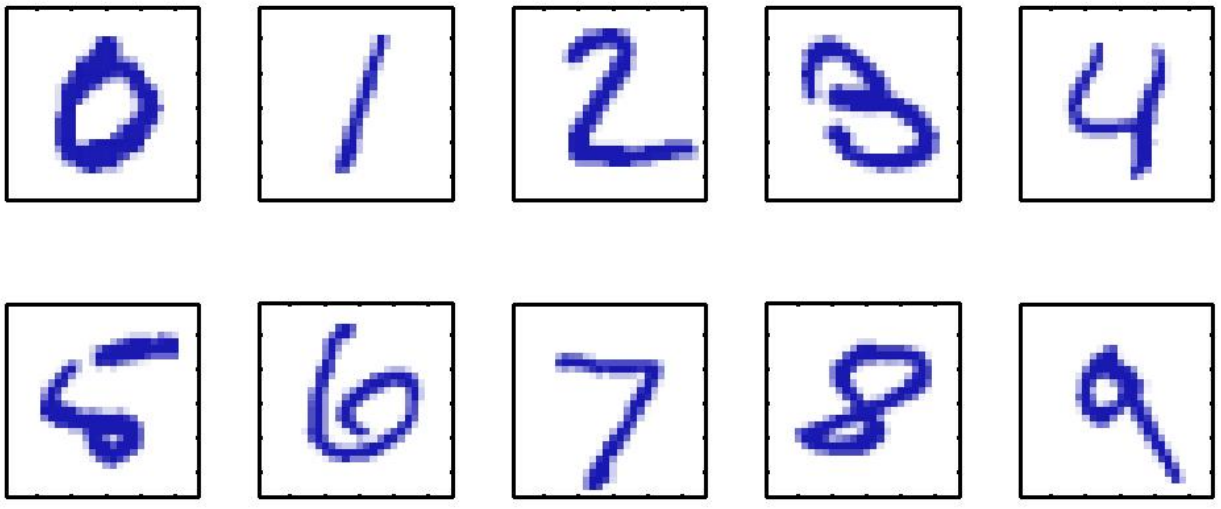
Pattern Recognition and Machine Learning: Introduction

Libao Jin

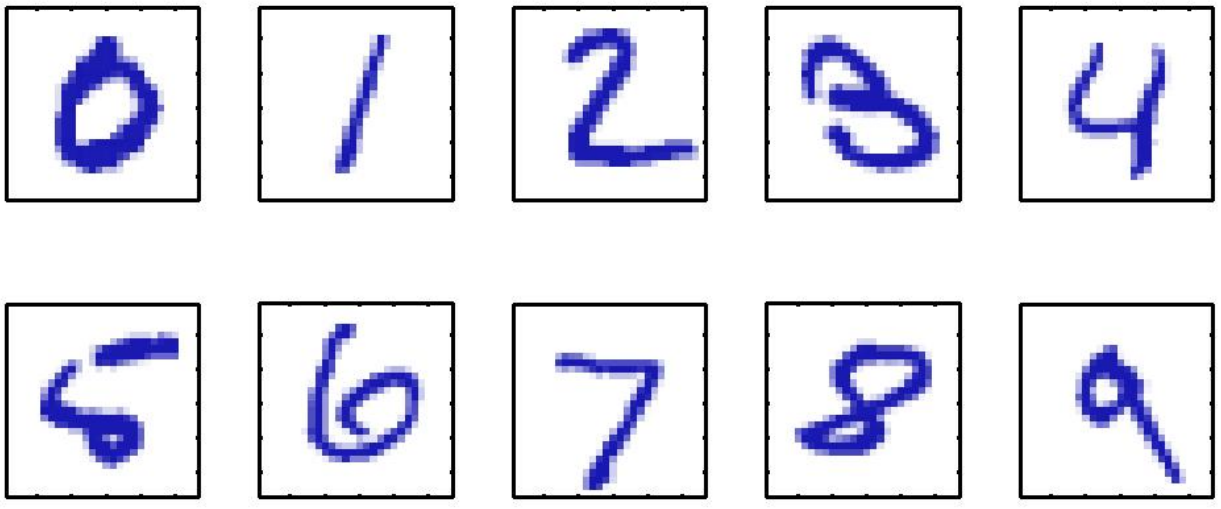
November 17, 2016

Example: Handwritten Digit Recognition



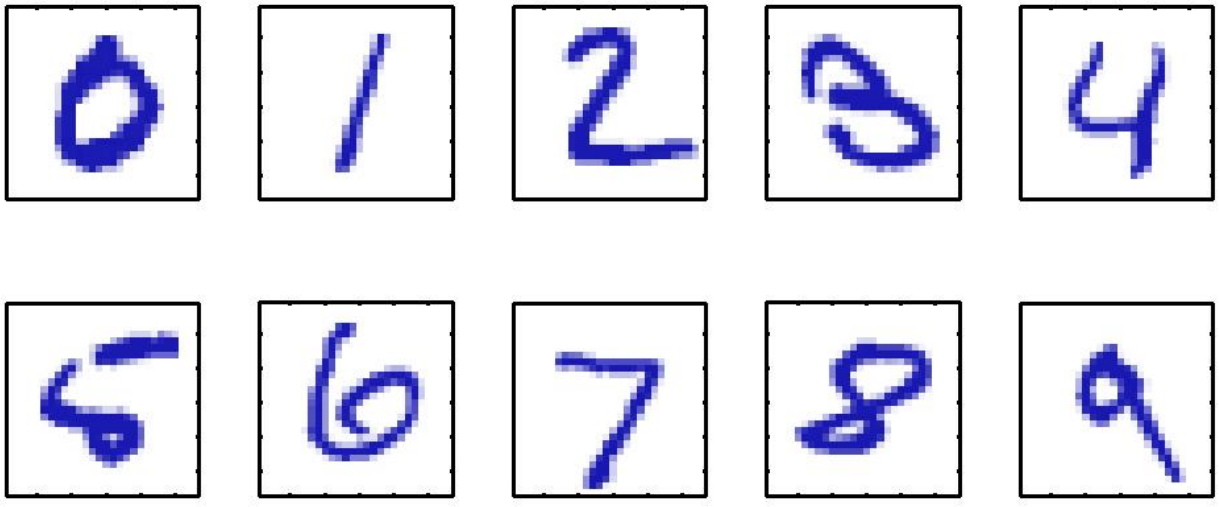
- Training Set: $\mathbf{x}$, to tune the parameters of an adaptive model

Example: Handwritten Digit Recognition



- Training Set: $\mathbf{x}$, to tune the parameters of an adaptive model
- Target Vector: $\mathbf{t}$, to express the category of a digit

Example: Handwritten Digit Recognition



- Training Set: $\mathbf{x}$, to tune the parameters of an adaptive model
- Target Vector: $\mathbf{t}$, to express the category of a digit
- Note that there is one such target vector $\mathbf{t}$ for each digit image $\mathbf{x}$

The Result of Running the Machine Learning Algorithm

- $\mathbf{y} = \mathbf{y}(\mathbf{x})$, which encoded in the same way as the target vectors

