Data Mining Techniques

CS 6220 - Section 3 - Fall 2016

Lecture 2: Regression

Jan-Willem van de Meent
(*credit*: Yijun Zhao, Marc Toussaint, Bishop)

Administrativa

Instructor

Jan-Willem van de Meent

Email: j.vandemeent@northeastern.edu
Phone: +1 617 373-7696
Office Hours: 478 WVH, Wed 1.30pm - 2.30pm

Teaching Assistants

Yuan Zhong

E-mail: yzhong@ccs.neu.edu
Office Hours: WVH 462, Wed 3pm - 5pm

Kamlendra Kumar

E-mail: kumark@zimbra.ccs.neu.edu
Office Hours: WVH 462, Fri 3pm - 5pm

Administrativa

Course Website

http://www.ccs.neu.edu/course/cs6220f16/sec3/

Piazza

https://piazza.com/northeastern/fall2016/cs622003/home

Project Guidelines (Vote next week)

http://www.ccs.neu.edu/course/cs6220f16/sec3/project/

Question

What would ***you*** like to get out of this course?

Linear Regression

Regression Examples

Features **Continuous Value**

$$\mathbf{x} \Longrightarrow y$$

- {age, major, gender, race} ⇒ GPA
- {income, credit score, profession} ⇒ Loan Amount
- {college,major,GPA} ⇒ Future Income

Example: Boston Housing Data

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0.0	0.524	6.012	66.6	5.5605	5.0	311.0	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0.0	0.524	6.172	96.1	5.9505	5.0	311.0	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0.0	0.524	5.631	100.0	6.0821	5.0	311.0	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0.0	0.524	6.004	85.9	6.5921	5.0	311.0	15.2	386.71	17.10	18.9
10	0.22489	12.5	7.87	0.0	0.524	6.377	94.3	6.3467	5.0	311.0	15.2	392.52	20.45	15.0

UC Irvine Machine Learning Repository

(*good source for project datasets*)

https://archive.ics.uci.edu/ml/datasets/Housing

Example: Boston Housing Data

1. **CRIM**: per capita crime rate by town
2. **ZN**: proportion of residential land zoned for lots over 25,000 sq.ft.
3. **INDUS**: proportion of non-retail business acres per town
4. **CHAS**: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. **NOX**: nitric oxides concentration (parts per 10 million)
6. **RM**: average number of rooms per dwelling
7. **AGE**: proportion of owner-occupied units built prior to 1940
8. **DIS**: weighted distances to five Boston employment centres
9. **RAD**: index of accessibility to radial highways
10. **TAX**: full-value property-tax rate per $10,000
11. **PTRATIO**: pupil-teacher ratio by town
12. **B**: 1000(Bk - 0.63)^2 where Bk is the proportion of african americans by town
13. **LSTAT**: % lower status of the population
14. **MEDV**: Median value of owner-occupied homes in $1000's

Example: Boston Housing Data

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0.0	0.524	6.012	66.6	5.5605	5.0	311.0	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0.0	0.524	6.172	96.1	5.9505	5.0	311.0	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0.0	0.524	5.631	100.0	6.0821	5.0	311.0	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0.0	0.524	6.004	85.9	6.5921	5.0	311.0	15.2	386.71	17.10	18.9
10	0.22489	12.5	7.87	0.0	0.524	6.377	94.3	6.3467	5.0	311.0	15.2	392.52	20.45	15.0

CRIM: per capita crime rate by town

Example: Boston Housing Data

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0.0	0.524	6.012	66.6	5.5605	5.0	311.0	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0.0	0.524	6.172	96.1	5.9505	5.0	311.0	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0.0	0.524	5.631	100.0	6.0821	5.0	311.0	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0.0	0.524	6.004	85.9	6.5921	5.0	311.0	15.2	386.71	17.10	18.9
10	0.22489	12.5	7.87	0.0	0.524	6.377	94.3	6.3467	5.0	311.0	15.2	392.52	20.45	15.0

CHAS: Charles River dummy variable
(= 1 if tract bounds river; 0 otherwise)

Example: Boston Housing Data

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0.0	0.524	6.012	66.6	5.5605	5.0	311.0	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0.0	0.524	6.172	96.1	5.9505	5.0	311.0	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0.0	0.524	5.631	100.0	6.0821	5.0	311.0	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0.0	0.524	6.004	85.9	6.5921	5.0	311.0	15.2	386.71	17.10	18.9
10	0.22489	12.5	7.87	0.0	0.524	6.377	94.3	6.3467	5.0	311.0	15.2	392.52	20.45	15.0

MEDV: Median value of owner-occupied homes in $1000's

Example: Boston Housing Data

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0.0	0.524	6.012	66.6	5.5605	5.0	311.0	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0.0	0.524	6.172	96.1	5.9505	5.0	311.0	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0.0	0.524	5.631	100.0	6.0821	5.0	311.0	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0.0	0.524	6.004	85.9	6.5921	5.0	311.0	15.2	386.71	17.10	18.9
10	0.22489	12.5	7.87	0.0	0.524	6.377	94.3	6.3467	5.0	311.0	15.2	392.52	20.45	15.0

N data points

D features

Regression: Problem Setup

Given *N* observations

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$$

learn a function

$$y_i = f(\boldsymbol{x}_i) \qquad \forall i = 1, 2, \ldots, N$$

and for a new input **x*** predict

$$y^* = f(\boldsymbol{x}^*)$$

Linear Regression

Assume *f* is a linear combination of *D* features

$$y = w_0 + w_1 x_1 + \ldots + w_D x_D = \boldsymbol{w}^\top \boldsymbol{x}$$

were **x** and **w** are defined as

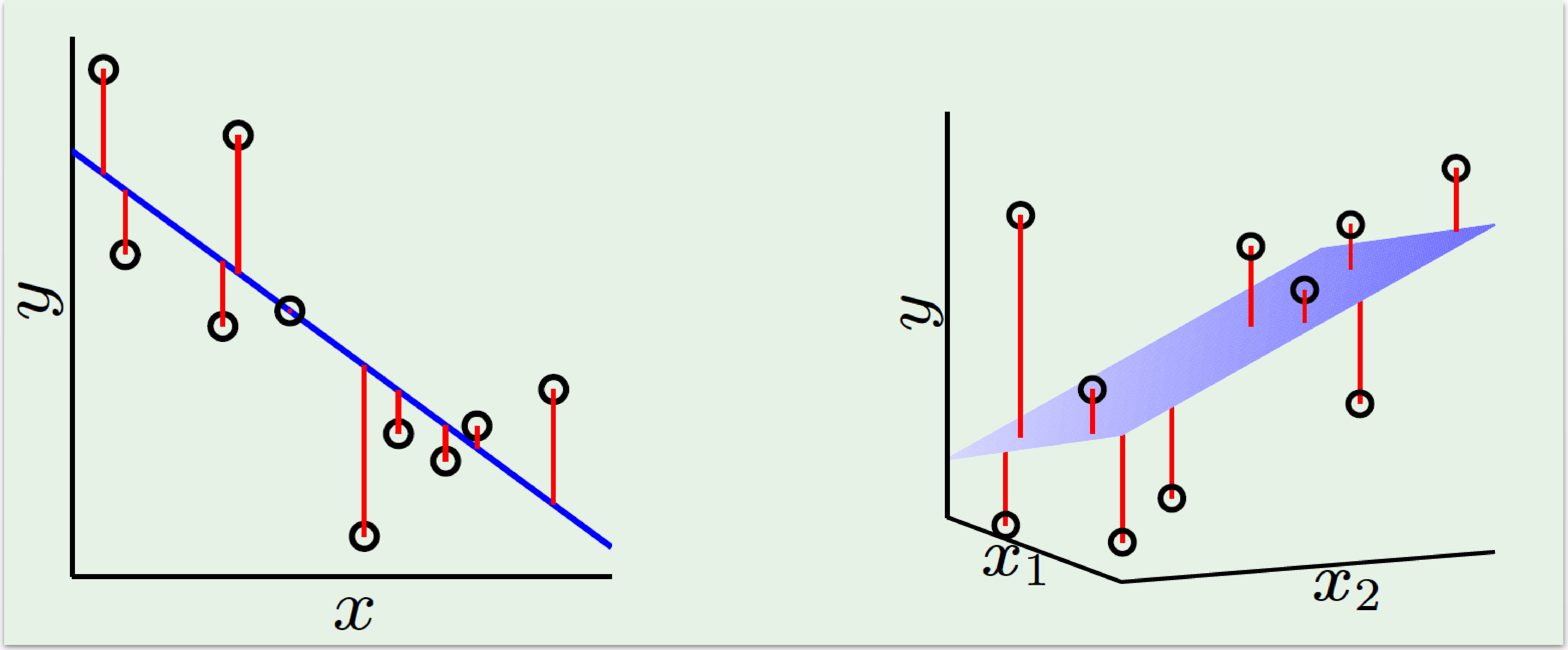
$$\boldsymbol{x} = (1, x_1, \ldots, x_D) \qquad \boldsymbol{w} = (w_0, w_1, \ldots w_D)$$

for N points we write

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} \qquad \boldsymbol{y} = (y_1, \ldots, y_N) \qquad \boldsymbol{X} = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_N^\top)$$

Learning task: Estimate **w**

Linear Regression



Error Measure

Mean Squared Error (MSE):

$$E(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n - y_n)^2$$

$$= \frac{1}{N}\|\,\mathbf{X}\mathbf{w} - \mathbf{y}\,\|^2$$

where

$$\mathbf{X} = \begin{bmatrix} — \mathbf{x}_1{}^T — \\ — \mathbf{x}_2{}^T — \\ \cdots \\ — \mathbf{x}_N{}^T — \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1{}^T \\ y_2{}^T \\ \cdots \\ y_N{}^T \end{bmatrix}$$

Minimizing the Error

$$E(\mathbf{w}) = \frac{1}{N} \parallel \mathbf{X}\mathbf{w} - \mathbf{y} \parallel^2$$

$$\nabla E(\mathbf{w}) = \frac{2}{N}\mathbf{X}^{\mathsf{T}}(\mathbf{X}\mathbf{w} - \mathbf{y}) = \mathbf{0}$$

$$\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathbf{w} = \mathbf{X}^{\mathsf{T}}\mathbf{y}$$

$$\mathbf{w} = \mathbf{X}^{\dagger}\mathbf{y}$$

where $\mathbf{X}^{\dagger} = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}$ is the 'pseudo-inverse' of $\mathbf{X}$

Minimizing the Error

$$E(\mathbf{w}) = \frac{1}{N} \| \mathbf{X}\mathbf{w} - \mathbf{y} \|^2$$

$$\nabla E(\mathbf{w}) = \frac{2}{N}\mathbf{X}^{\mathsf{T}}(\mathbf{X}\mathbf{w} - \mathbf{y}) = \mathbf{0}$$

$$\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathbf{w} = \mathbf{X}^{\mathsf{T}}\mathbf{y}$$

$$\mathbf{w} = \mathbf{X}^{\dagger}\mathbf{y}$$

where $\mathbf{X}^{\dagger} = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}$ is the 'pseudo-inverse' of $\mathbf{X}$

Matrix Cookbook (on course website)

