EEL5840 Fundamental Machine Learning Homework 1

Hudanyun Sheng

Question 1 Solution:

MATLAB code is shown below:

```
upBound = 1; %set the upper bound for the training data
lowBound = 0; %set the lower bound for the training data
num = 100; % # of desired training data
step = (upBound - lowBound)/(num-1);
x_train = lowBound+step/2:step:upBound+step/2;
x_test = lowBound+step/2+0.1:step:upBound+step/2+0.1;
mu = 0;
sigma = 0.1;
e_train = normrnd(mu, sigma, [1, num]);
e_test = normrnd(mu, sigma, [1, num]);
t = (sin(2*pi*x_train) + e_train)'; % labels of the train data
t_test = (sin(2*pi*x_test) + e_test)';
max_M = 10;
ERMS_train = zeros(1, max_M+1);
ERMS_test = zeros(1, max_M+1);
for M = 0:max_M
    X = (x_train'.^ (0:1:M));
    X_test = (x_test'.^ (0:1:M));
    w = ((X'*X)\X')*t;
    y1 = sin(2*pi*x_train); % the true value of the sine function
    y2 = t'; % the true label of training data (sine function with error)
    y3 = (X*w)'; % the fitted polynomial
    y4 = (X_test*w)'; % the predicted value of test data
    y5 = t_test'; % the label of test set
    figure
```

```
    plot(x_train,y1', 'Linewidth', 3);
    hold on
    scatter(x_train,y2);
    hold on
    plot(x_train,y3, 'Linewidth', 3);
    title(['Training Data and Estimated Polynomial when M=' num2str(M)])
    legend('True Sine Function','Training Data', 'Estimated Polynomial')
    xlabel('x')
    ylabel('t')
    print(gcf,'-djpeg',[num2str(M),'train.jpeg']);
    hold off
    figure
    plot(x_test,y4, 'Linewidth', 3);
    hold on
    scatter(x_test, y5)
    title(['Test Data and Predicted Value when M=' num2str(M)])
    legend('Test Data', 'Predicted value of test')
    xlabel('x')
    ylabel('t')
    print(gcf,'-djpeg',[num2str(M),'test.jpeg']);
    hold off
    ERMS_train(M+1) = sqrt(mean((y2 - y3).^2));
    ERMS_test(M+1) = sqrt(mean((y4 - y5).^2));
end
figure(101)
plot(0:1:max_M, ERMS_train, 'o-', 'Linewidth', 3)
hold on
plot(0:1:max_M, ERMS_test, 'o-', 'Linewidth', 3)
legend('train', 'test')
```

```
xlabel('M')
ylabel('E_{RMS}')
print(gcf,'-djpeg','ERMS.jpeg');
hold off
```

I generated 100 training data from the range $[0, 1]$, based on the sine curve, with selected step size, and added zero-mean Gaussian noise to the training set. I generated the test data set in the same way, while the range changed to $[0.1, 1.1]$. And I changed the order of the model from 0 to 10, the selected results are shown below:

