INM431: Machine Learning

Curve fitting and fundamentals

Pranava Madhyastha (pranava.madhyastha@city.ac.uk)

Based on materials by Artur S. d'Avila Garcez + Bishop's book

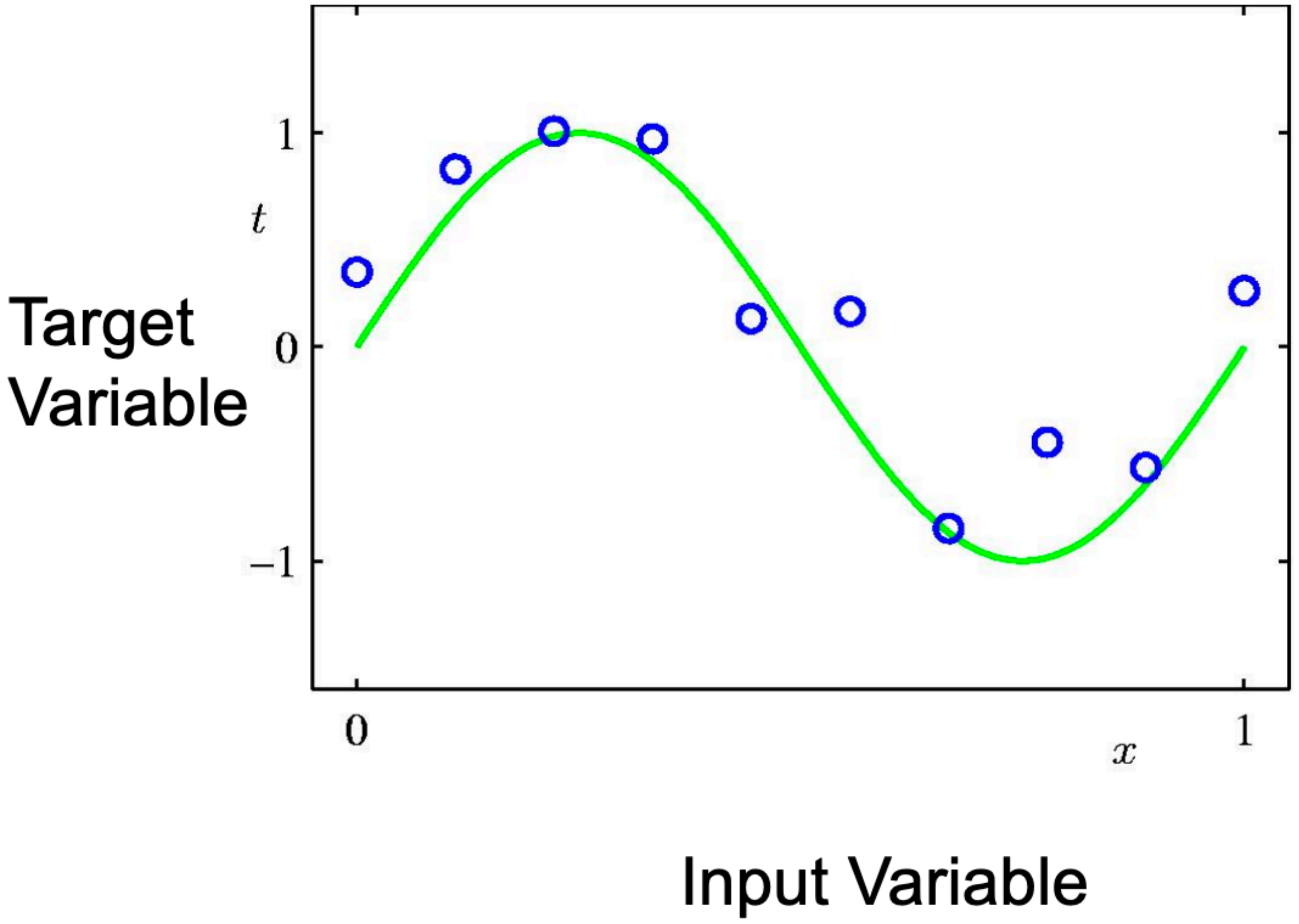
Synthetic data

Data generated from the function: $\sin(2\pi x)$

(x is the input value)

+

Random noise in target values



Input values $\{x_n\}$ generated uniformly in range $(0,1)$. Corresponding target values $\{t_n\}$obtained by first computing corresponding values $\sin(2\pi x)$ of then adding random noise with a Gaussian distribution with std.deviation of 0.3

