INTRODUCTION TO MACHINE LEARNING COMPSCI 4ML3

LECTURE 4

HASSAN ASHTIANI

MATRIX FORM OLS

- $\Delta = \begin{pmatrix} \Delta_1 \\ \dots \\ \Delta_n \end{pmatrix} = \begin{pmatrix} x_1^1 & \cdots & x_d^1 \\ \vdots & \ddots & \vdots \\ x_1^n & \cdots & x_d^n \end{pmatrix} \begin{pmatrix} w_1 \\ \dots \\ w_d \end{pmatrix} - \begin{pmatrix} y^1 \\ \dots \\ y^n \end{pmatrix}$

$$\underset{W \in \mathbb{R}^{d\times 1}}{\text{MIN}} \sum_{i=1}^{n} (\Delta_i)^2 = \underset{W \in \mathbb{R}^{d\times 1}}{\text{MIN}} \|\Delta\|_2^2 =$$

$$\boldsymbol{\min_{W \in \mathbb{R}^{d\times 1}} \|XW - Y\|_2^2}$$

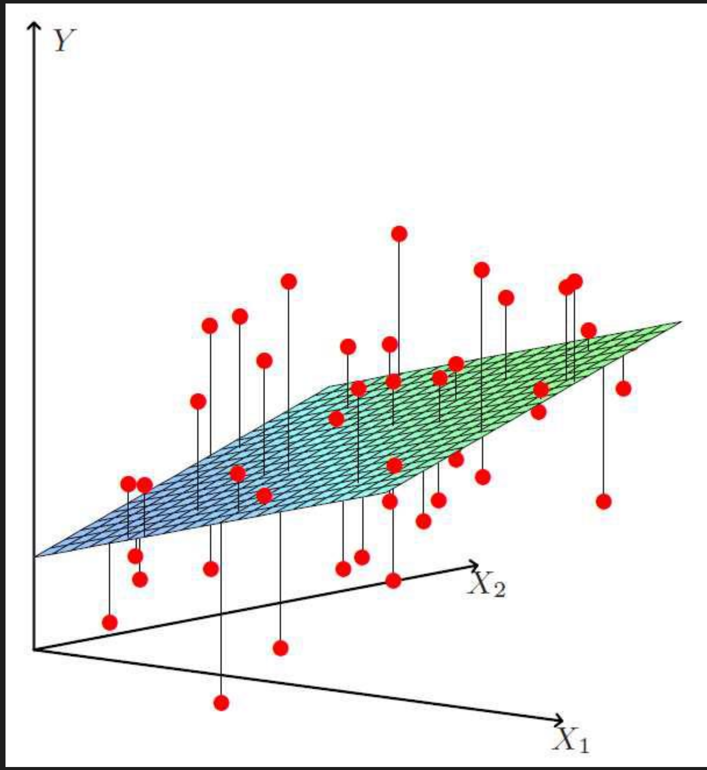
$$W^{LS} = (X^T X)^{-1} X^T Y$$

BIAS/INTERCEPT TERM

WE ARE MISSING THE BIAS TERM (w_0)

$$\min_{w_0, w_1, \ldots, w_d \in \mathbb{R}} \sum_{i=1}^{n} (w_1 x_1^i + \cdots + w_d x_d^i + w_0 - y^i)^2$$

$$\min_{w_0 \in \mathbb{R}, W \in \mathbb{R}^{d \times 1}} \left\| XW + \begin{pmatrix} w_0 \\ w_0 \\ \cdots \\ w_0 \end{pmatrix} - Y \right\|_2^2$$