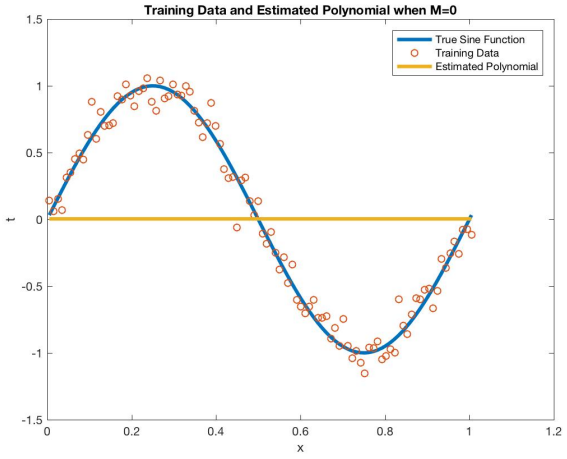

Figure 1: Training: M = 0

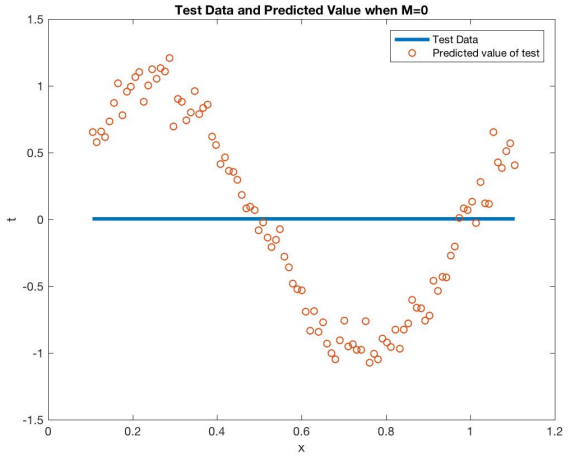

Figure 2: Test: M = 0

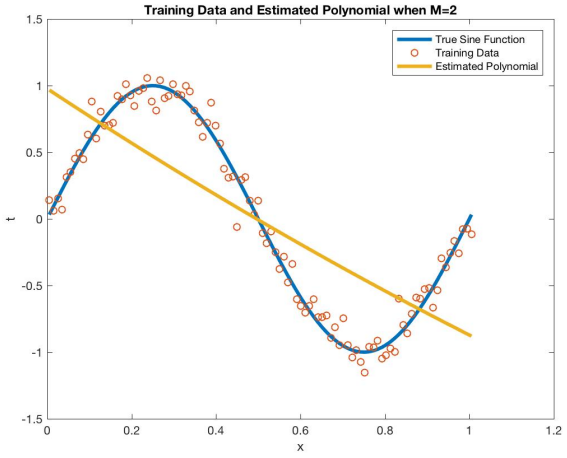

Figure 3: Training: M = 2

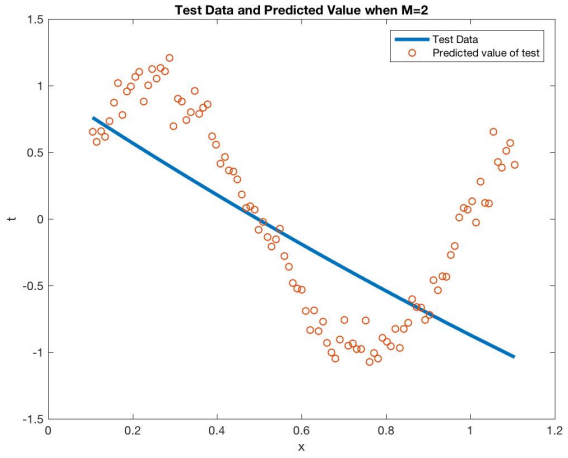

Figure 4: Test: M = 2

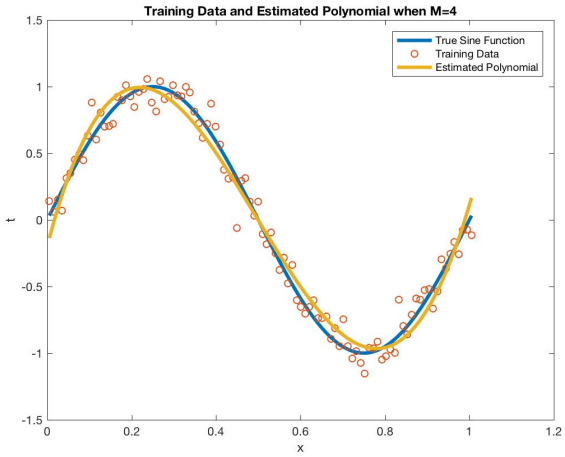

Figure 5: Training: M = 4

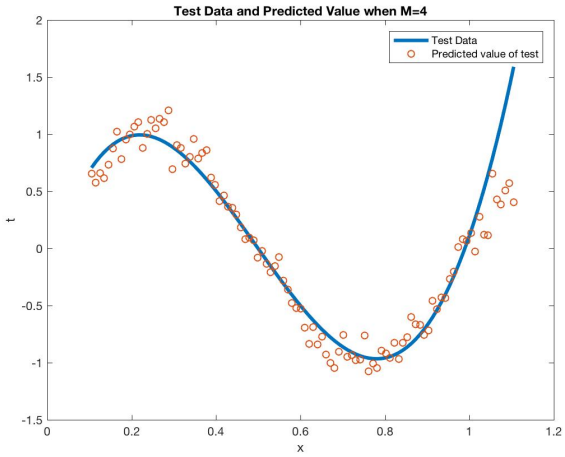

Figure 6: Test: M = 4

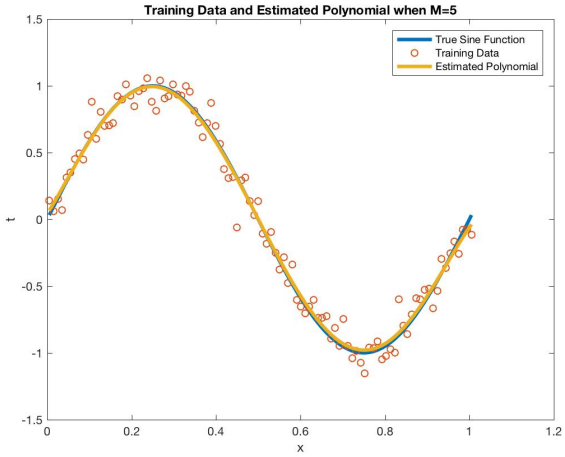

Figure 7: Training: M = 5

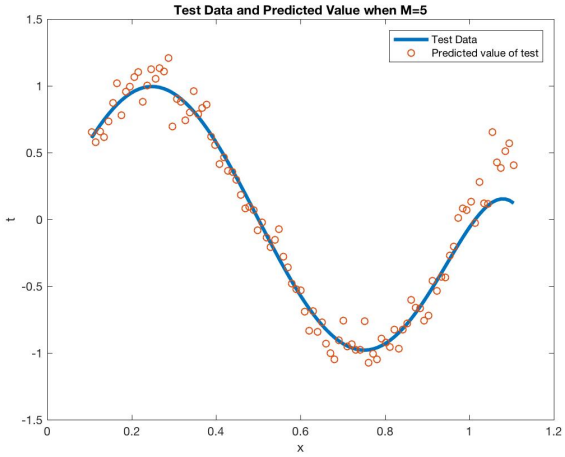

Figure 8: Test: M = 5

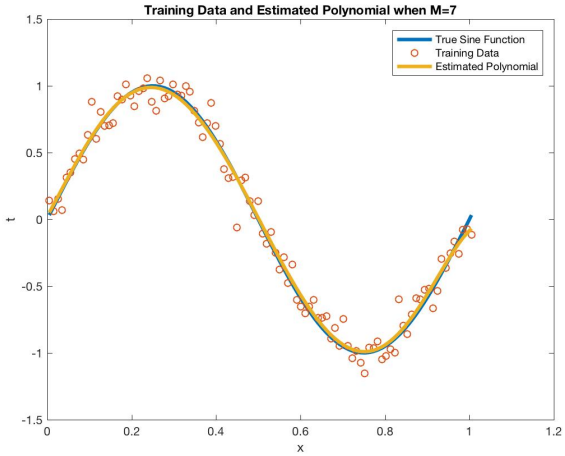

Figure 9: Training: M = 7