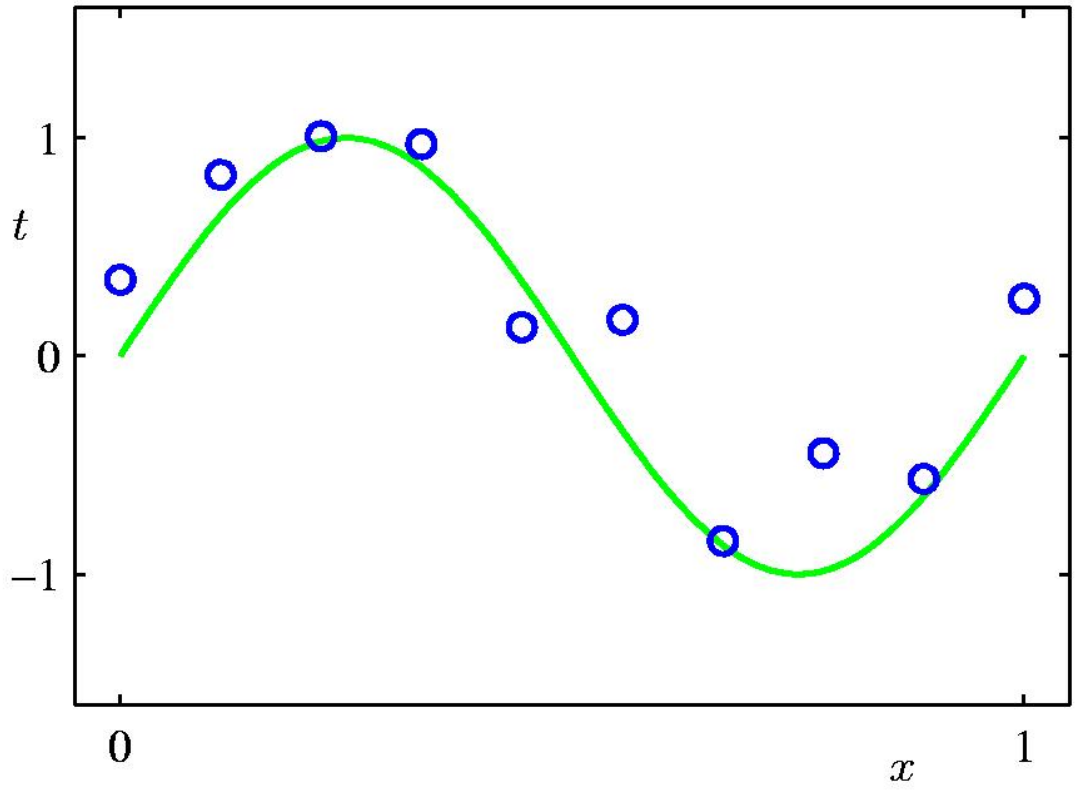
The Result of Running the Machine Learning Algorithm

- $\mathbf{y} = \mathbf{y}(\mathbf{x})$, which encoded in the same way as the target vectors
- Once the model is trained it can then determine the identity of new digit images, which are said to comprise a *test set*

The Result of Running the Machine Learning Algorithm

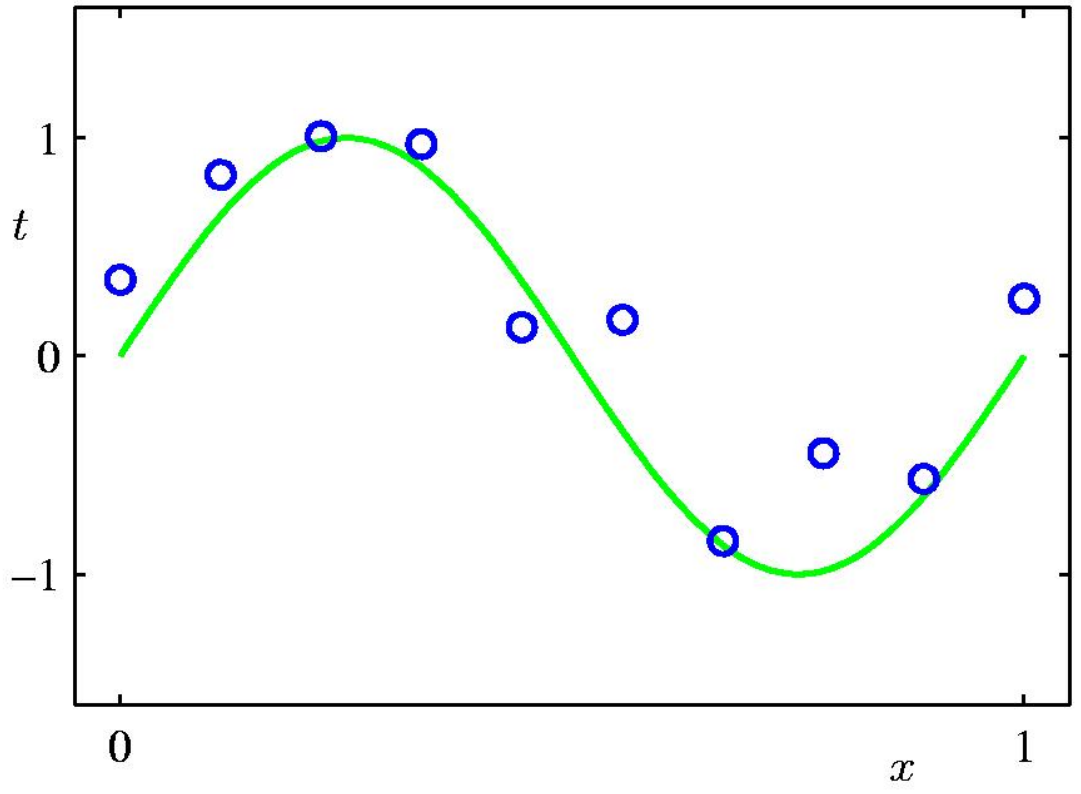
- $\mathbf{y} = \mathbf{y}(\mathbf{x})$, which encoded in the same way as the target vectors
- Once the model is trained it can then determine the identity of new digit images, which are said to comprise a *test set*
- In practical applications, training data can comprise only a tiny fraction of all possible input vectors, and so generalization is a central goal in pattern recognition