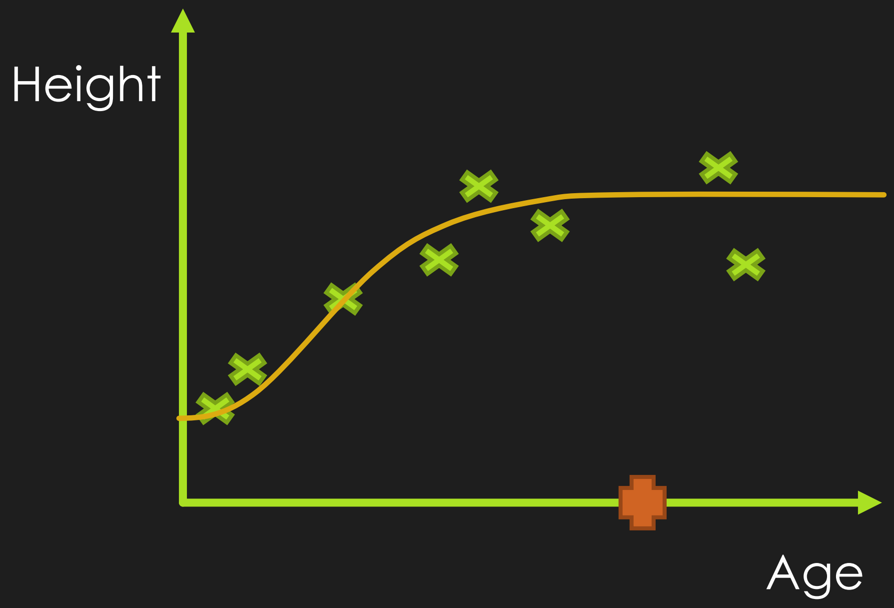


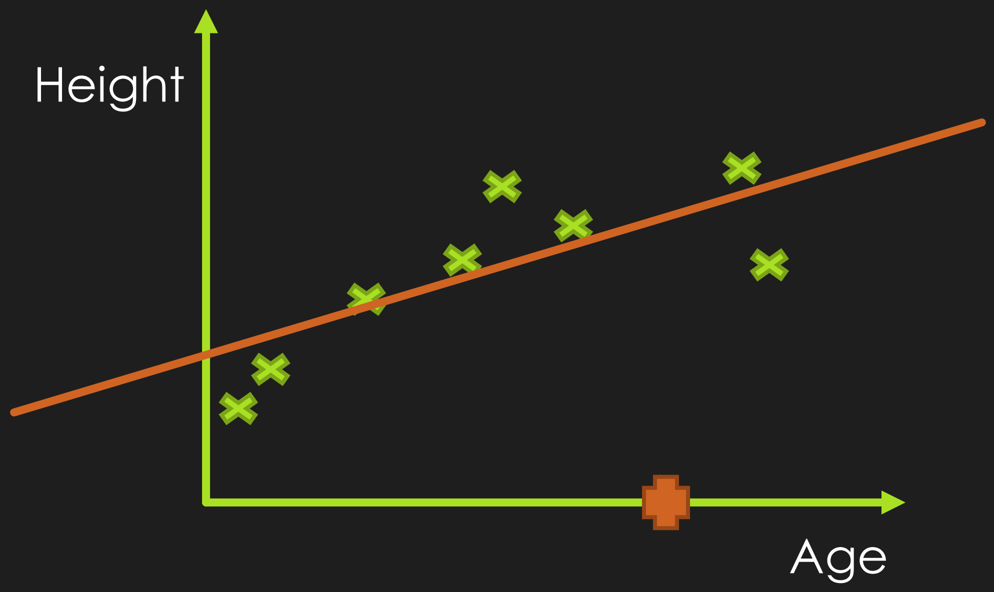
BIAS/INTERCEPT TERM

- ADD A NEW AUXILIARY DIMENSION TO THE DATA
- $X_{n\times(d+1)} = \begin{pmatrix} x_1^1 & \cdots & x_d^1 & 1 \\ \vdots & \ddots & \vdots & 1 \\ x_1^n & \cdots & x_d^n & 1 \end{pmatrix}$, $W_{(d+1)\times 1} = \begin{pmatrix} w_1 \\ \cdots \\ w_d \\ w_0 \end{pmatrix}$
- SOLVE OLS: $\min_{W\in\mathbb{R}^{(d+1)\times 1}} \|XW - Y\|_2^2$
- w_0 WILL BE THE BIAS TERM!

"NON-LINEAR" DATA?

- FOR EXAMPLE, WHAT IS THE BEST DEGREE 2 POLYNOMIAL?





