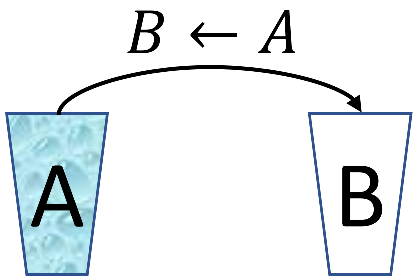
Introduction to Machine Learning

Sadegh Eskandari
Department of Computer Science, University of Guilan

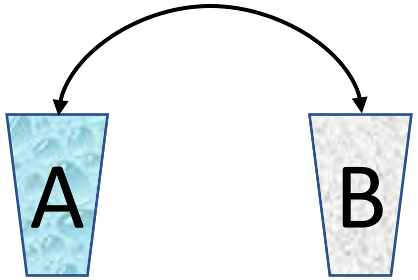
Introduction

Consider a robot with a single capability: pouring one glass into another

Operator



Question: how the robot can swap the contents of two glasses?



Problem

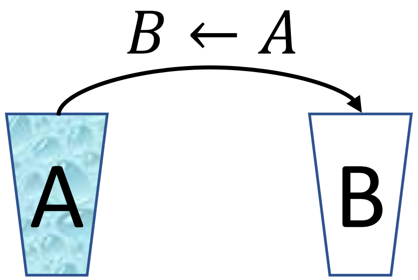
$$C \leftarrow A$$
$$A \leftarrow B$$
$$B \leftarrow C$$

Algorithm

Introduction

Consider a robot with a single capability: pouring one glass into another

Operator



A, B

$\Downarrow$

$\max(A, B)$

Problem

Another Question: How to find the max of two glasses?

The problem is unsolvable by the robot. Why?

- The comparison operation is not defined for the robot
- To solve the problem we should change the operator

Introduction

Operator (Computer)

Capabilities:

- Input/Output (I/O)
- memory W/R
- Some basic arithmetic and logical operations (+,-,*,/, %, and, or, not, …)

$$A, B$$

$$\Downarrow$$

$$\max(A, B)$$

Problem

Problem: How to find the max of two glasses?

```
read A,B
if A>B:
    max = A
else:
    max = B
print max
```

Algorithm

Introduction

- A problem is said to be Decidable if we can always construct an algorithm that can solve the problem correctly.
- An example of undecidable problems:

  Can one algorithm specify the output of another algorithm?

- Decidability does not mean simplicity!
  - Traveling Salesman Problem (TSP): simple to program but hard to execute
  - Recognizing dogs and cats in an image: simple to do but hard to program

Introduction

Traveling Salesman Problem (TSP)

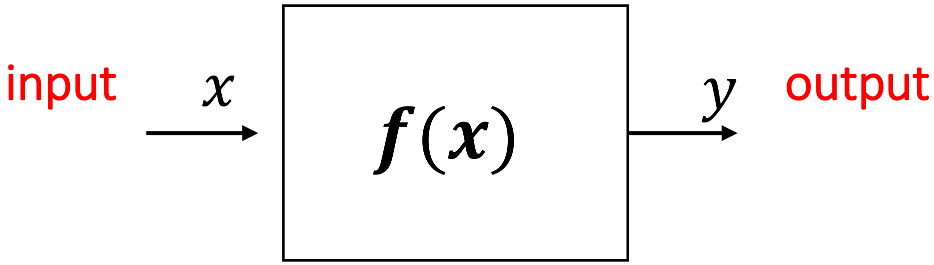
- For a given weighted complete graph with n nodes, find the Hamilton circuit with minimum length.
- An algorithm should compare $(n-1)!$ circuits to find the best one.
- Time required to run this algorithm on a good computer:
  - $n = 4$ then $time \approx 0.000000007s$
  - $n = 99$ then $time \approx 3.1 \times 10^{140}\ years$ ☹