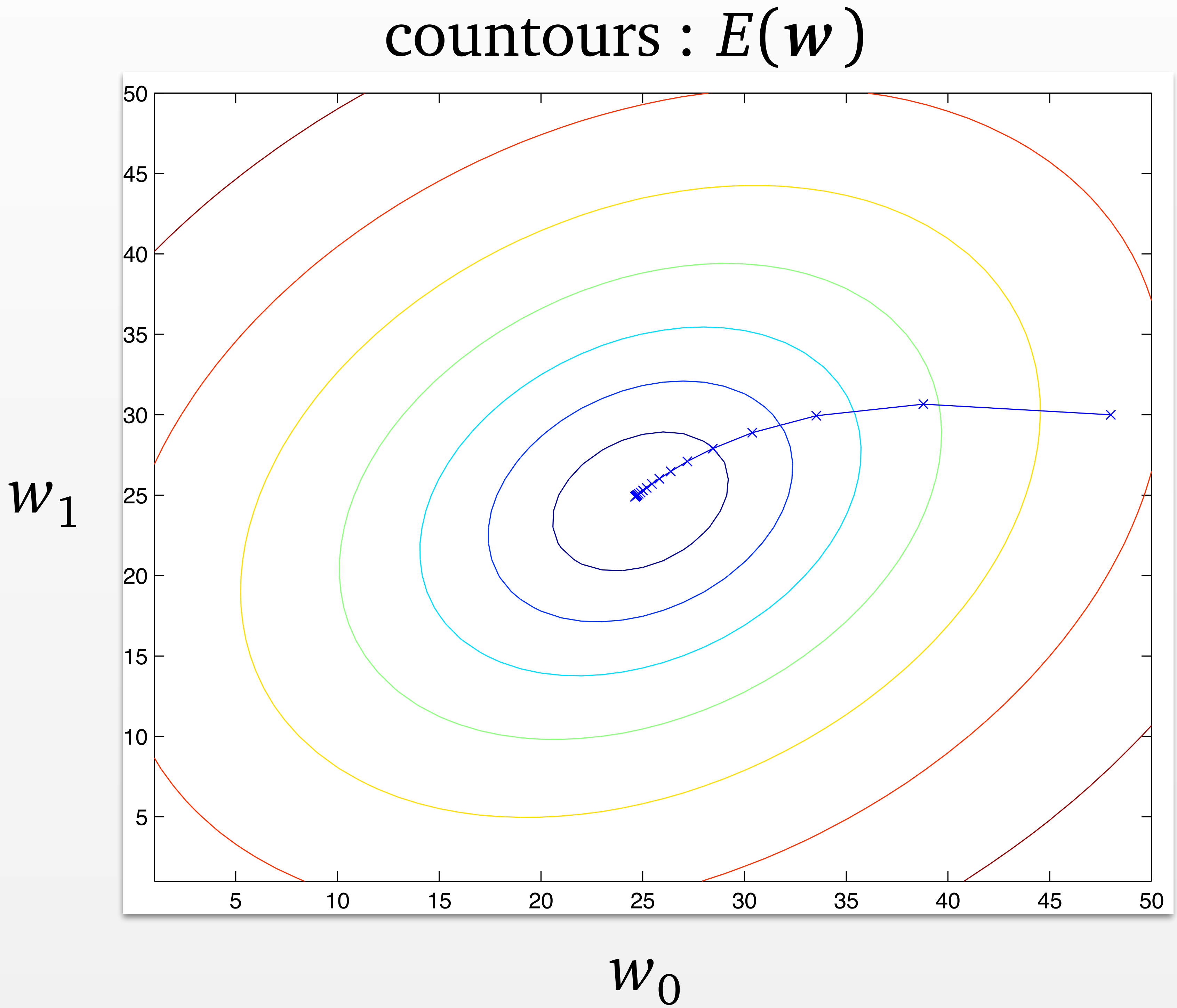
Ordinary Least Squares

- Construct matrix $\mathbf{X}$ and the vector $\mathbf{y}$ from the dataset $\{(\mathbf{x}_1, y_1), \mathbf{x_2}, y_2), \ldots, (\mathbf{x_N}, y_N)\}$ (each $\mathbf{x}$ includes $x_0 = 1$) as follows:

$$\mathbf{X} = \begin{bmatrix} — \mathbf{x_1^T} — \\ — \mathbf{x_2^T} — \\ \ldots \\ — \mathbf{x_N^T} — \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1^T \\ y_2^T \\ \ldots \\ y_N^T \end{bmatrix}$$

- Compute $\mathbf{X}^\dagger = (\mathbf{X^T X})^{-1}\mathbf{X^T}$
- Return $\mathbf{w} = \mathbf{X}^\dagger \mathbf{y}$

Gradient Descent



Least Mean Squares

(a.k.a. gradient descent)

- Initialize the $\mathbf{w}(0)$ for time $t = 0$
- for $t = 0, 1, 2, \ldots$ do
- Compute the gradient $\mathbf{g}_t = \triangledown E(\mathbf{w}(t))$
- Set the direction to move, $\mathbf{v}_t = -\mathbf{g}_t$
- Update $\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \mathbf{v}_t$
- Iterate until it is time to stop
- Return the final weights $\mathbf{w}$

Question

When would you want
to use OLS, when LMS?

Computational Complexity

Ordinary least squares (OMS)

$$\boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}(\boldsymbol{X}^\top \boldsymbol{y})$$

$(\boldsymbol{X}^\top \boldsymbol{y})$	$O(DN)$
$(\boldsymbol{X}^\top \boldsymbol{X})$	$O(D^2 N)$
$(\boldsymbol{X}^\top \boldsymbol{X})^{-1}$	$O(D^3)$

Least Mean Squares (LMS)

$$\nabla E(\boldsymbol{w}) = \frac{2}{N}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

$\boldsymbol{X}\boldsymbol{w}$	$O(DN)$
$\boldsymbol{X}(\boldsymbol{w} - \boldsymbol{y})$	$O(N)$
$\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$	$O(DN)$

Computational Complexity

Ordinary least squares (OMS)

$$\boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}(\boldsymbol{X}^\top \boldsymbol{y})$$

$(\boldsymbol{X}^\top \boldsymbol{y})$	$O(DN)$
$(\boldsymbol{X}^\top \boldsymbol{X})$	$O(D^2 N)$
$(\boldsymbol{X}^\top \boldsymbol{X})^{-1}$	$O(D^3)$

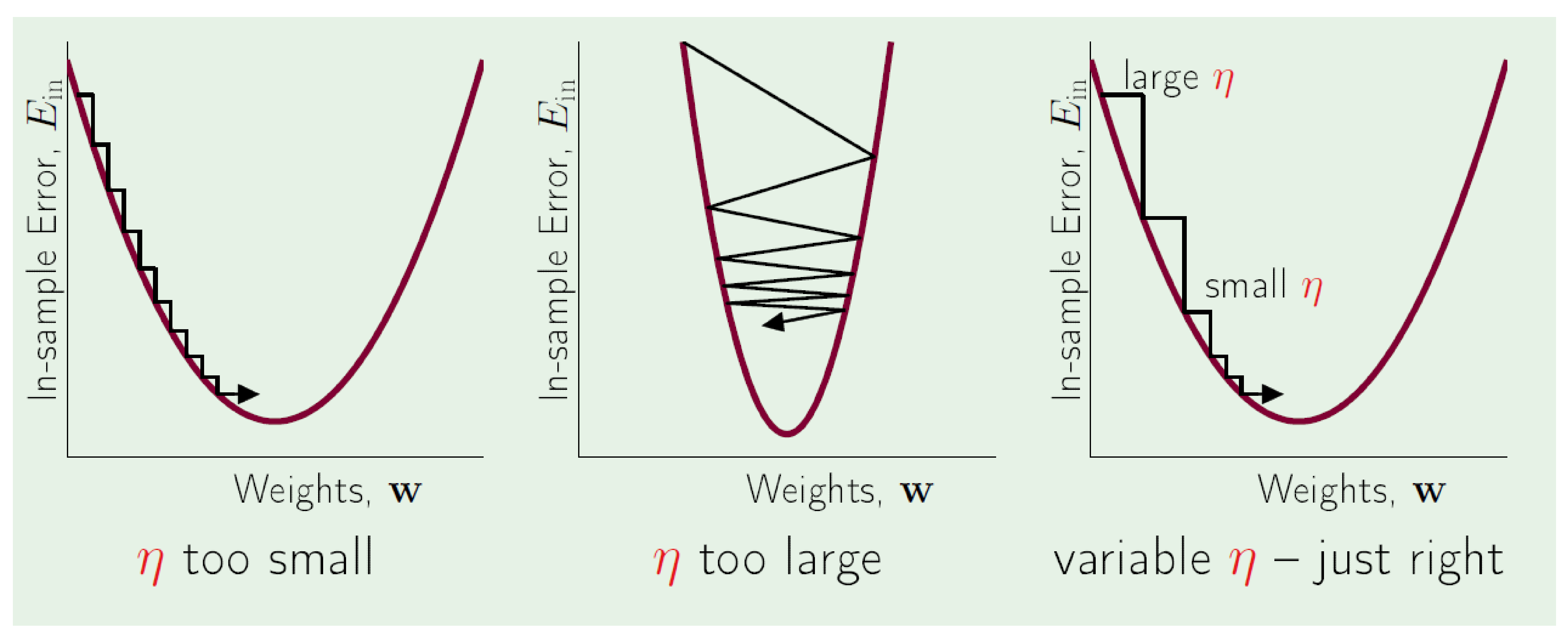
Least Mean Squares (LMS)

$$\nabla E(\boldsymbol{w}) = \frac{2}{N}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

$\boldsymbol{X}\boldsymbol{w}$	$O(DN)$
$\boldsymbol{X}(\boldsymbol{w} - \boldsymbol{y})$	$O(N)$
$\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$	$O(DN)$

