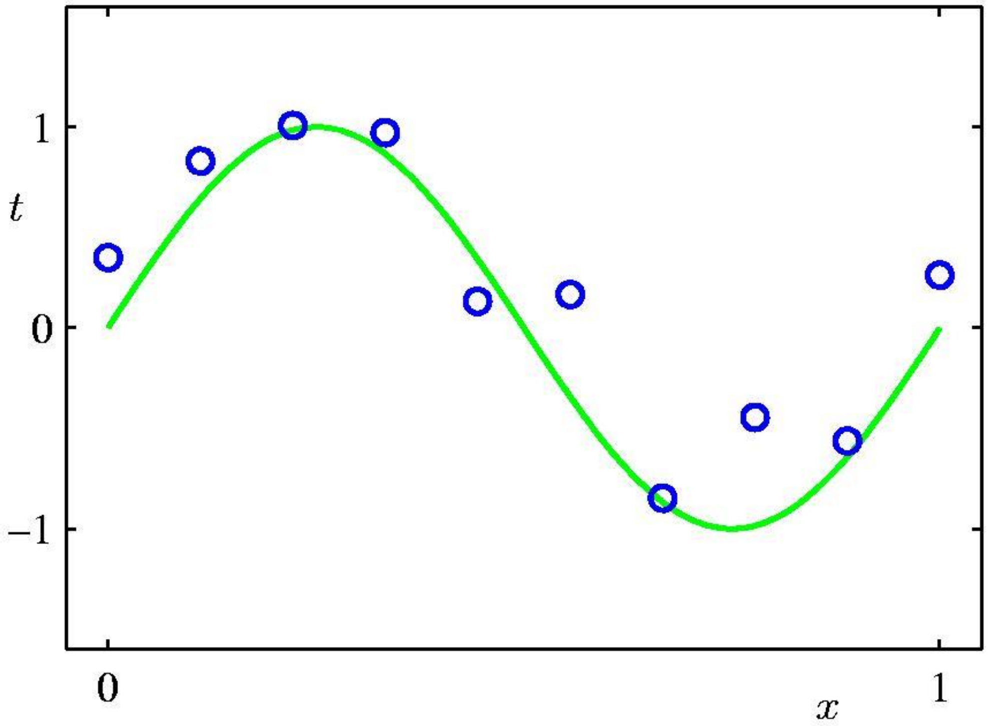
NATIONAL UNIVERSITY OF
SCIENCES AND TECHNOLGOY

Machine Learning

Dr. Muhammad Imran Malik

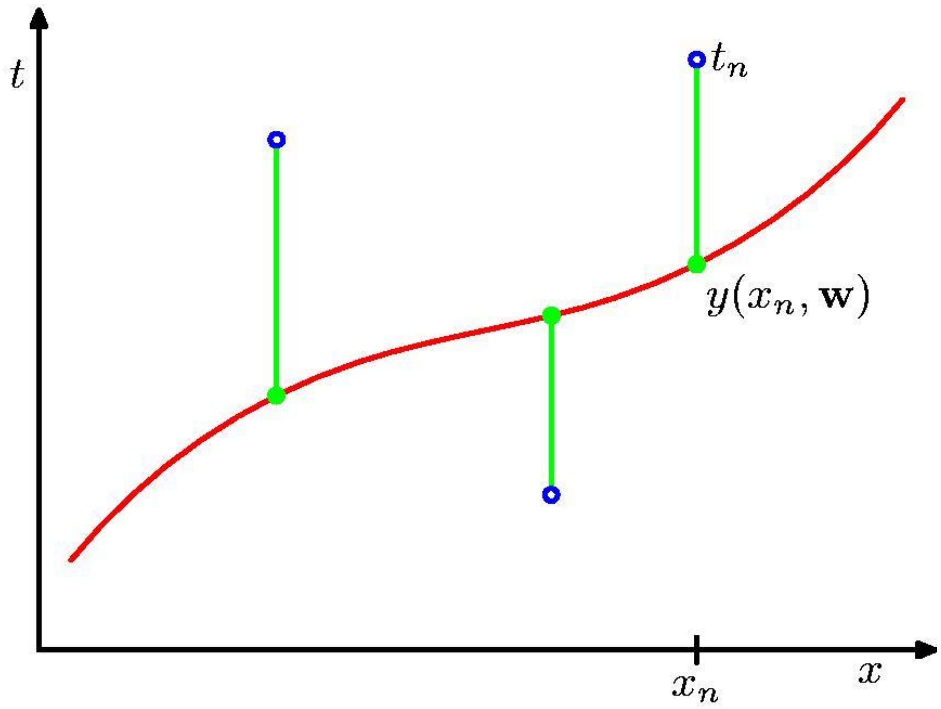
Assistant Professor
School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST), Islamabad