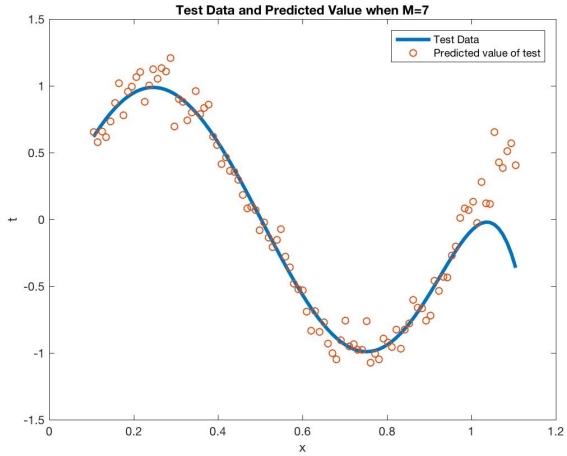

Figure 10: Test: M = 7

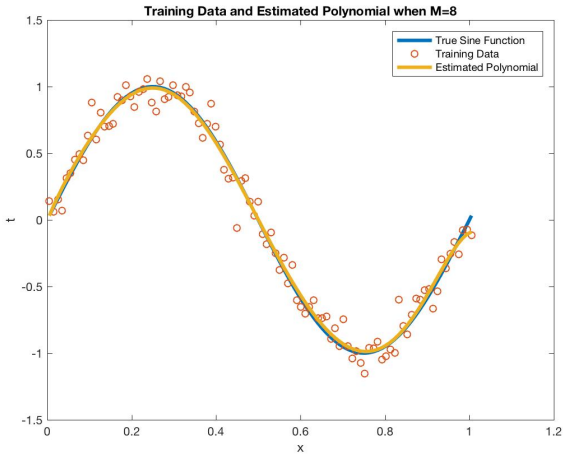

Figure 11: Training: M = 8

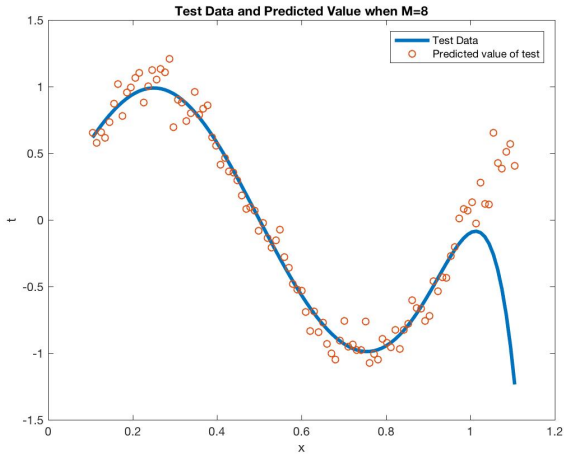

Figure 12: Test: M = 8

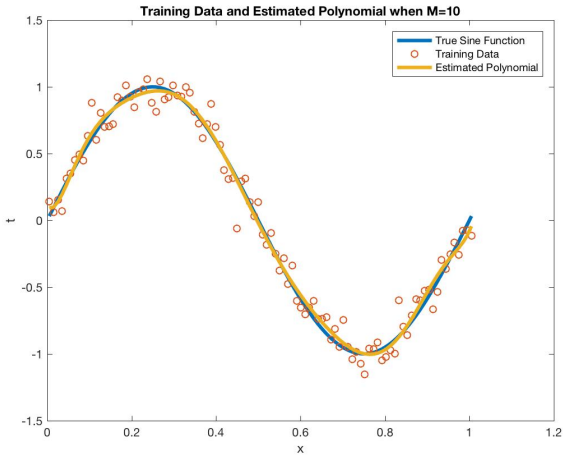

Figure 13: Training: M = 10

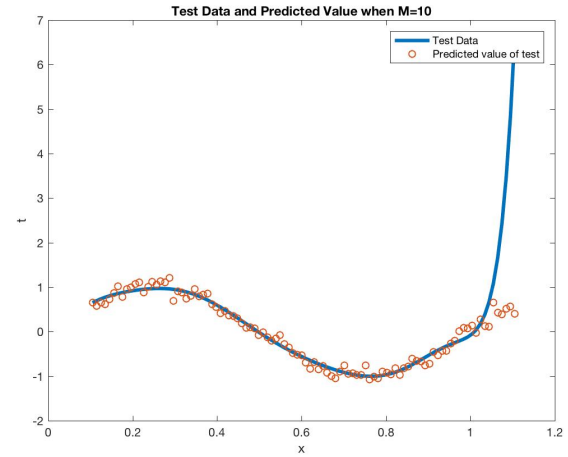

Figure 14: Test: M = 10

Actually, as the the model order M increases, it is intuitively by the graphs that the polynomial fits the training data better, while it fits the test data poorer every time. As it can also told through the graphs of the E_{RMS} evaluating for both