Introduction

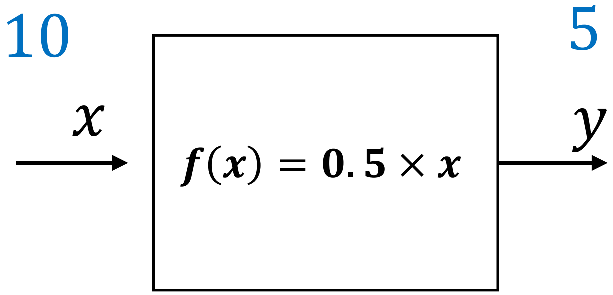
Dogs vs Cats

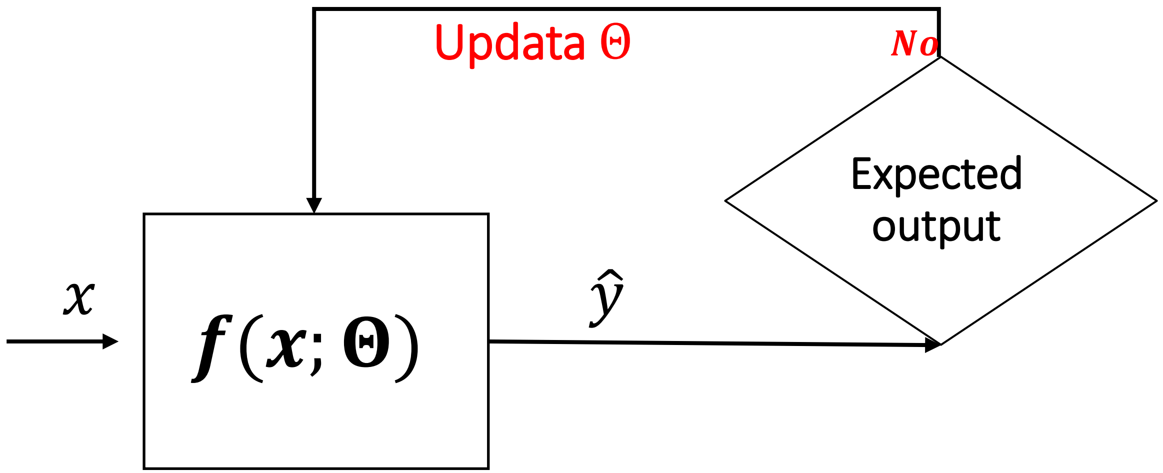
An effective approach: Machine Learning

Algorithms that Can Learn



A typical algorithm or function

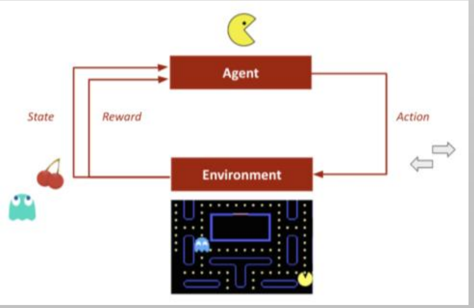




A Parametric Algorithm (Model)

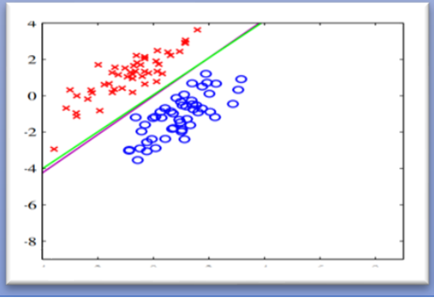
Algorithms that Can Learn

(Reinforcement)

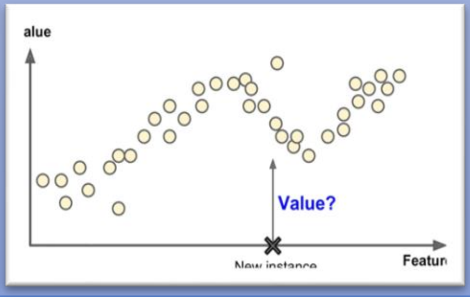


(Supervised)

