [9 15] respectively shown as blue curves, fitted to the data set shown in 1 with a regularization term 0.1

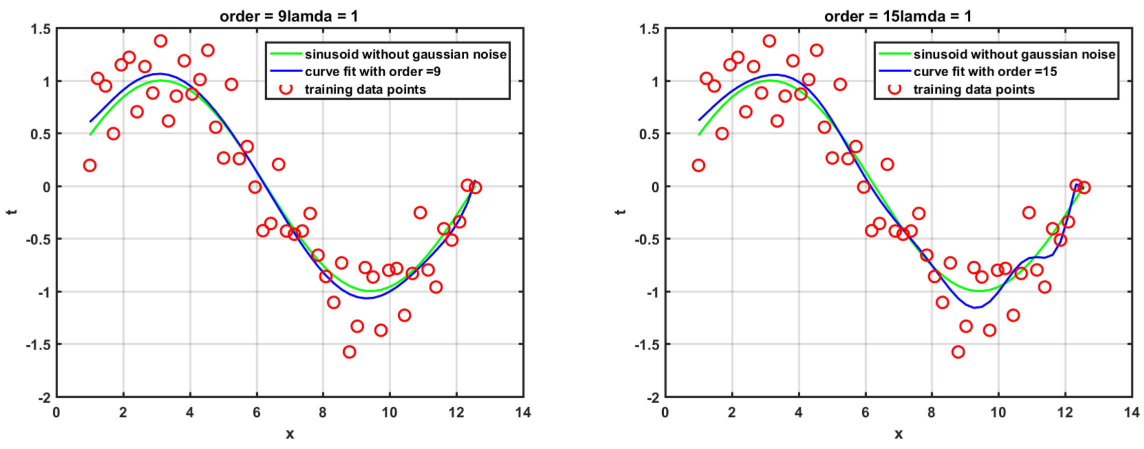


Figure 8: Plot of polynomials having orders M = [9 15] respectively shown as blue curves, fitted to the data set shown in 1 with a regularization term 1

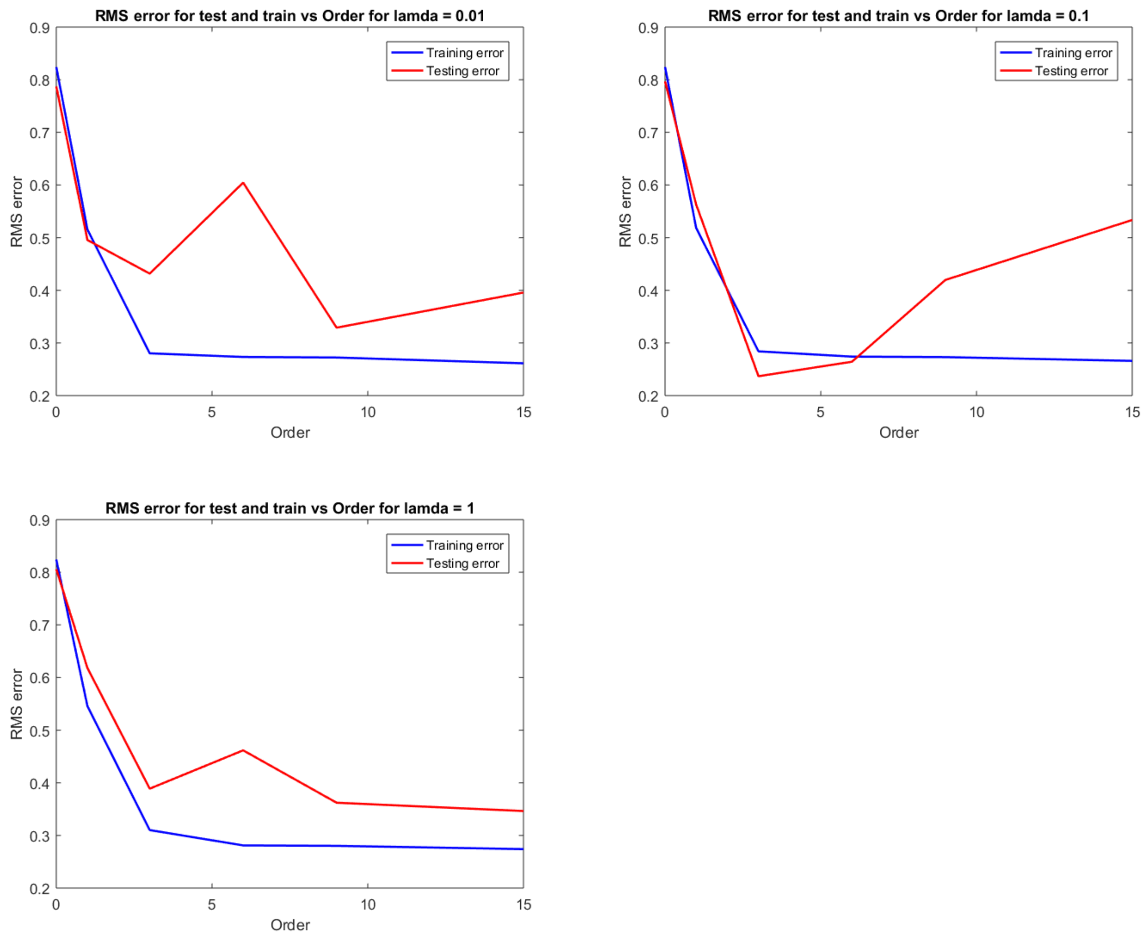


Figure 9: Plot of root-mean-square error, evaluated on the training set and independent test set for various values of order M with various values of λ

3.3 Error Minimization using MLE

With the assumption of uncertainty over the target values, which is represented as a Gaussian distribution, the goal of MLE is to predict the mean and variance of this distribution. The data points are assumed to independently and identically distributed (i.i.d) .The mean of the distribution keeps varying as it is equal to the output of the hypothesis function for every data point. The variance of the distribution is calculated by maximizing the likelihood function. The variance of the distribution depends on the value of mean, and thereby, the data.
From figures 10, 11 and 12 it can be observed that almost all the target data points in 1 falls within the shaded boundary around the fitted curve. Hence, it is evident that the uncertainty over the target data points has been taken care of. From comparison of figures 10,11 and 12 with figures 2, 3 and 4 respectively, we can observe that the fitted curves are similar. This proves that error minimization is equivalent to likelihood maximization. But, the difference between method 1 and this method is evident by figure 13. MLE performs better on the test data set. In fact, its performance is similar to method 2, error minimization with regularization.

