Cure Fitting/Linear Regression

Yifan Guo

January 23, 2019

Abstract

In order to grasp the Bayesian modeling framework, this project solves the linear regression problem by two different approaches: 1) direct error minimization and 2) Bayesian approach. After complete the program, I will summerize, compare and contrast the results of two different methods.

Contents

1 Introduction

In this project, I will accomplish the curve fitting problem through MATLAB. First, I create some random sample points, which are obey the Gaussian distribution. Then I use four methods to solve the curve fitting problem. They are 1)error minimization of SSE 2)error minimization with the regularization term 3)the ML (maximal likelihood) estimator of the Bayesian approach 4)the MAP (maximum a posteriori) estimator of the Bayesian approach. Finally, I obtain the result and make a comparison between them.

2 Approach

2.1 Data

For this project, I use the simulation data. I create a sinusoidal function and add a Gaussian noise to it. Then I get a random sequence so that I can do the curve fitting and linear regression.

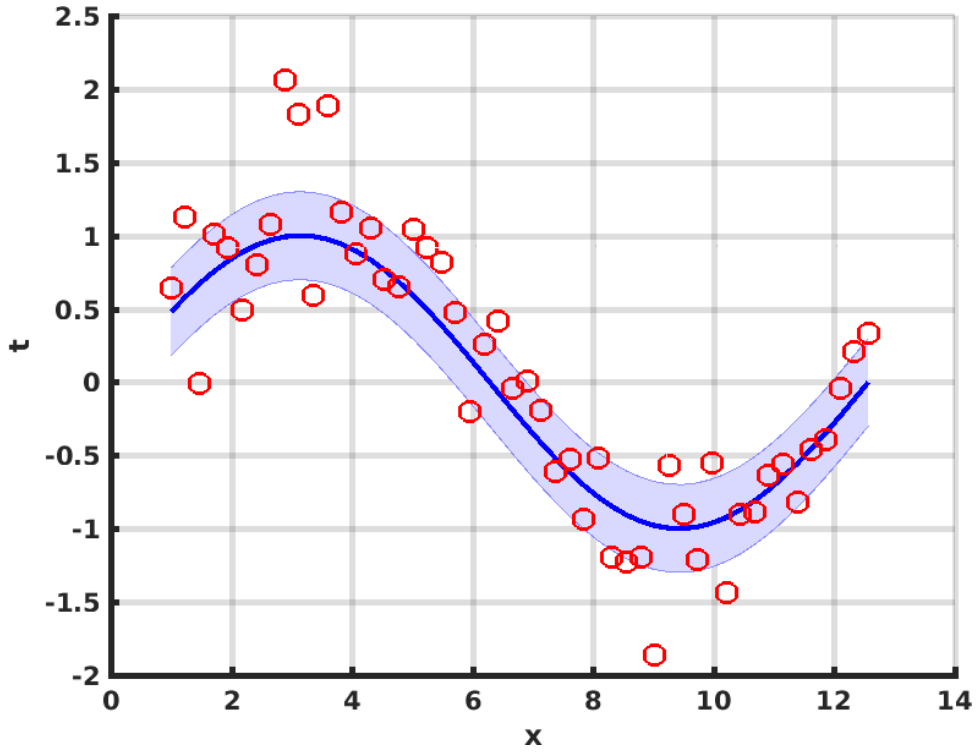

Figure 1: Plot of a training data set of $N = 50$ points, shown as red circles,each comprising an observation of the input variable x along with the corresponding target variable t. The blue curve shows thefunction $sin(\pi x)$ used to generate the data. Our goal is to predict the value of t for some new value of x, without knowledge of the blue curve.

2.2 Methods[1]

2.2.1 Regression using error minimization of SSE

In order to fit the data, we shall use a polynomial function of the form

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{1}$$

where M is the order of the polynomial.
This error function is given by the sum of the squares of the errors between the predictions $y(x_n, \mathbf{w})$ for each data point x_n and the corresponding target values t_n, so that we minimize

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

To minimize this, we shall change equation 2 into a matrix form,

$$\mathbf{W} = [w_0 w_1 \ldots w_M]^T \tag{3}$$

$$T = [t_1 t_2 \ldots t_M]^T \tag{4}$$

$$X = \begin{bmatrix} x_1^0 x_1^1 & \cdots & x_1^M \\ \vdots & \ddots & \vdots \\ x_N^0 x_N^1 & \cdots & x_N^M \end{bmatrix} \tag{5}$$

