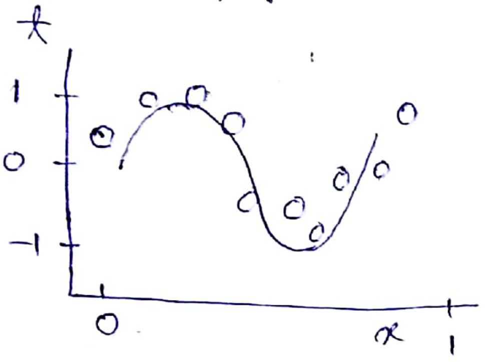
1.1 Example: Polynomial curve fitting

We shall use a simple regression problem to motivate a number of key concepts. Suppose we are given a training set comprising N observations of x, written as $\mathbf{x} \equiv (x_1, x_2, \ldots x_N)^T$ together with corresponding observations of the values of t, denoted as $\mathbf{t} \equiv (t_1, t_2 \ldots t_N)^T$. The data is distributed as shown in fig. 1.



Plot shows training data of $N = 10$ points shown in circles each comprising an observation of the input variable x along with t. The curve shows the function $\sin(2\pi x)$ used to generate data. The points are generated by adding normal noise to the points on curve.

fig-1.

Our goal is to predict $\hat{t}$ of the target variable for some new $\hat{x}$ of the input variable. This is a difficult problem as we need to generalise from a finite set of data points. observed data is corrupted by noise so we'll discuss probability in next section for capturing the uncertainty. for the moment, we proceed with this simple approach of curve fitting,

$$y(x, \mathbf{W}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j \qquad \ldots(1.1)$$

where M is the power (or order) of polynomial.

The polynomial coefficients $w_0, w_1 \ldots w_M$ are collectively denoted by vector $\mathbf{W}$.

We can fit the polynomial by minimizing the error function that measures the misfit between the function and data. We can use sum of the squares of the errors between the prediction and true value for datapoints as error function, so,

$$E(\mathbf{W}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{W}) - t_n\}^2 \qquad \ldots(1.2)$$

where ½ factor is used for mathematical convenience.

We can solve the problem of curve fitting by choosing the value of W for which E(W) is as small as possible.