Lecture Outline

Reading: Chapter 1

- Example: Polynomial Curve Fitting

1.1 Example: Polynomial Curve Fitting

Introduction

Suppose we observe a real-valued input variable x and we wish to use this observation to predict the value of a real-valued, continuous target variable t, i.e. regression.

Now, suppose we are given a training set comprising N observations of x written $\mathbf{x} \equiv (x_1, \ldots, x_N)^T$ together with corresponding (noisy) target values of t written $\mathbf{t} \equiv (t_1, \ldots, t_N)^T$. Fig. 1 shows a plot of a training set comprising $N = 10$ data points. The input data $\mathbf{x}$ was generated by choosing $0 \leq x_n \leq 1$, and the target data $\mathbf{t}$ was generated with $t_n = \sin(2\pi x_n) + v_n$ where v_n is additive noise.

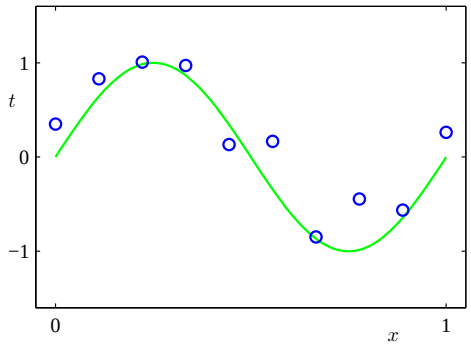


Figure 1:

Our goal is to exploit this training set in order to make predictions of the value $\hat{t}$ of the target variable for some new value $\hat{x}$ of the input variable [we do not assume knowledge of the underlying function $\sin(2\pi x_n)$].

Figure 2: Training and Test Stages

Curve Fitting

In this simple example, we consider an approach based on *curve fitting*. In particular, we shall fit the data using a polynomial function (model) of the form

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M \tag{1}$$

where M is the *order* of the polynomial (model) and $\mathbf{w} = [w_0, \ldots, w_M]^T$ is the vector of polynomial (model) coefficients. Note that although $y(x, \mathbf{w})$ is a nonlinear function of x it is a linear function of $\mathbf{w}$.

Definition: Functions which are linear in the unknown parameters are called *linear models*.

The coefficient values will be determined by fitting the polynomial to the training data. This can be done by minimizing an *error function* or *cost function* that measures the misfit between the function $y(x, \mathbf{w})$, for any given value of $\mathbf{w}$ and the training data.

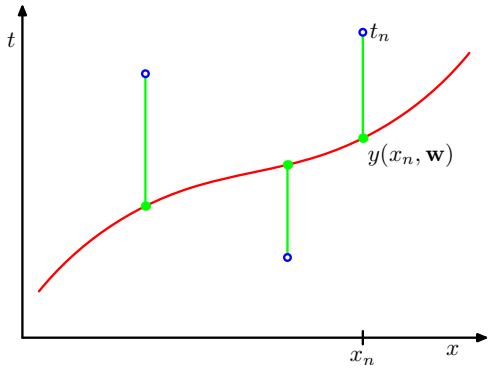


Figure 3:

One simple choice of error function is the total squared error

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

This error function is nonnegative and equal to zero if and only if $y(x, \mathbf{w})$ passes through each of the training data points (see Fig. 3). We solve the curve fitting problem by choosing $\mathbf{w}$ which minimizes $E(\mathbf{w})$. Because the error function is a quadratic function of $\mathbf{w}$, its derivatives with respect to $\mathbf{w}$ will be linear and so the minimization problem has a unique, closed-form solution $\mathbf{w}^*$. The actual solution for $\mathbf{w}^*$ is given in Prob. 1.1; a more general solution is given in Section 3.1.

Model Order Selection

There remains the problem of choosing the order M of the polynomial (model). In Fig. 4, we notice for

- $M = 0$ and $M = 1$ the polynomial provides a poor fit
- $M = 3$ polynomial appears to provide the best fit to the function $\sin(2\pi x)$ of the examples shown.
- $M = 9$ we obtain excellent fit to the training data (polynomial passes through each point) but the polynomial is a poor fit to the underlying process $\sin(2\pi x)$. This behavior is known as *over-fitting*.

We can obtain some quantitative insight into the dependence of the generalization performance on M by considering a separate test set comprising of 100 data points generated according to $t_n = \sin(2\pi x_n) + v_n$.