Polynomial Curve Fitting



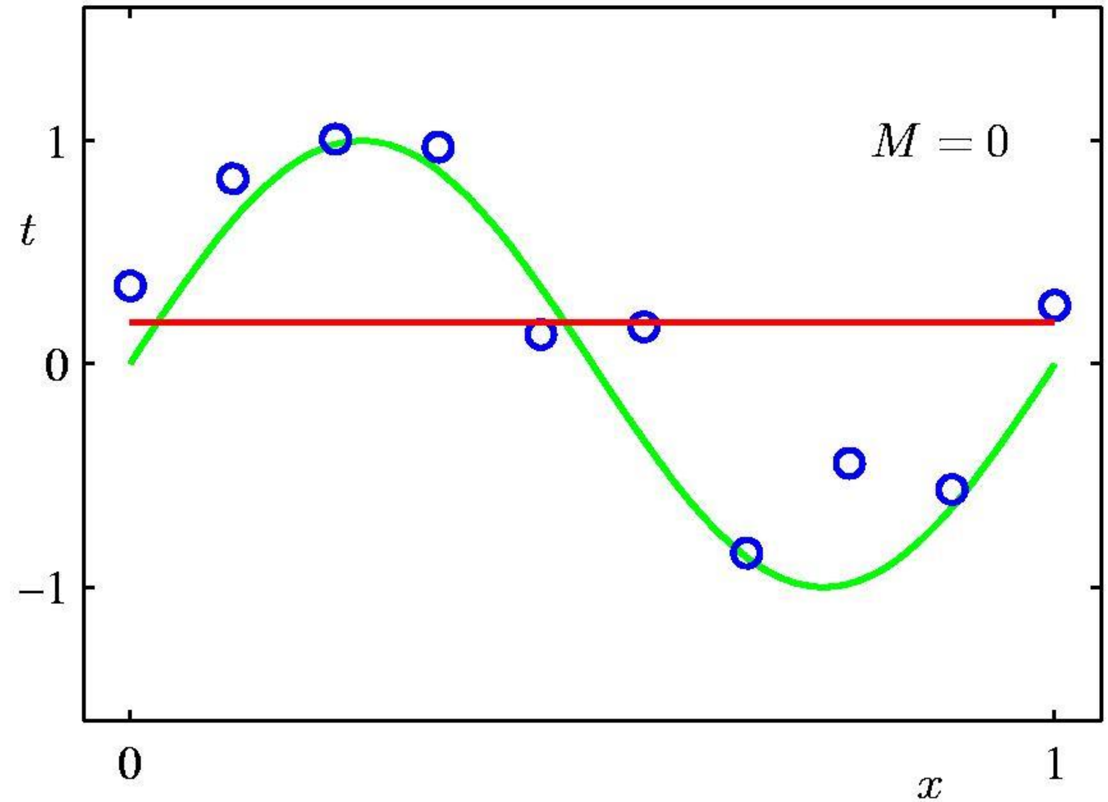
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sum-of-Squares Error Function



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

0^{th} Order Polynomial



Fitting a Polynomial

$$\begin{bmatrix} N & \sum_{i=1}^{N} x_i & \cdots & \sum_{i=1}^{N} x_i^k \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \cdots & \sum_{i=1}^{N} x_i^{k+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{N} x_i^k & \sum_{i=1}^{N} x_i^{k+1} & \cdots & \sum_{i=1}^{N} x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i y_i \\ \vdots \\ \sum_{i=1}^{N} x_i^k y_i \end{bmatrix}$$

a : Polynomial coefficient

k : The degree of the polynomial

N : The number of points to be regressed

ϵ : Error

Fitting a Polynomial

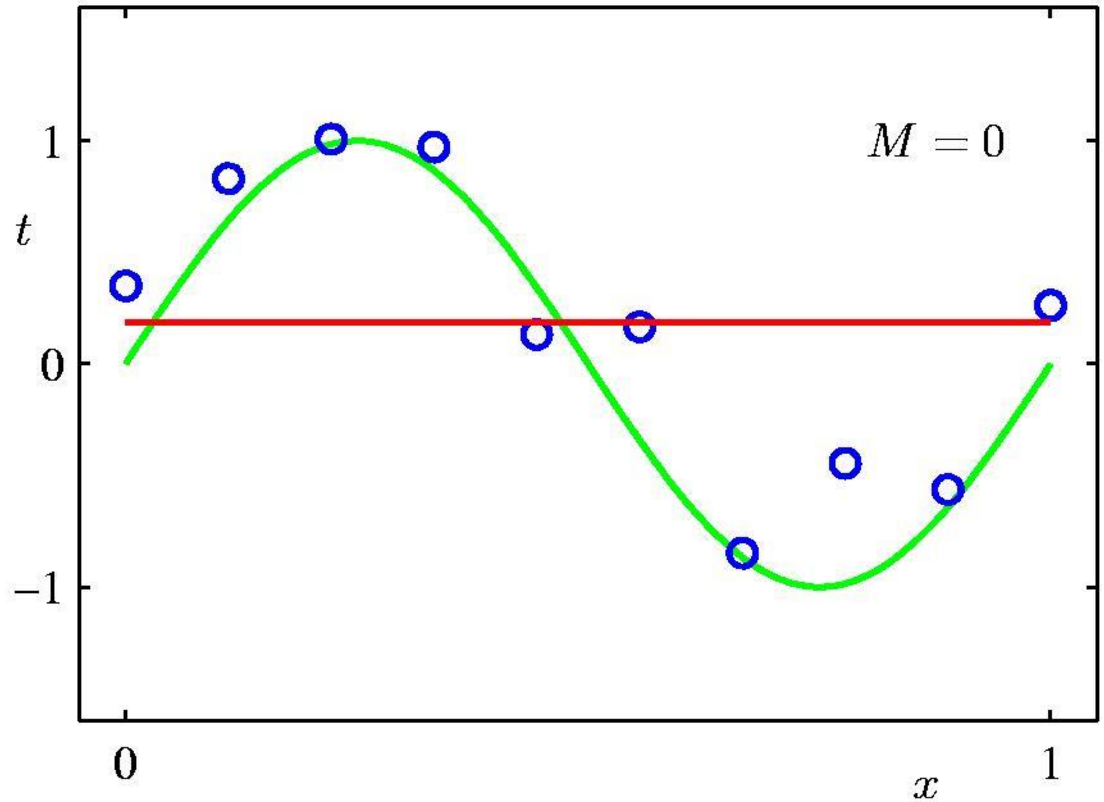
Cramer's rule allows you to solve the linear system of equations to find the regression coefficients using the determinants of the square matrix M. Each of the coefficients a_k may be determined using the following equation:

$$a_k = \frac{det(M_i)}{det(M)}$$

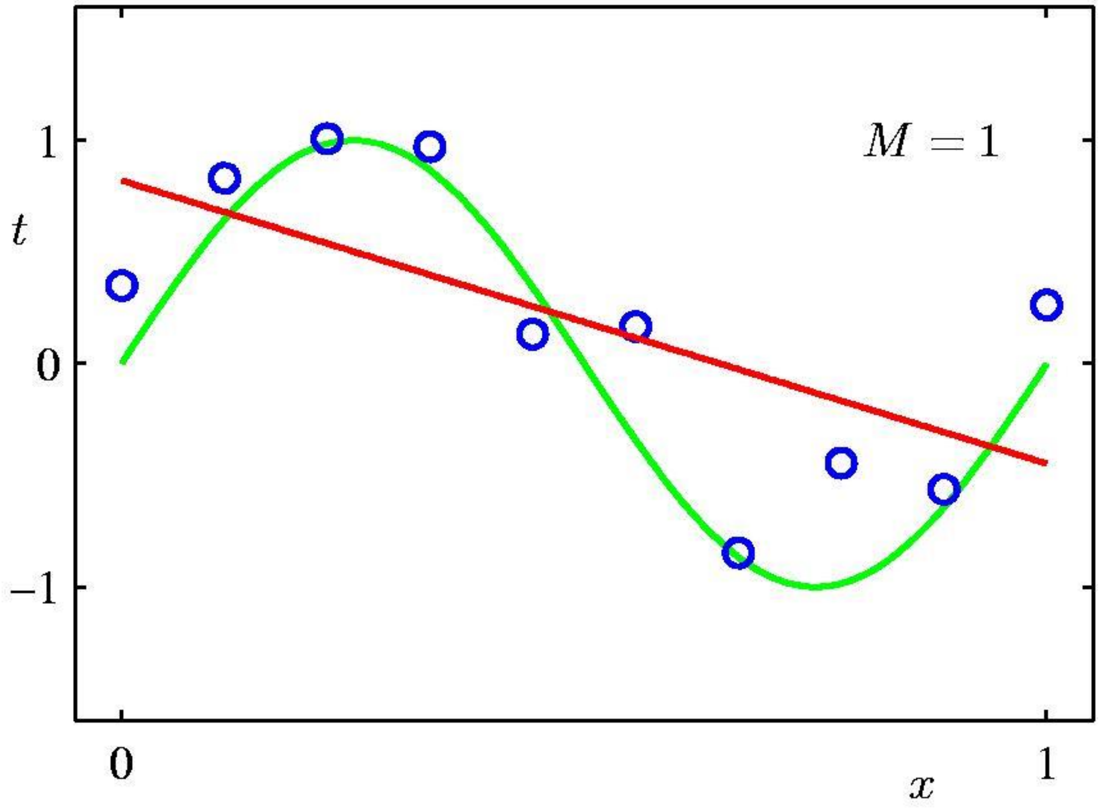
Fitting a Polynomial

$$M_0 = \begin{bmatrix} \sum_{i=1}^{N} y_i & \sum_{i=1}^{N} x_i & \cdots & \sum_{i=1}^{N} x_i^k \\ \sum_{i=1}^{N} x_i y_i & \sum_{i=1}^{N} x_i^2 & \cdots & \sum_{i=1}^{N} x_i^{k+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{N} x_i^k y_i & \sum_{i=1}^{N} x_i^{k+1} & \cdots & \sum_{i=1}^{N} x_i^{2k} \end{bmatrix}$$