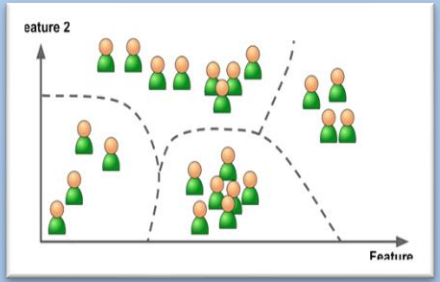
(Classification)



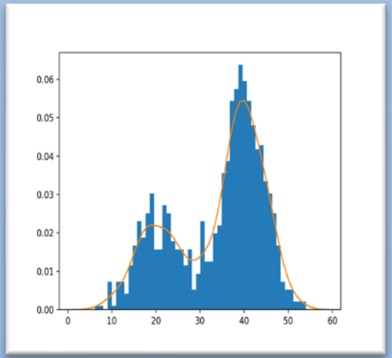
(Regression)



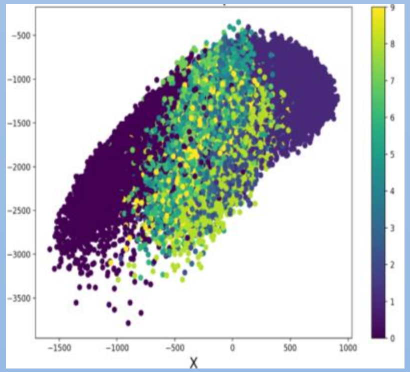
(Unsupervised)



(Clustering)



(Density Estimation)



(Visualization)

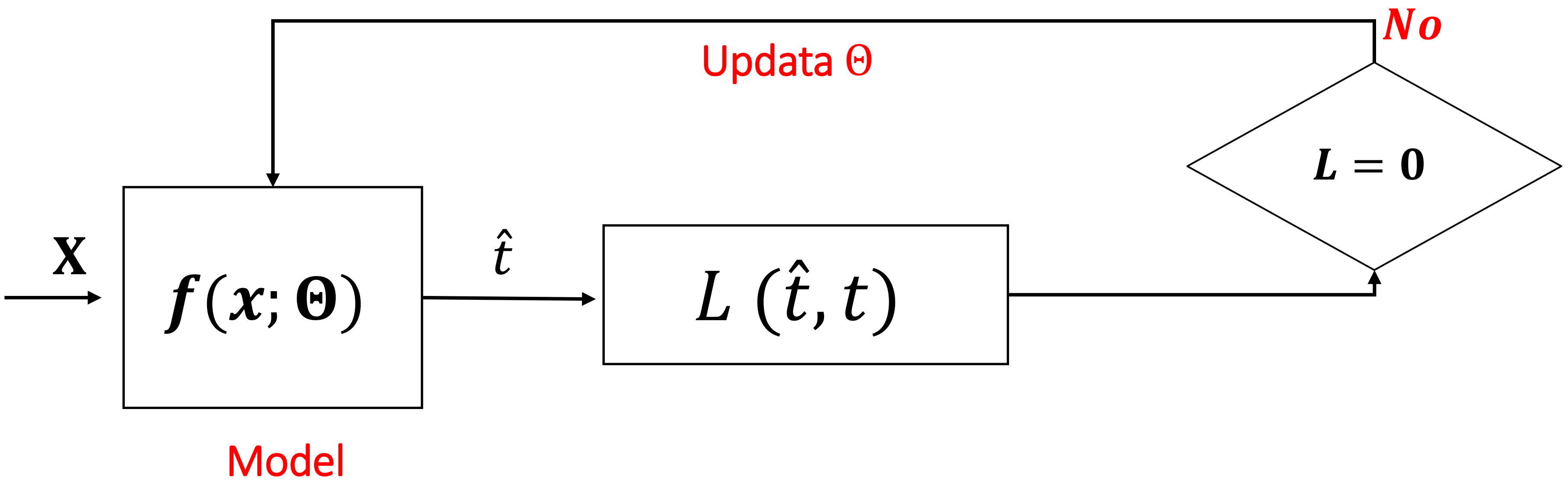
Supervised Learning

- Suppose that we are given a training set comprising N observations of random variable x (training set) :