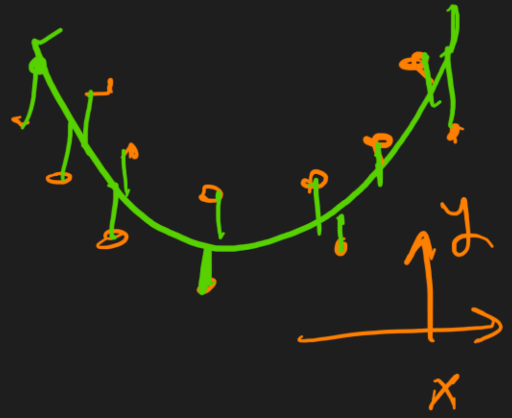
- HOW CAN WE REUSE THE "LEAST-SQUARES MACHINERY"?

IDEA: DATA TRANSFORMATION

- We increased the flexibility of our predictor by a form of data transformation/augmentation
- $X'_{n\times(d+1)} = \begin{pmatrix} x_1^1 & \cdots & x_d^1 & 1 \\ \vdots & \ddots & \vdots & 1 \\ x_1^n & \cdots & x_d^n & 1 \end{pmatrix}$
- Can we use the same idea to make our predictor even more flexible (non-linear)?

EXAMPLE

$$\hat{y} = a x^2 + b x + c$$



$(x^1)^T$

$$X = \begin{pmatrix} x^1 \\ \vdots \\ x^n \end{pmatrix} \rightarrow X' = \begin{pmatrix} x^1 & (x^1)^2 & 1 \\ \vdots & \vdots & 1 \\ x^n & (x^n)^2 & 1 \end{pmatrix}, \quad w' = \begin{pmatrix} b \\ a \\ c \end{pmatrix}$$

$$(x^1)^T w' = b\, x^1 + a (x^1)^2 + c$$

$$\rightarrow \binom{2+1}{1} = 3 \qquad d = 1 \quad M = 2$$

$\hat{y} = ax^2 + bx + c$

log

is $\hat{y}$ a linear function of x^2 and x

LEAST-SQUARES FOR POLYNOMIALS

- IDEA: $ax^2 + bx + c$ IS STILL LINEAR WITH RESPECT TO THE PARAMETERS! (W.R.T. a, b AND c)
- INSTEAD OF $X_{n\times 1} = \begin{pmatrix} x^1 \\ \dots \\ x^n \end{pmatrix}$ USE $X'_{n\times 3} = \begin{pmatrix} x^1 & (x^1)^2 & 1 \\ \dots & \dots & \dots \\ x^n & (x^n)^2 & 1 \end{pmatrix}$
- TREAT $X_{n\times 3}$ AS IF IT WAS YOUR ORIGINAL INPUT DATA
- WE CAN EXTEND THIS TO HIGHER DEGREE POLYNOMIALS SIMILARLY, E.G., $ax^3 + bx^2 + cx + d$
- NOTEBOOK EXAMPLE

MULTIVARIATE POLYNOMIALS

$d_1 = 2$

$\downarrow$

$d_2 = 6$

- HOW ABOUT WHEN x IS MULTIVARIATE ITSELF?
  - $\underline{w_1 x_1} + \underline{w_2 x_2} + \underline{w_3 x_1 x_2} + \underline{w_4 (x_1)^2} + \underline{w_5 (x_2)^2} + \underline{w_6}$
  - INSTEAD OF (x_1, x_2) USE $(x_1 \quad x_2 \quad x_1 x_2 \quad (x_1)^2 \quad (x_2)^2 \quad 1)$
- TREAT THE NEW X AS (A HIGHER-DIMENSIONAL) INPUT