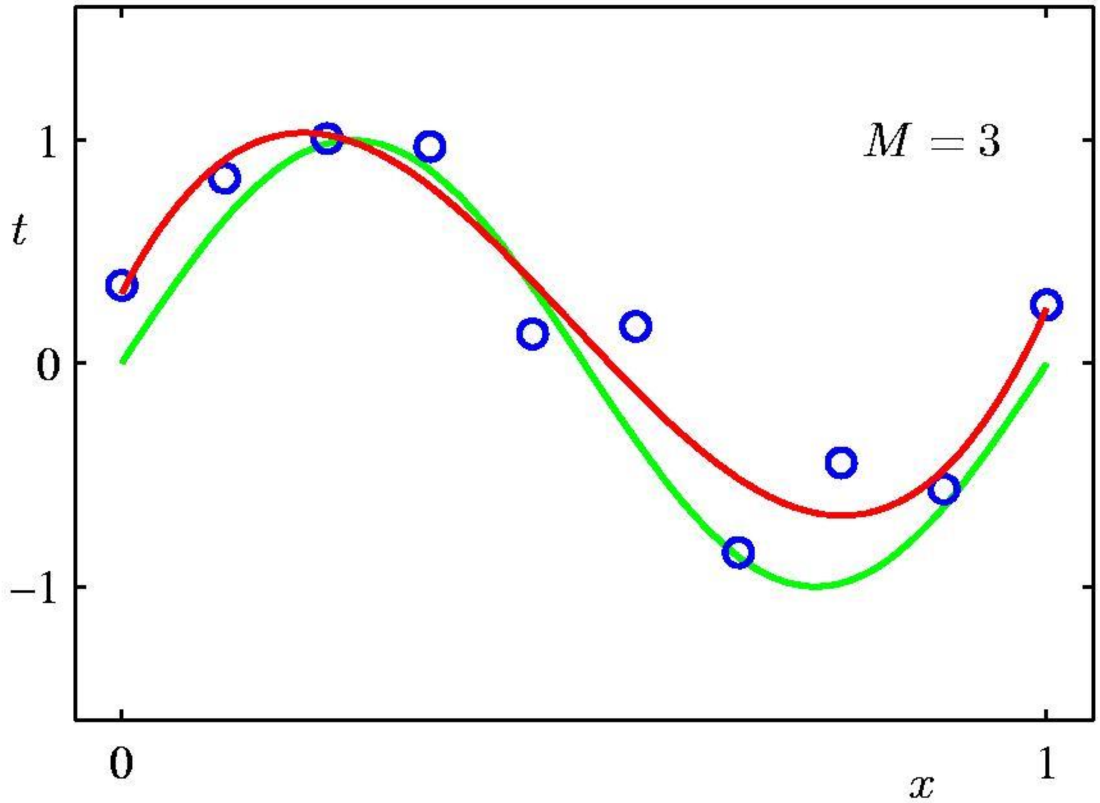
0th Order Polynomial



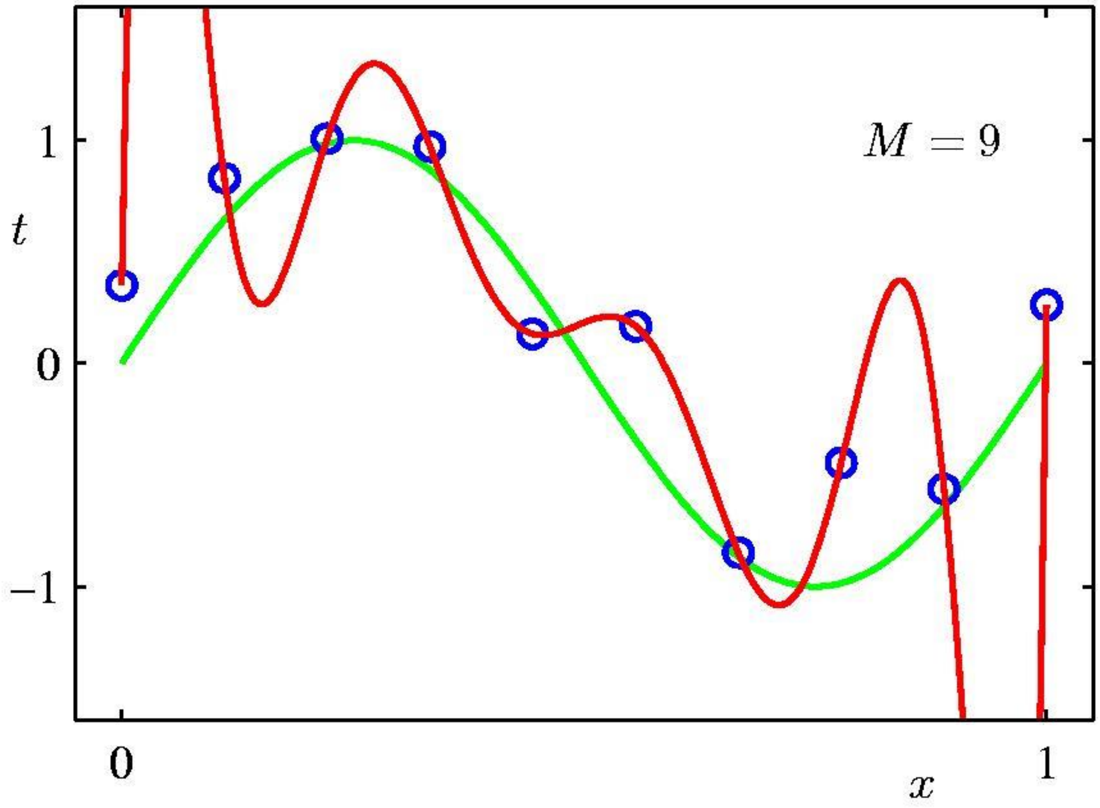
1st Order Polynomial



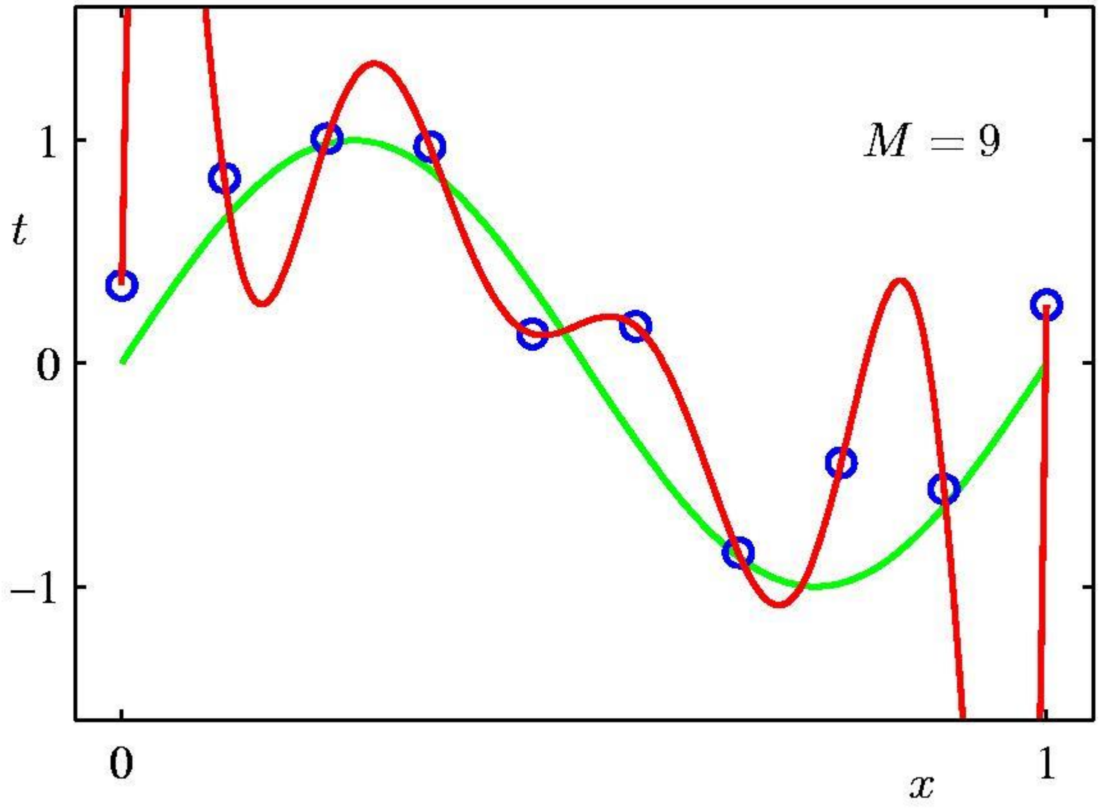
3rd Order Polynomial