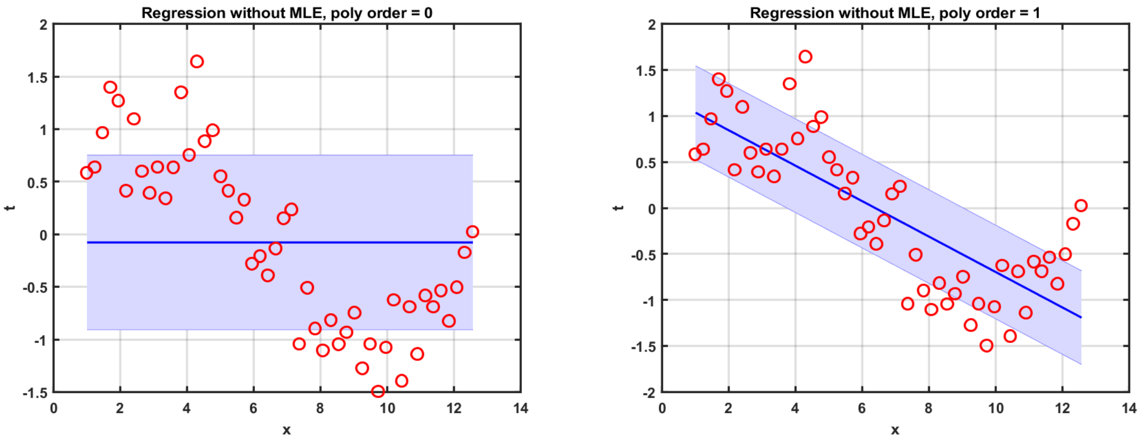


Figure 10: Plot of polynomials having orders M = [0 1] respectively shown as blue curves, fitted to the data set shown in 1. The blue shaded area indicates the variance of the Gaussian distribution around the curve, i.e., it indicates how the target values are distributed around the curve.

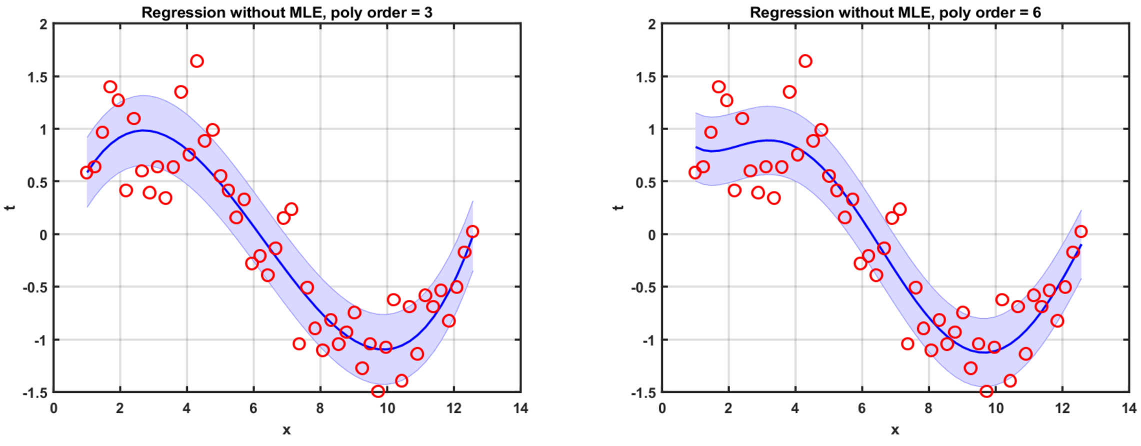


Figure 11: Plot of polynomials having orders M = [3 6] respectively shown as blue curves, fitted to the data set shown in 1. The blue shaded area indicates the variance of the Gaussian distribution around the curve, i.e., it indicates how the target values are distributed around the curve.

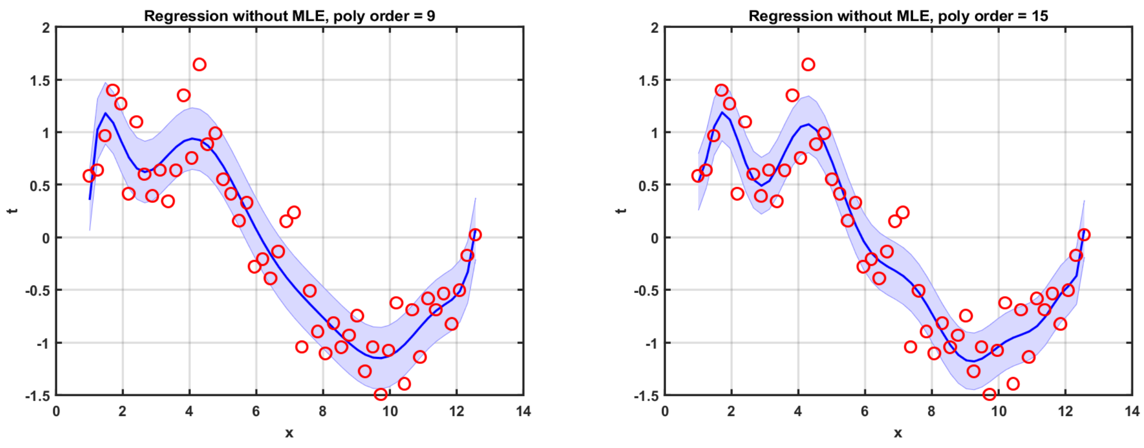


Figure 12: Plot of polynomials having orders M = [9 15] respectively shown as blue curves, fitted to the data set shown in 1. The blue shaded area indicates the variance of the Gaussian distribution around the curve, i.e., it indicates how the target values are distributed around the curve.