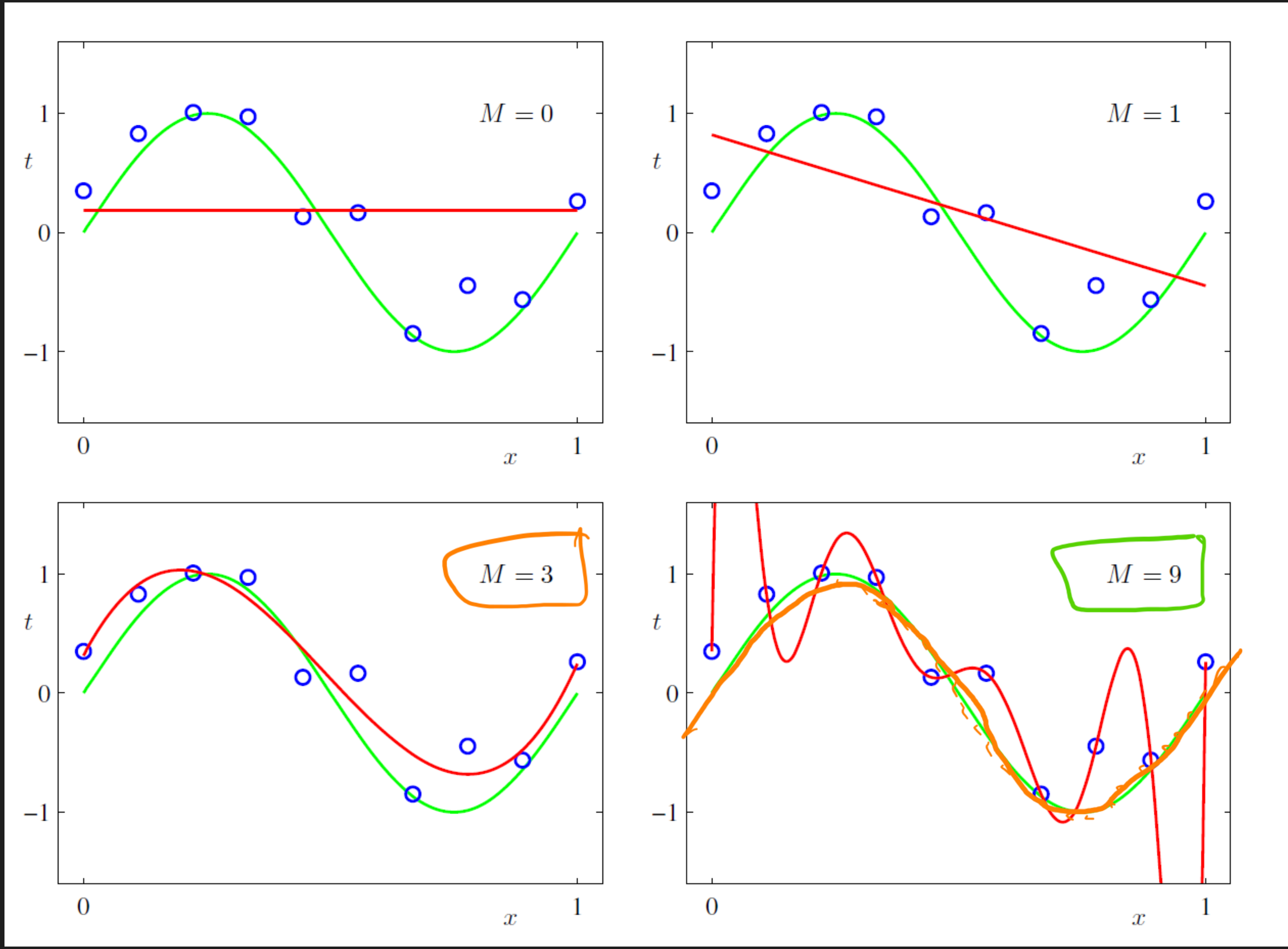
- INPUT DIMENSION: d
- DEGREE OF POLYNOMIAL: M
- NUMBER OF TERMS (MONOMIALS) OF DEGREE AT MOST M ≈