Polynomial Curve Fitting



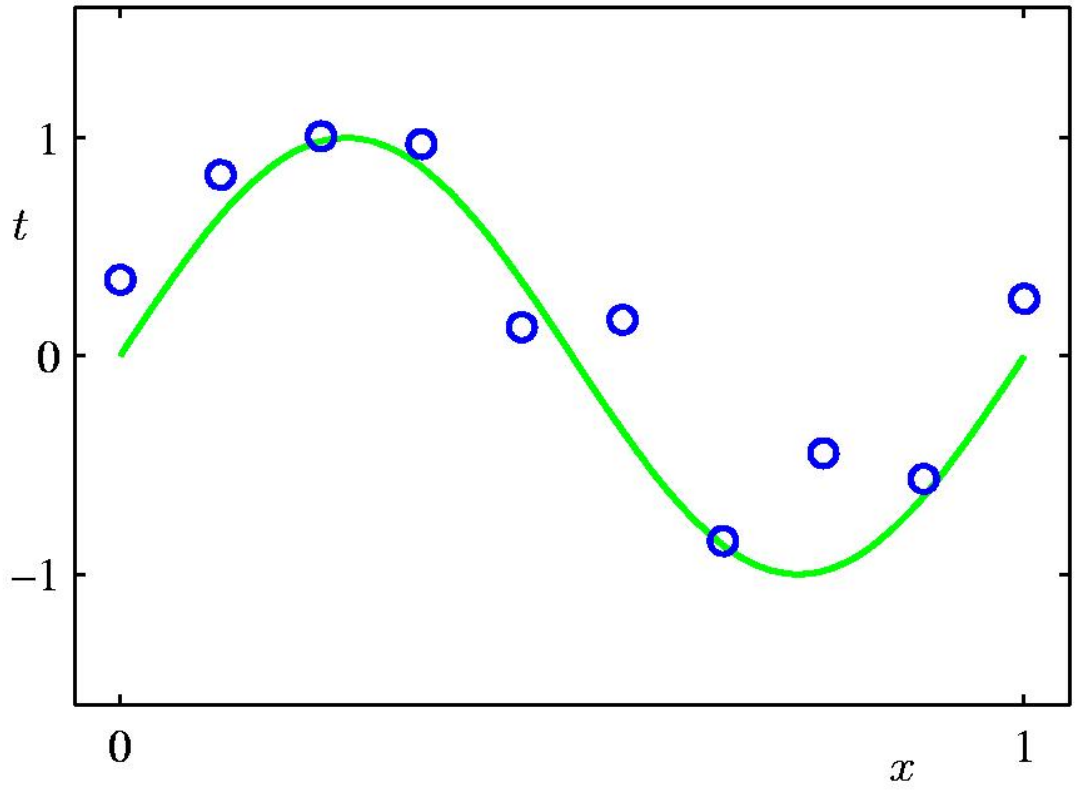
- Training Set (blue circles): $\mathbf{x} \equiv (x_1, \ldots, x_N)^T$

Polynomial Curve Fitting



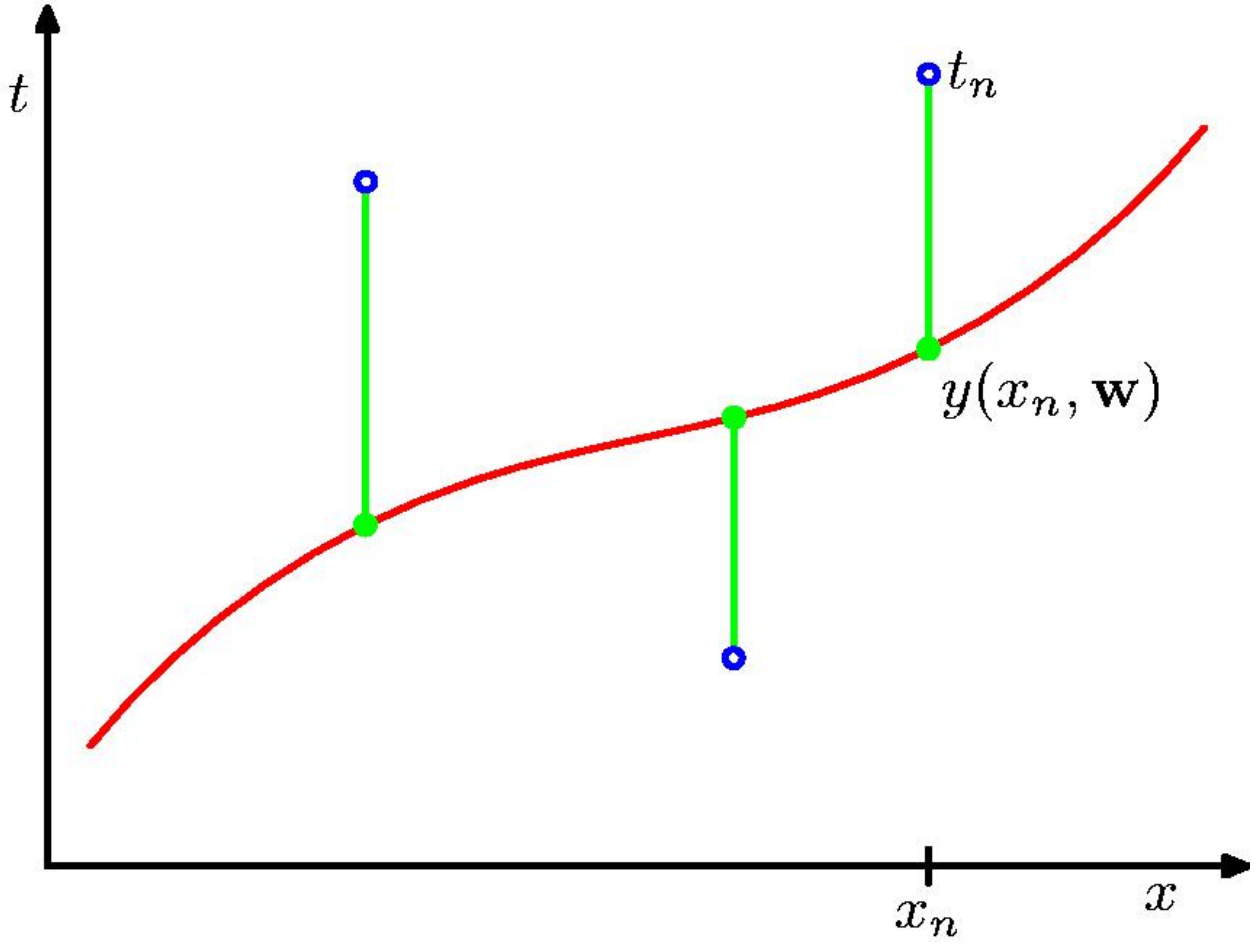
- Training Set (blue circles): $\mathbf{x} \equiv (x_1, \ldots, x_N)^T$
- Target Vector (green line): $\mathbf{t} \equiv (t_1, \ldots, t_N)^T$

Polynomial Curve Fitting



- Training Set (blue circles): $\mathbf{x} \equiv (x_1, \ldots, x_N)^T$
- Target Vector (green line): $\mathbf{t} \equiv (t_1, \ldots, t_N)^T$
- $y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$