From above, we get

$$E(\mathbf{w}) = \frac{1}{2}(XW - T)^T (XW - T) \tag{6}$$

take derivative on the equation 6

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = X^T (XW - T) \tag{7}$$

hen we get the $\mathbf{w}^*$, which drive to the minimum error function.

$$\mathbf{w}^* = X^{-1} T \tag{8}$$

Take $\mathbf{w}^*$ into $y(x, \mathbf{w})$, then we get the fitting curve $y(x, \mathbf{w}^*)$.

2.2.2 Regression using error minimization with the regularization term

The method of using error minimization of SSE may casue over-fitting problem sometimes. When that problem happen, we have two methods to solve it, one is getting more data and the other is adding a regularization term. Getting more data will cost a lot, so we usually use the second method to solve the over-fitting problem.
Based on the error function, we get a new error function of the form

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} ||\mathbf{w}||^2 \tag{9}$$

Like above, we use the same method to get $\mathbf{w}^*$ which can drive the minimum $\tilde{E}(\mathbf{w})$. First, we change the error function into a matrix form

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}(XW - T)^T(XW - T) + \frac{\lambda}{2}W^TW \tag{10}$$

then take derivative on both side

$$\frac{\partial\tilde{E}(\mathbf{w})}{\partial\mathbf{w}} = (X^TX + \lambda)W - X^TT \tag{11}$$

Finally, we get the $\mathbf{w}^*$

$$\mathbf{w}^* = \frac{X^TT}{X^TX + \lambda} \tag{12}$$

Take $\mathbf{w}^*$ into $y(x, \mathbf{w})$, then we get the fitting curve $y(x, \mathbf{w}^*)$.

2.2.3 Regression using the ML (maximal likelihood) estimator of the Bayesian approach

The goal in the curve fitting problem is to be able to make predictions for the target variable t given some new value of the input variable x. We can express our uncertainty over the value of the target variable using a probability distribution. For this purpose, we shall assume that, given the value of x, the corresponding value of t has a Gaussian distribution with a mean equal to the value y(x,w). Thus we have

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1}) \tag{13}$$

where, for consistency with the notation in later chapters, we have defined a precision parameter β corresponding to the inverse variance of the distribution.
If the data are assumed to be drawn independently from the distribution, then the likelihood function is given by

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N}\mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1}) \tag{14}$$

Then take log on both sides, we obtain the log likelihood function in the form

$$lnp(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - |t_n\}^2 + \frac{N}{2}ln\beta - \frac{N}{2}ln(2\pi) \tag{15}$$

Consider first the determination of the maximum likelihood solution for the polynomial coefficients, which will be denoted by $\mathbf{w}_{ML}$. Cause we only consider $\mathbf{w}$, maximizing the likelihood is equal to minimizing the SSE error function(2)

$$\mathbf{w}_{ML} = X^{-1}T \tag{16}$$

We can also use maximum likelihood to determine the precision parameter β of the Gaussian conditional distribution.

$$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}_{ML}) - |t_n\}^2 \tag{17}$$

Finally, we can make predictions for new values of x depend on the distribution below

$$p(t|x, \mathbf{w}_{ML}, \beta_{ML}) = \mathcal{N}(t|y(x, \mathbf{w}_{ML}), \beta_{ML}^{-1}) \tag{18}$$

2.2.4 Regression using the MAP (maximum a posteriori) estimator of the Bayesian approach

In this technique, we should first consider a Gaussian distribution of the form

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|0, \alpha^{-1}\mathbf{I}) = (\frac{\alpha}{2\pi})^{\frac{M+1}{2}} exp\{-\frac{\alpha}{2}\mathbf{w}^T w\} \tag{19}$$

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

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha) \tag{20}$$

We can now determine $\mathbf{w}$ by finding the most probable value of $\mathbf{w}$ given the data, in other words by maximizing the posterior distribution. We find that the maximum of the posterior is given by the minimum of

$$\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} \tag{21}$$

Thus we see that maximizing the posterior distribution is equivalent to minimizing the regularized sum-of-squares error function encountered earlier in the form (9), with a regularization parameter given by $\lambda = \frac{\alpha}{\beta}$.