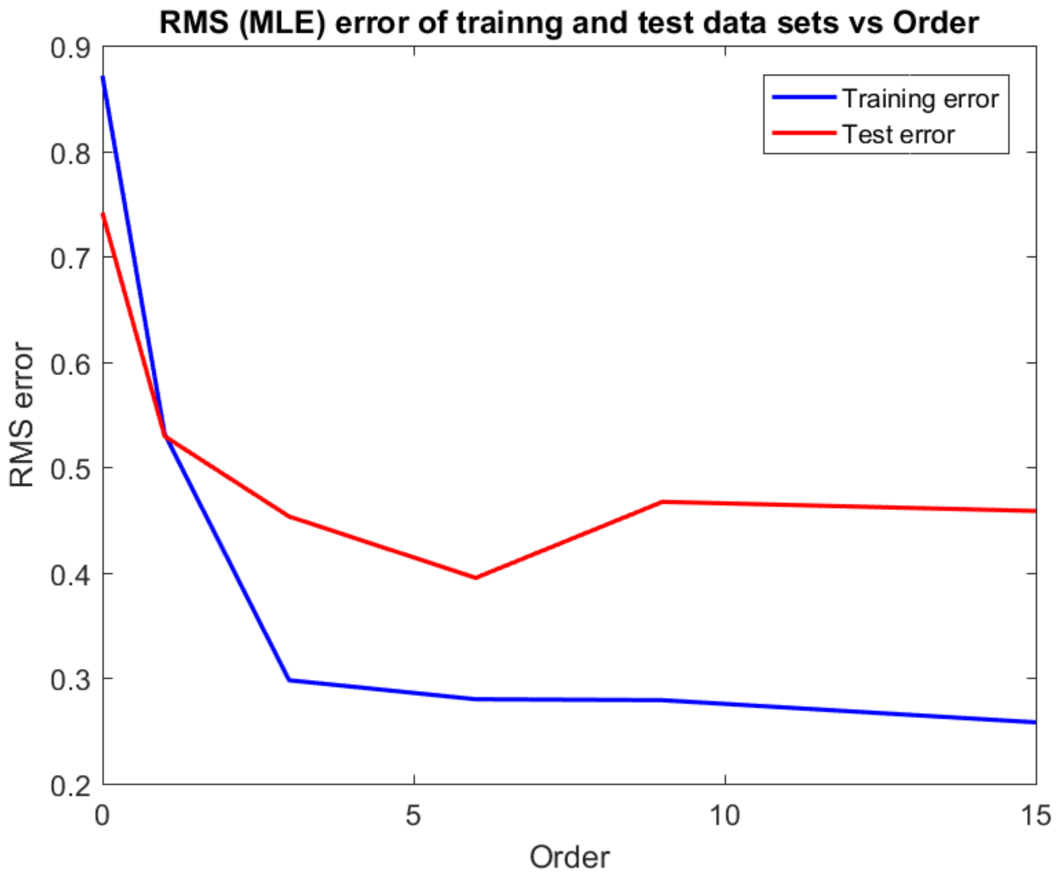


Figure 13: Plot of root-mean-square error for MLE, evaluated on the training set and independent test set for various values of order M.

3.4 Error Minimization using MAP

Unlike MLE, in addition to a likelihood Gaussian distribution over target values, MAP also considers a prior Gaussian probability distribution over the weights. Hence, MAP uses the posterior, which is proportional to the product of these two distributions, to estimate the optimum weights for the hypothesis function. The optimum weight vector depends on both the variance of the target value distribution β and the prior hyper-parameter α. Hence, these two parameters can therefore be used to achieve better prediction accuracy on unseen data. Hence, MAP is analogous to MLE with regularization, with the regularization term equal to $\frac{\alpha}{\beta}$.
Maximum likelihood approach systematically underestimates the variance of the distribution, which is taken care of in MAP. [1]
For this problem, β is chosen to be $\frac{1}{0.3^2} = 11.1$ and α is chosen to be $[0.005\ 0.1]$.
The curves are plotted for M = [3 6 9 15].
Visualizing figures 14 and 15, it is evident that the curves fit the desired response with minimum error. Figure 16 shows that the curve fits the desired response even when the polynomial order is equal to 9 without any oscillations around the desired trajectory. As the order increases, the oscillations around the desired curve increases, as in 17, but the oscillations are smaller when compared to MLE, as in 12
However, the most significant difference is shown in 18, where the Test error is below the Training error for all orders and both the values of α. This implies that the hypothesis function has learnt the optimum values of weights for each order,i.e., prediction over unseen data has maximum accuracy when MAP Estimation is applied. This is the best result among all the methods.

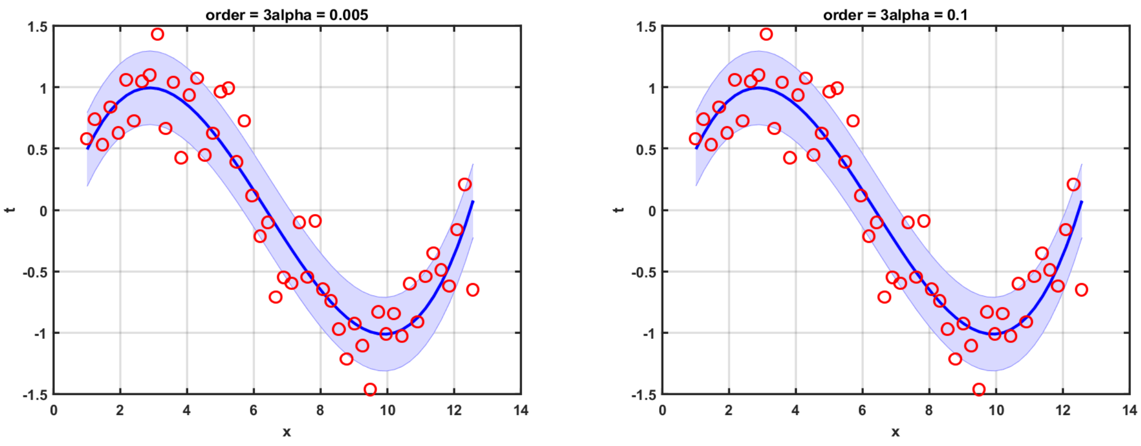


Figure 14: Plot of polynomials having orders M = [3] respectively shown as blue curves, fitted to the data set shown in 1. The blue shaded area indicates the variance of the Gaussian distribution around the curve, i.e., it indicates how the target values are distributed around the curve.

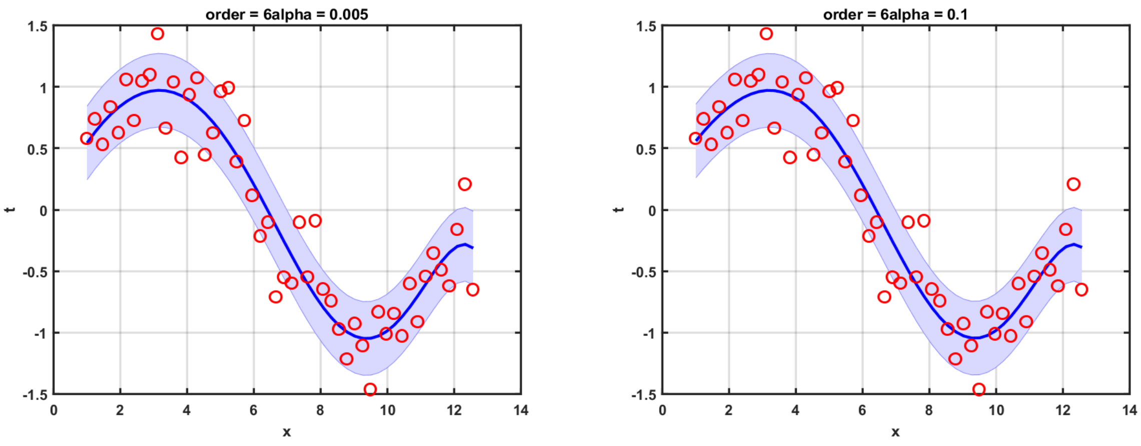


Figure 15: Plot of polynomials having orders M = [6] respectively shown as blue curves, fitted to the data set shown in 1. The blue shaded area indicates the variance of the Gaussian distribution around the curve, i.e., it indicates how the target values are distributed around the curve.