training set and test set (Figure 15). RMS(root-mean-square) error is defined by: $E_{RMS} = \sqrt{2E(\boldsymbol{w})/N}$. If the model order is too small, say, $M \leq 2$, the polynomial curve fitting has the problem of underfitting, i.e., the model is unable to capture the underlying trend of the data. While if the model is quite large, the polynomial curve fitting has the problem of overfitting, i.e., the model fits the training set nearly perfect, but it is too complex and it has poor predictive performance for the test set(e.g. Figure 11, 12, 13 and 14).

Based on the results I got, I would say using model order $M = 5$ to get minimum of both training and test RMS error.

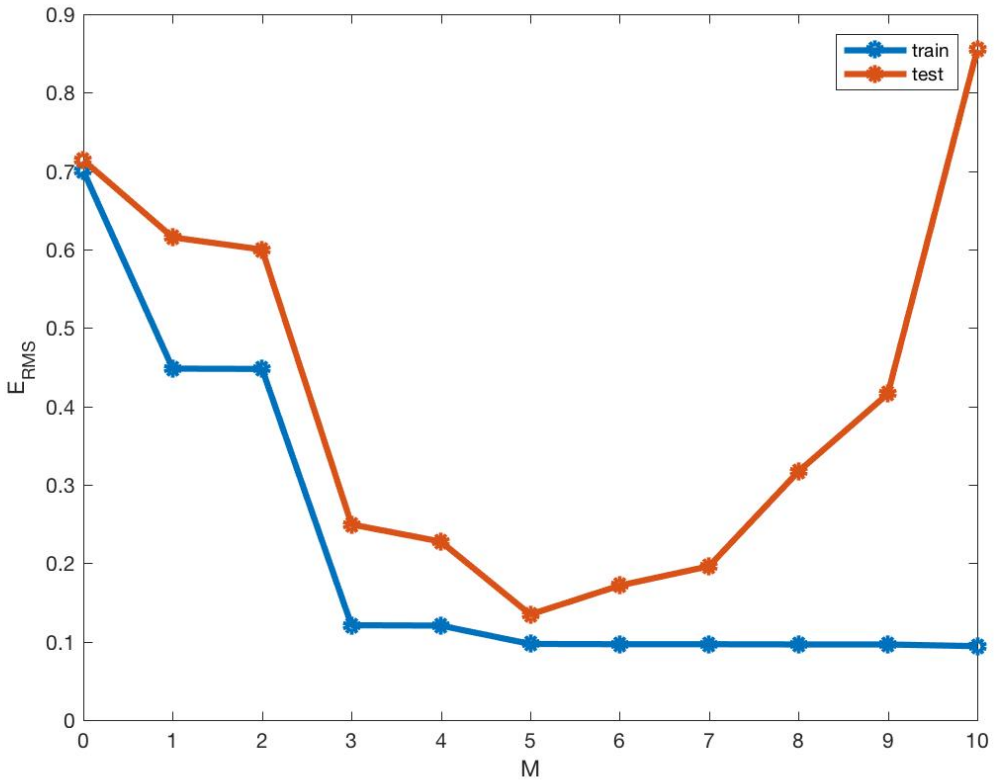

Figure 15: Graphs of the root-mean-square error evaluating for both training set and test set for various values of M

Question 2 Solution:

Based on the known sizes of matrices or vectors: $\mathbf{X}$: $M \times N$, $\mathbf{Y}$: $N \times N$, $\mathbf{a}$: $M \times 1$, $\mathbf{b}$: $N \times 1$

We got the sizes of corresponding transposes: $\mathbf{X^T}$: $N \times M$, $\mathbf{Y^T}$: $N \times N$, $\mathbf{a^T}$: $1 \times M$, $\mathbf{b^T}$: $1 \times N$.