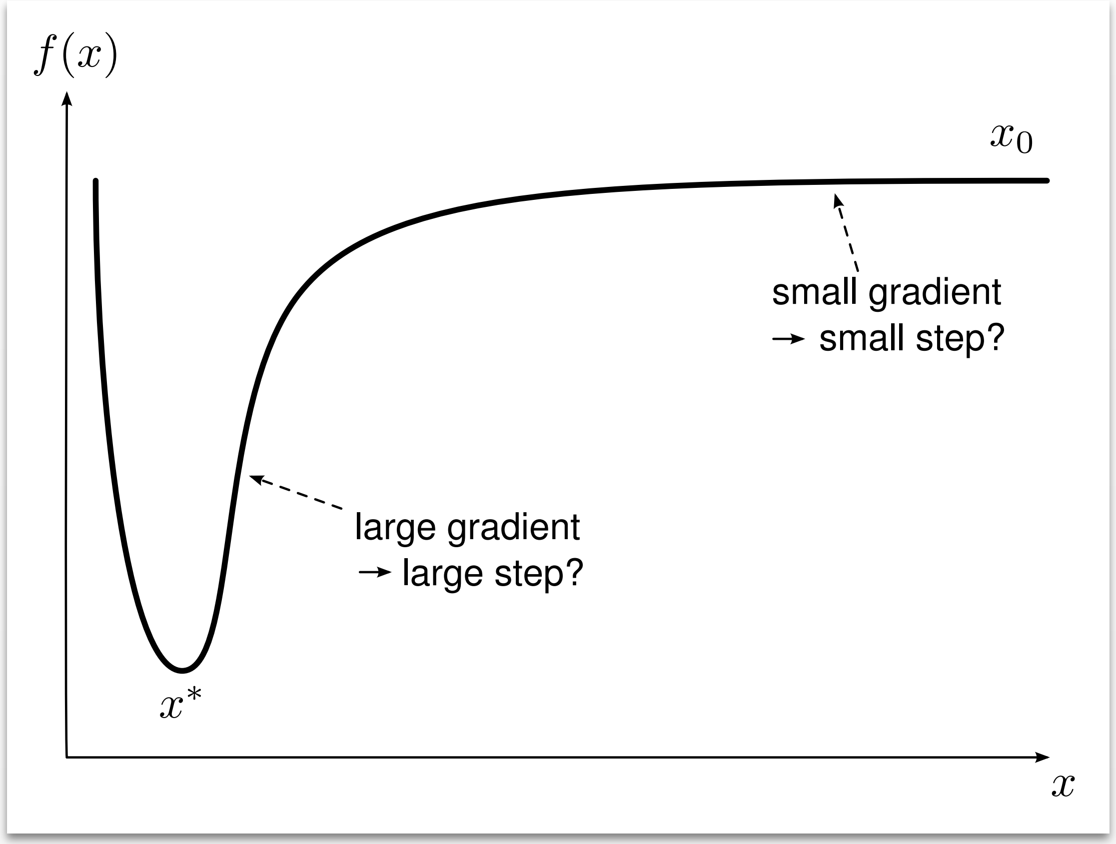
OMS is expensive when D is large

Effect of step size



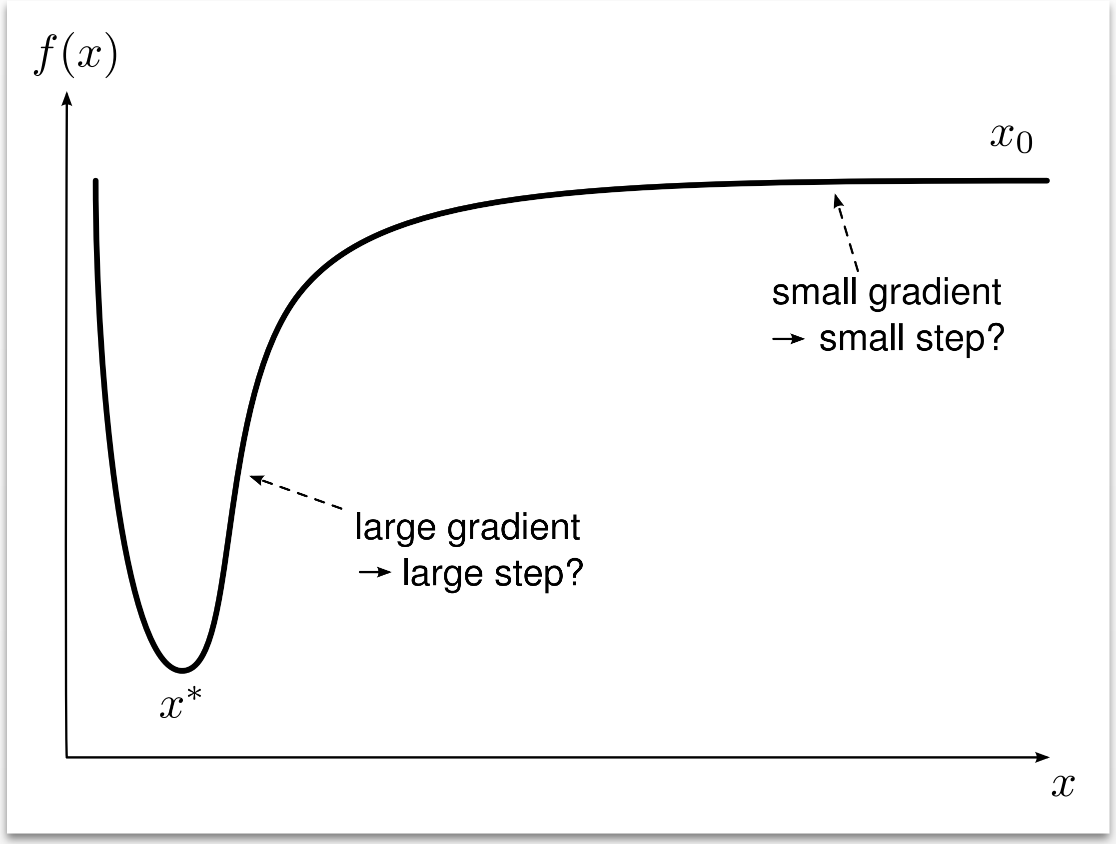
Choosing Stepsize

Set step size proportional to $\nabla f(x)$?



Choosing Stepsize

Set step size proportional to $\nabla f(x)$?



Two commonly used techniques

1. Stepsize adaptation
2. Line search

Stepsize Adaptation

Input: initial $x \in \mathbb{R}^n$, functions $f(x)$ and $\nabla f(x)$, initial stepsize α, tolerance θ

Output: x

repeat
  $y \leftarrow x - \alpha \frac{\nabla f(x)}{|\nabla f(x)|}$
  if [**then**step is accepted] $f(y) \leq f(x)$
    $x \leftarrow y$
    $\alpha \leftarrow 1.2\alpha$ *// increase stepsize*
  else[step is rejected]
    $\alpha \leftarrow 0.5\alpha$ *// decrease stepsize*
  end if
until $|y - x| < \theta$ [perhaps for 10 iterations in sequence]

("magic numbers")

Second Order Methods

Compute ***Hessian*** matrix of second derivatives

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} \\[2.2ex] \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_n} \\[2.2ex] \vdots & \vdots & \ddots & \vdots \\[2.2ex] \dfrac{\partial^2 f}{\partial x_n\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Second Order Methods

Broyden-Fletcher-Goldfarb-Shanno (BFGS) method:

Input: initial $x \in \mathbb{R}^n$, functions $f(x)$, $\nabla f(x)$, tolerance θ
Output: x

initialize $H^{-1} = \mathbf{I}_n$
repeat
    compute $\Delta = -H^{-1} \nabla f(x)$
    perform a line search $\min_\alpha f(x + \alpha\Delta)$
    $\Delta \leftarrow \alpha\Delta$
    $y \leftarrow \nabla f(x + \Delta) - \nabla f(x)$
    $x \leftarrow x + \Delta$
    update $H^{-1} \leftarrow \left(\mathbf{I} - \frac{y\Delta^\top}{\Delta^\top y}\right)^\top H^{-1} \left(\mathbf{I} - \frac{y\Delta^\top}{\Delta^\top y}\right) + \frac{\Delta\Delta^\top}{\Delta^\top y}$
until $\|\Delta\|_\infty < \theta$

Memory-limited version: L-BFGS

Stochastic Gradient Descent

What if *N* is really large?

Batch gradient descent (evaluates all data)

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \alpha_t \nabla_{\boldsymbol{w}} E(\boldsymbol{y}; \boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}_{t-1}}$$

Minibatch gradient descent (evaluates subset)

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \alpha_t \nabla_{\boldsymbol{w}} E(\boldsymbol{y}_t; \boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}_{t-1}} \qquad \boldsymbol{y}_t \subset \boldsymbol{y}$$

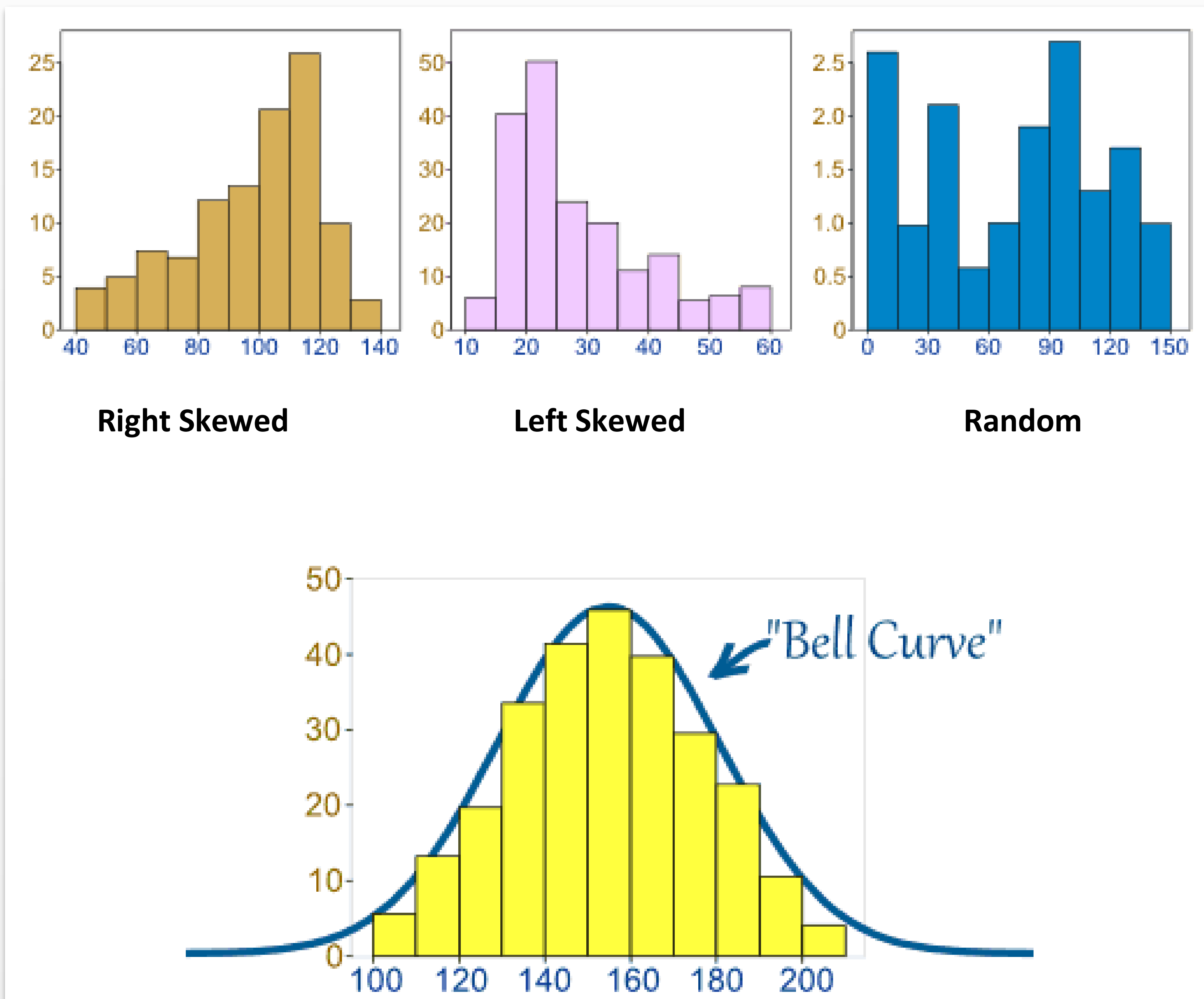
Converges under Robbins-Monro conditions

$$\sum_{t=1}^{\infty} \alpha_t = \infty \qquad \sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