9th Order Polynomial



9^{th} Order Polynomial

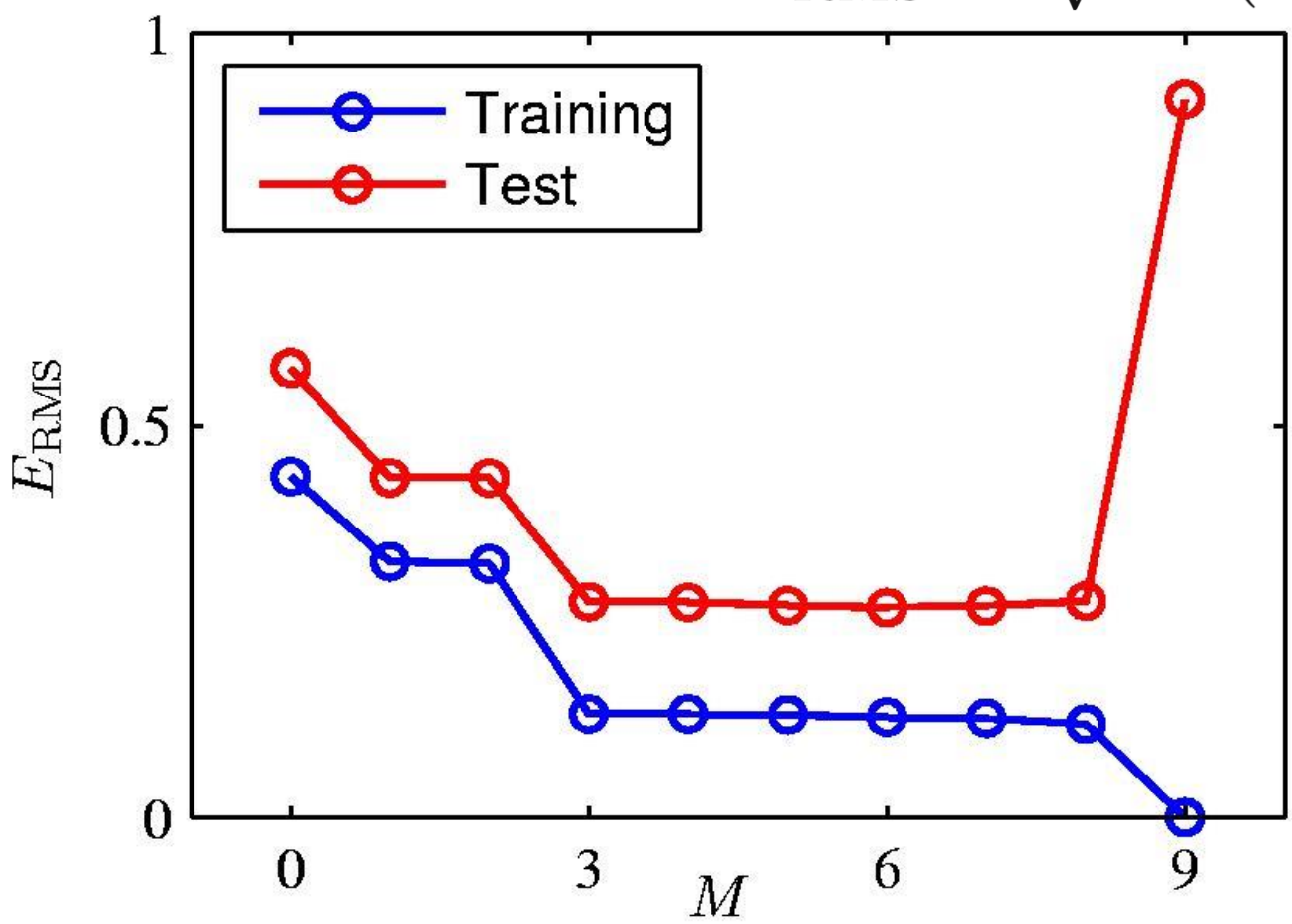


What about Generalization?

Typical Challenges in Machine Learning

Over-fit