Sum-of-Squares Error Function



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

Minimize Sum-of-Squares Error Function

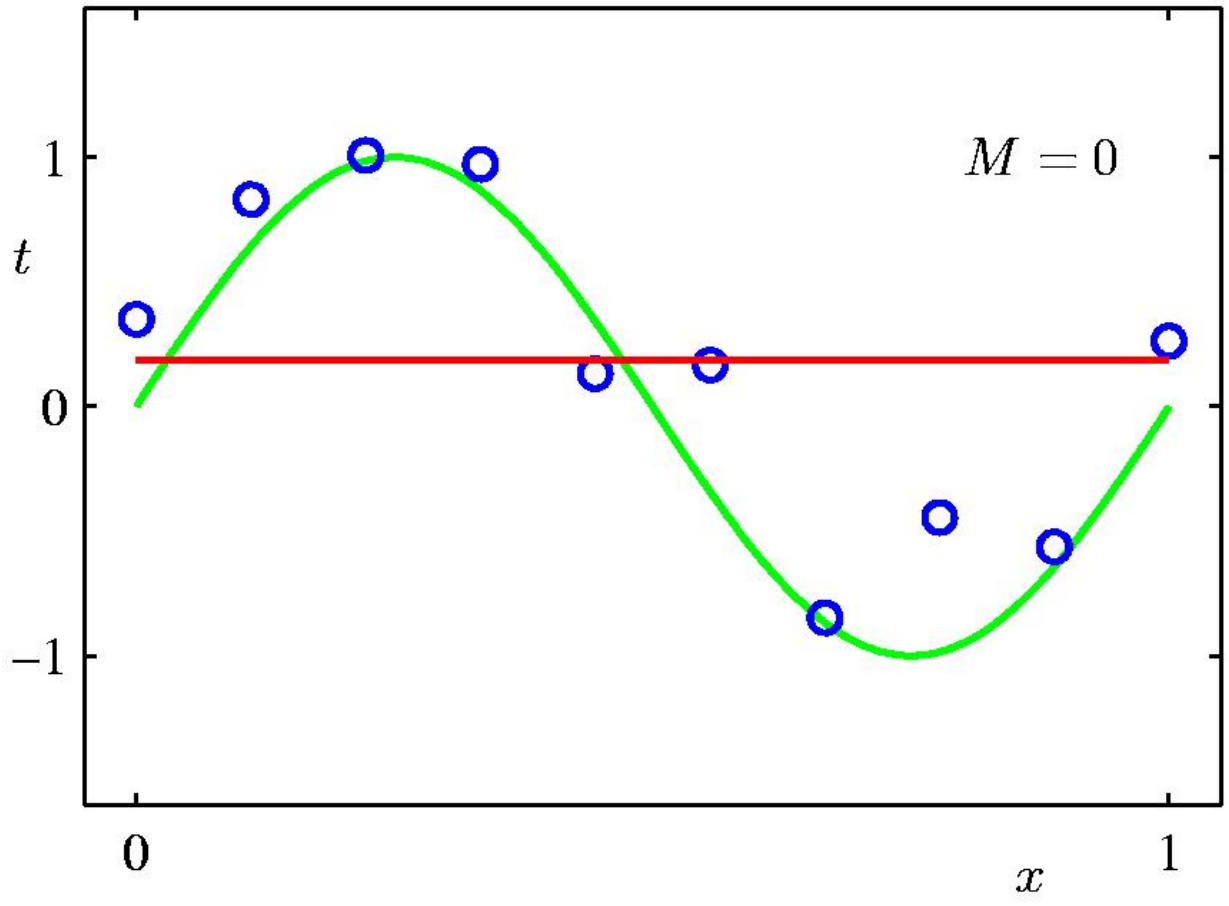
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 = \frac{1}{2}\sum_{n=1}^{N}\left(\sum_{j=0}^{M} w_j x_n^j - t_n\right)^2$$

$$\frac{\partial E(\mathbf{w})}{\partial w_j} = \sum_{n=1}^{N}\left(\sum_{j=0}^{M} w_j x_n^j - t_n\right) x_n^j$$

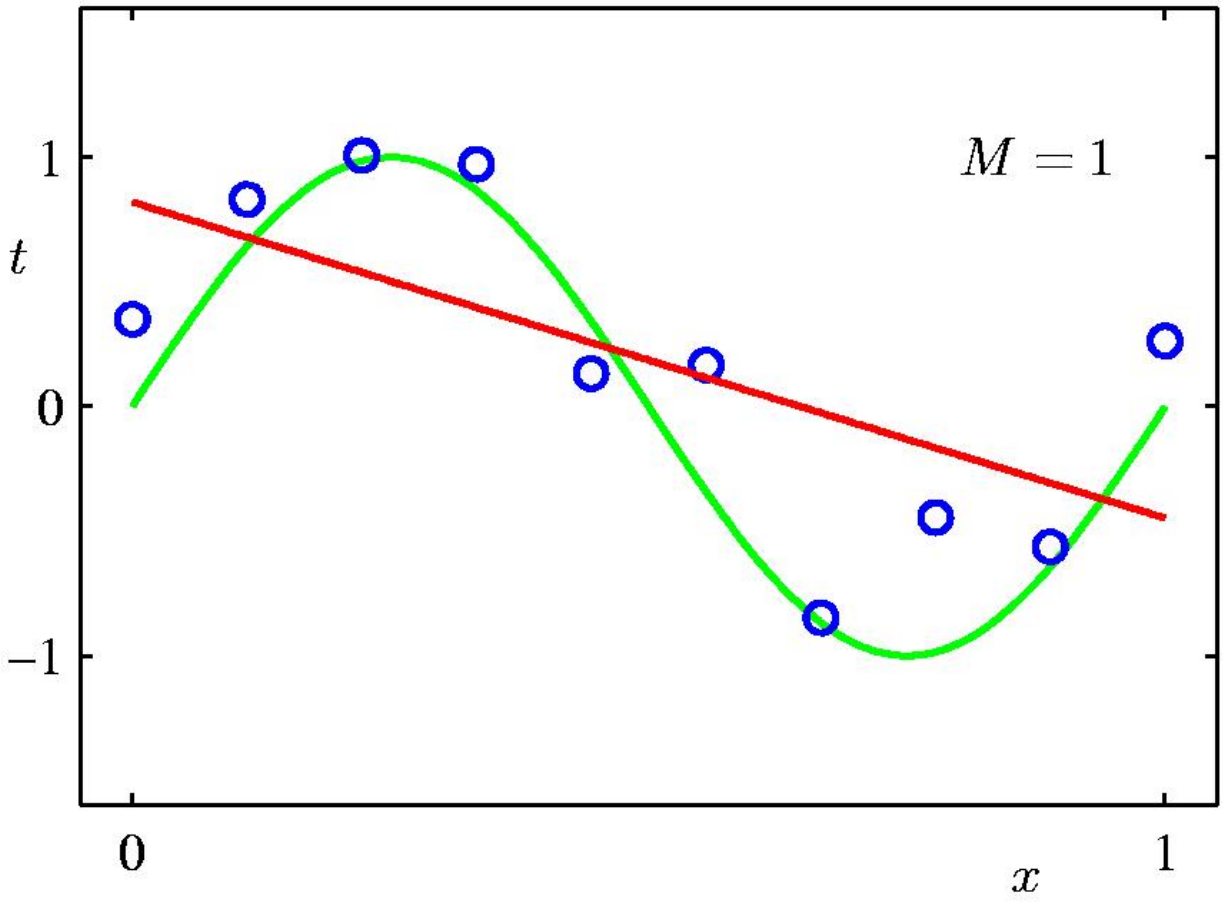
$$= \begin{bmatrix} x_1^j & \cdots & x_N^j \end{bmatrix} \left(\begin{bmatrix} x_1^0 & x_1 & \cdots & x_1^M \\ x_2^0 & x_2 & \cdots & x_2^M \\ \vdots & \vdots & \ddots & \vdots \\ x_N^0 & x_N & \cdots & x_N^M \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} \right)$$

$$= \begin{bmatrix} x_1^j & \cdots & x_N^j \end{bmatrix} (X\mathbf{w} - \mathbf{t}) = 0 \Rightarrow \mathbf{w} = (X^T X)^{-1} X^T \mathbf{t}$$