Training set

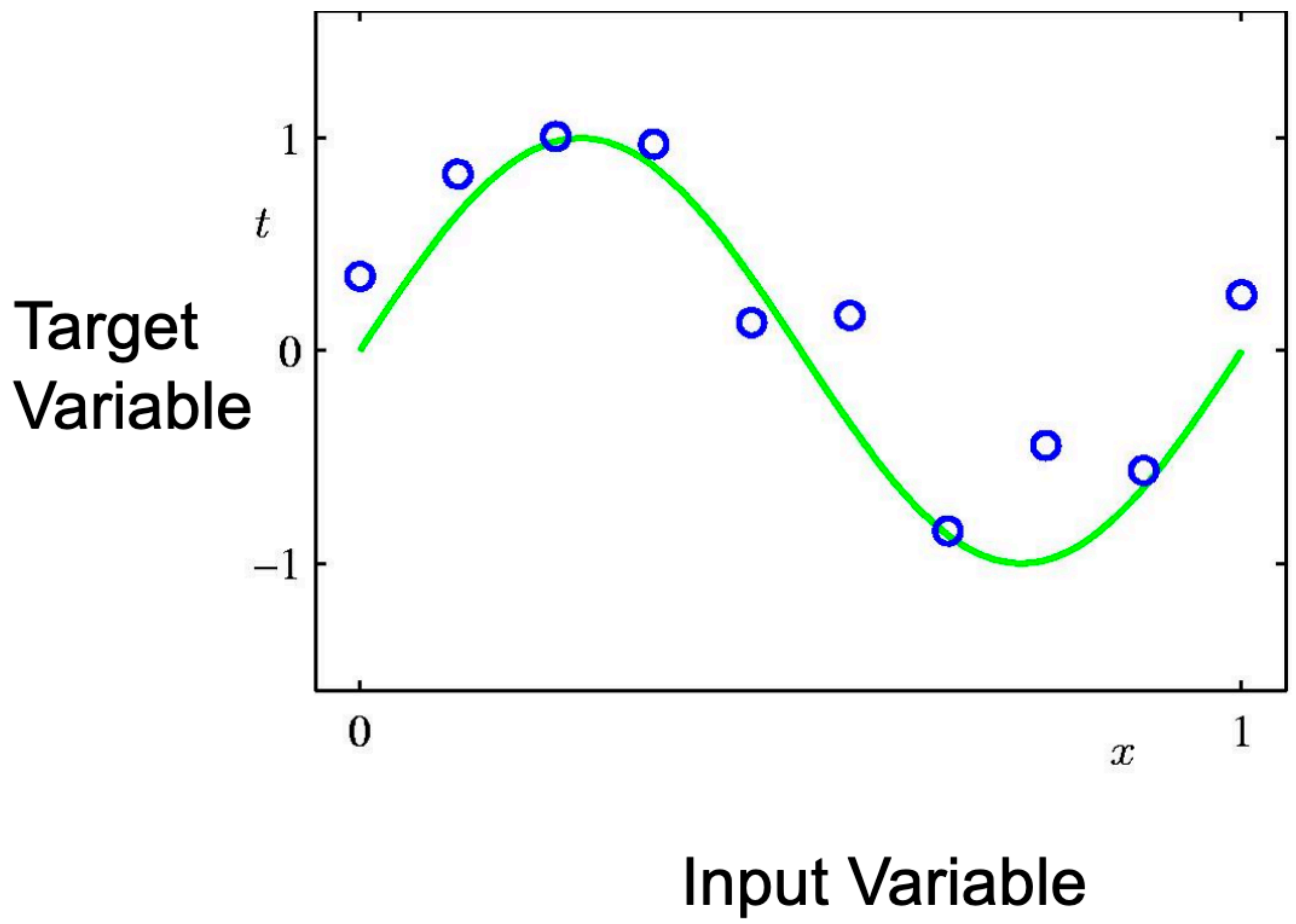
N observations of x

$x = (x_1, \cdots, x_n)$

$t = (t_1, \cdots, t_n)$

Goal is to exploit training set to predict $\hat{t}$ for some new value $\hat{x}$

Inherently a difficult problem

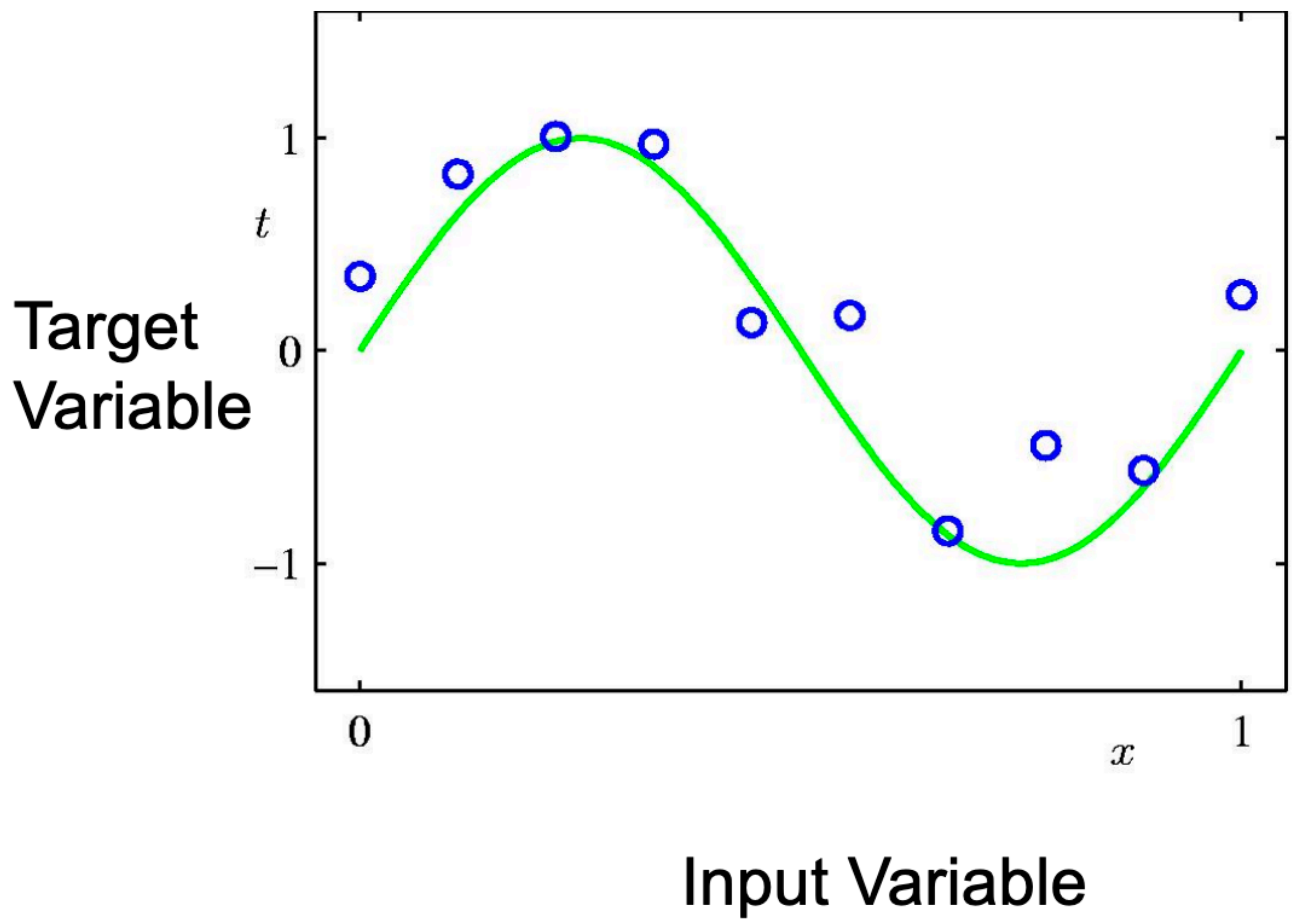


Polynomial function

Fit the data using a polynomial function

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

M is the order of the polynomial



Would higher orders of polynomials fit better?

Would that be the right choice?

Coefficients $w_0, \cdots, w_M$ are collectively denoted by vector $\mathbf{w}$

It is a nonlinear function of x, but a linear function of the unknown parameters w

How do we select the best function?

Error/loss function!