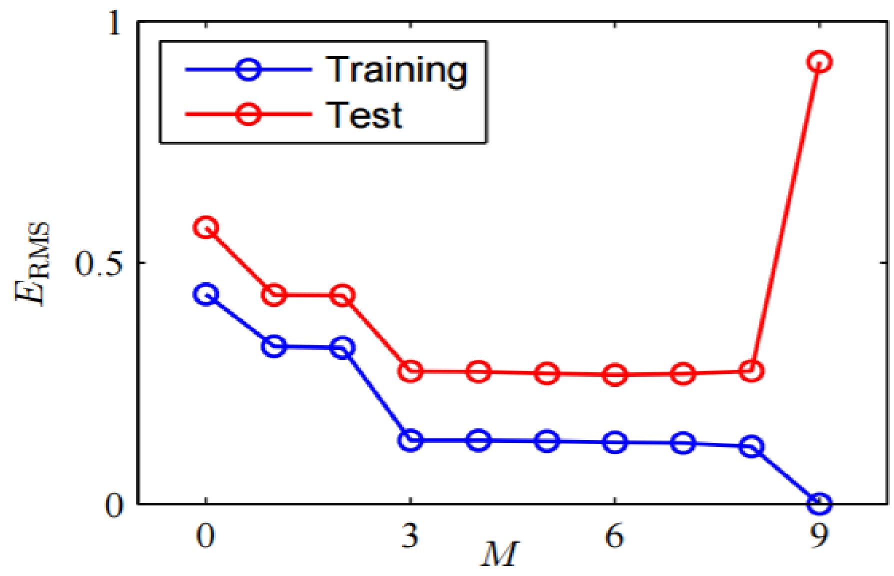
$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^{\star})/N}$$



Overfitting

Fix the training set size, vary H complexity (e.g., degree of polynomials)