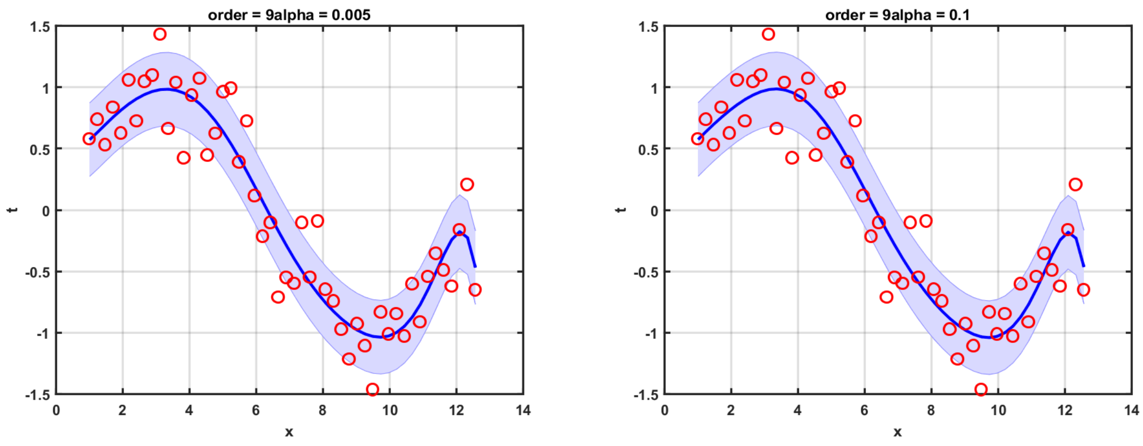


Figure 16: Plot of polynomials having orders M = [9] respectively shown as blue curves, fitted to the data set shown in 1. The blue shaded area indicates the variance of the Gaussian distribution around the curve, i.e., it indicates how the target values are distributed around the curve.

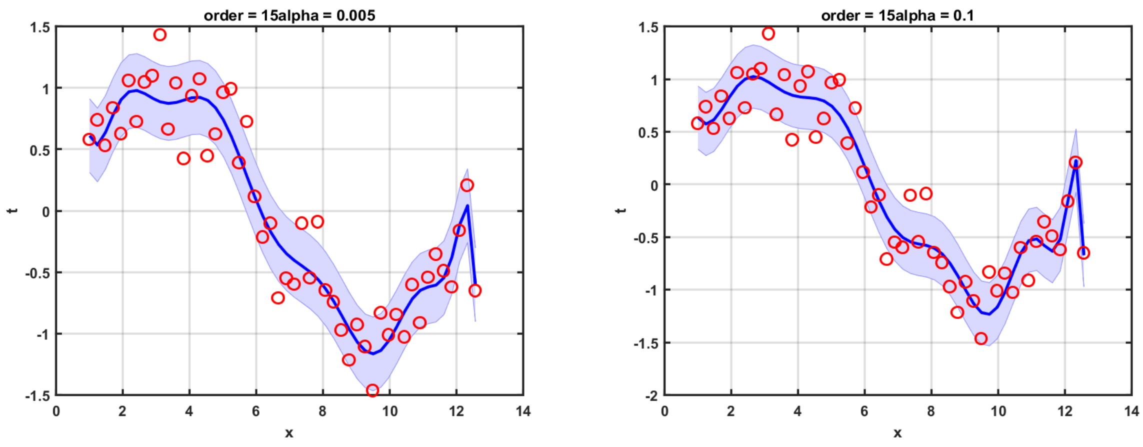
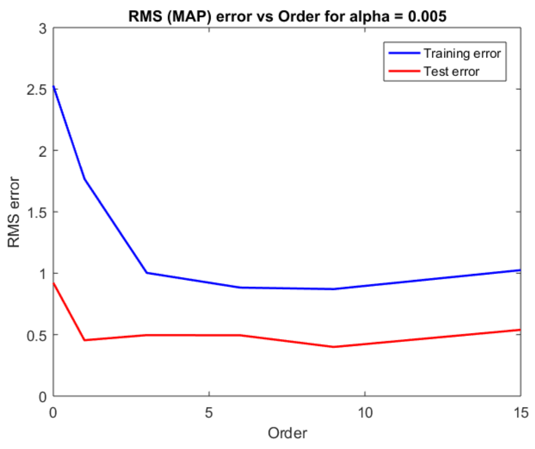


Figure 17: Plot of polynomials having orders M = [15] respectively shown as blue curves, fitted to the data set shown in 1. The blue shaded area indicates the variance of the Gaussian distribution around the curve, i.e., it indicates how the target values are distributed around the curve.



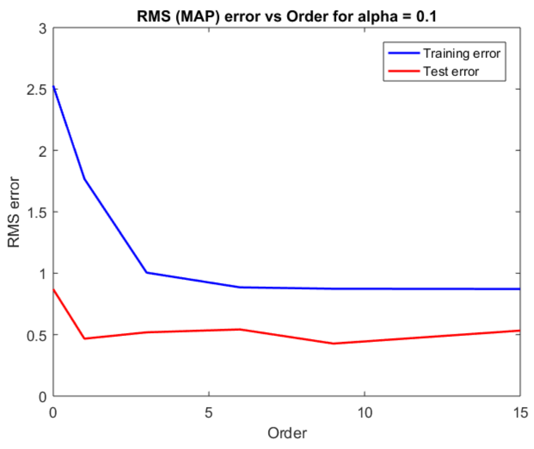


Figure 18: Plot of root-mean-square error for MAP, evaluated on the training set and independent test set for various values of order M and $\alpha = 0.005$ and 0.1

3.5 Extras

1) Method 2, error minimization with regularization has been used to generate plots for M = [3 6 9 15] for $\log(\lambda) = -18$,$\log(\lambda) = -15$ and $\log(\lambda) = -13$. Figures 19 to 23 show these plots.

2) A table 24 of order M vs Weights w has been generated using method 1, regression using error minimization, to observe the increase in variance factor of the weights with order.