$$\binom{M+d}{d} = \binom{M+d}{M}$$

$d=2$
$M=2$ $\rightarrow$ $\binom{4}{2}=6$ ✓

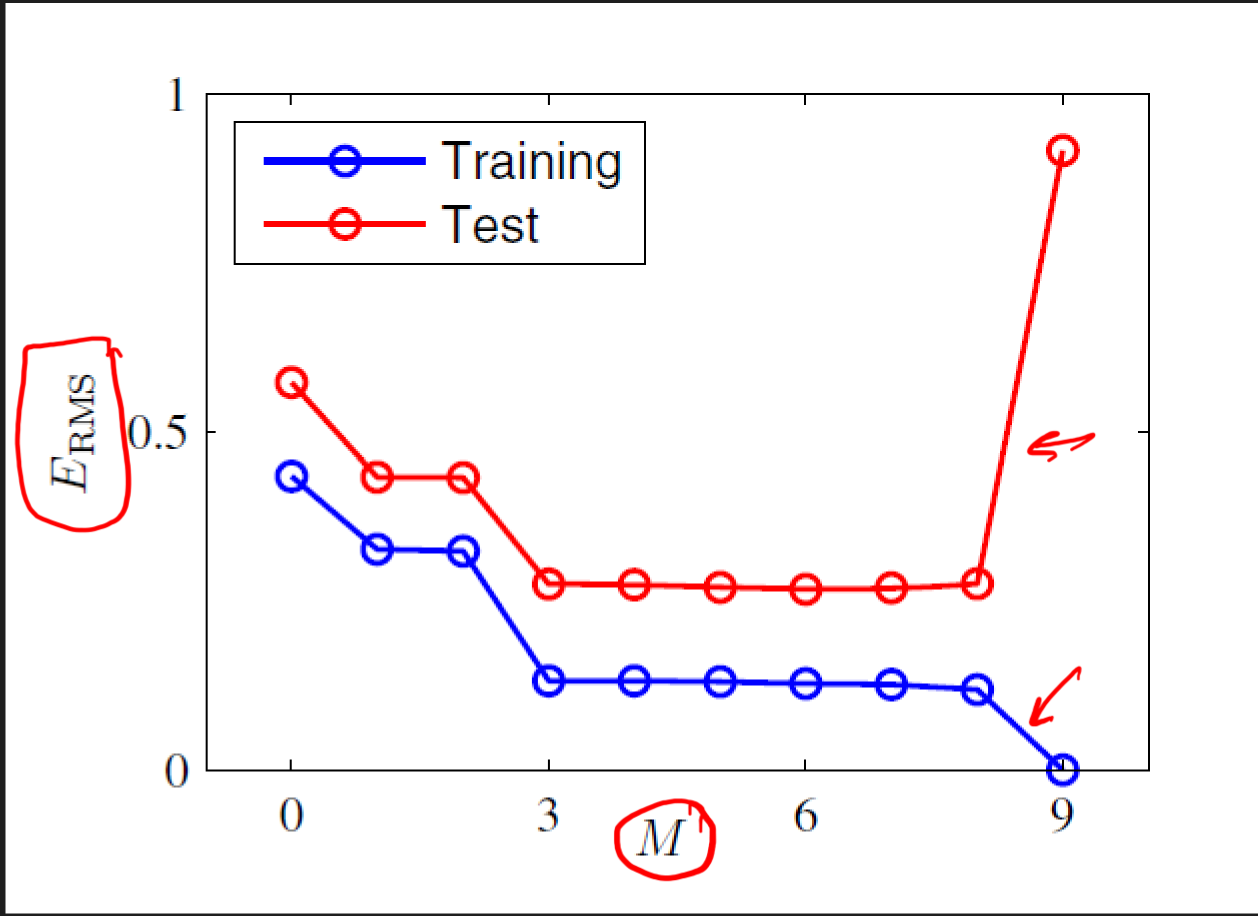
n balls

OVERFITTING



OVERFITTING

- DIVIDE THE DATA RANDOMLY TO "TRAIN" AND "TEST" SETS
- ROOT-MEAN-SQUARE ERROR FOR EACH SET:
- $\sqrt{\frac{\|\hat{Y}-Y\|_2^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n}(y_i-\widehat{y_i})^2}{n}}$