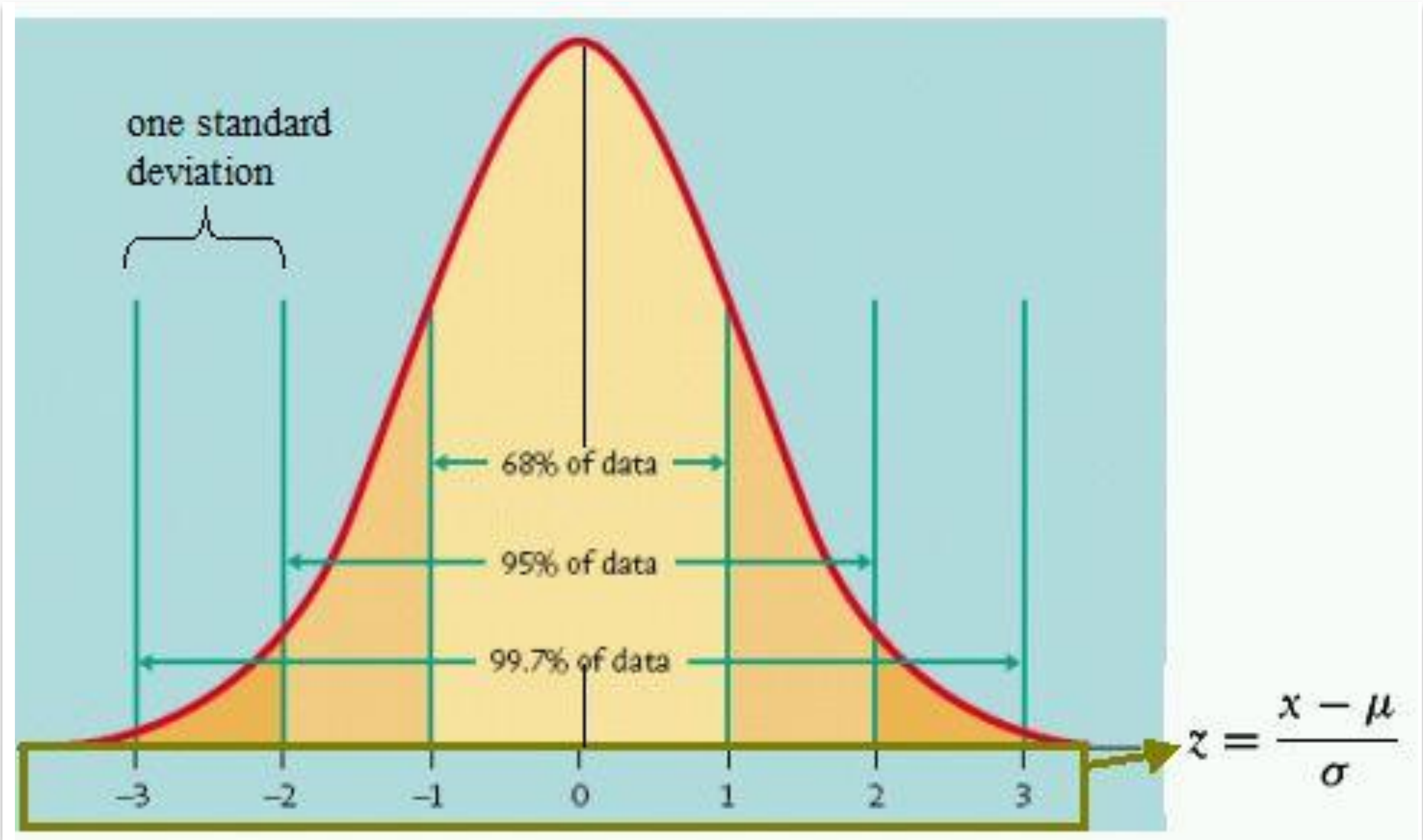
$$\alpha_t = \frac{\alpha_0}{(\tau+t)^{\kappa}}$$

Probabilistic Interpretation

Normal Distribution

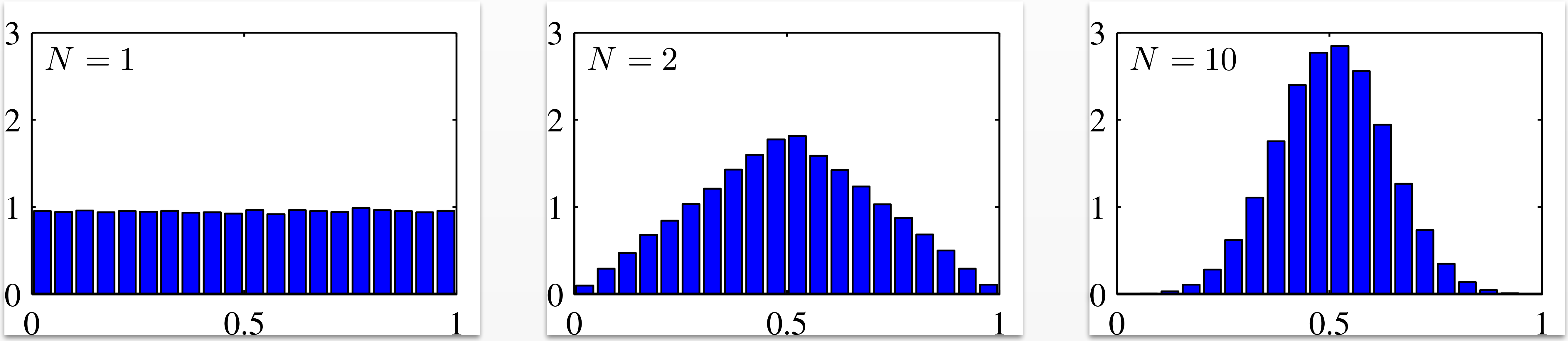


Normal Distribution



Density: $f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{1}{2}(x-\mu)^2/\sigma^2}$

Central Limit Theorem

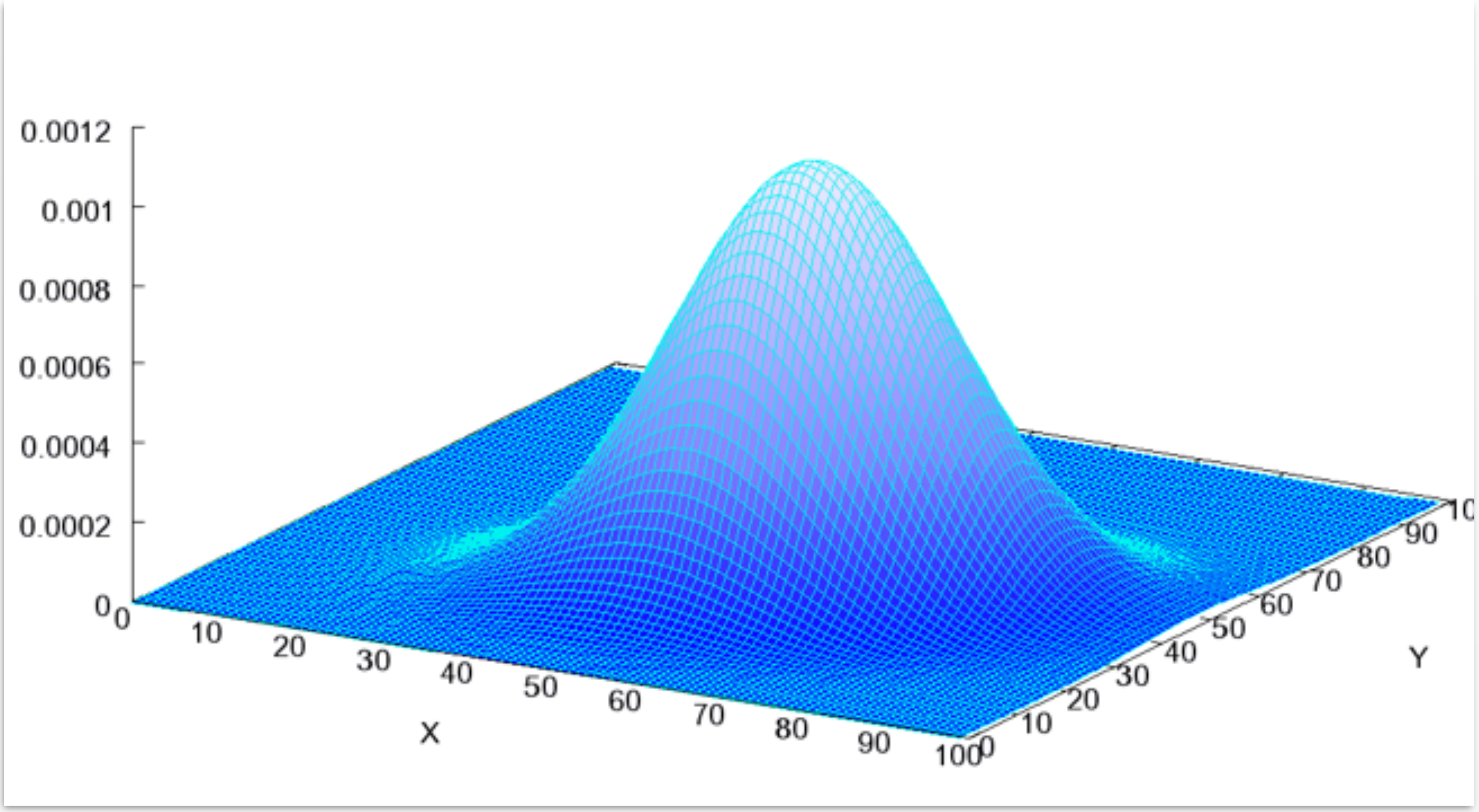


If $y_1, \ldots, y_n$ are

1. Independent identically distributed (i.i.d.)
2. Have finite variance $0 < \sigma_y^2 < \infty$

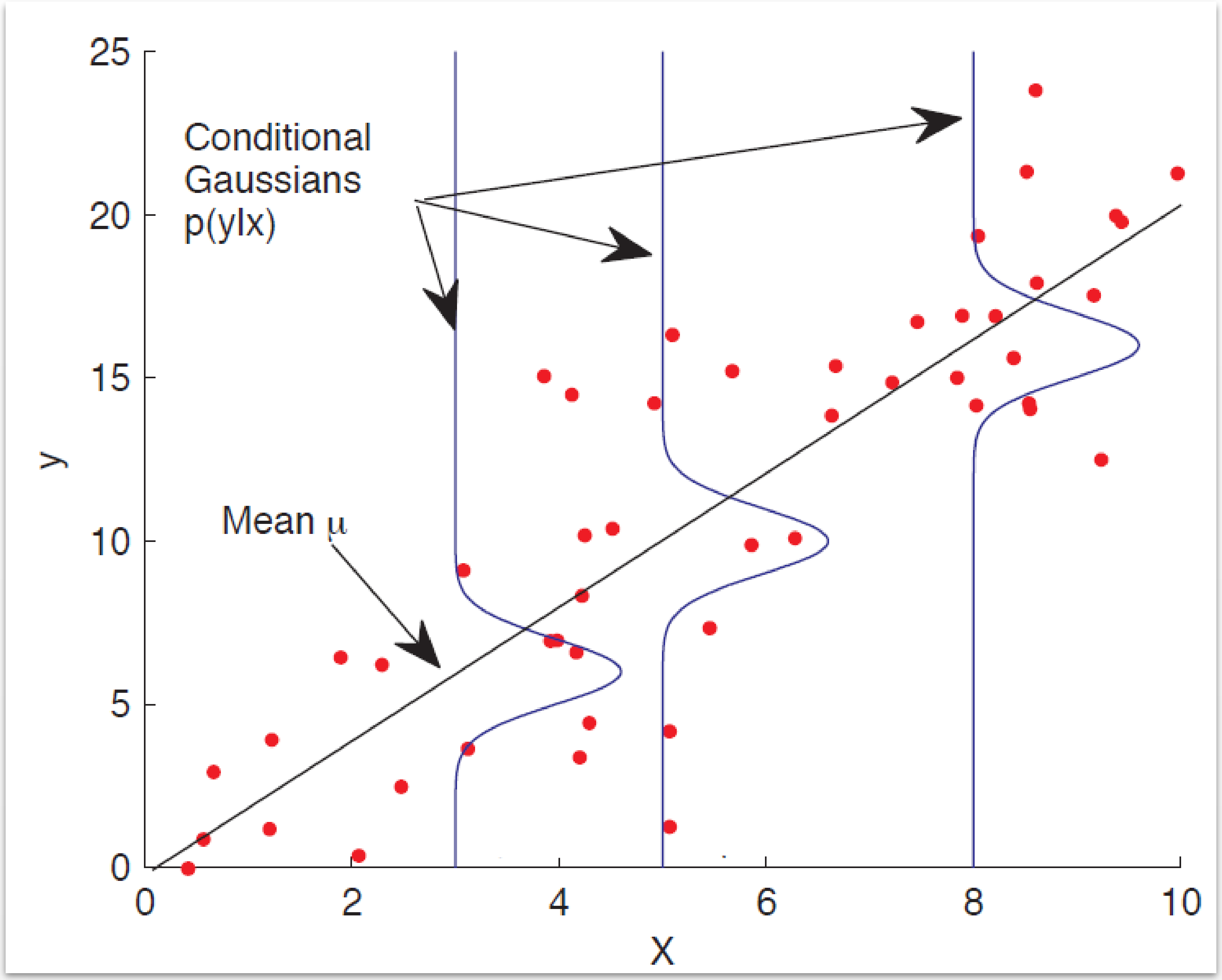
$$f(\bar{y}) = \text{Normal}(\bar{y}\,;\, \mu_y, \sigma_y^2/N) \qquad \bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n$$