3) Plots for a constant order M = 9 and different number of data points have been generated using method 1, regression using error minimization. These plots show that over-fitting occurs for very sparse data, which can be avoided by increasing the number of data points. Invoking the thumb rule that the order chosen should be such that the number of data points should be at least 5 times the order, we can observe that 9^{th} order polynomial fits 100 data points without over-fitting and it causes over-fitting for 10 data points.

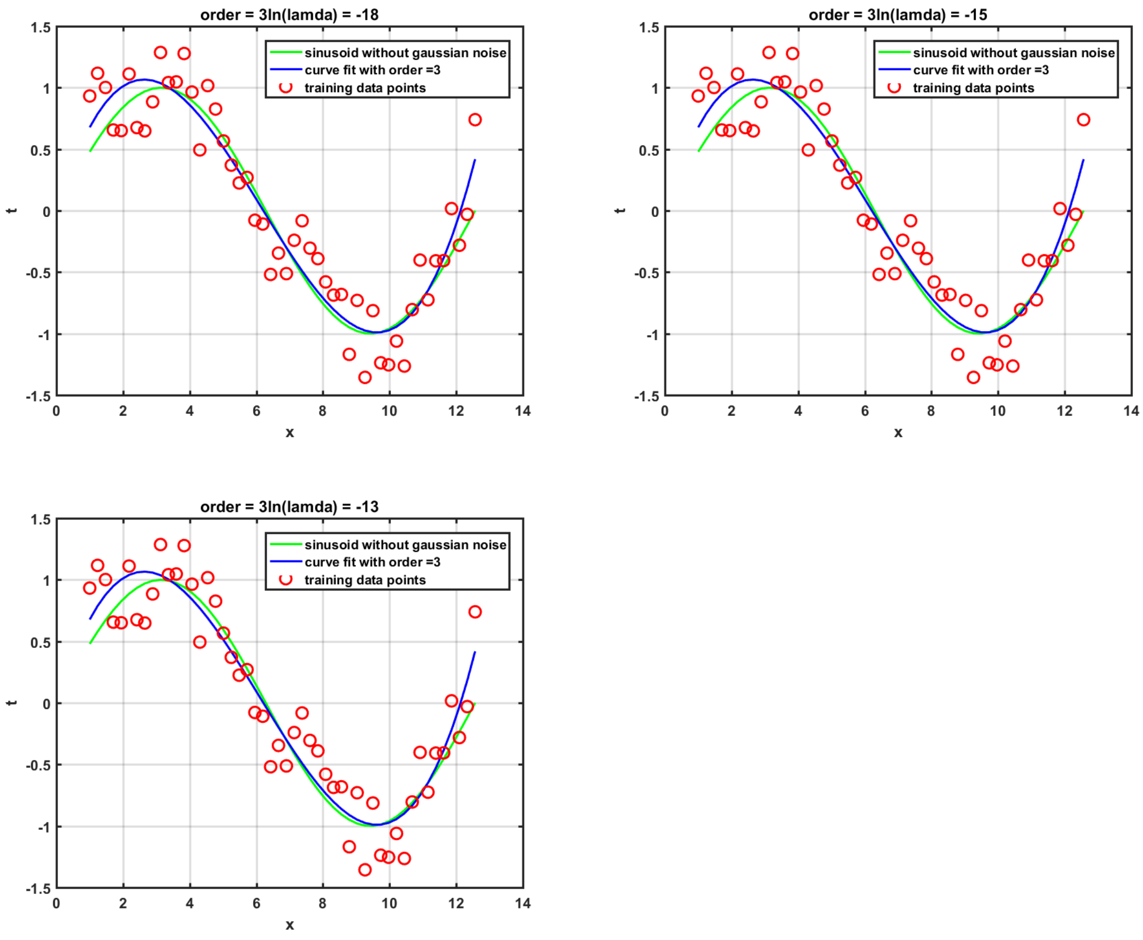
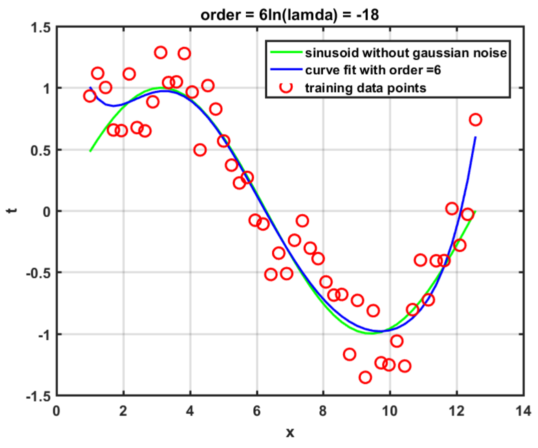
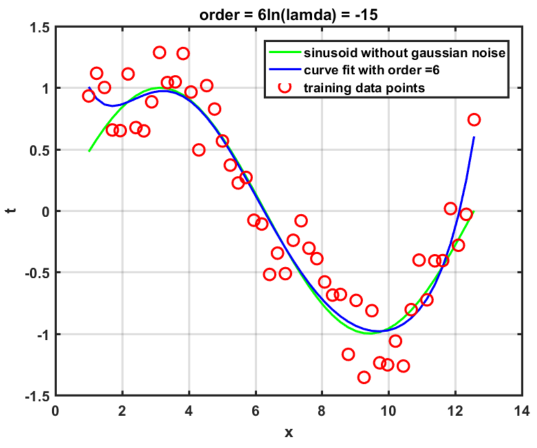


Figure 19: Plot of polynomials having orders M = [3] respectively shown as blue curves, fitted to the data set shown in 1 for all three alpha values.





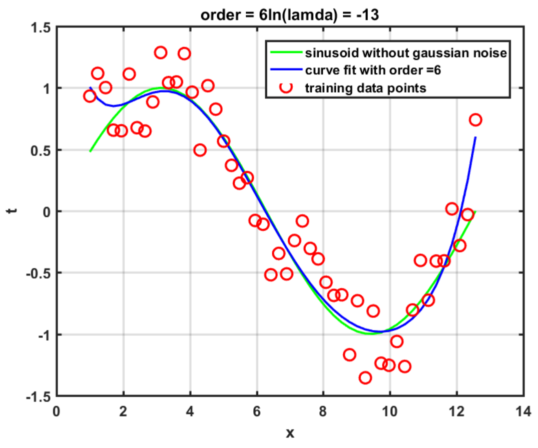


Figure 20: Plot of polynomials having orders M = [6] respectively shown as blue curves, fitted to the data set shown in 1 for all three alpha values.