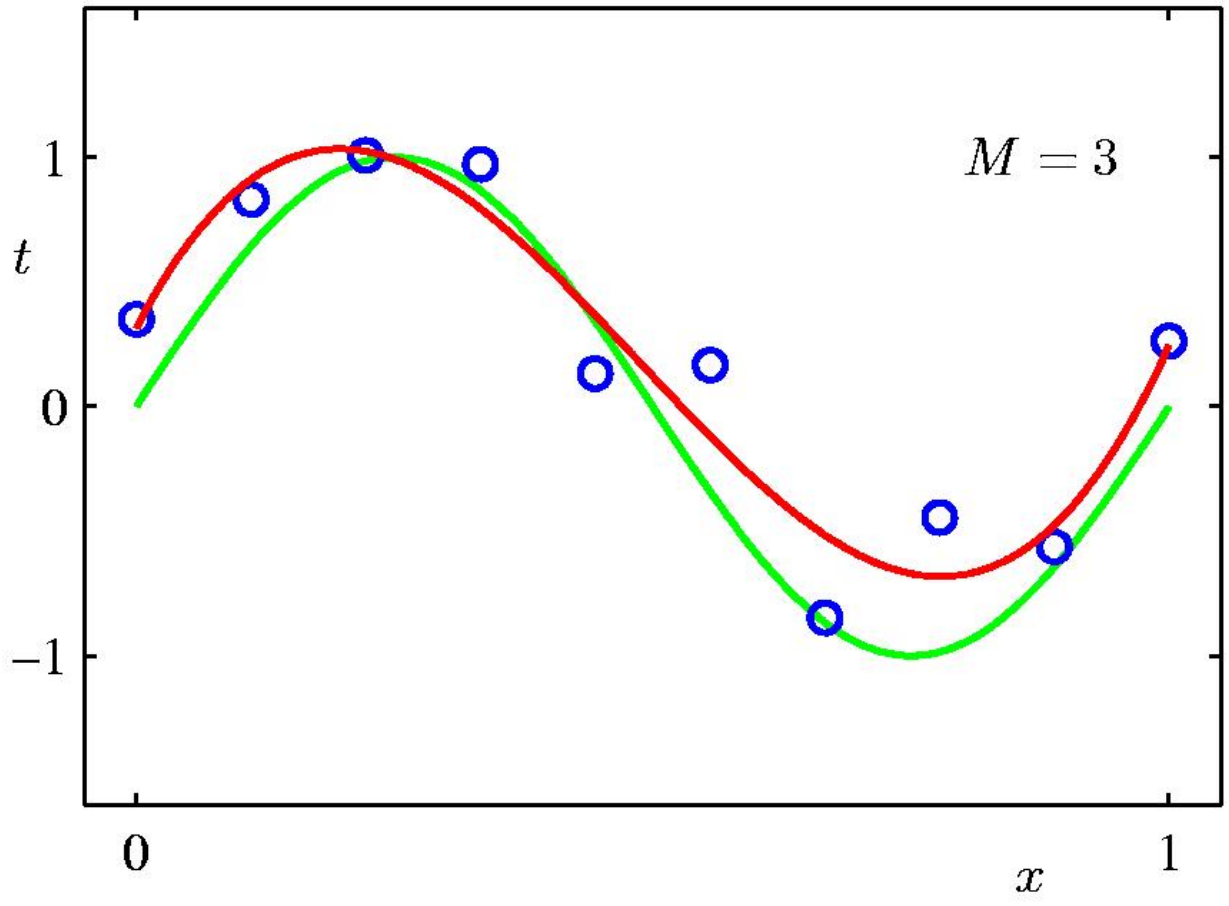
0^{th} Order Polynomial



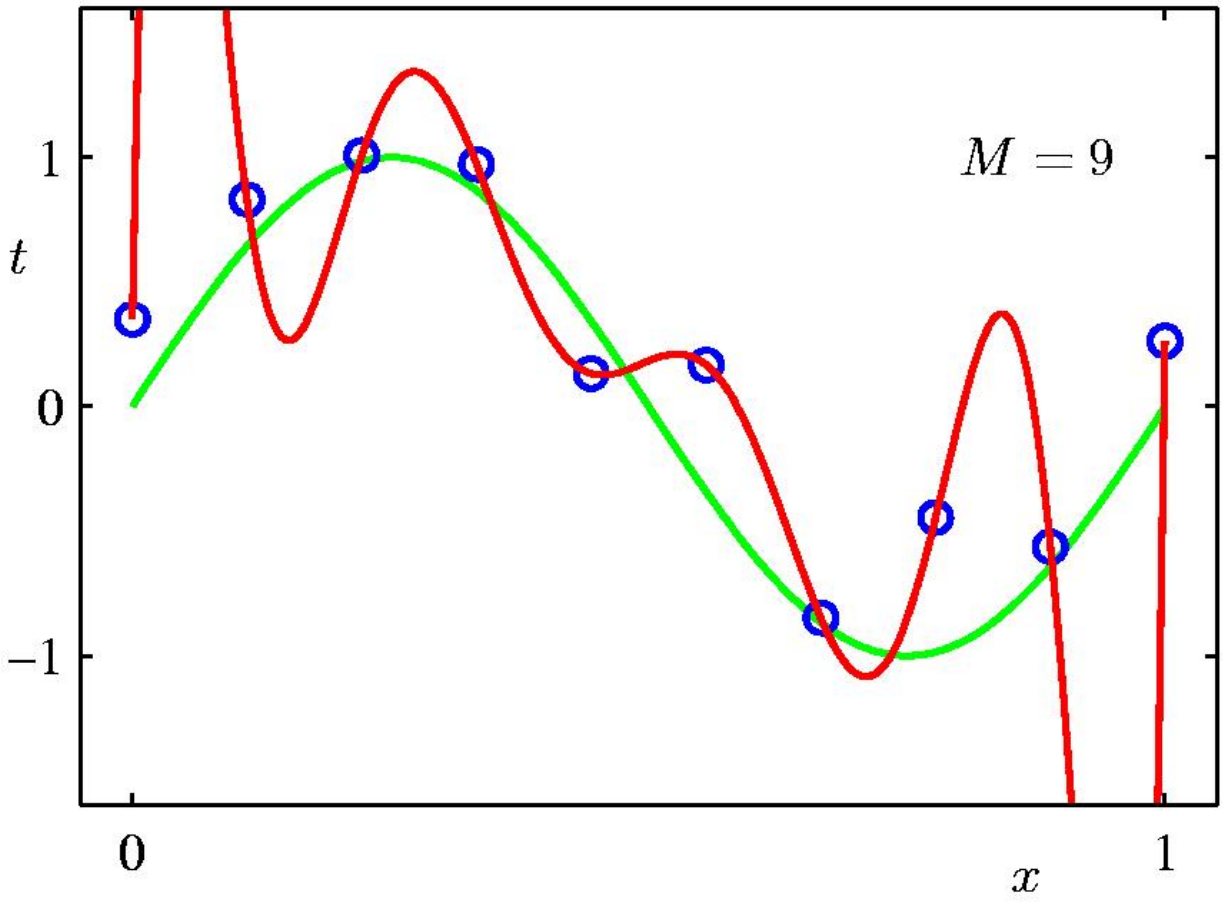
1st Order Polynomial



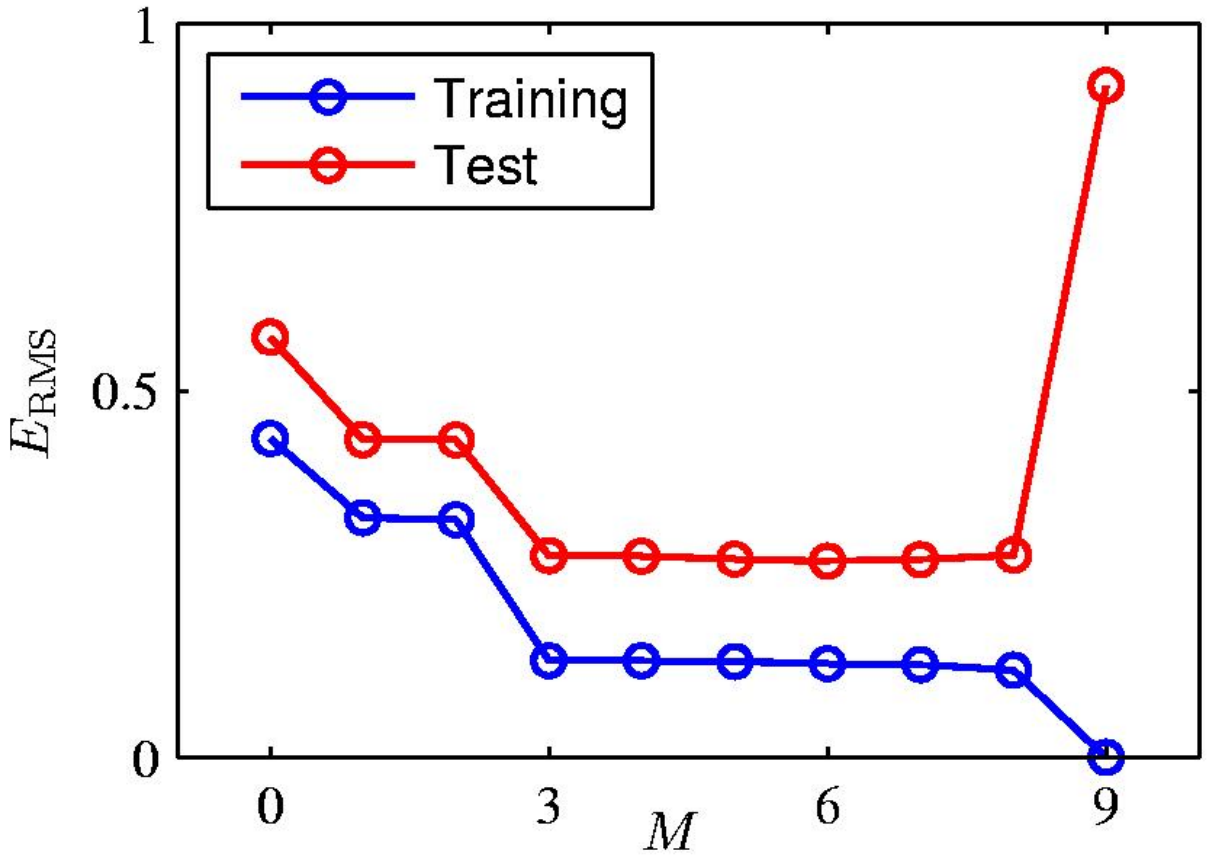