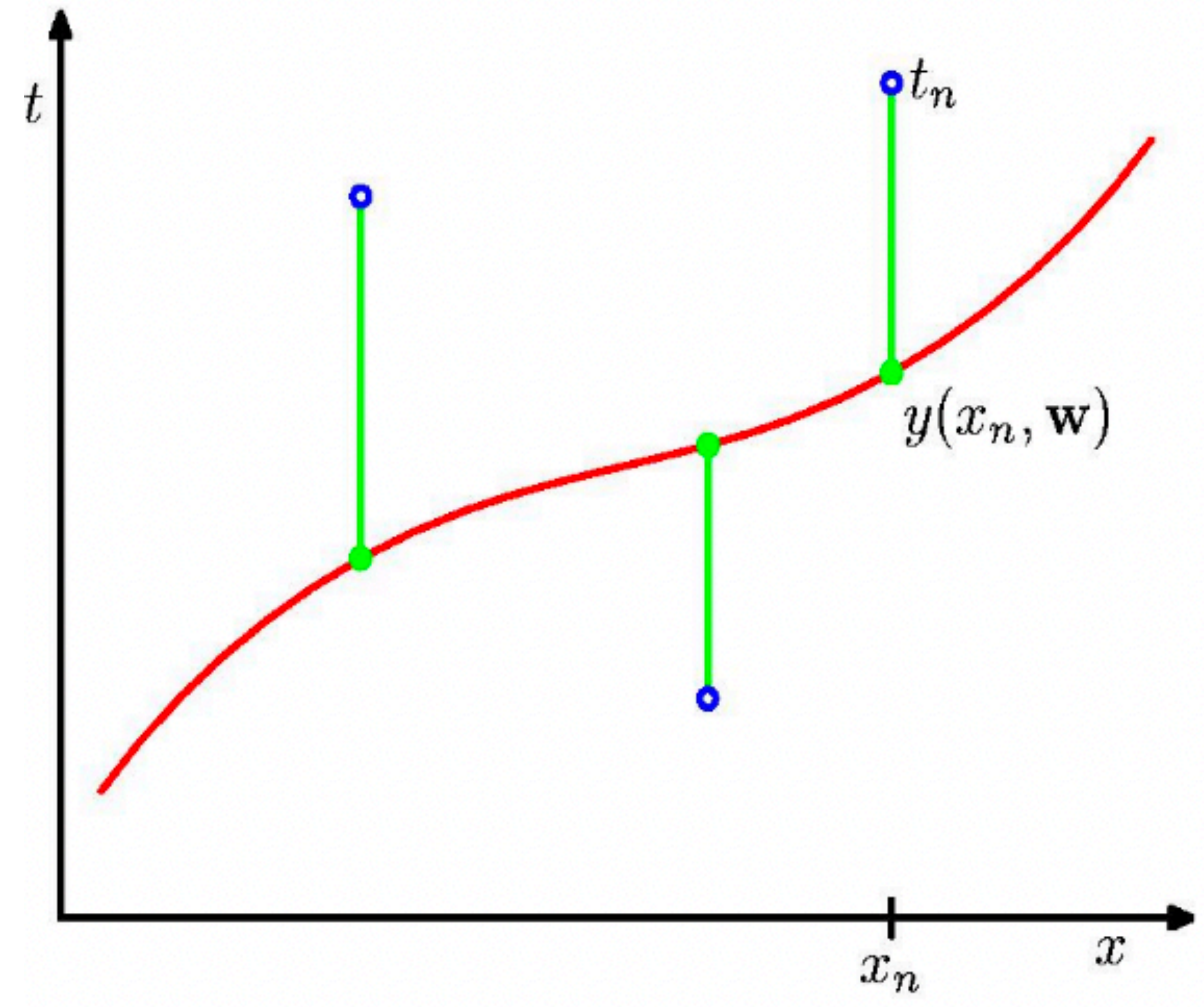
We can obtain a fit by minimising an error function

Sum of squares of the errors between the predictions $f(x_n, w)$ for each data point $x_n$and target value t_n:

$$L(w) = \frac{1}{2}\sum_{n=1}^{N}(f(x_n, w) - t_n)^2$$

Solve by choosing value of w for which the error is as small as possible

Red line is best polynomial fit



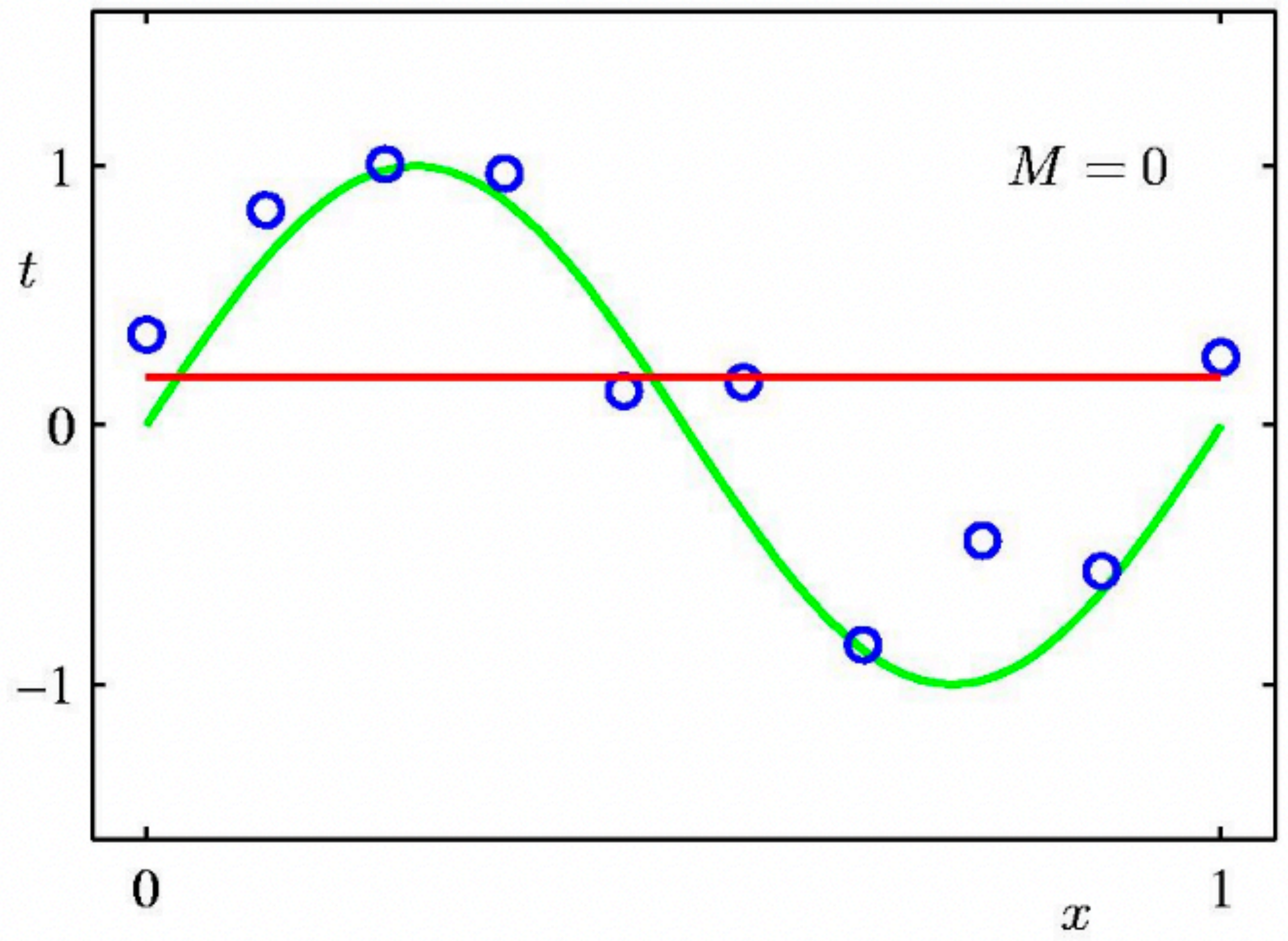
Minimising the loss