Multivariate Normal



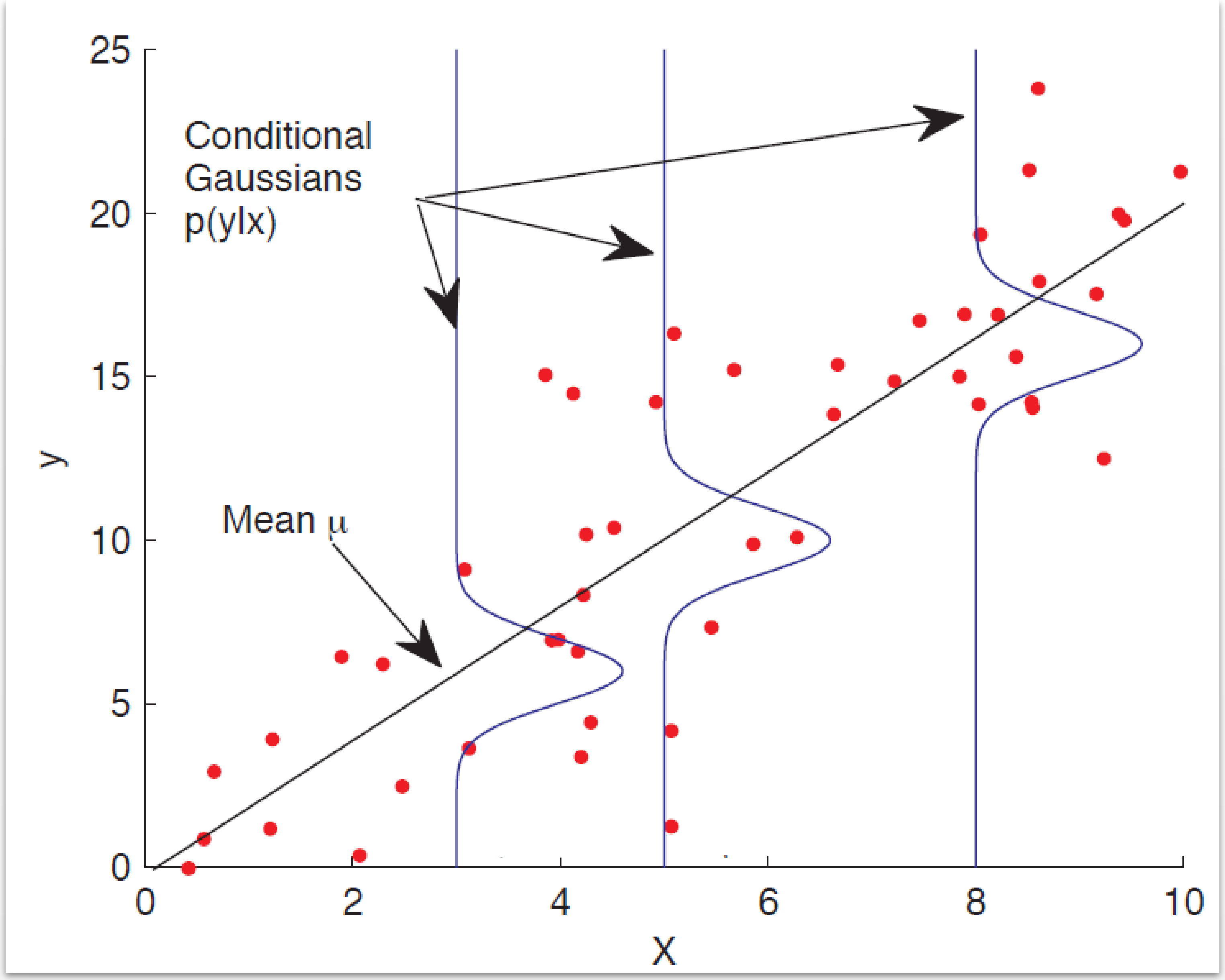
Density: $f(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} \exp^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$

Regression: Probabilistic Interpretation



$$y_n = a x_n + b + \sigma \epsilon_n \qquad \epsilon \sim \text{Normal}(0, 1)$$

Regression: Probabilistic Interpretation



$$\mu_n = \boldsymbol{w}^\top \boldsymbol{x}_n \qquad y_n \sim \text{Normal}(\mu_n, \Sigma)$$

Regression: Probabilistic Interpretation

Joint probability of *N* independent data points

$$p(y_1,\ldots,y_N) = \prod_{n=1}^{N} p(y_n)$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}^N} \prod_{n=1}^{N} \exp^{-\frac{1}{2}(x-\mu)^2/\sigma^2}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}^N} \exp^{-\frac{1}{2}\sum_{n=1}^{N}(x-\mu)^2/\sigma^2}$$

Regression: Probabilistic Interpretation

Log joint probability of *N* independent data points

$$\log p(y_1, \ldots, y_N) = \sum_{n=1}^{N} \log p(y_n)$$

$$= -\frac{1}{2}\left[N \log(2\pi\sigma^2) + \sum_{n=1}^{N} \frac{(y_n - \mu_n)^2}{\sigma^2}\right]$$

Regression: Probabilistic Interpretation

Log joint probability of N independent data points

$$\log p(y_1, \ldots, y_N) = \sum_{n=1}^{N} \log p(y_n)$$

$$= -\frac{1}{2}\left[N \log(2\pi\sigma^2) + \sum_{n=1}^{N} \frac{(y_n - \boldsymbol{w}^\top x_n)^2}{\sigma^2}\right]$$

Regression: Probabilistic Interpretation

Log joint probability of *N* independent data points

$$
\begin{aligned}
\log p(y_1, \ldots, y_N) &= \sum_{n=1}^{N} \log p(y_n) \\
&= -\frac{1}{2}\left[N \log(2\pi\sigma^2) + \sum_{n=1}^{N} \frac{(y_n - \boldsymbol{w}^\top x_n)^2}{\sigma^2} \right] \\
&= -\frac{N}{2}\left[\text{const} + E(\boldsymbol{w})\right]
\end{aligned}
$$

Regression: Probabilistic Interpretation

Log joint probability of *N* independent data points

$$\log p(y_1, \ldots, y_N) = \sum_{n=1}^{N} \log p(y_n)$$

$$= -\frac{1}{2}\left[N \log(2\pi\sigma^2) + \sum_{n=1}^{N} \frac{(y_n - \boldsymbol{w}^\top x_n)^2}{\sigma^2}\right]$$

$$= -\frac{N}{2}\left[\text{const} + E(\boldsymbol{w})\right]$$

$$\underset{\boldsymbol{w}}{\operatorname{argmax}}\; p(y_1, \ldots, y_N; \boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, E(\boldsymbol{w})$$

Maximum Likelihood

Basis function regression

Linear regression

$$y = w_0 + w_1 \boldsymbol{x}_1 + \ldots + w_D \boldsymbol{x}_D = \boldsymbol{w}^T \boldsymbol{x}$$