3^{rd} Order Polynomial



9^{th} Order Polynomial



Over-fitting

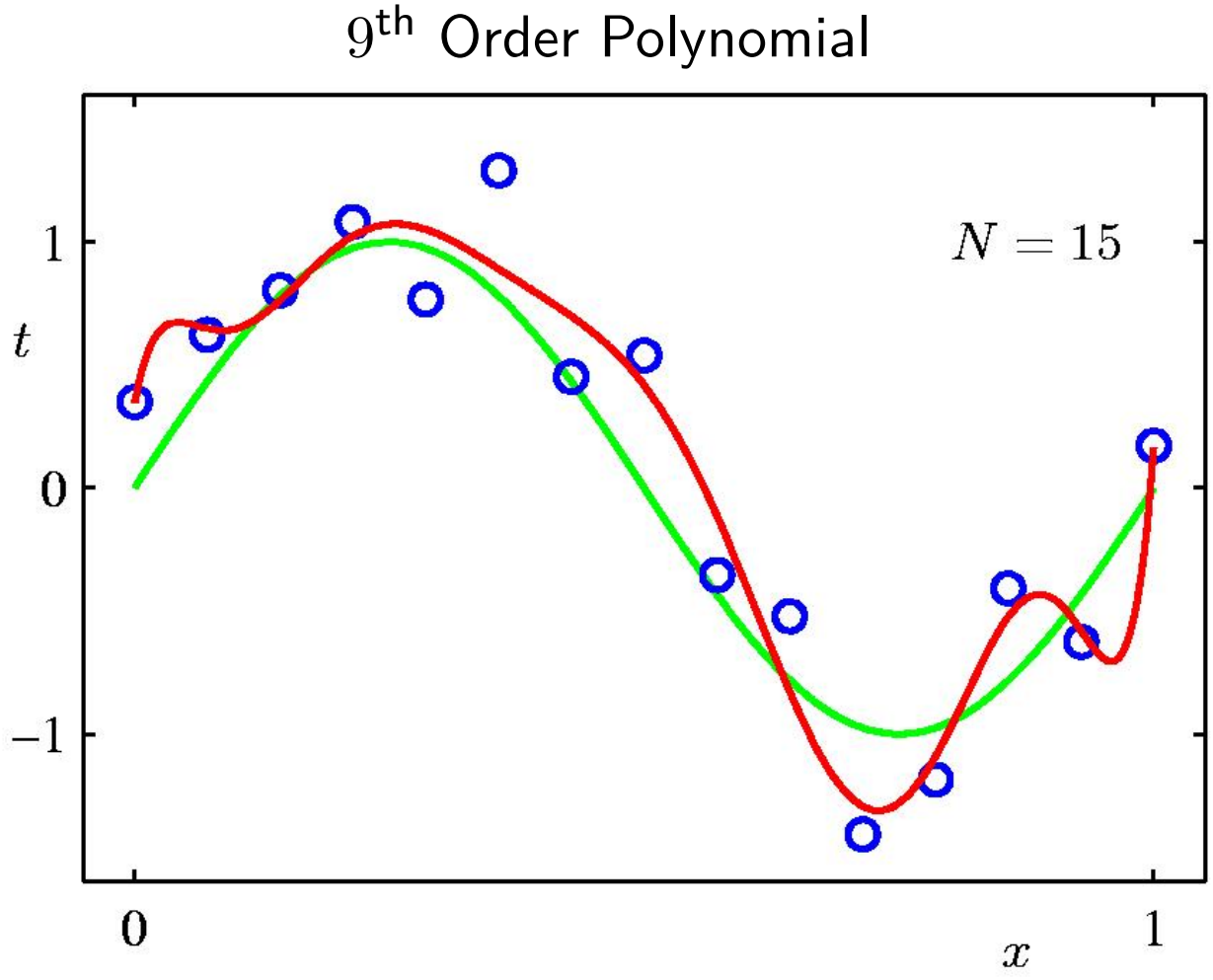


Root-Mean-Square (RMS) Error: $E_{RMS} = \sqrt{2E(\mathbf{w}^*)/N}$