$$\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N)^{\boldsymbol{T}}$$

- Moreover, for each observation $\boldsymbol{x}_i$ we are given a target value t_i (training target):

$$\mathbf{t} = (t_1, t_2, \dots, t_N)^{\boldsymbol{T}}$$



Example: Polynomial Curve Fitting

- $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$ is generated uniformaly in $[0,1]$.
- $\boldsymbol{t} = \{t_i |\ t_i = \sin(2\pi x) + \mathcal{N}(0,0.3)\,, i = 1, 2, \dots, N\}$

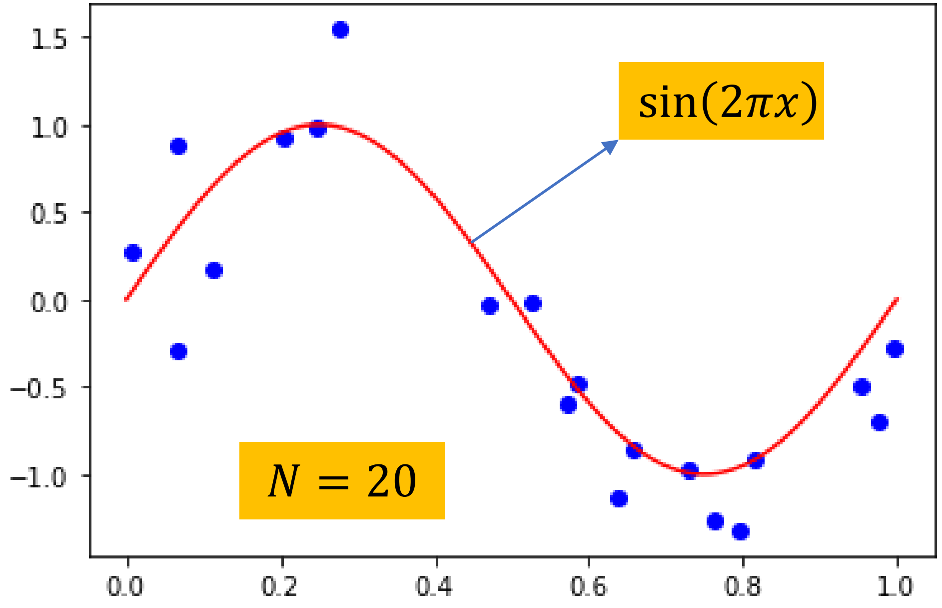
- The generating function in not known and the aim is to estimate it such that:
  - ❑ The estimated function should describe the training data
  - ❑ The estimated function should generalize to new data

