2025春

智能信号处理导论

实验I：线性回归

Introduction to Intelligent Signal Processing

Part II: Fundamentals of Machine Learning

李 杨

电子工程系，新技术楼705

86418051-7051，li.yang@hit.edu.cn

http://homepage.hit.edu.cn/liyang

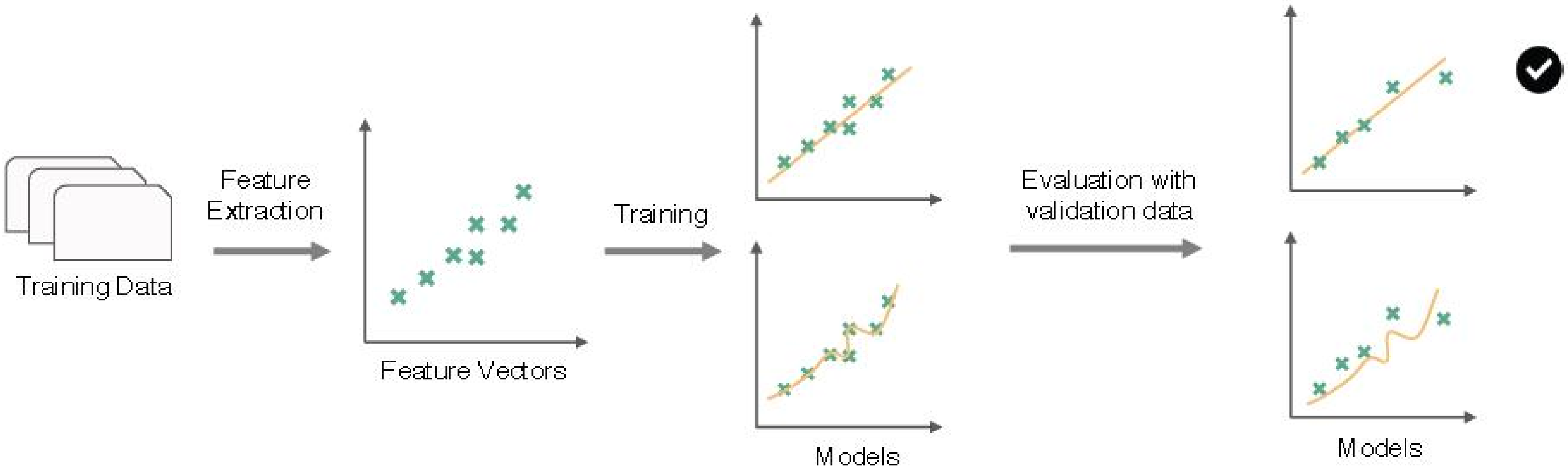
2.2.4 Example: Linear Regression

- **线性回归**：我们的目标是建立一个系统，将向量$x \in \mathbf{R}^n$作为输入，预测标量$y \in \mathbf{R}$作为输出。线性回归的输出是其输入的线性函数。令$\hat{y}$表示模型预测y应该取的值。我们定义输出为

$$\hat{y} = \omega^T x$$

$\omega \in \mathbf{R}^n$：参数向量（决定每个预测如何影响预测的权重）

- **定义任务*T***：通过输出$\hat{y} = \omega^T x$ 从x预测y。 ——Tracking



Performance Measure, *P*

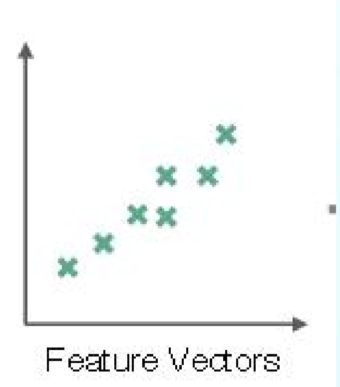
- 测试集（test set）： 不用来训练模型，而是评估模型性能如何
- 将输入的设计矩阵记作 $\mathbf{x}^{(\text{test})}$ 回归目标向量记作 $\mathbf{y}^{(\text{test})}$
- 假设我们有m个输入样本组成的设计矩阵。
- 每个样本对应的正确值y组成的回归目标向量。