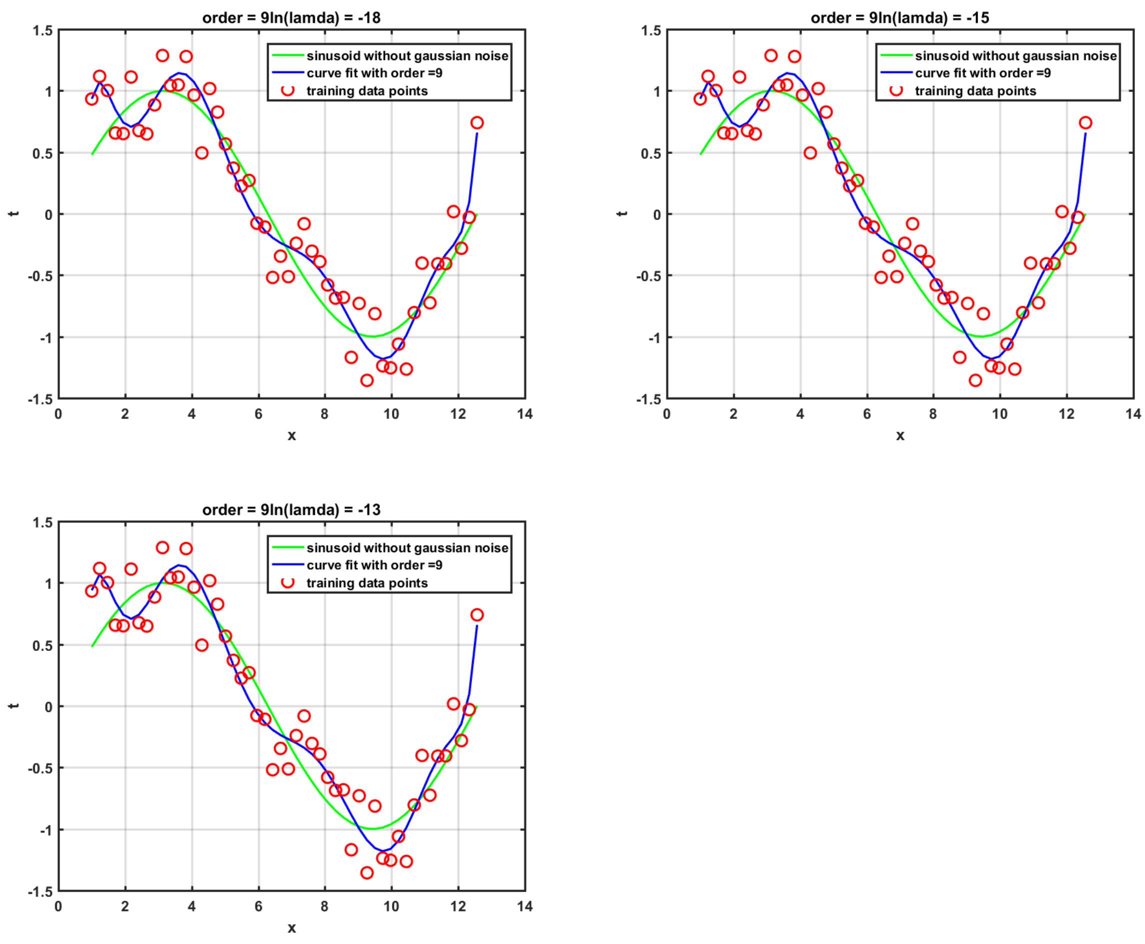


Figure 21: Plot of polynomials having orders M = [9] respectively shown as blue curves, fitted to the data set shown in 1 for all three alpha values.

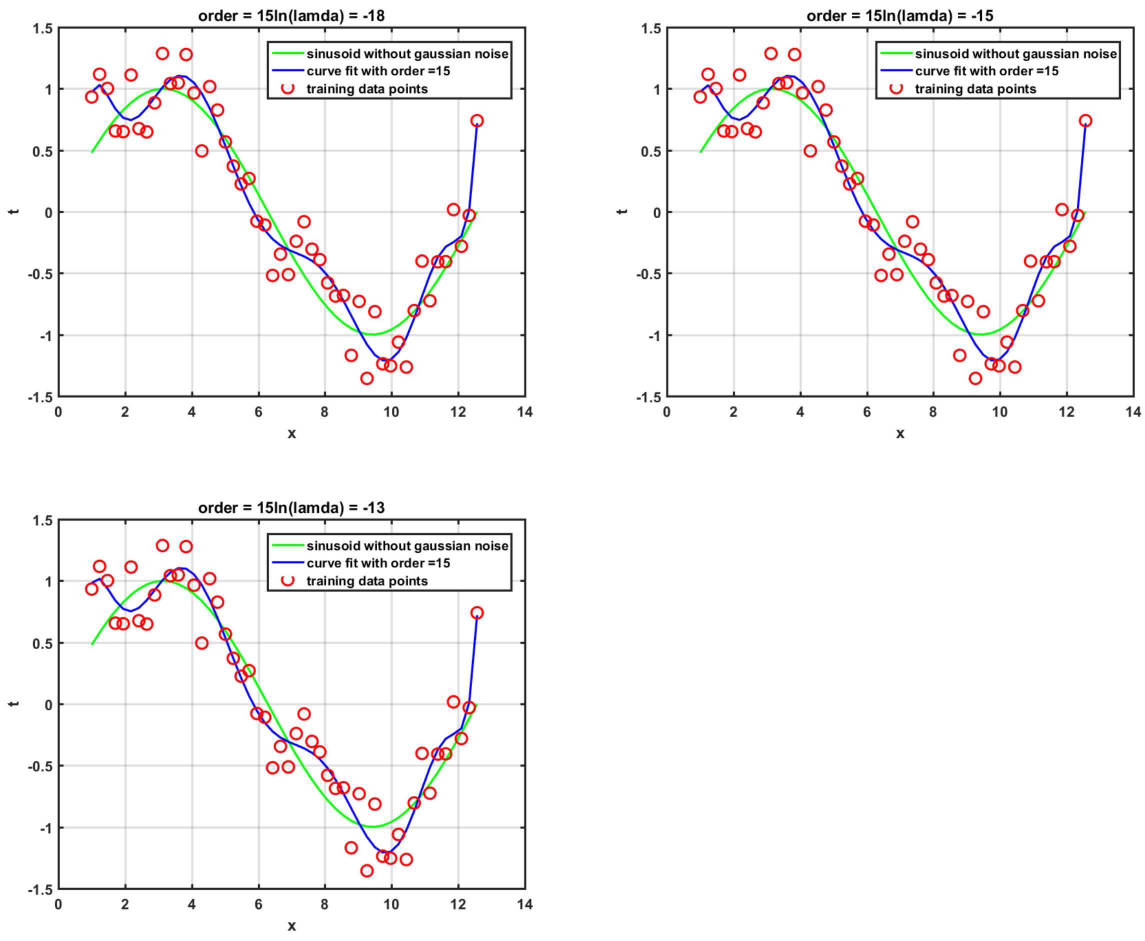


Figure 22: Plot of polynomials having orders M = [15] respectively shown as blue curves, fitted to the data set shown in 1 for all three alpha values.