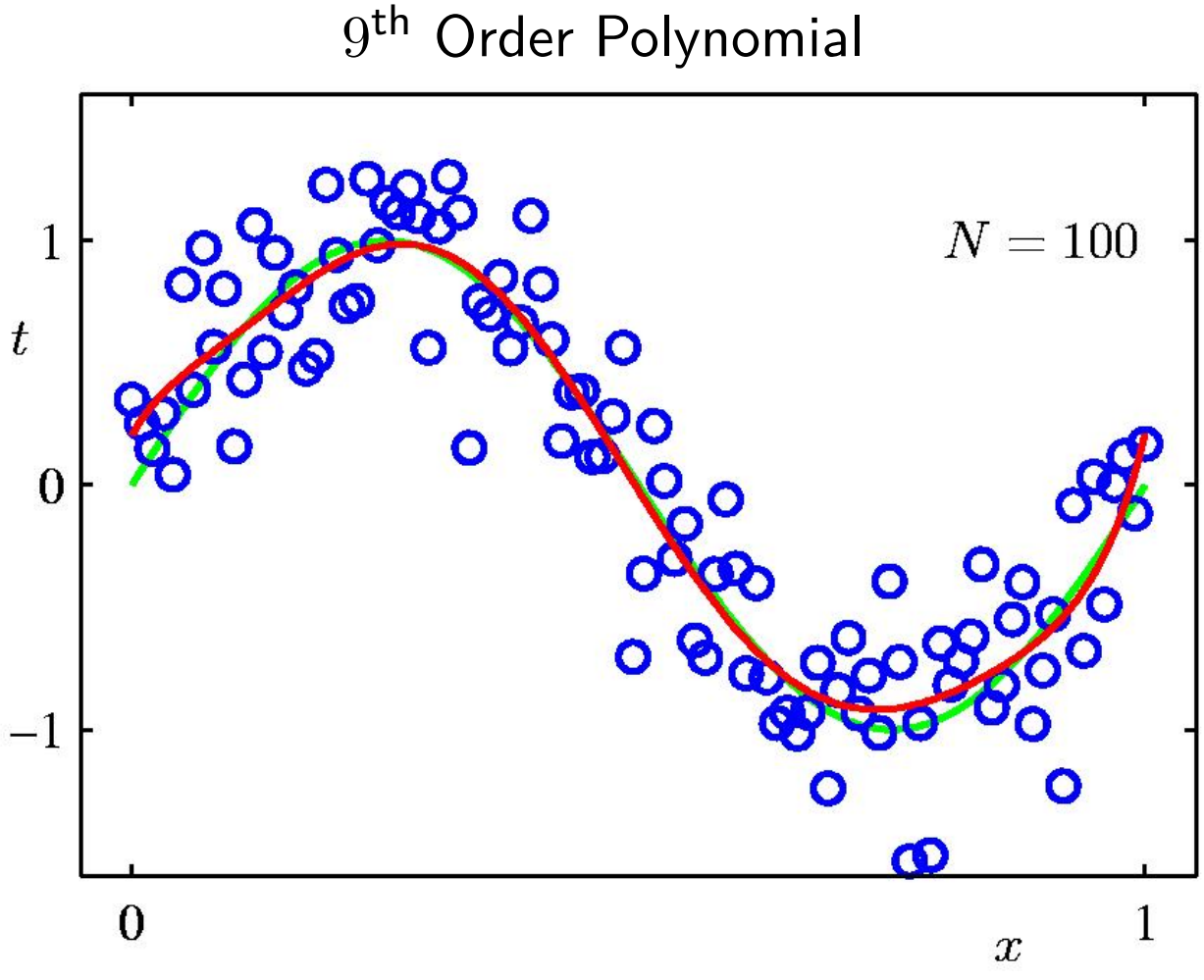
Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	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