样本1：	特征1	特征2	特征3	......	特征K
样本2：	特征1	特征2	特征3	......	特征K
......	......	......	......	......	......
样本M*N：	特征1	特征2	特征3	......	特征K

- 均方误差（mean squared error）：一种度量方法

若$\hat{y}^{(\text{test})}$表示模型在测试集上的预测值，则

$$\mathrm{MSE}_{\text{test}} = \frac{1}{m}\sum_i \left(\hat{\boldsymbol{y}}^{(\text{test})} - \boldsymbol{y}^{(\text{test})}\right)_i^2 .$$



当 $\hat{\boldsymbol{y}}^{(\text{test})} = \boldsymbol{y}^{(\text{test})}$ 时，误差为0.

- 欧氏距离

$$\mathrm{MSE}_{\text{test}} = \frac{1}{m}\left\|\hat{\boldsymbol{y}}^{(\text{test})} - \boldsymbol{y}^{(\text{test})}\right\|_2^2$$

*p*范数 $\|\boldsymbol{x}\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$

补充：赋范线性空间

- 赋范线性空间(normed linear space)是在线性空间中引进一种与代数运算相联系的度量，即由向量范数诱导出的度量。赋范线性空间称为Banach空间，是指由范数导出的度量是完备的。
- 定义：设X是线性空间，如果X中每一个元素x都有一个实数$||x||$与之对应，且满足：

（1）$||x||=0$当且仅当$x=\theta$；

（2）对任何$x\in X$及$\alpha\in C$（实数或复数），$||\alpha x||= \alpha||x||$；

（3）对任意$x,y\in X$，$||x+y||\le ||x||+ ||y||$。

则称X为赋范线性空间，称$||x||$为元素的范数。

p范数 $\|\boldsymbol{x}\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$

可定义元素x和y的距离：$\rho(x,y)=||x-y||$ $(x,y\in X)$

p范数 $\|\boldsymbol{x}\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$

Experience, *E*

$$\mathrm{MSE}_{\text{test}} = \frac{1}{m}\left\|\hat{\boldsymbol{y}}^{(\text{test})} - \boldsymbol{y}^{(\text{test})}\right\|_2^2$$

$$\hat{\boldsymbol{y}} = \boldsymbol{\omega}^T \boldsymbol{x}$$

■ **构建ML算法：**

✓ 设计算法，通过观察训练集获得经验，减少$\mathrm{MSE}_{\text{test}}$，以改进权重ω

✓ 通过最小化训练集上的均方误差$\mathrm{MSE}_{\text{train}}$

✓ 求导

$$\nabla_{\boldsymbol{w}} \mathrm{MSE}_{\text{train}} = 0$$

$$\Rightarrow \nabla_{\boldsymbol{w}} \frac{1}{m}\left\|\hat{\boldsymbol{y}}^{(\text{train})} - \boldsymbol{y}^{(\text{train})}\right\|_2^2 = 0 \quad \because \mathrm{MSE}_{\text{test}} = \frac{1}{m}\left\|\hat{\boldsymbol{y}}^{(\text{test})} - \boldsymbol{y}^{(\text{test})}\right\|_2^2$$

$$\Rightarrow \frac{1}{m}\nabla_{\boldsymbol{w}}\left\|\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right\|_2^2 = 0 \quad \because \hat{\boldsymbol{y}} = \boldsymbol{\omega}^T \boldsymbol{x}$$

$$\Rightarrow \nabla_{\boldsymbol{w}}\left(\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right)^\top\left(\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right) = 0$$

$$\Rightarrow \nabla_{\boldsymbol{w}}\left(\underline{\boldsymbol{w}^\top \boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\boldsymbol{w}} - \underline{2\boldsymbol{w}^\top \boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})}} + \boldsymbol{y}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})}\right) = 0$$

$$\Rightarrow 2\boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - 2\boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})} = 0$$

$$\Rightarrow \boldsymbol{w} = \left(\boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\right)^{-1}\boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})}$$