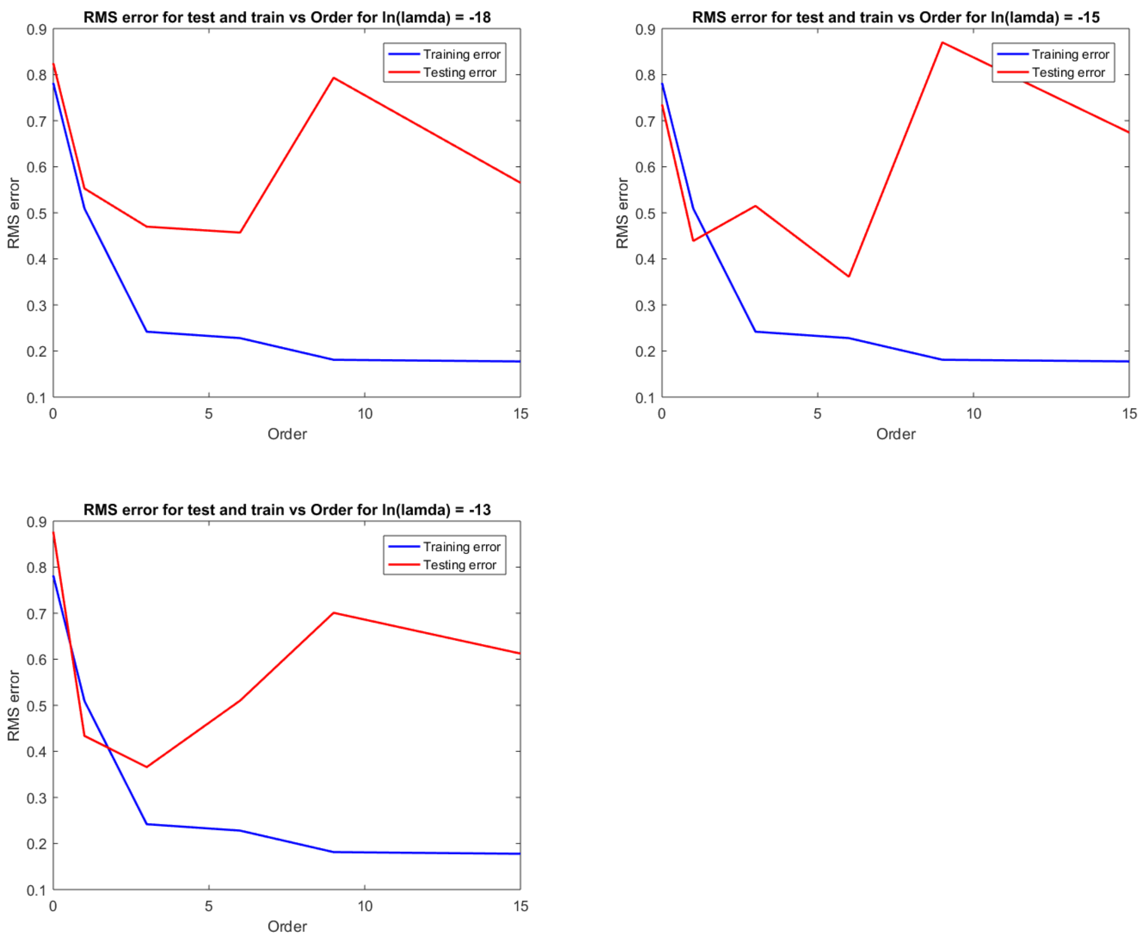


Figure 23: Plot of root-mean-square error for MAP, evaluated on the training set and independent test set for various values of order M and all λ values.