3 Results

3.1 Regression using error minimization of SSE

For this method, we get the fitting cuve as below

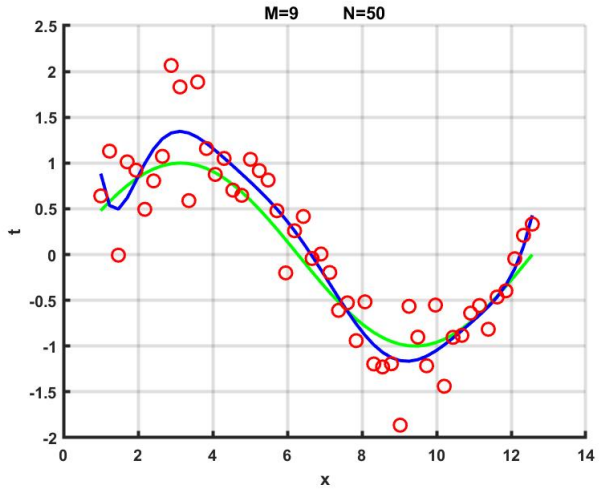

Figure 2: Plots of the solutions obtained by minimizing the sum-of-squares error function using the M = 9 polynomial for N = 50 data points

Futhermore, I fix the order of polynomial (M=9) and vary the number of sample points(N=15,100).

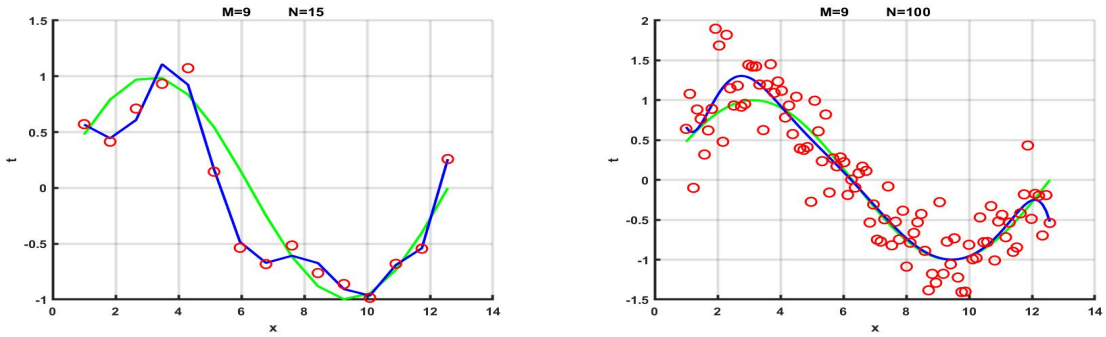

Figure 3: Plots of the solutions obtained by minimizing the sum-of-squares error function using the M = 9 polynomial for N = 15 data points (left plot) and N = 100 data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem

As we can see in the above plot, more data wiil result in a greater fitting curve. What's more, I fix the number of sample points and vary the order of polynomial M(M=0,1,3,6,9).Then I make a table to show the coefficient **w** under different order M.

	0	1	3	6	9
w_0^*	0.0693	1.3898	-0.7816	1.3372	12.4146
w_1^*		-0.1947	1.4667	-1.6323	-26.8202
w_2^*			-0.3114	1.2307	23.4287
w_3^*			0.0161	-0.3348	-10.4728
w_4^*				0.0399	2.7329
w_5^*				-0.0022	-0.4404
w_6^*				0.0000	0.0443
w_7^*					-0.0027
w_8^*					0.0001
w_9^*					-0.0000

Table 1: Table of the coefficients $\mathbf{w}^*$ for polynomials of various order.