——正规方程（Normal Equation）

Experience, *E*

p范数 $||\boldsymbol{x}||_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$

$$\Rightarrow \boldsymbol{w} = \left(\boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\right)^{-1}\boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})}$$

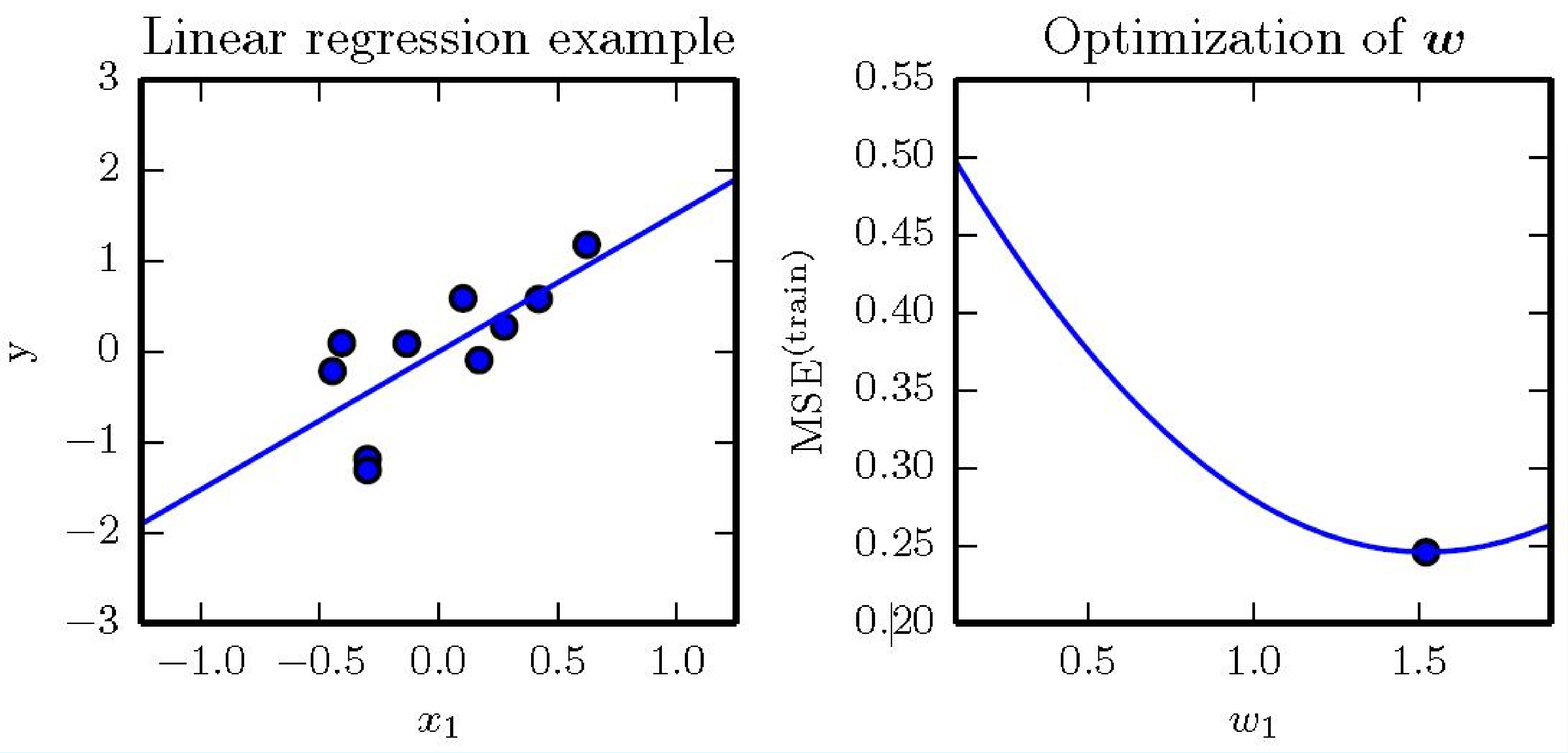
评价
使用测试集

$$\text{MSE}_{\text{test}} = \frac{1}{m}\sum_i (\hat{\boldsymbol{y}}^{(\text{test})} - \boldsymbol{y}^{(\text{test})})_i^2.$$