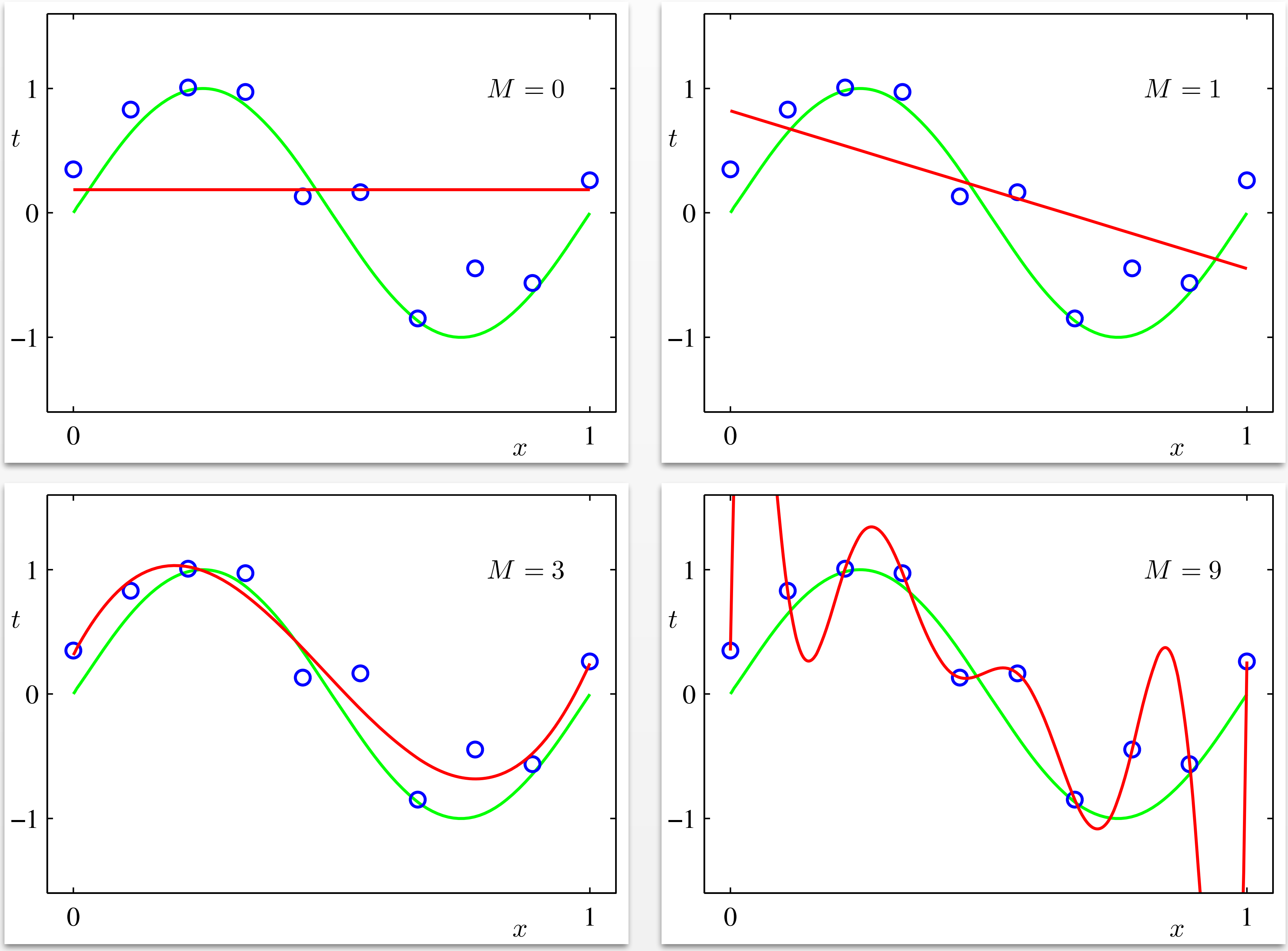
Basis function regression

$$y = w_0 + w_1 \phi_1(\boldsymbol{x}) + \ldots + w_D \phi_D(\boldsymbol{x})$$

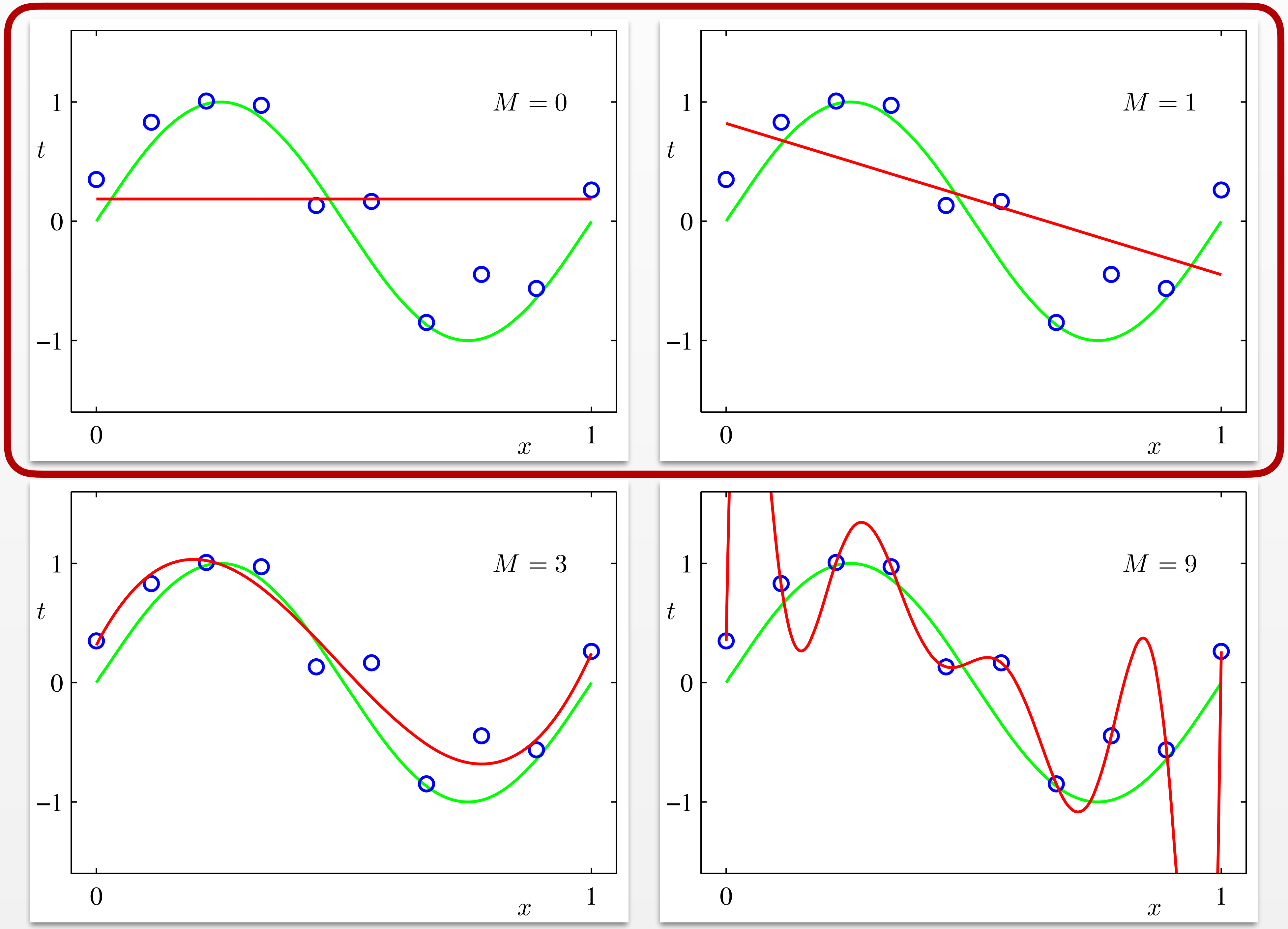
Polynomial regression

$$\boldsymbol{x}_d := \phi_d(\boldsymbol{x}) \qquad \phi_d(\boldsymbol{x}) := \boldsymbol{x}^d$$

Polynomial Regression

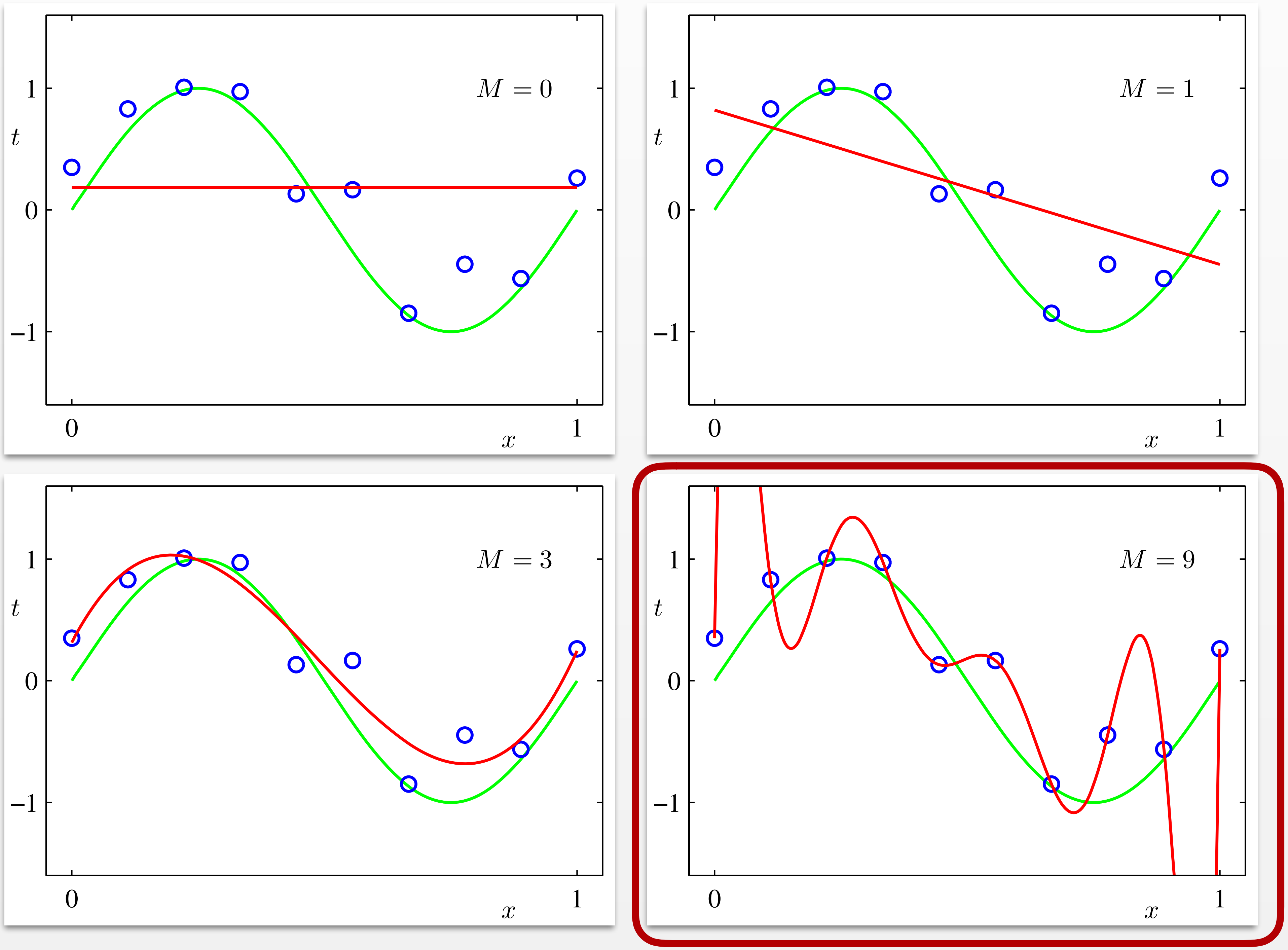


Polynomial Regression



Underfit

Polynomial Regression



Overfit

Regularization

$L2$ regularization (ridge regression) minimizes:

$$E(\mathbf{w}) = \quad \| \mathbf{X}\mathbf{w} - \mathbf{y} \|^2 + \lambda \| \mathbf{w} \|^2$$

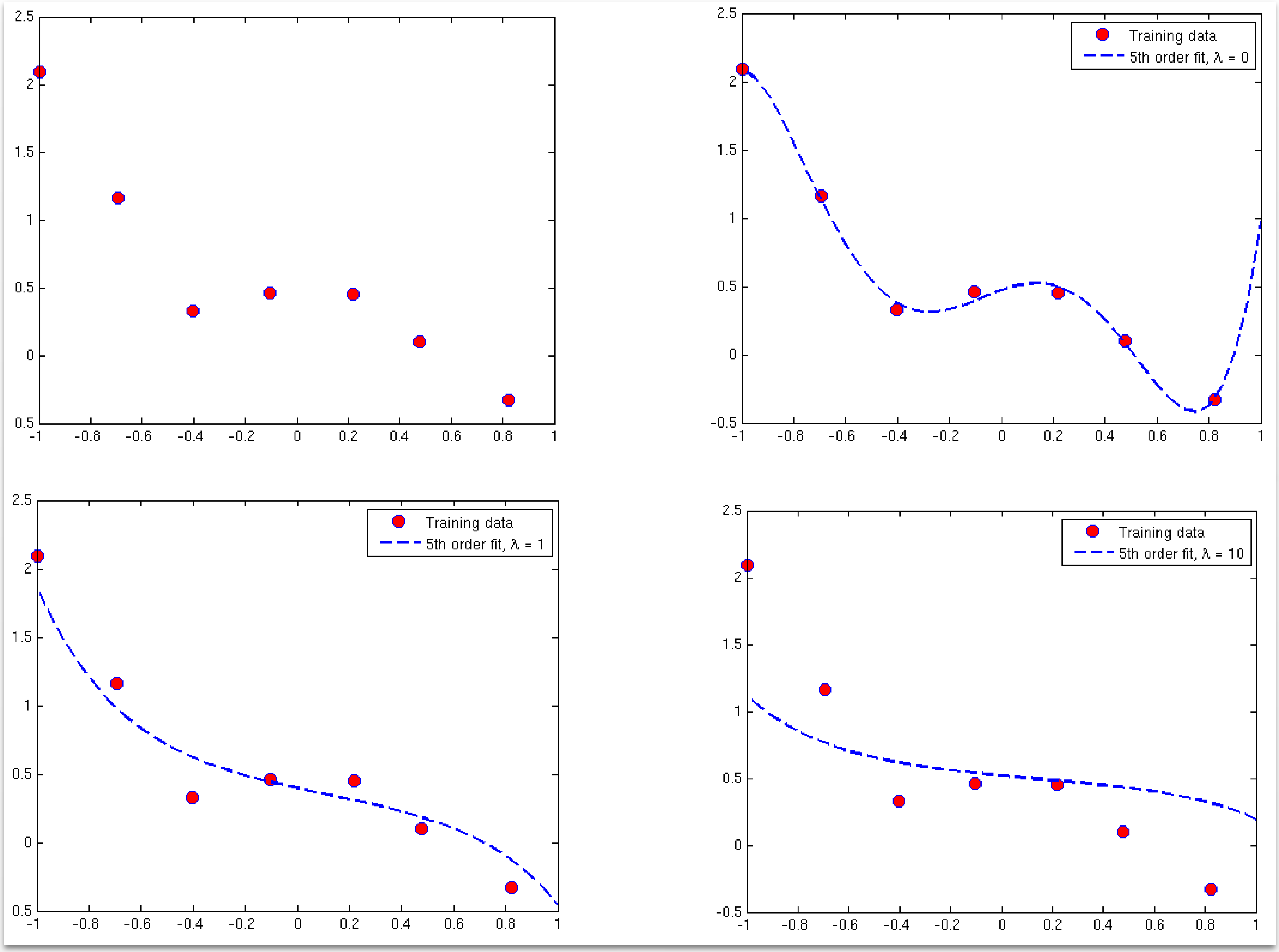
where $\lambda \geq 0$ and $\| \mathbf{w} \|^2 = \mathbf{w}^\mathbf{T}\mathbf{w}$