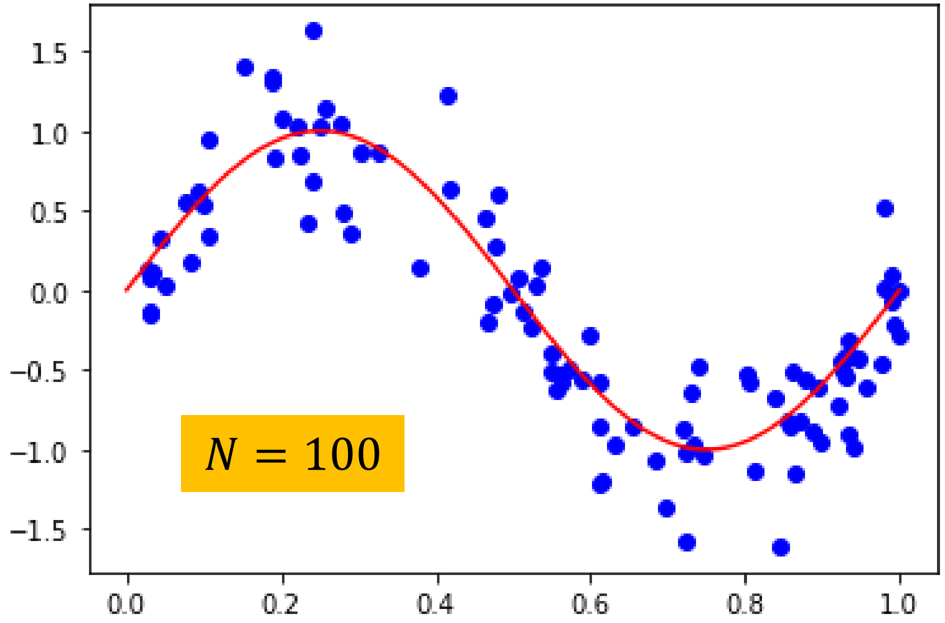
- In particular, we shall fit the data using a polynomial function of the form

$$y(x; \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$$

  - ❑ M: the order of polynomial
  - ❑ $w \equiv [w_0, w_1, \dots, w_M]$: The model parameters (unknown in advance)

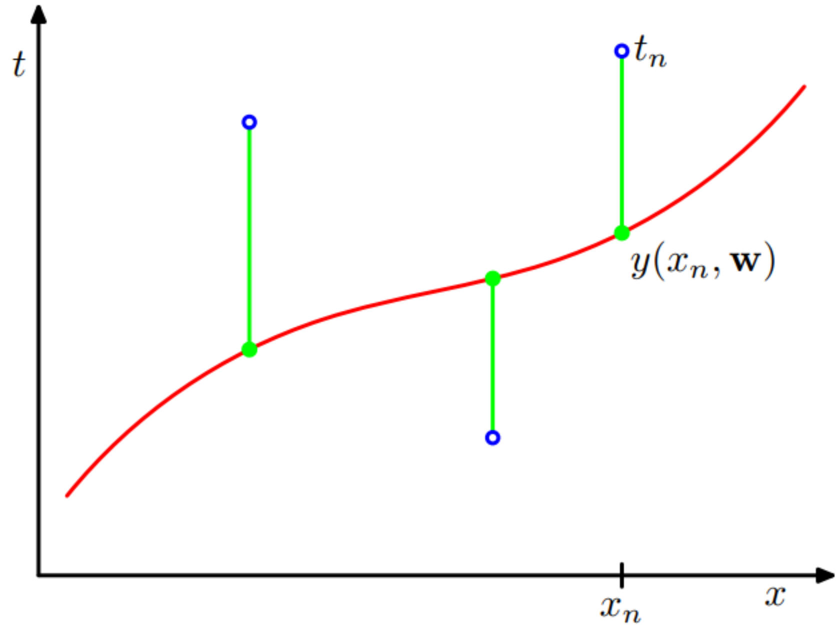
- $y(x, \boldsymbol{w})$ is a linear function of the coefficients $\boldsymbol{w}$. Such functions are called linear models.