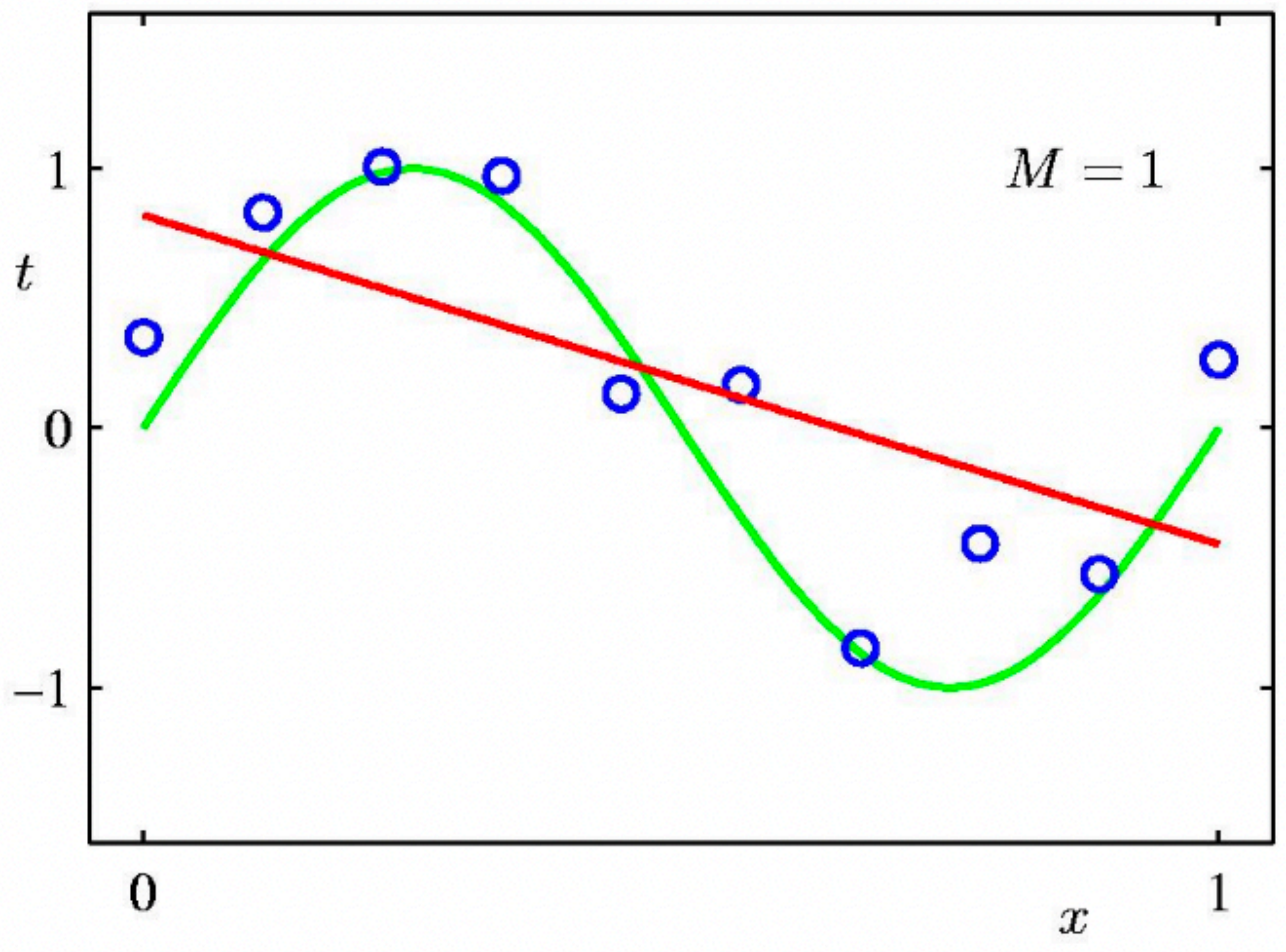
Loss function is a quadratic w

The derivative with respect to coefficients will be linear in w

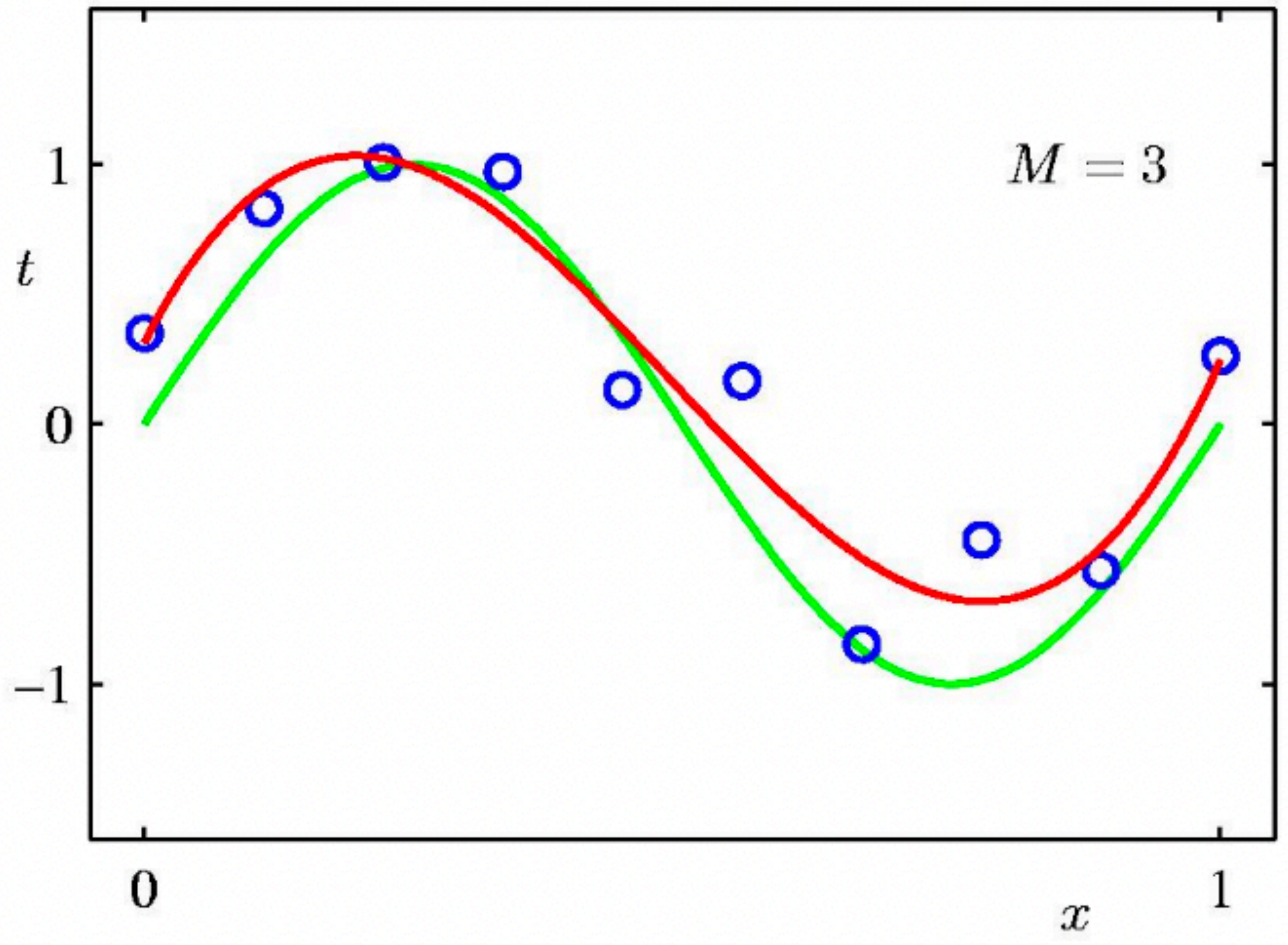
The loss function has a unique solution which can be found with a unique minimum