$L1$ regularization (LASSO) minimizes:

$$E(\mathbf{w}) = \quad \| \mathbf{X}\mathbf{w} - \mathbf{y} \|^2 + \lambda |\mathbf{w}|_1$$

where $\lambda \geq 0$ and $|\mathbf{w}|_1 = \sum_{i=1}^{D} |\omega_i|$

Regularization



Regularization

$L2$: closed form solution

$$\mathbf{w} = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$$

$L1$: No closed form solution. Use quadratic programming:

$$\text{minimize} \parallel \mathbf{X}\mathbf{w} - \mathbf{y} \parallel^2 \quad s.t. \quad \parallel \mathbf{w} \parallel_1 \leq s$$

Review: Bias-Variance Trade-off

Maximum likelihood estimator

$$\hat{f} := \operatorname*{argmax}_{\tilde{f}} p(\boldsymbol{y} \mid \tilde{f})$$

Bias-variance decomposition
(*expected value over <u>possible data points</u>*)

$$\mathbb{E}[(y - \hat{f}(\boldsymbol{x}))^2] = \text{Bias}[\hat{f}(\boldsymbol{x})]^2 + \text{Var}[\hat{f}(\boldsymbol{x})] + \sigma^2$$

$$\text{Bias}[\hat{f}(\boldsymbol{x})] = \mathbb{E}[\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})]$$

$$\text{Var}[\hat{f}(\boldsymbol{x})] = \mathbb{E}[\hat{f}(\boldsymbol{x})^2] - \mathbb{E}[\hat{f}(\boldsymbol{x})]^2$$

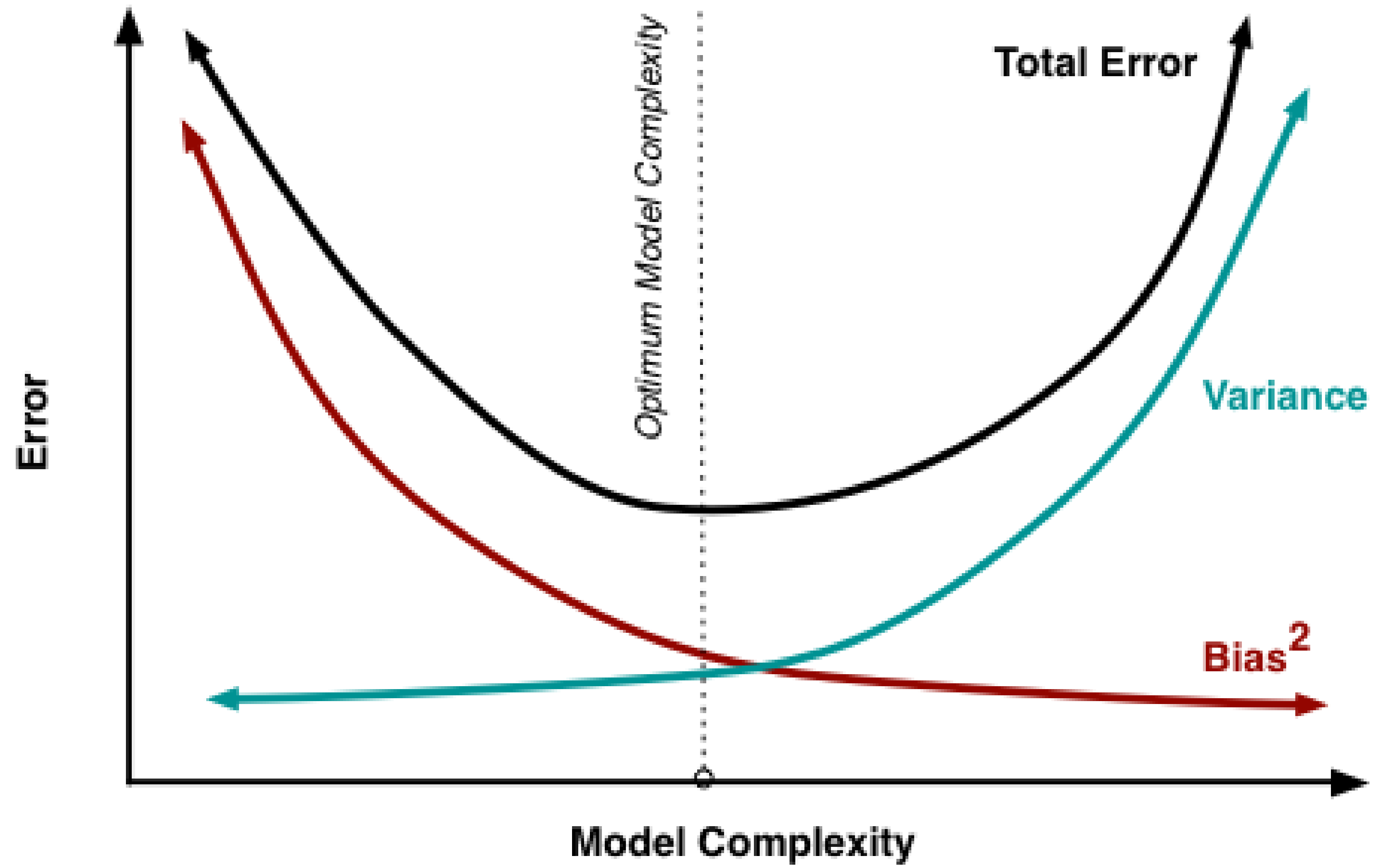
$$\sigma^2 = \mathbb{E}[y^2] - \mathbb{E}[f(\boldsymbol{x})]^2$$

Bias-Variance Trade-off

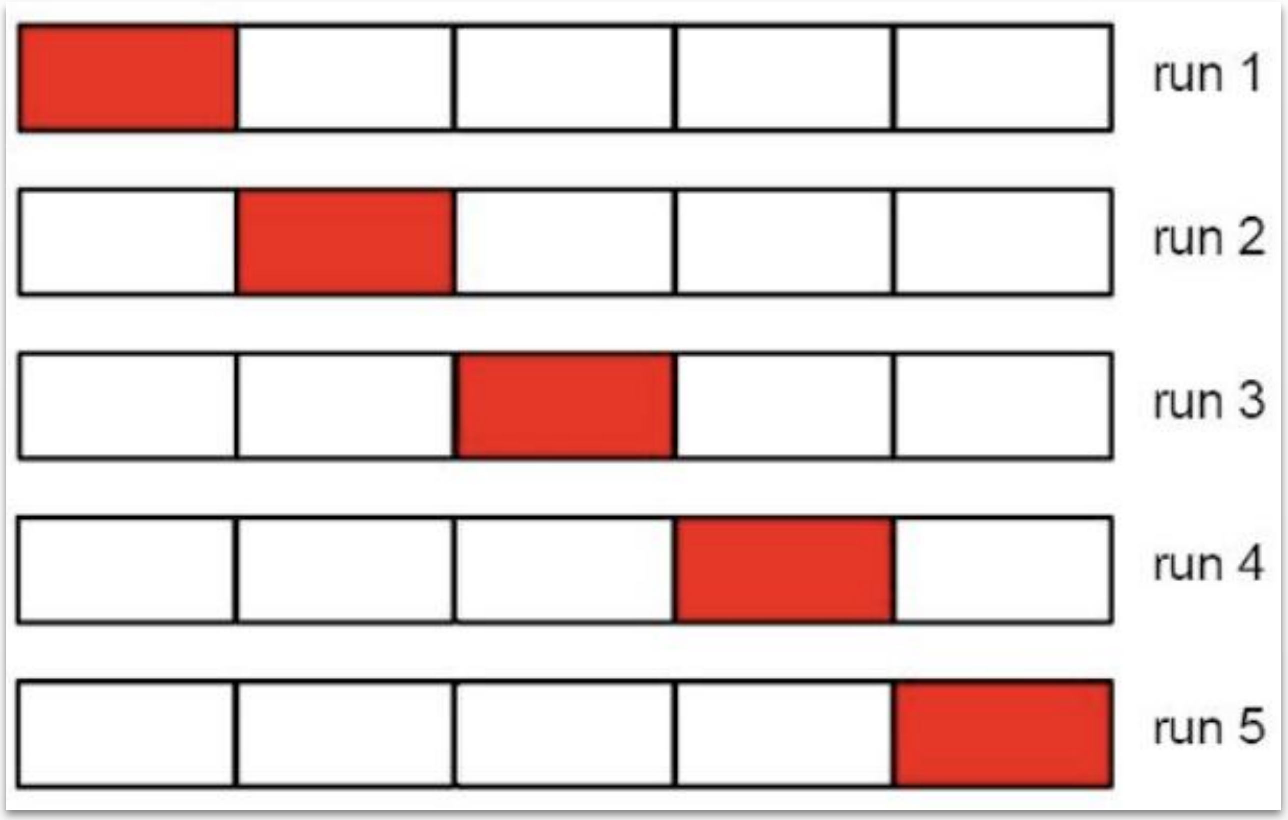
Often: low bias $\Rightarrow$ high variance

low variance $\Rightarrow$ high bias

Trade-off:

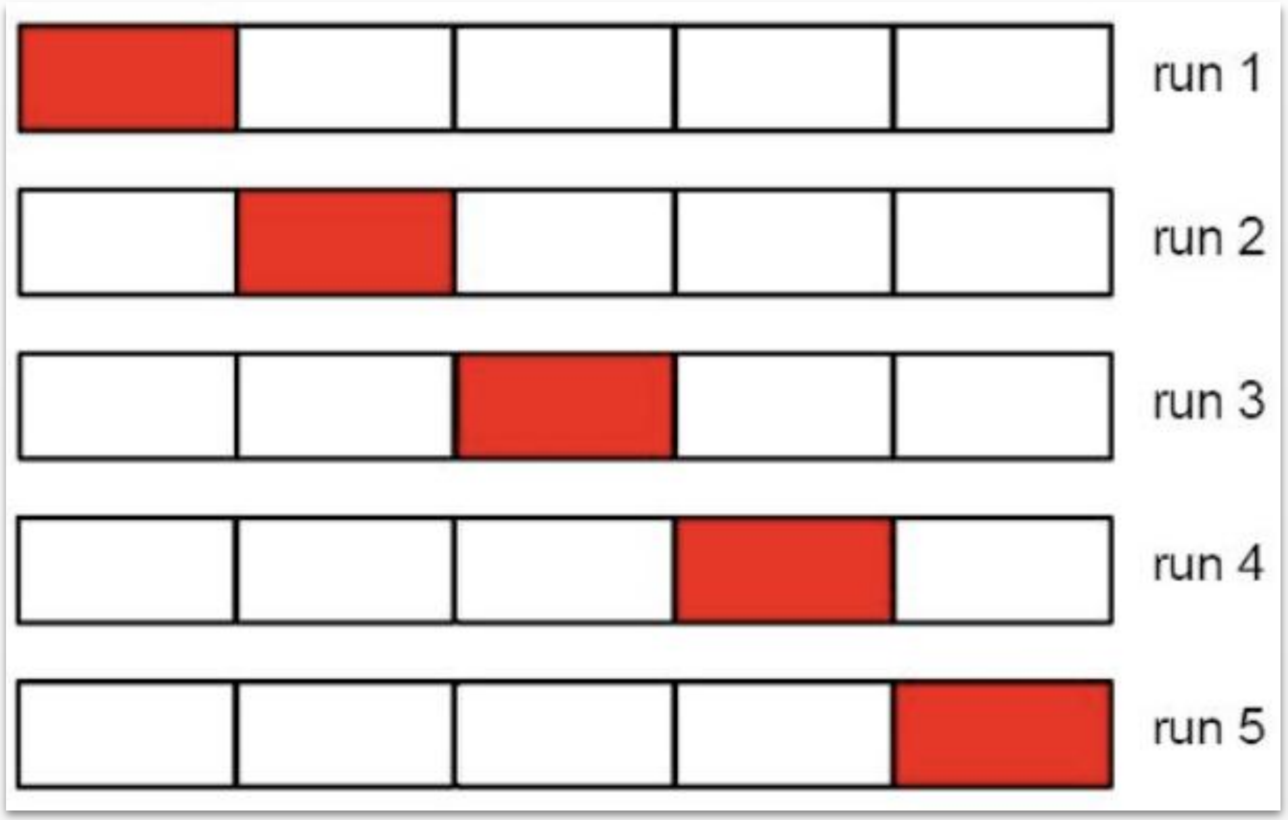


K-fold Cross-Validation



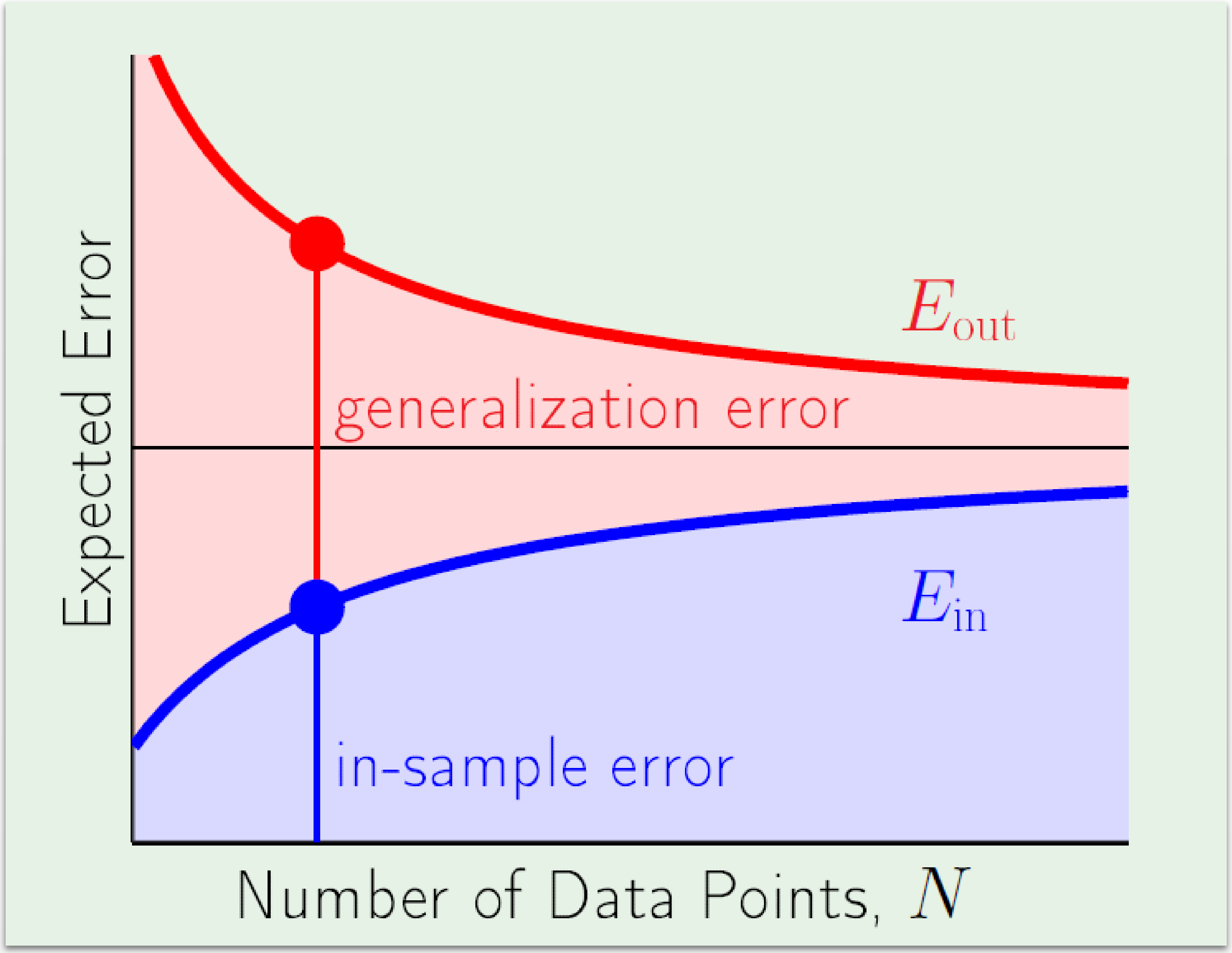
1. ***Divide*** dataset into K "folds"
2. ***Train*** on all except *k*-th fold
3. ***Test*** on *k*-th fold
4. ***Minimize*** test error w.r.t. λ

K-fold Cross-Validation

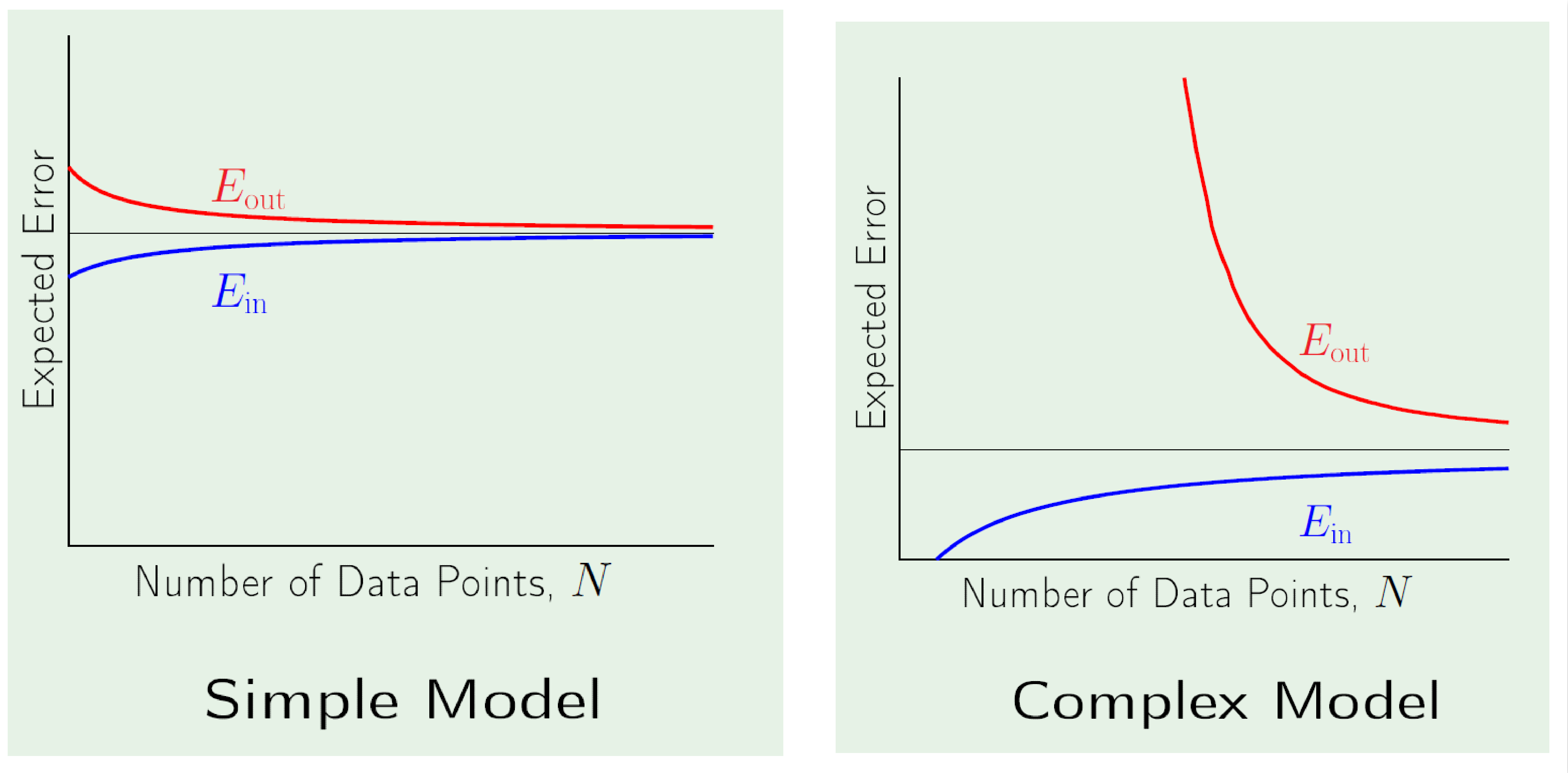


- *Choices for K*: 5, 10, *N* (leave-one-out)
- Cost of computation: *K* x number of λ

Learning Curve



Learning Curve



Loss Functions

squared loss:	$\frac{1}{2}(\boldsymbol{w}^\top \boldsymbol{x} - y)^2$	$y \in \mathbb{R}$
logistic loss:	$\log\left(1 + \exp(-y\boldsymbol{w}^\top \boldsymbol{x})\right)$	$y \in \{-1, +1\}$
hinge loss:	$\max\{0, 1 - y\boldsymbol{w}^\top \boldsymbol{x}\}$	$y \in \{-1, +1\}$