Example: Polynomial Curve Fitting

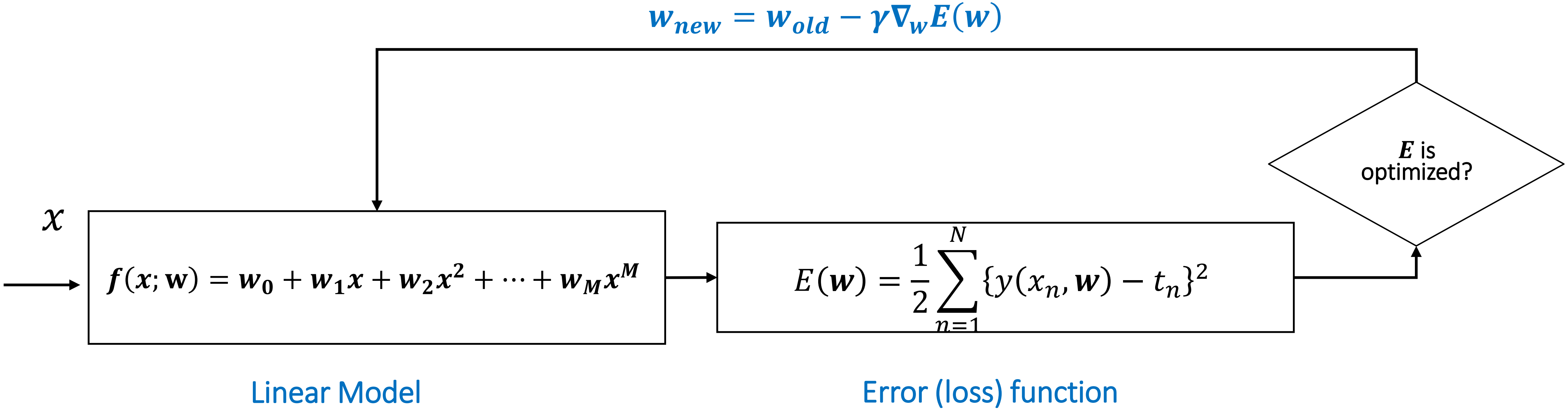
- An error function (loss function) is required to measure the misfit between the function $y(x, \boldsymbol{w})$, for any given $\boldsymbol{w}$, and the training data points.



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

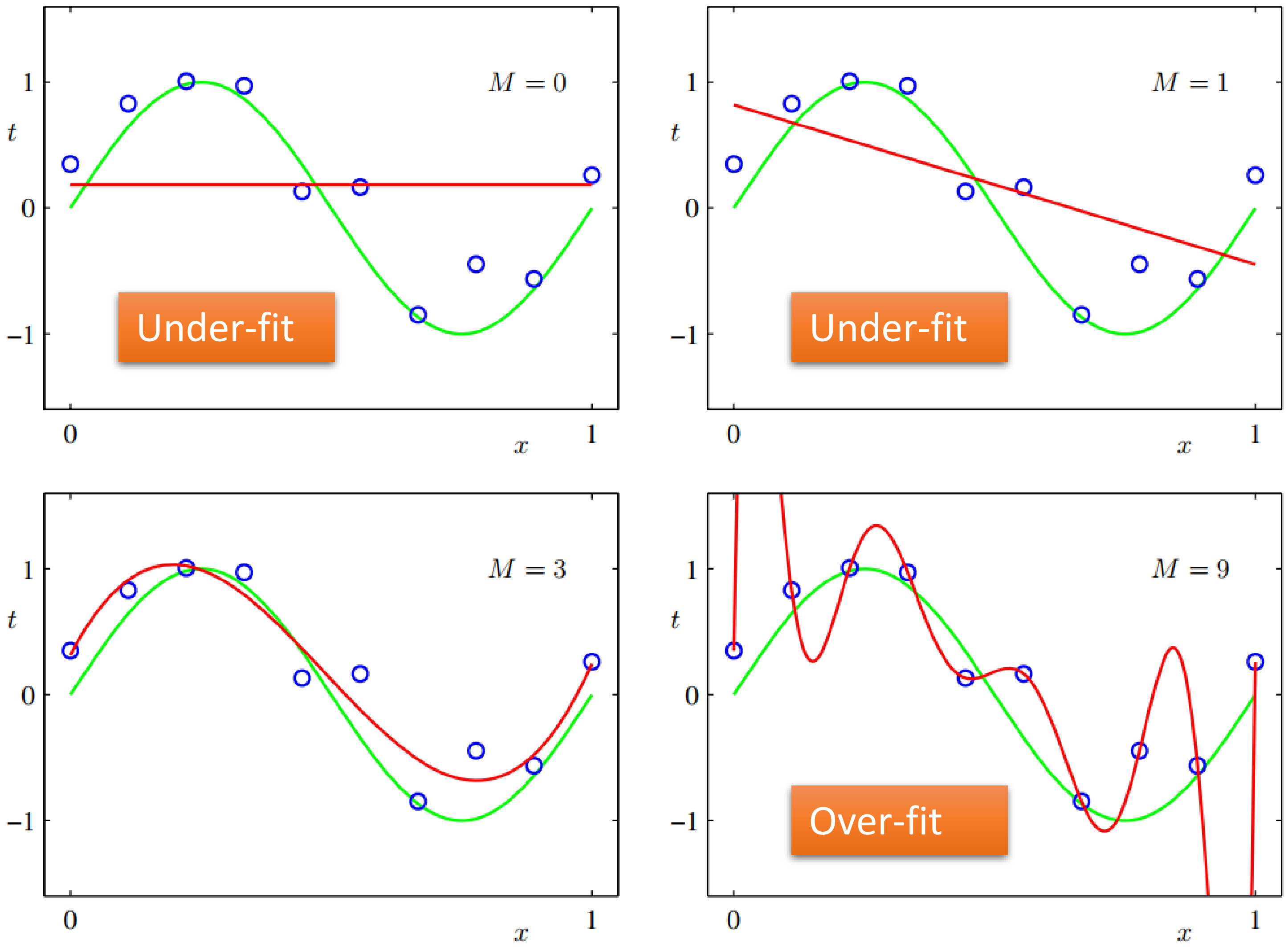
- $E(\boldsymbol{w})$ is a quadratic function of $\boldsymbol{w}$,
- Therefore $\frac{\partial E}{\partial \boldsymbol{w}}$ is linear in the elements of $\boldsymbol{w}$, and so the minimization of the error function has a unique solution, which can be found in closed form.