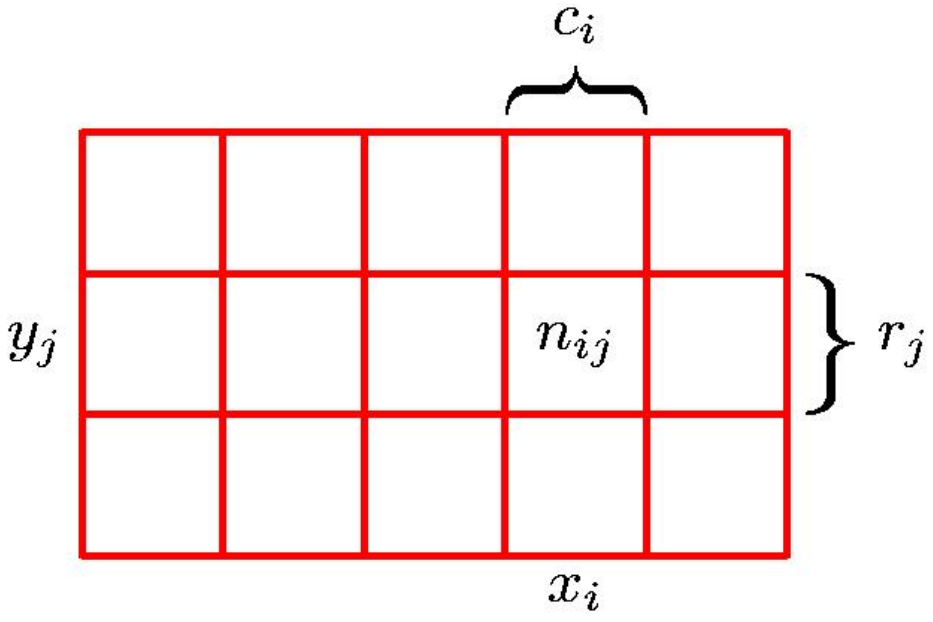
Data Set Size: $N = 15$



Data Set Size: $N = 100$



Probability Theory

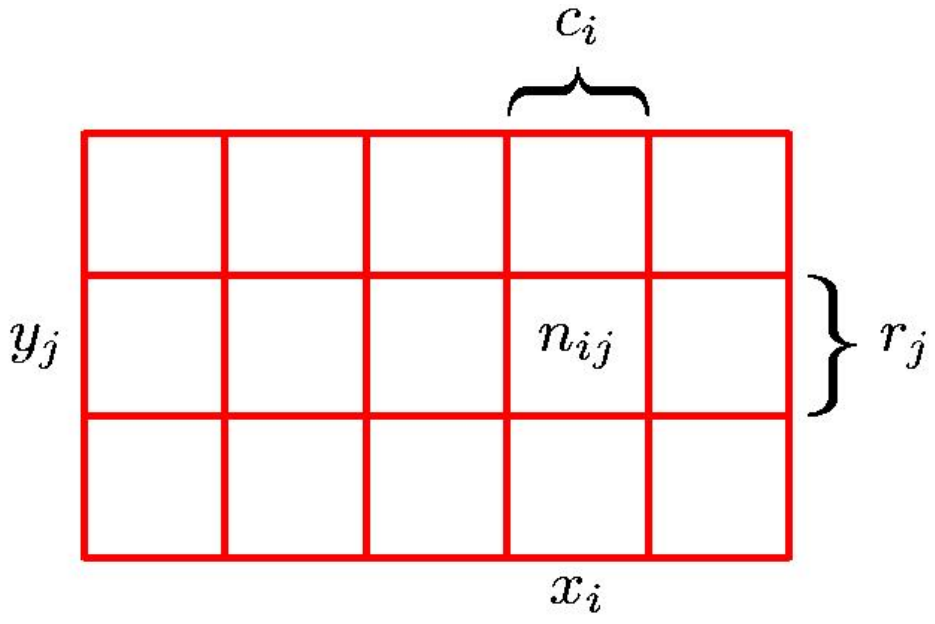


Marginal Probability: $p(X = x_i) = \frac{c_i}{N}$.

Joint Probability: $p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$.

Conditional Probability: $p(Y = y_j|X = x_i) = \frac{n_{ij}}{c_i}$.

Probability Theory



Sum Rule:

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N} \sum_{j=1}^{L} n_{ij} = \sum_{j=1}^{L} p(X = x_i, Y = y_j).$$

Product Rule:

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N} = p(Y = y_j | X = x_i) p(X = x_i).$$

The Rules of Probability

- Sum Rule $\quad p(X) = \sum_{Y} p(X, Y)$
- Product Rule $\quad p(X, Y) = p(Y|X)p(X)$

Bayes' Theorem

By Product Rule, we have

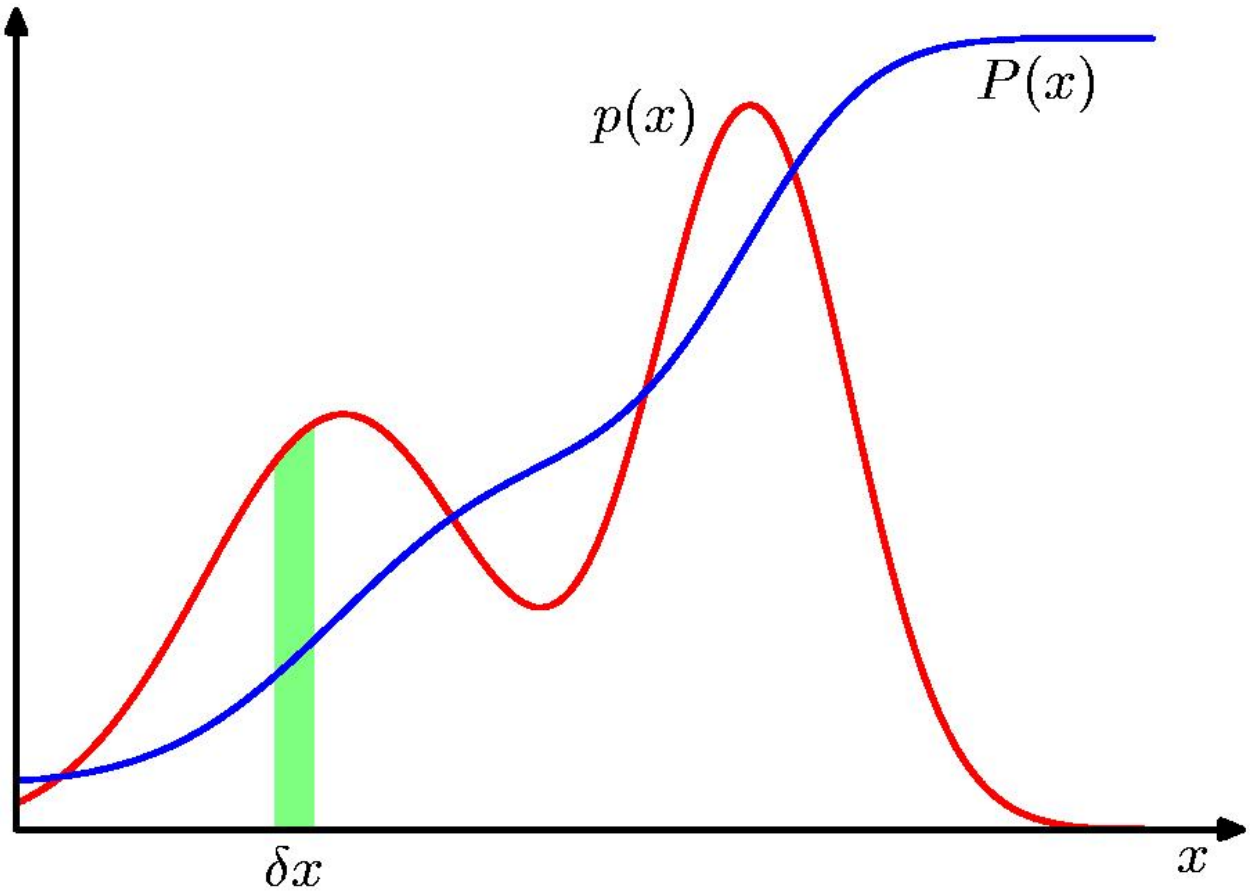
$$p(X, Y) = p(Y, X) \Rightarrow p(Y|X)p(X) = p(X|Y)p(Y)$$

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

$$p(X) = \sum_{Y} P(X|Y)p(Y)$$

posterior $\propto$ likelihood $\times$ prior

Probability Density



$$P(z) = \int_{-\infty}^{z} p(x)dx$$

$$p(x) \geq 0 \qquad \int_{-\infty}^{\infty} p(x)dx = 1$$

Expectations

$\mathbb{E}[f] = \sum_x p(x) f(x)$	$\mathbb{E}[f] = \int p(x) f(x) dx$
$\mathbb{E}[f\|y] = \sum_x p(x\|y) f(x)$	Conditional Expectation
$\mathbb{E}[f] \approx \frac{1}{N} \sum_{n=1}^{N} f(x_n)$	Approximate Expectation

Variances and Covariances

$$\mathrm{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2.$$

$$\mathrm{cov}[x, y] = \mathbb{E}_{x,y}[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}] = \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y].$$

$$\mathrm{cov}[\mathbf{x}, \mathbf{y}] = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^T - \mathbb{E}[\mathbf{y}^T]\}] = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^T].$$