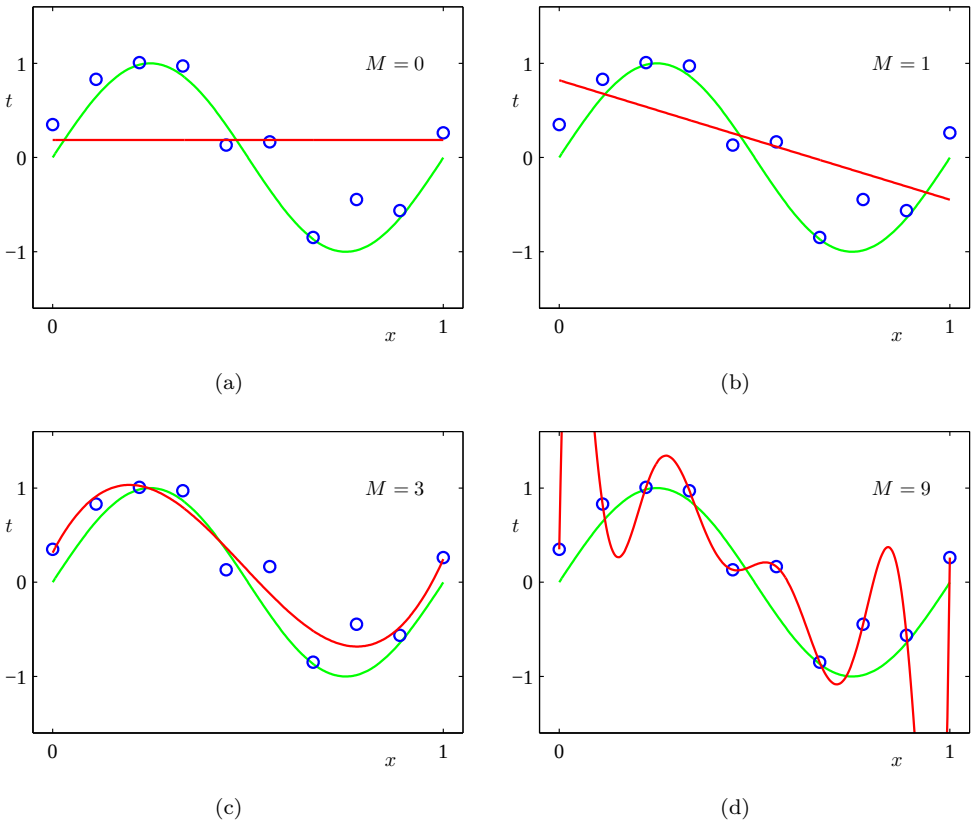


Figure 4:

For each choice of M and $\mathbf{w}^*$ (determined by training data), we measure $E(\mathbf{w}^*)$ given by (2) for each of the training and test datasets. Alternately, we can compute the root-mean-square error by

$$E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N} \tag{3}$$

in which the division by N allows us to compare different sizes of data sets on equal footing. Graphs of the training and test set RMS errors are shown for various M in Fig. 5. The test results measure how well we are doing in predicting values of t for new data observations x, i.e. generalization.

We notice that for

- $M \leq 2$, the polynomials give large E_{RMS} since the polynomial does not have enough degrees of freedom to model the sinusoid's oscillations
- $3 \leq M \leq 8$, the polynomial achieves the smallest E_{RMS} and fits the sinusoid well
- $M \geq 9$, the E_{RMS} is large since $y(x, \mathbf{w})$ exhibits wild oscillations which do not fit the sinusoid.

For $M = 9$, despite the large E_{RMS} for the test data, the E_{RMS} for the training data is zero, since we expect with a polynomial with 10 coefficients to exactly fit 10 data points. One would expect that since $\sin(2\pi x)$ can be expanded as a power series, results should improve as M increases. The problem is that with such a limited number of training data points (10 in this case), as M increases, the polynomial is becoming *increasingly tuned to the noise* which can be seen by examining the coefficients and noting their large values.

For a given model complexity, the over-fitting problem becomes less severe as the size of the data set increases as shown in Fig. 6.