$$\Rightarrow w = \left(X^{(\text{train})\top} X^{(\text{train})}\right)^{-1} X^{(\text{train})\top} y^{(\text{train})}$$

Example of the Linear Regression Learning Algorithm

- 训练集包括十个数据点，每点包含一个特征，只有一个要学习的参数w_1。
- (左) 观察到线性回归学习w_1，使直线$y=w_1x$尽量接近穿过所有的训练点。
- (右) 标注点表示由学习到的w_1的值，它可以最小化训练集上的均方误差。

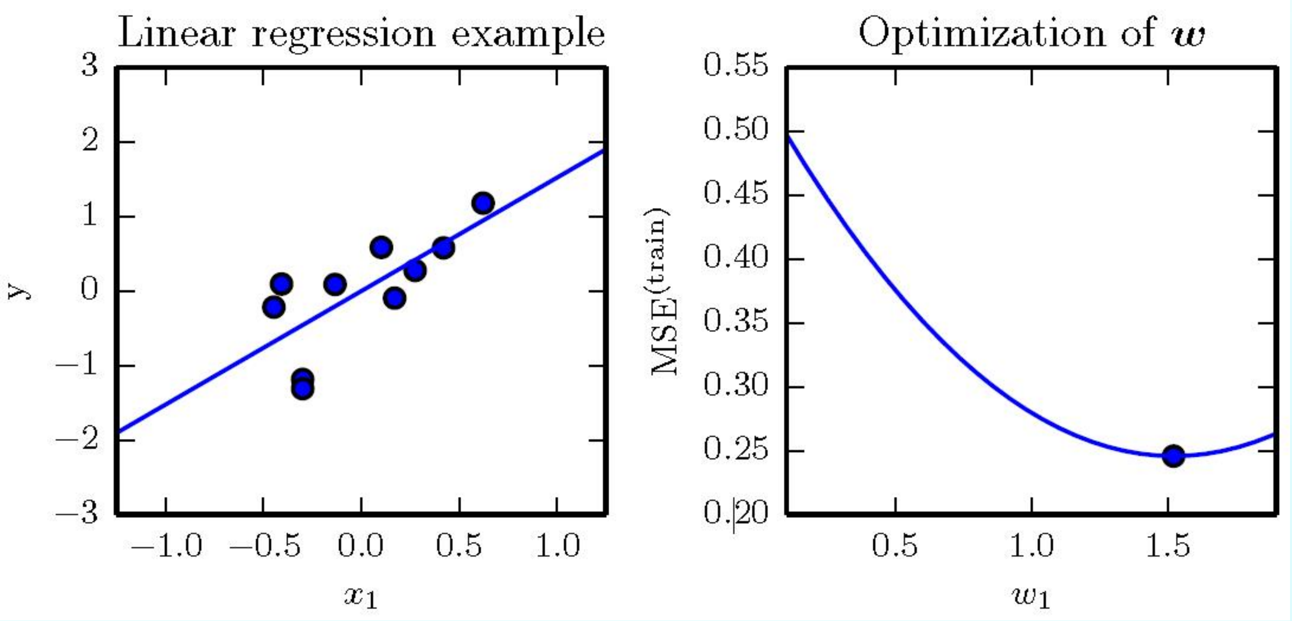


2.2.4 Example: Linear Regression

□ Lab 1: How to Design a Simulation like that?