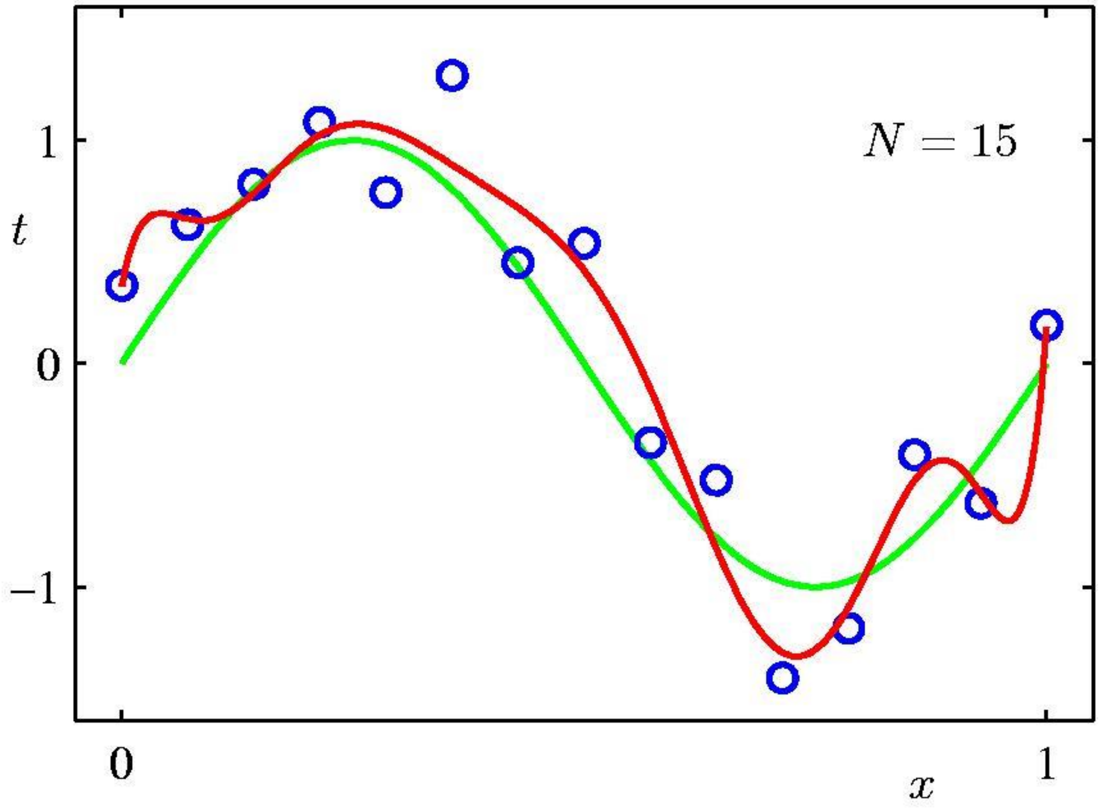
Example from Bishop, Figure 1.5



For any given N, some h of sufficient complexity fits the data but may have very bad generalization error!!

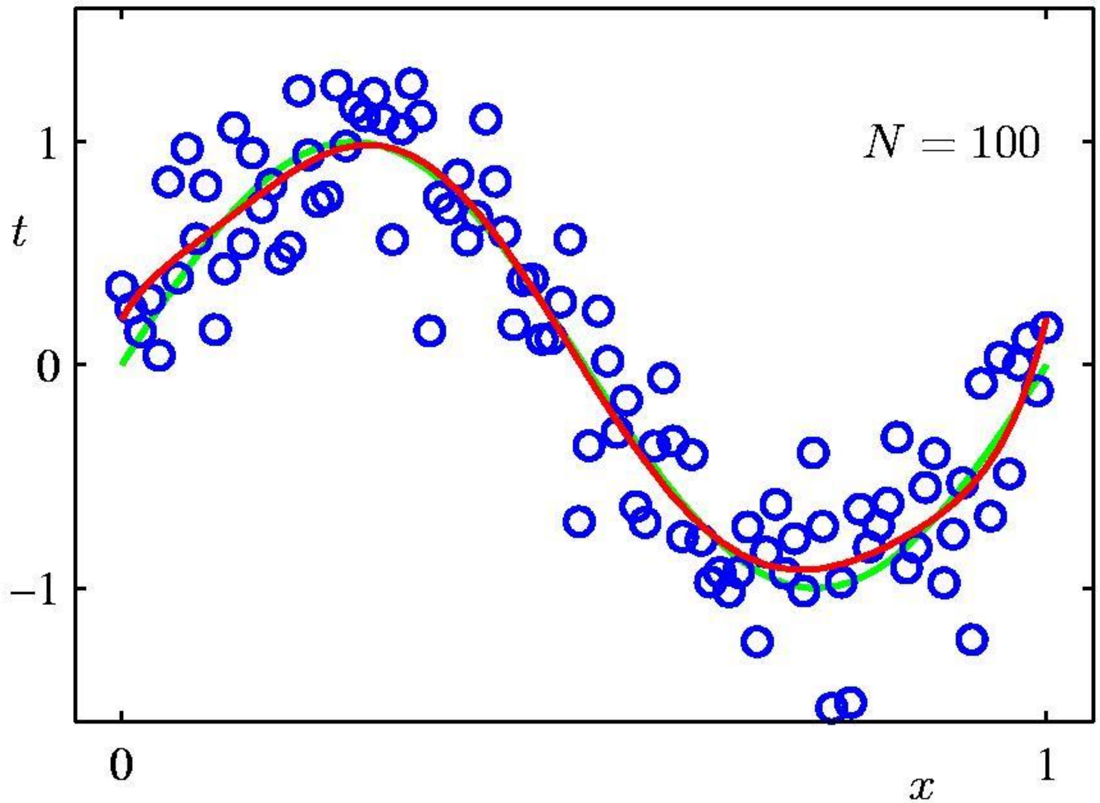
Data Set Size:

9th Order Polynomial fitted on 15 samples



Data Set Size:

9^{th} Order Polynomial fitted on 100 samples

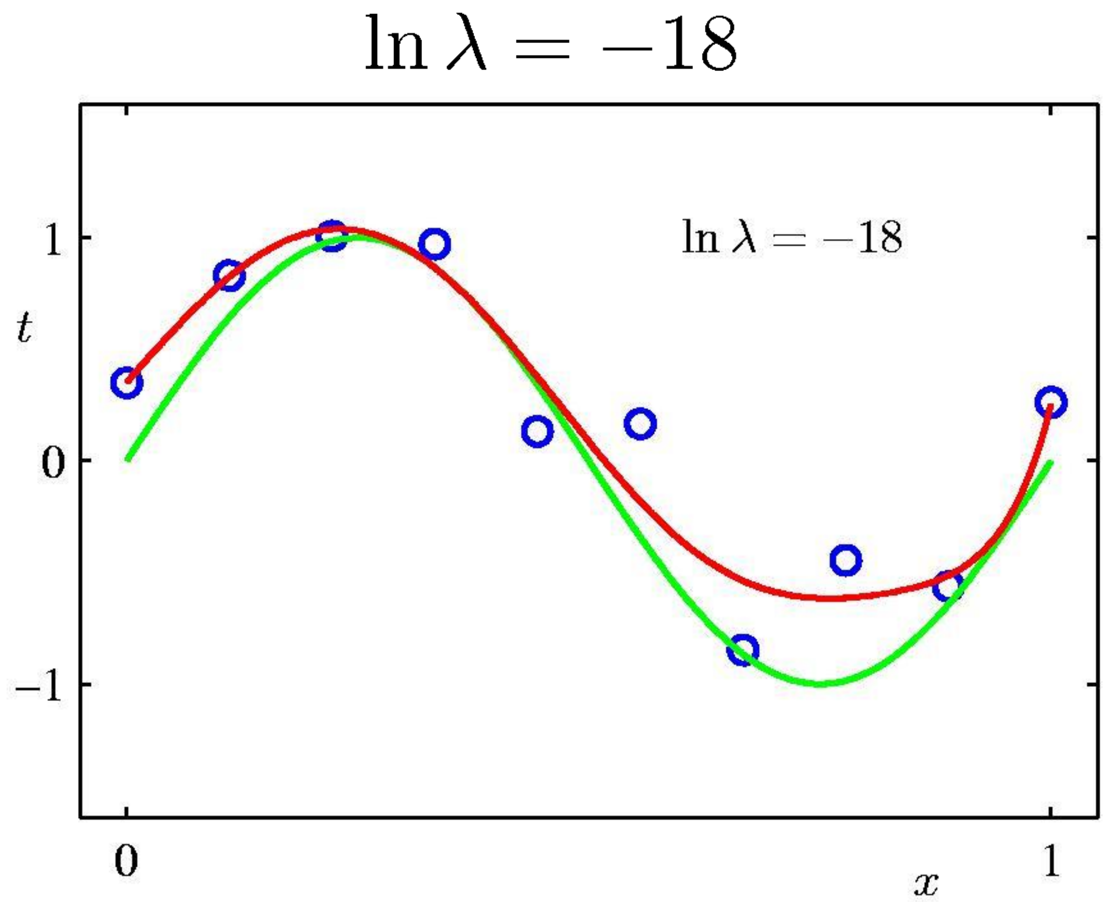


Regularization

- Penalize large coefficient values

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

Regularization:



Cross-validation

- Regularization helps but still need to pick λ
- Want to minimize test-set error, but we have no test set!
- Idea: make one (a validation set) by pretending we can't see the labels
- Try different values of λ, learn $\hat{h}^{\lambda}$ on rest of data,

test $\hat{h}^{\lambda}$ on validation set, pick best λ, train on all

- Problem: small validation set $\Rightarrow$ large error in estimated loss
- large validation set $\Rightarrow$ small training set $\Rightarrow$ bad $\hat{h}^{\lambda}$

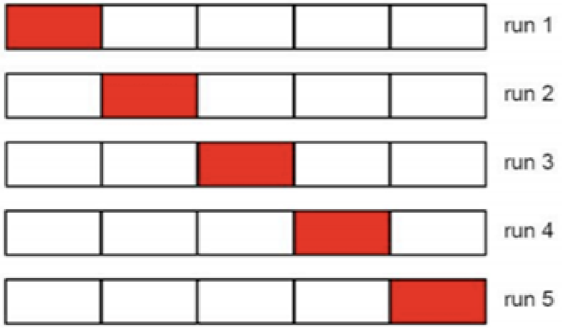
Cross-validation contd.

- K-fold cross-validation: divide data into *K* blocks
- For k=1 to k

 train on blocks except *kth* block, test on *kth* block

average the results, choose best λ

Common cases: *K=5,10* or *K=N (LOOCV)*



High computational cost*: K folds* × many choices of model *or* λ