Best fit: model selection

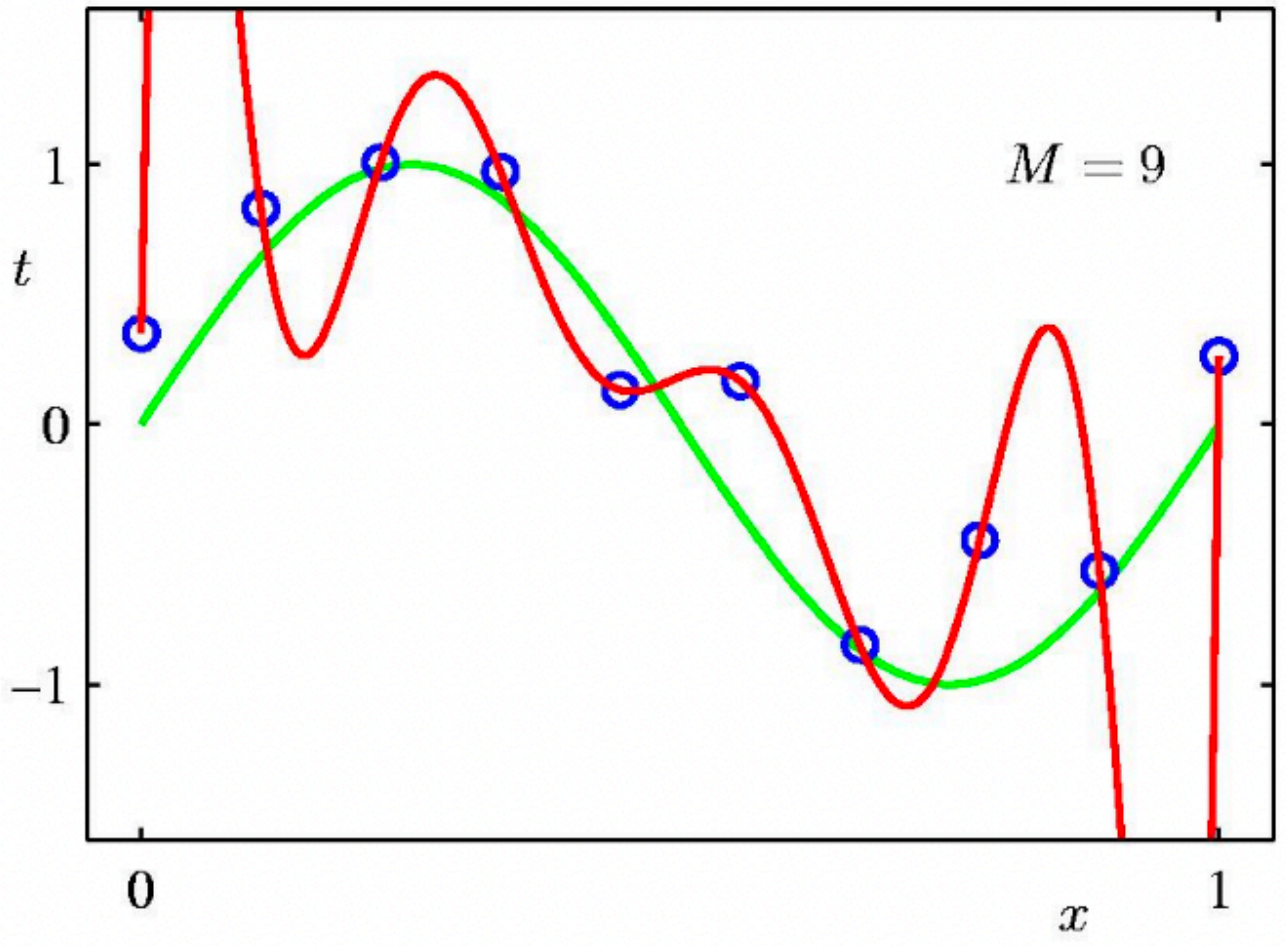




← Poor representations of $\sin(2\pi x)$



← Best Fit to $\sin(2\pi x)$



Over Fit
Poor representation of $\sin(2\pi x)$

Generalisation

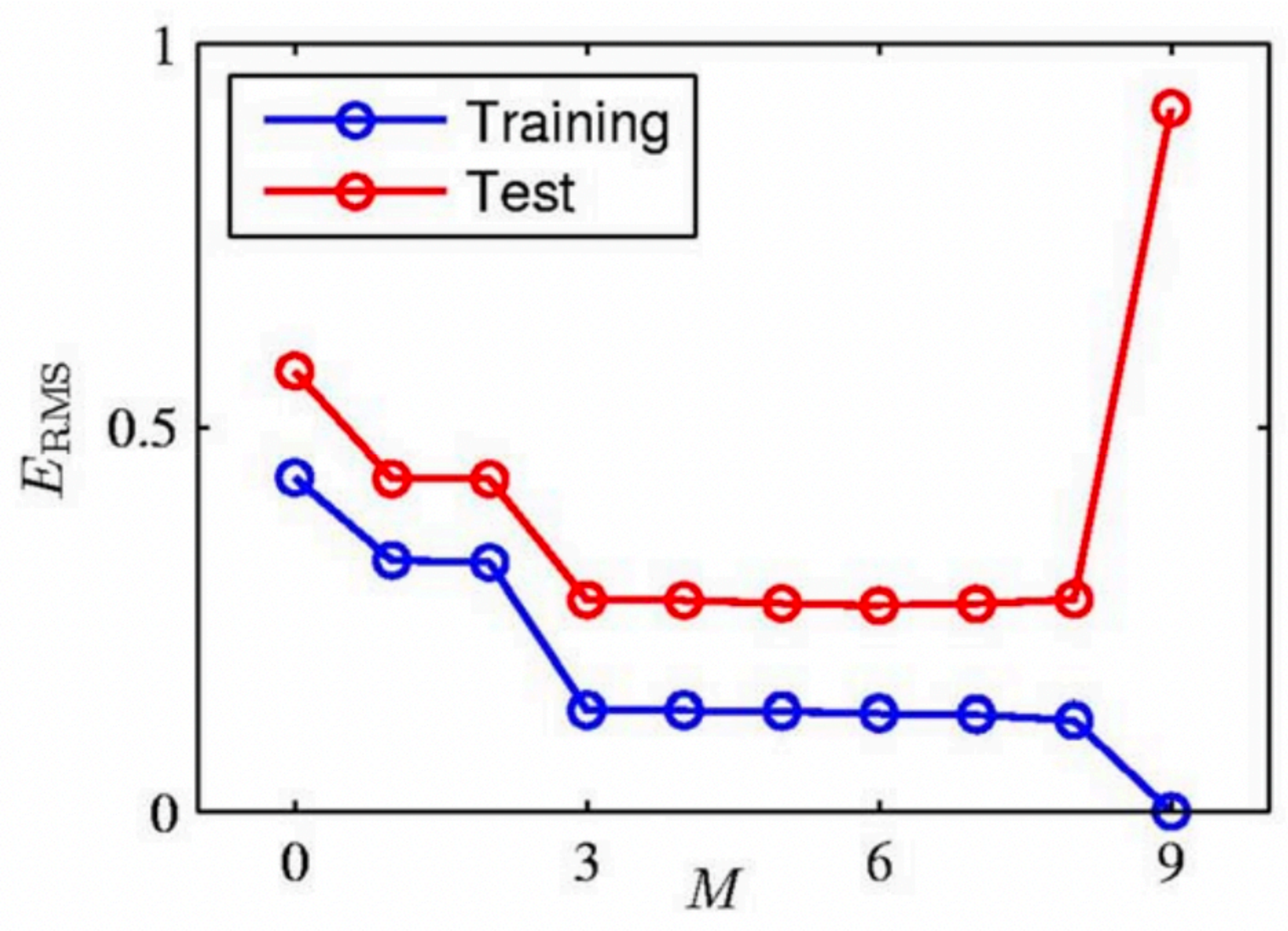
Let us take a separate unseen (during fit) dataset of 100 points