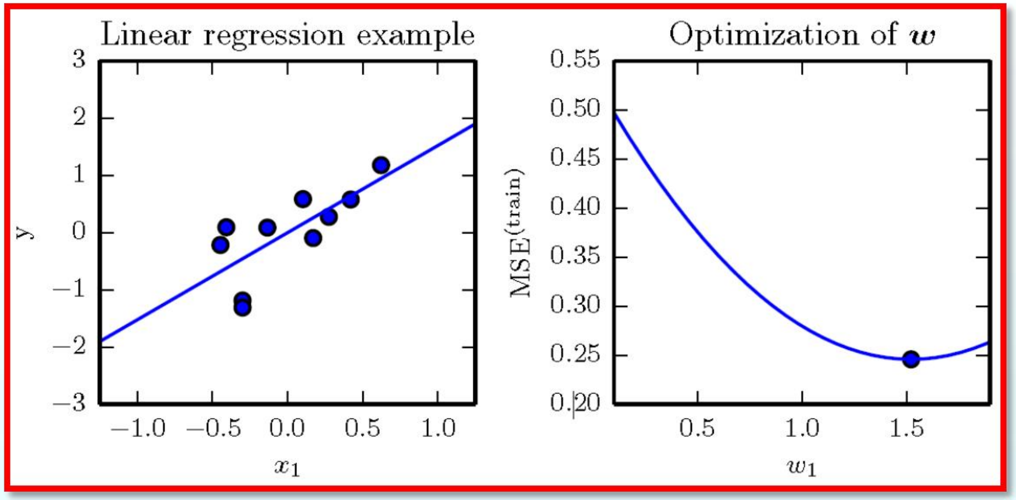
✓ **List all Steps of the Procedure**

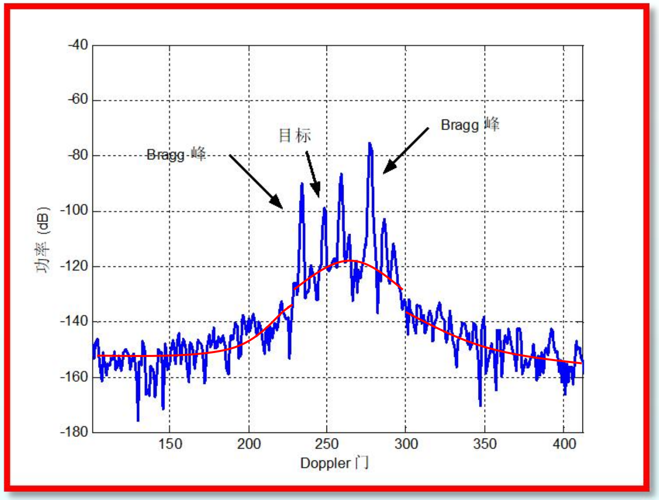
✓ **Plot the Flow Chart**



2.2.4 Example: Linear Regression

➢ Lab 1-1: How to Design a Simulation like that?





- ✓ **Step1：看懂问题是什么**
- ✓ **Step2：弄清数据来源**
- ✓ **Step3：明确处理过程和算法**
- ✓ **Step4：注意评价和分析**