3.2 Regression using error minimization with the regularization term

For this method, we getting the fitting cure as below

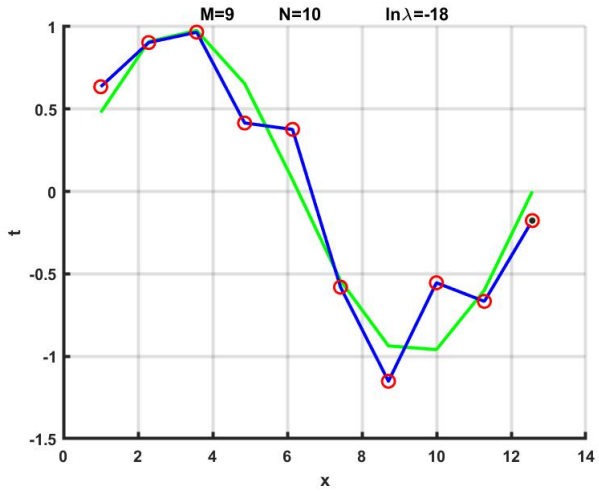

Figure 4: Plots of M=9 polynomials fitted to the data N=10 using the regularized error function for value of the regularization parameter λ corresponding to $ln\lambda = -18$

Then, I compare the different values of parameter λ, $ln\lambda = -18$, $ln\lambda = -15$, $ln\lambda = -13$,$ln\lambda = 0$

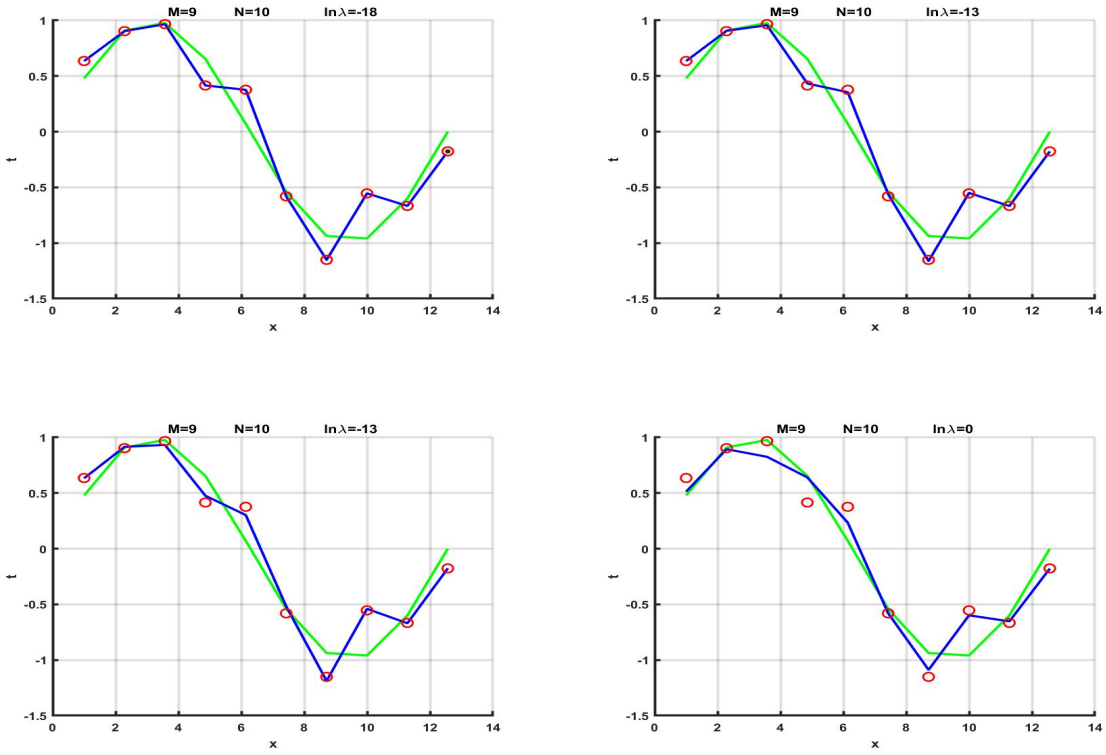

Figure 5: Plots of M=9 polynomials fitted to the data N=10 using the regularized error function for four values of the regularization parameter λ corresponding to $ln\lambda = -18$, $ln\lambda = -15$, $ln\lambda = -13$, $ln\lambda = 0$

But actually, from the plots above ,we can't see how parameter λ influence the fitting curve, cause λ is not large enough to govern the error function. Then I change λ corre-

sponding to $ln\lambda = 20$, which means λ will do a great influence on the fitting curve. And we can see the fitting curve is bad.