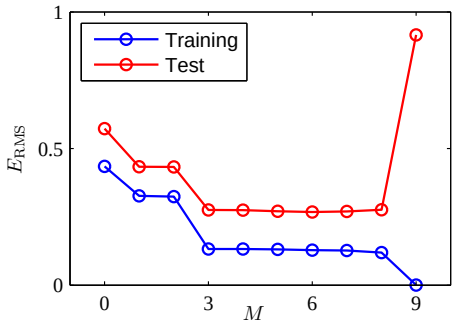
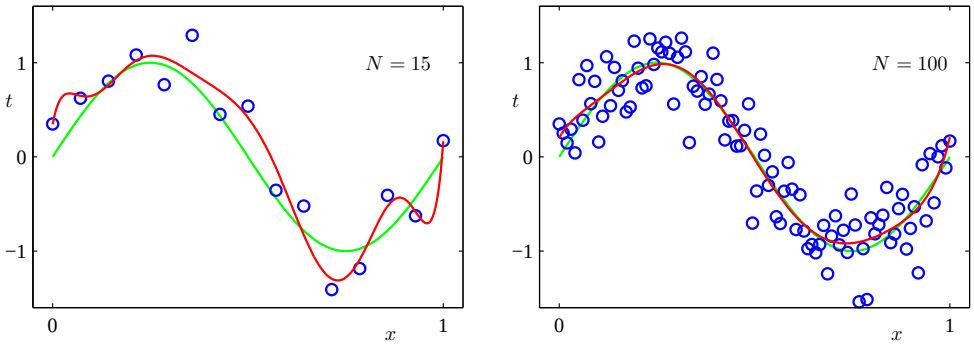


Figure 5:



(a) (b)

Figure 6: $M = 9$

Regularization

One technique that is often used to control the over-fitting problem (large coefficient values) is that of *regularization*, which adds a penalty term to the error function in order to discourage the coefficients from reaching large values:

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}[y(x,\mathbf{w}) - t_n]^2 + \frac{\lambda}{2}||\mathbf{w}||^2 \tag{4}$$

where $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$ and the value λ governs the relative importance of the regularization term compared with the squared error term. For a given set of training data, the coefficients $\mathbf{w}$ in (4) can be solved in closed form. Results of regularization are shown in Fig. 7. The actual solution for $\mathbf{w}^*$ with regularization is given in Prob. 1.2; a more general solution is given in Section 3.1.

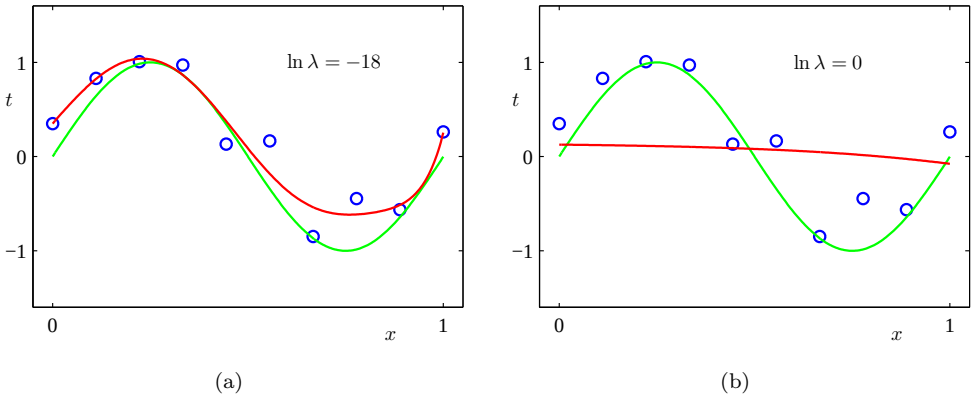


Figure 7: Illustrations of relatively light regularization ($\ln \lambda = -18$ or $\lambda = 1.5 \times 10^8$) and relatively heavy regularization ($\ln \lambda = 0$ or $\lambda = 1$) for $M = 9$.

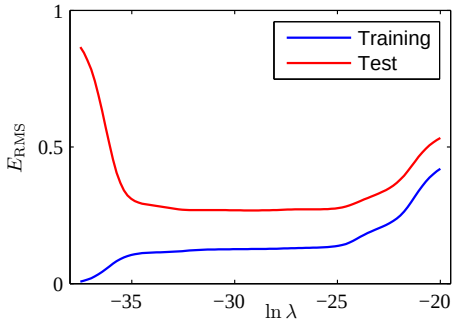


Figure 8: E_{RMS} vs. $\ln \lambda$ for $M = 9$