For each value of M evaluate

$$L(w^*) = \frac{1}{2}\sum_{n=1}^{N}(f(x_n, w^*) - t_n)^2$$

Do this for both training data and test data

The RMS error: $E_{RMS} = \sqrt{\frac{2 \times L(w^*)}{N}}$

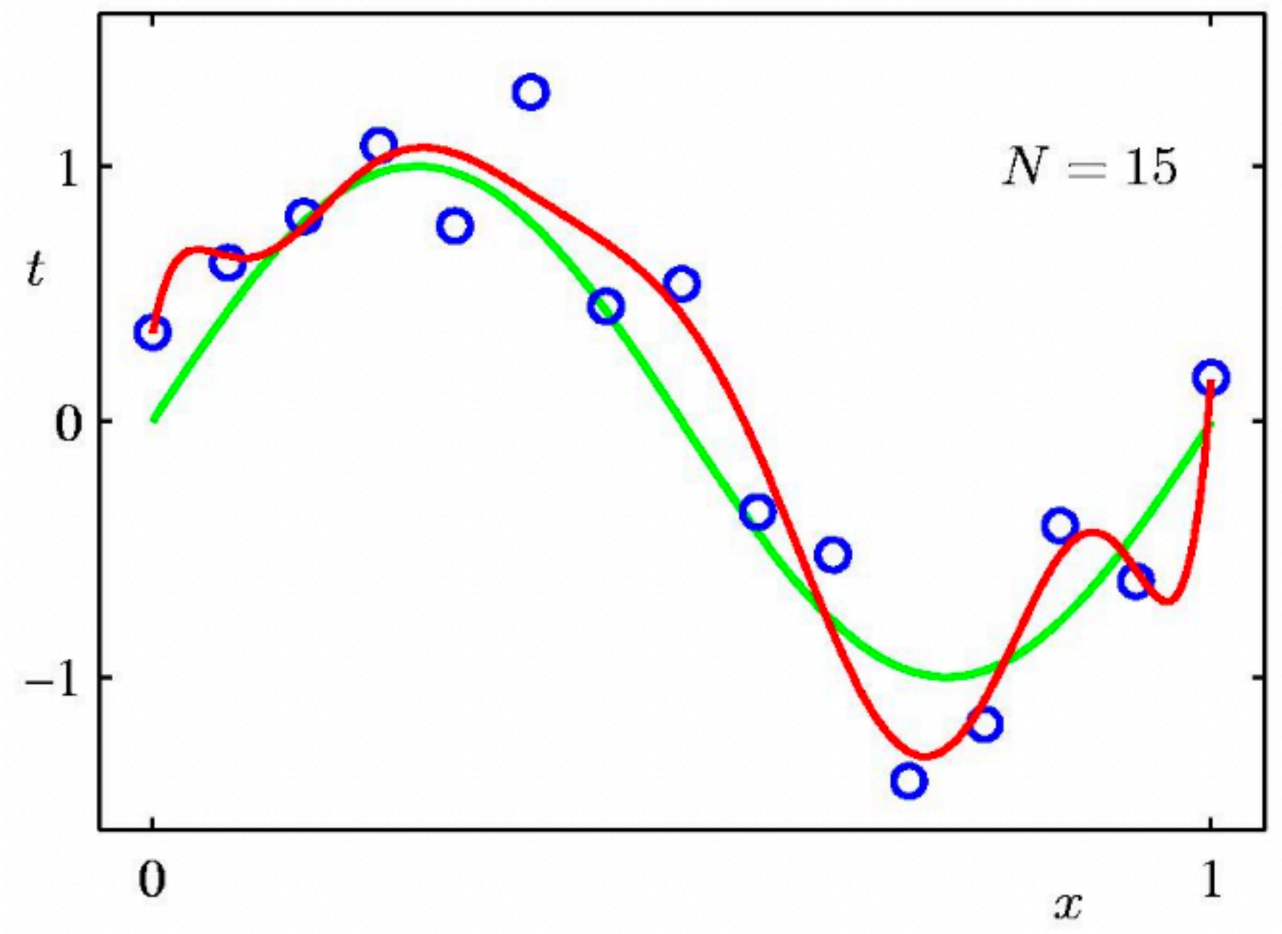


Best fit: model selection

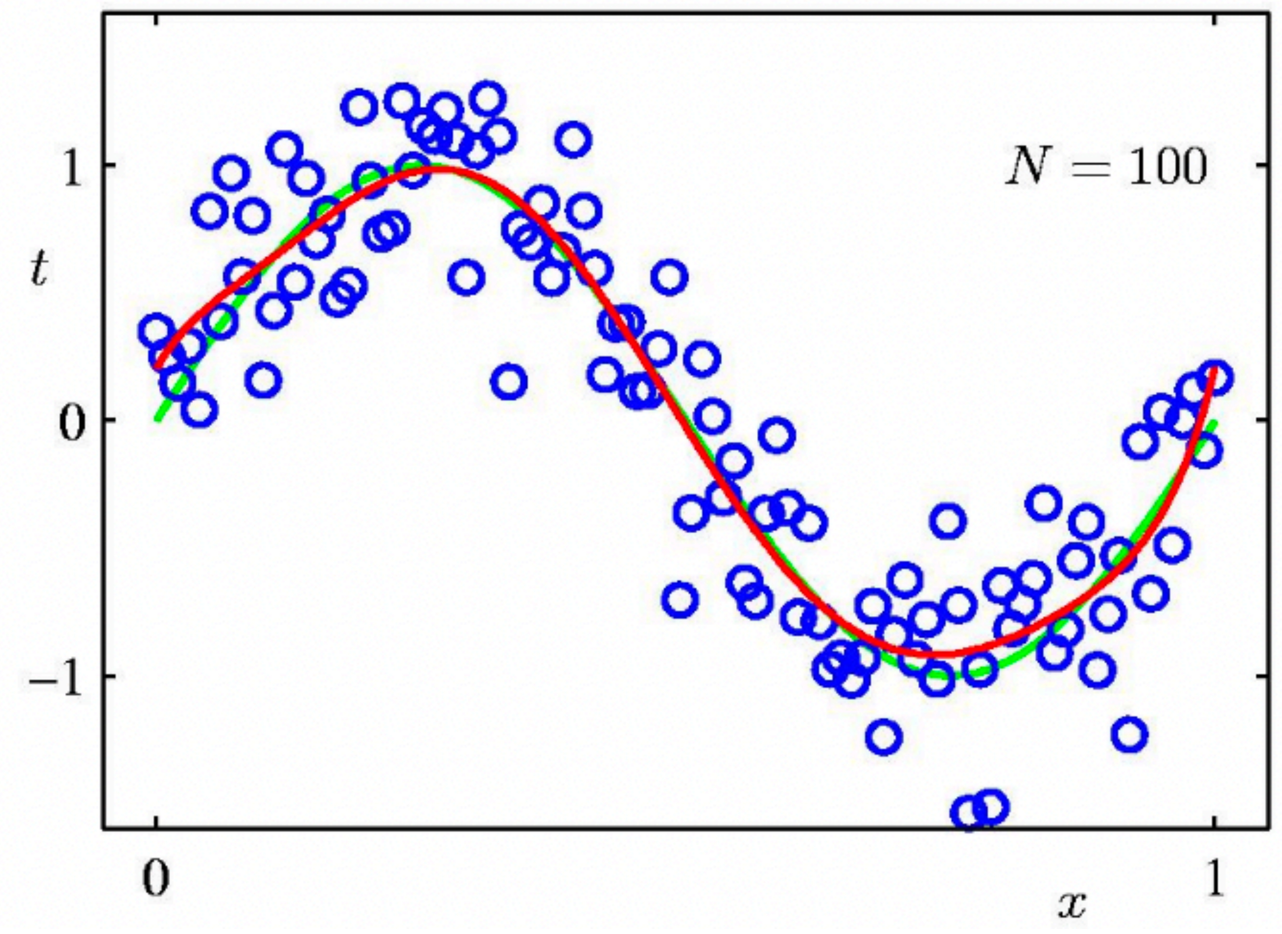
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

What happens when we increase data?