Example: Polynomial Curve Fitting

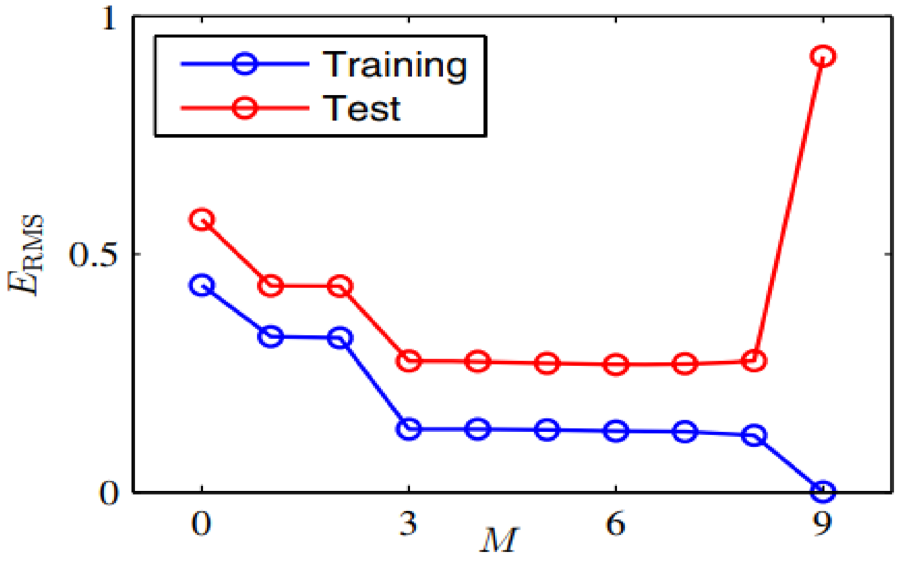


Example: Polynomial Curve Fitting

Model Selection (Model Comparison)