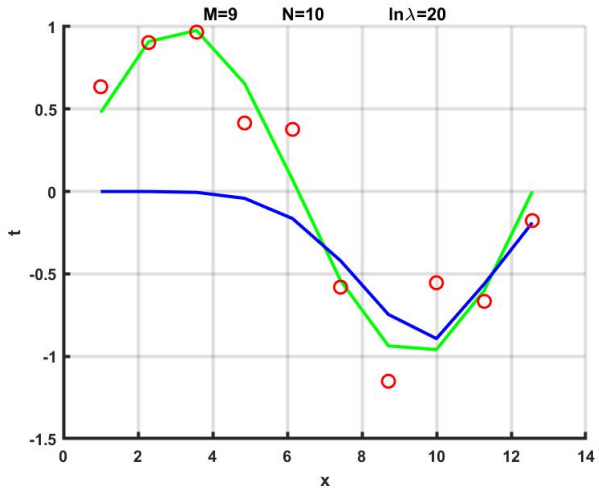

Figure 6: Plots of M=9 polynomials fitted to the data N=10 using the regularized error function for value of the regularization parameter λ corresponding to $ln\lambda = 20$

3.3 Regression using the ML (maximal likelihood) estimator of the Bayesian approach

As for this method, I get the fitting curve as below

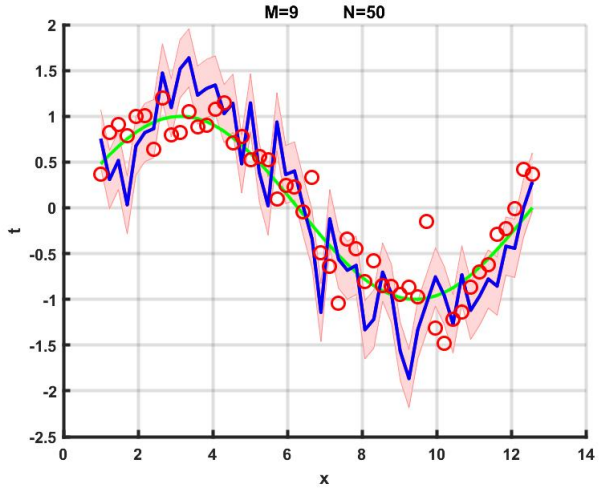

Figure 7: Plots of M=9 polynomials fitted to the data N=50 using the ML (maximal likelihood) estimator of the Bayesian approach

3.4 Regression using the MAP (maximum a posteriori) estimator of the Bayesian approach

As for this method, I get the fitting curve as below

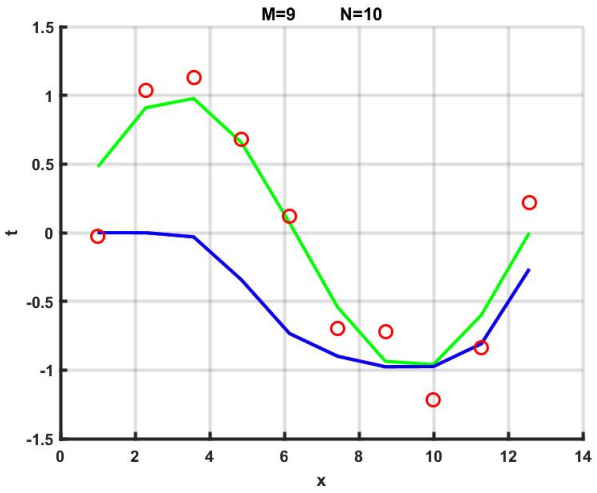

Figure 8: The predictive distribution resulting from a Bayesian treatment of polynomial curve fitting using an $M = 9$ polynomial, with the fixed parameters $\alpha = 5 \times 10^{36}$ and $\beta = 2.5 \times 10^{20}$

In this two methods, we can see the fitting curve is not good enough, in my opinion, it is for the reason that the value of x is much bigger than the value of t. Thus when I calculate the mean value, it is not accurate.

4 Conclusion

In order to solve the curve fitting problem, I use four methods. Each of them can solve the problem to some extent. For the method of SSE, which I used first, is not good enough, cause it cannot deal with the over-fitting problem. Accutally, in this problem, I think the method of error function is more useful, but the method of Bayesian approach is more helpful and principled in the later reserch of pattern recognition.

References

[1] Christopher M. Bishop. Pattern Recognition and Machine Learning 2006 Springer Science+Business Media, LLC

1, 3