Overfitting problem is less severe as size of data set increases



Larger data set allows us to fit more complex functions



Least squares

$$L(w) = \frac{1}{2}\sum_{n=1}^{N}(f(x_n, w) - t_n)^2$$

Model complexity ∝ problem complexity

Maximising the likelihood of the data

Regularisation of Least Squares

We can exploit relatively complex models with data sets of limited size

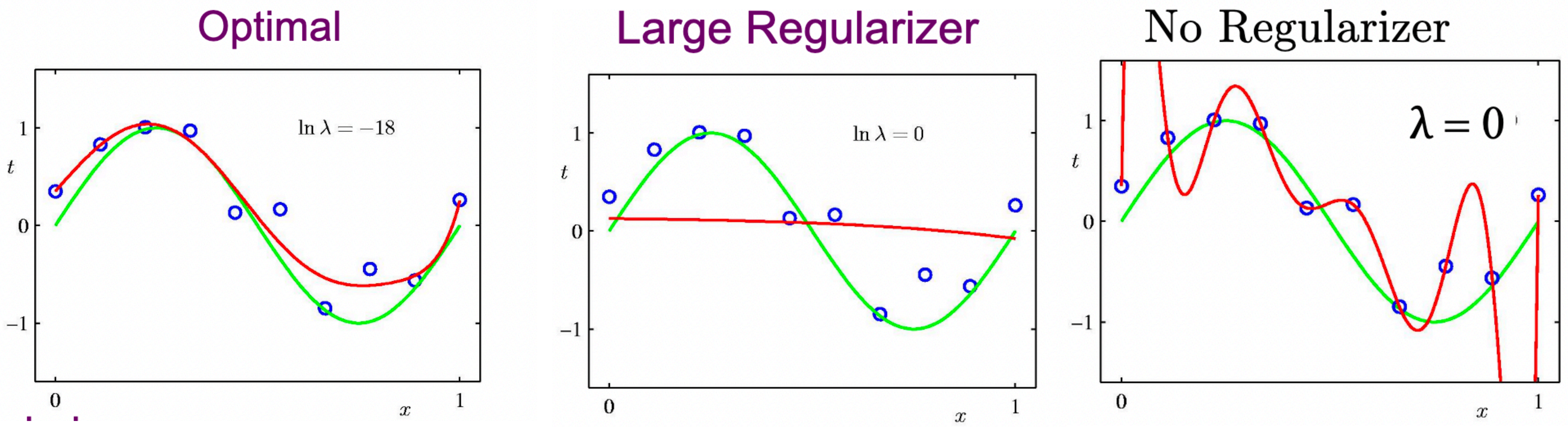
Use a penalty term with the loss function to discourage coefficients from reaching large values (regularisation/shrinkage/weight decay):

$$\hat{L}(w) = \frac{1}{2}\sum_{n=1}^{N}(f(x_n, w) - t_n)^2 + \frac{\lambda}{2}||w||^2$$

where, $||w||^2 = w^\top w = (w_0^2 + w_1^2 + \cdots + w_m^2)$

Effect of regularisation

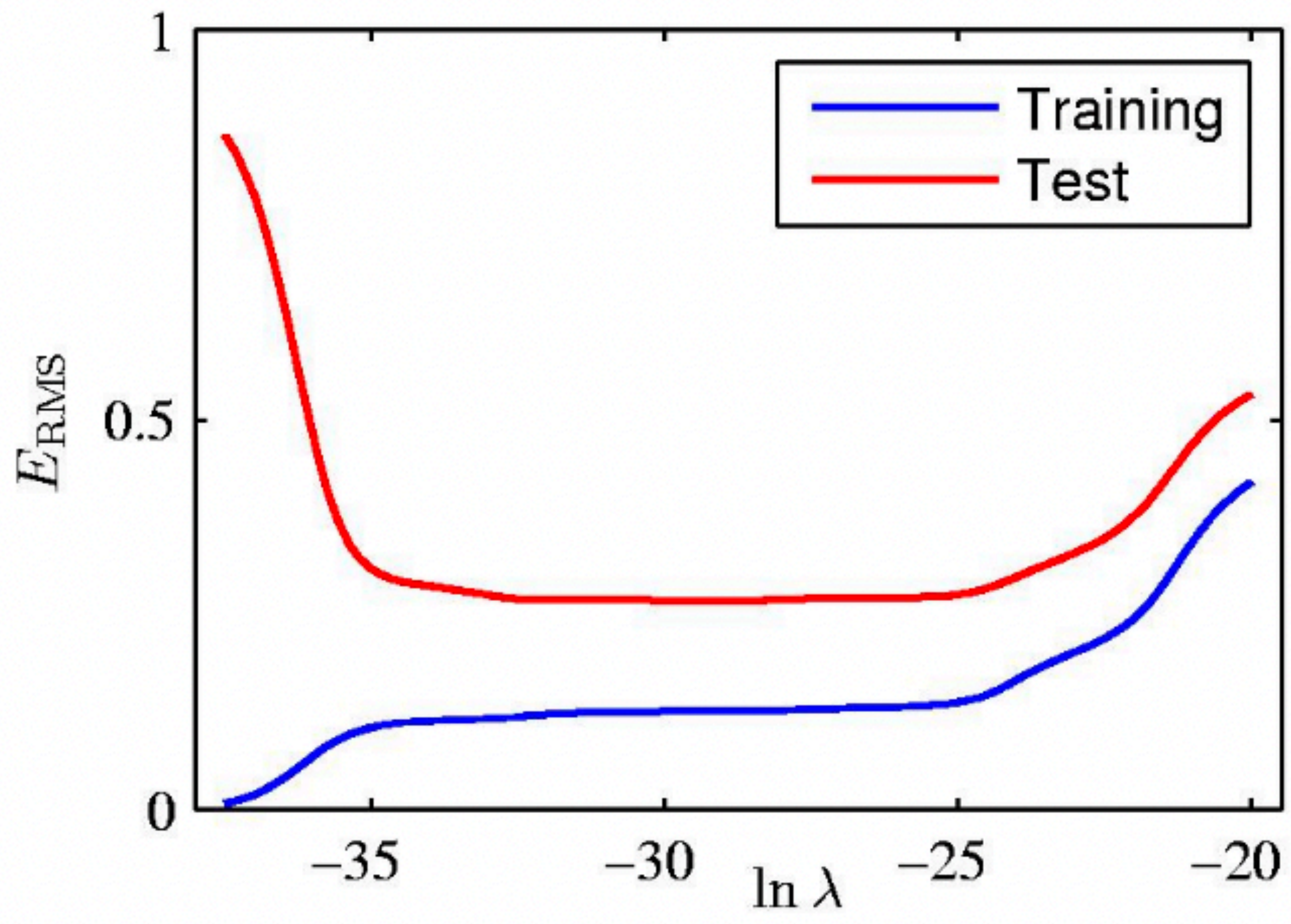
M = 9



Effect of regularisation: loss

Regularisation coefficient controls the complexity of the model

Hence the degree of overfitting



Using a validation set

Divide the total dataset into three subsets:

– Training data is used for learning the parameters of the model.

– Validation data is not used for learning but is used for deciding what type of model and what amount of regularisation works best.

– Test data is used to get a final, unbiased estimate of how well the learner works. We expect this estimate of the generalisation error to be worse than on the validation data.

We could then re-divide the total dataset to get another estimate of the generalisation error