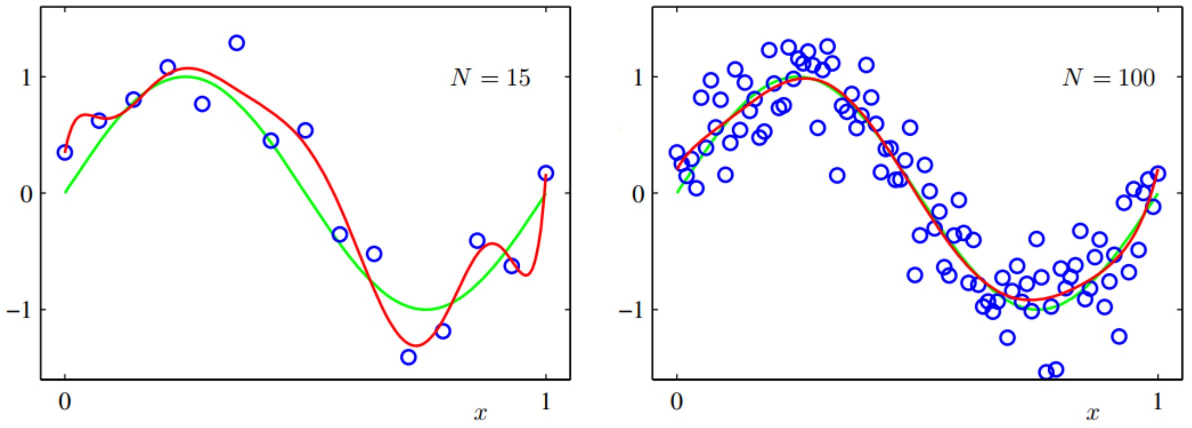
	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43



Example: Polynomial Curve Fitting

Model Selection (Model Comparison)

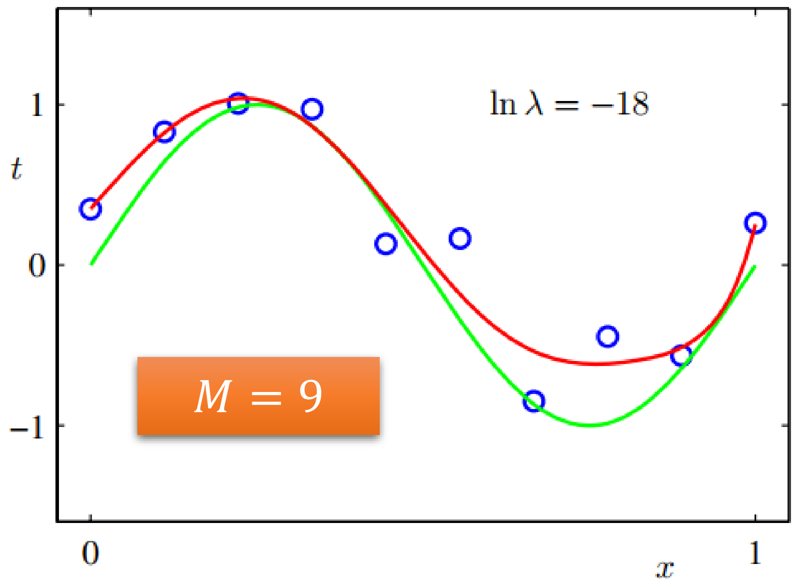
- For a given model complexity, the over-fitting problem become less severe as the size of the data set increases.



- One technique that to control the over-fitting phenomenon regularization, which involves adding a penalty term to the error function.

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

Where $\|\boldsymbol{w}\|^2 = \boldsymbol{w}^T\boldsymbol{w} = w_0^2 + w_1^2 + \cdots + w_M^2$



Example: Polynomial Curve Fitting

- The least squares approach is a specific case of *maximum likelihood* (will be discussed later)
- The over-fitting problem is a general property of maximum likelihood.
- By adopting a *Bayesian* approach, the over-fitting problem can be avoided.
- In a Bayesian model the *effective* number of parameters adapts automatically to the size of the data set.