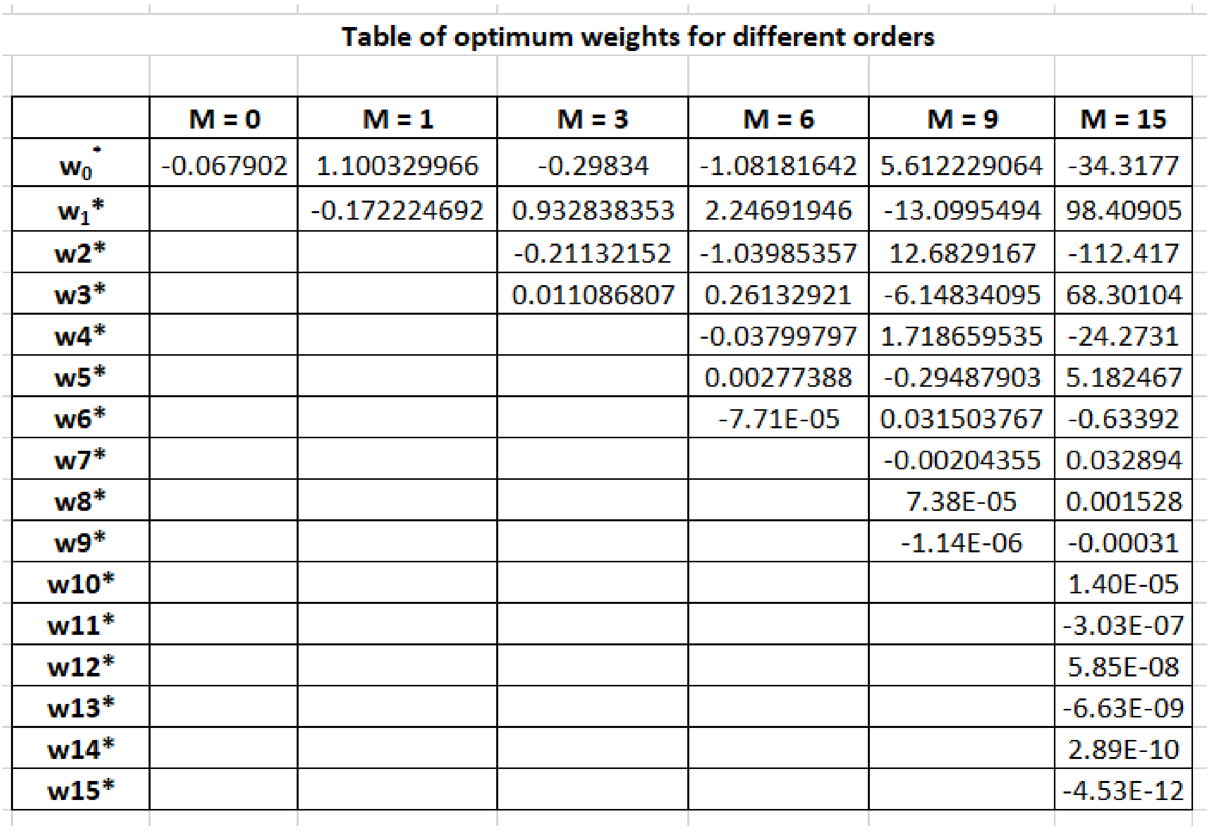


Table of optimum weights for different orders

	M = 0	M = 1	M = 3	M = 6	M = 9	M = 15
w_0^*	-0.067902	1.100329966	-0.29834	-1.08181642	5.612229064	-34.3177
w_1^*		-0.172224692	0.932838353	2.24691946	-13.0995494	98.40905
w2*			-0.21132152	-1.03985357	12.6829167	-112.417
w3*			0.011086807	0.26132921	-6.14834095	68.30104
w4*				-0.03799797	1.718659535	-24.2731
w5*				0.00277388	-0.29487903	5.182467
w6*				-7.71E-05	0.031503767	-0.63392
w7*					-0.00204355	0.032894
w8*					7.38E-05	0.001528
w9*					-1.14E-06	-0.00031
w10*						1.40E-05
w11*						-3.03E-07
w12*						5.85E-08
w13*						-6.63E-09
w14*						2.89E-10
w15*						-4.53E-12

Figure 24: Table of order M vs weights w

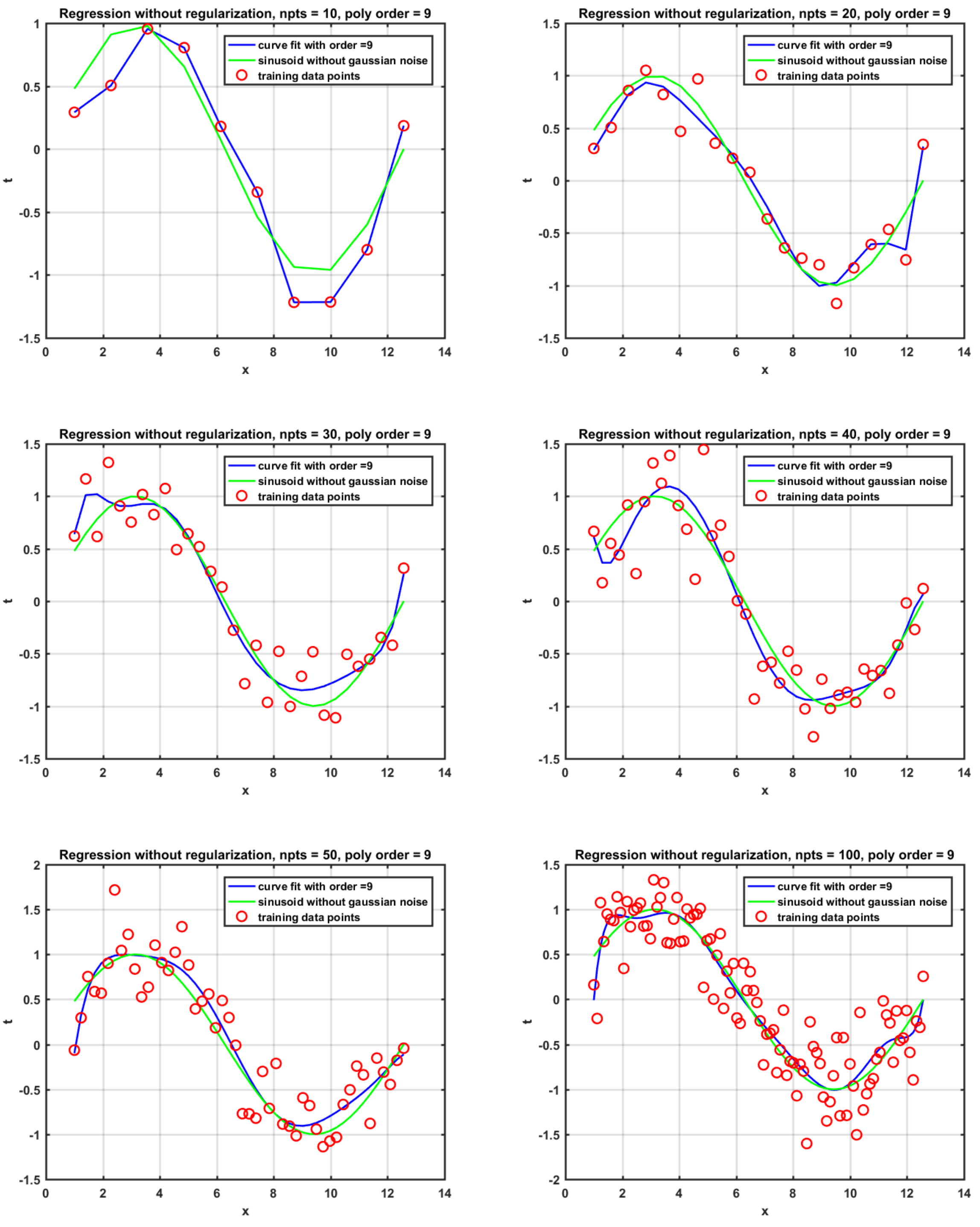


Figure 25: Plots of the solutions obtained by minimizing the sum-of-squares error function using the M = 9 polynomial for N = [10 20 30 40 50 100]. We see that increasing the size of the data set reduces the over-fitting problem

4 Conclusion

From the results observed, the following conclusions can be made:

1) Regularization of error function gives us better control over the regression model as the closed form solution is no longer dependent only on data. It also helps in preventing over-fitting. Higher order hypothesis function can be chosen even when the data is sparse.

2) Over-fitting of data can also be avoided when the number of training data points are increased. (ref : 25)

3) Without regularization, the variance factor of the weights increases with order. (ref: 24).

4) MAP estimation is analogous to MLE with regularization, with a penalty factor of $\frac{\alpha}{\beta}$.

5) MAP Estimaton performs the best among all the other methods, and has the least RMS error for both test and training data sets. This is due to the fact that the closed form solution of MAP, unlike that of MLE, is not entirely data driven. The prior hyper-parameter plays a vital role in reducing the error and providing a better prediction on unseen data.

References

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

[2] Prince, Simon JD *Computer vision: models, learning, and inference.* 2012 3, 5, 6, 19 3, 7