And s is a scalar.

1. The operation $\mathbf{XY}$ is defined. The size of the resulting answer is $M \times N$.
2. The operation $\mathbf{YX}$ is undefined.
3. The operation $\mathbf{YX^T}$ is defined. The size of the resulting answer is $N \times M$.
4. The operation $\mathbf{aX}$ is undefined.
5. The operation $\mathbf{a^TX}$ is defined. The size of the resulting answer is $1 \times N$.
6. The operation $\mathbf{aX^T}$ is undefined.
7. The operation $\mathbf{a^Tb}$ is undefined.

8. The operation $\mathbf{b^T b}$ is defined. The resulting answer is a scalar.

9. The operation $\mathbf{b b^T}$ is defined. The size of the resulting answer is $N \times N$.

10. The operation $s\mathbf{X} + \mathbf{Y}$ is undefined.

Question 3 Solution:

$\because rank(X) = r, \therefore rank(X^H) = r, \therefore rank(XX^H) = rank(X^H X) = r$. While if XX^H has the size $l \times l$, $X^H X$ may have the size $n \times n$ (XX^H and $X^H X$ do not have to have the same size).

Suppose XX^H has the eigenvalue λ and the eigenvector $\mathbf{u}$, we got $XX^H\mathbf{u} = \lambda\mathbf{u}$.

If we multiply X^H to the left of both sides of the above equation, we got: $X^H XX^H\mathbf{u} = X^H\lambda\mathbf{u}$.

Which is equivalent to $(X^H X)X^H\mathbf{u} = \lambda X^H\mathbf{u}$. Thus we can conclude that $X^H X$ and XX^H have the same eigenvalue λ. And if XX^H has the eigenvector $\mathbf{u}$, $X^H X$ has the eigenvector $X^H\mathbf{u}$.

Question 4 Solution:

1. $\frac{\partial f}{\partial \mathbf{x}} = 3(\frac{\partial \mathbf{x^T x}}{\partial \mathbf{x}}) + 4(\frac{\partial \boldsymbol{y}^T\boldsymbol{x}}{\partial \boldsymbol{x}}) - (\frac{\partial 1}{\partial \boldsymbol{x}}) = 3(\boldsymbol{x} + (\boldsymbol{x}^T)^T) + 4((\boldsymbol{y}^T)^T) + 0 = 6\boldsymbol{x} + 4\boldsymbol{y}$.
2. $\frac{\partial^2 f}{\partial \boldsymbol{x}^2} = \frac{\partial(6\boldsymbol{x} + 4\boldsymbol{y})}{\partial \boldsymbol{x}} = \mathbf{6} = 6\boldsymbol{I}$, $\boldsymbol{I}$ is the identity matrix.

Question 5 Solution:

1. $\frac{\partial f}{\partial \mathbf{x}} = -10[\mathbf{Qx} + (\mathbf{x^T Q})^\mathbf{T}] + (\mathbf{4y^T})^\mathbf{T} = -\mathbf{10}[\mathbf{Qx} + \mathbf{Q^T x}] + \mathbf{4y}$.

   $\because \mathbf{Q}$ is a symmetric matrix,

   $\therefore \mathbf{Q^T} = \mathbf{Q}$.

   $\therefore \frac{\partial f}{\partial x} = -20\boldsymbol{Q}\boldsymbol{x} + 4\boldsymbol{y}$.

2. $\frac{\partial^2 f}{\partial \boldsymbol{x}^2} = \frac{\partial(-20\boldsymbol{Q}\boldsymbol{x} + 4\boldsymbol{y})}{\partial \boldsymbol{x}} = -20\boldsymbol{Q}^T = -20\boldsymbol{Q}$.