MORE DATA, LESS OVER-FITTING

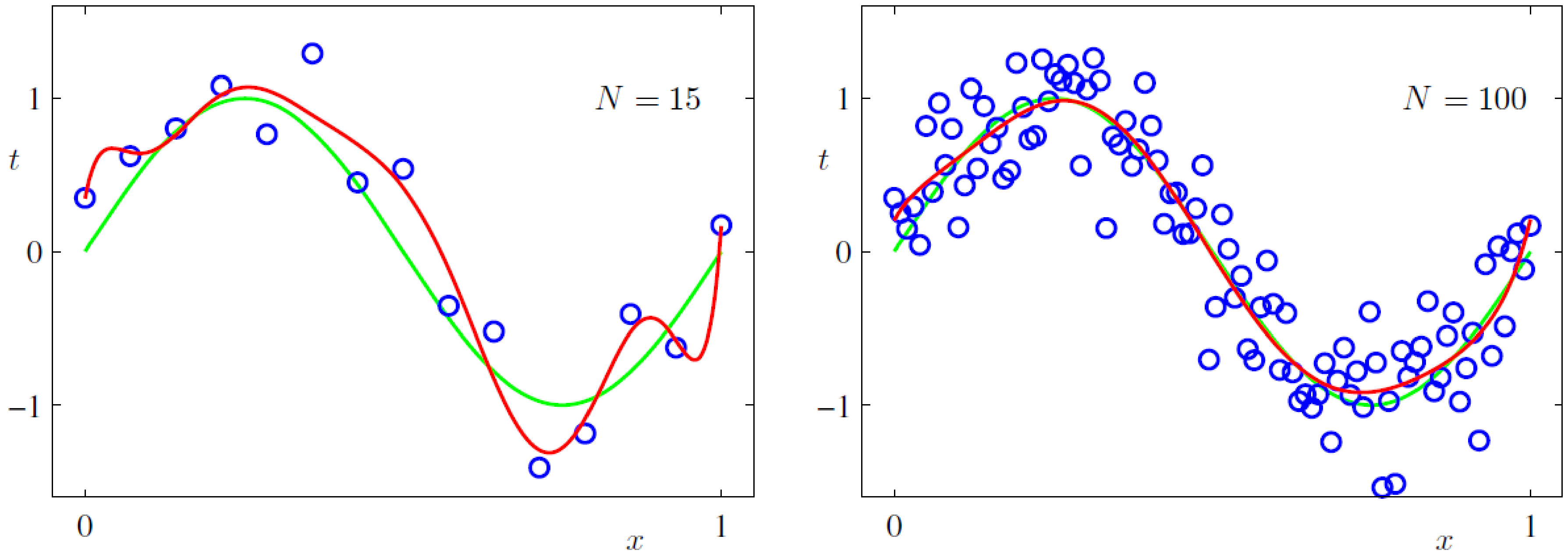


Figure 1.6 Plots of the solutions obtained by minimizing the sum-of-squares error function using the $M = 9$ polynomial for $N = 15$ data points (left plot) and $N = 100$ data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem.

THE TRADE-OFF

- A POWERFUL/FLEXIBLE CURVE-FITTING METHOD
  - SMALL TRAINING ERROR
  - REQUIRES MORE TRAINING DATA TO GENERALIZE
  - OTHERWISE LARGE TEST ERROR
- A LESS FLEXIBLE CURVE-FITTING METHOD
  - LARGER TRAINING ERROR
  - REQUIRES LESS TRAINING DATA
  - SMALLER DIFFERENCE BETWEEN TRAINING AND TEST ERROR
- THE SO-CALLED "BIAS-VARIANCE" TRADE-OFF

THE CASE OF MULTIVARIATE POLYNOMIALS

- Assume $M \gg d$
- Number of terms (monomials): $\approx \left(\frac{M}{d}\right)^d$
- #training samples $\approx$ #parameters $\approx \left(\frac{M}{d}\right)^d$
  - #training samples should increase exponentially with d
  - Susceptible to over-fitting…
  - An example of **<u>curse of dimensionality</u>**!
- We can say **<u>sample complexity</u>** of learning multivariate polynomials is exponential in d
  - Orthogonal to computational complexity

$\boxed{d \leq 1}$

$\binom{M+d}{M}$

$\binom{M+d}{1} \approx M+d$