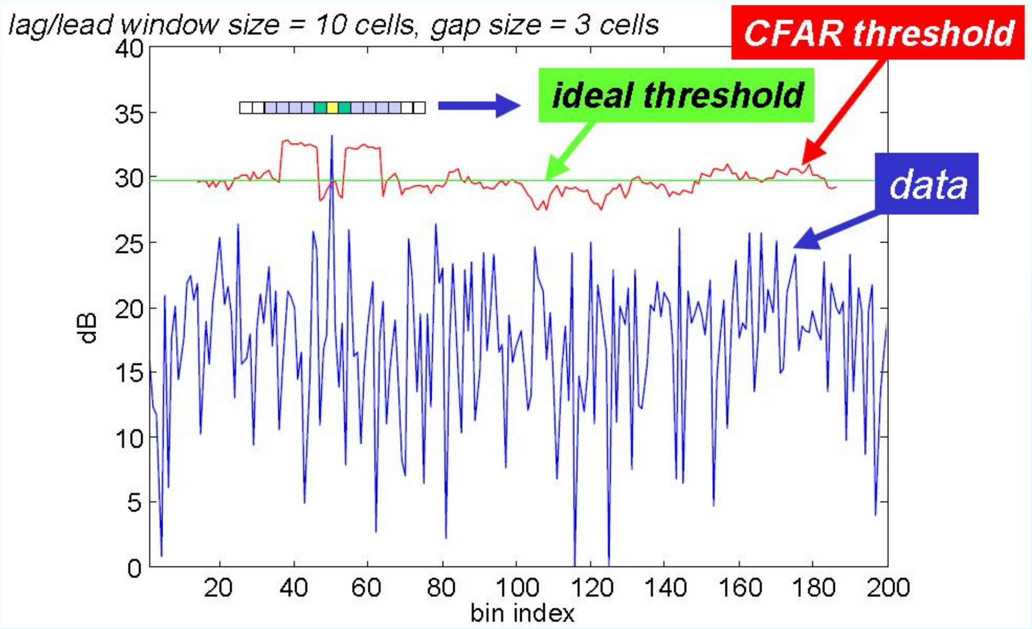
2.2.4 Example: Linear Regression

☐ Discussion

➢ **How to Design a Simulation like that?**

✓ **List all Steps of the Procedure**

✓ **Plot the Flow Chart**



✓Step1：看懂问题是什么

✓Step2：弄清数据来源

✓Step3：明确处理过程和算法

✓Step4：注意评价和分析

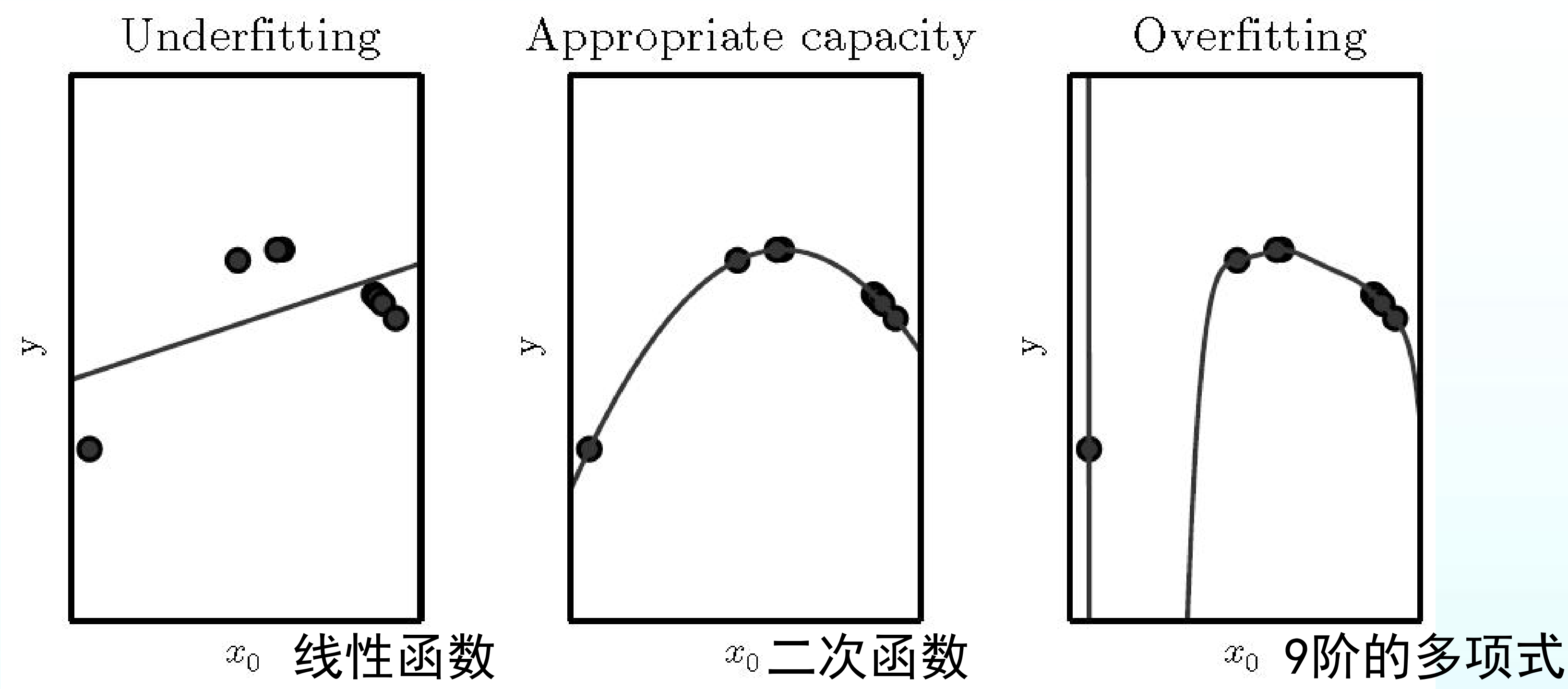
2.2.4 Example: Linear Regression

Lab 1-1

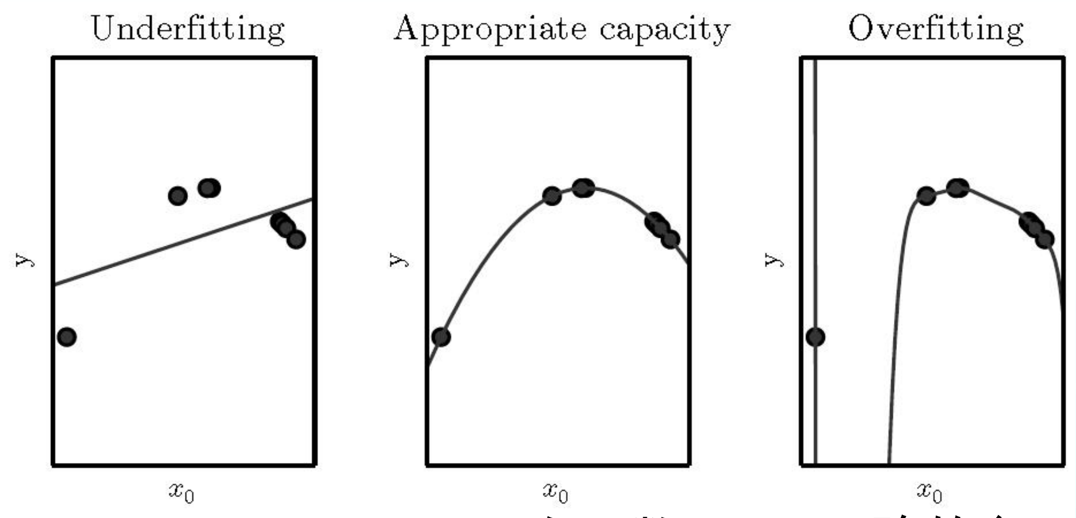
- **Try It （ Linear Regression ）in Lab 1 with Matlab@ on ~**

2.3 The Basic Principles of ML Algorithm Design

2.3.1 容量、过拟合、欠拟合



- 用三个模型拟合了这个训练集的样本。训练数据通过随机抽取x然后用二次函数确定性地生成y来合成的。（前提：无噪声）
- (左)线性函数拟合数据——欠拟合——无法捕捉数据中的曲率信息。
- (中)用二次函数拟合数据不会导致明显的欠拟合或者过拟合。
- (右)9阶的多项式拟合数据会导致过拟合。解此欠定的正规方程。得出的解能够精确地穿过所有的训练点——无法提取有效的结构信息。



线性函数　　　二次函数　　　9阶的多项式

- **改变模型容量——改变输入特征数——不是越多越好**
  - ✓ **容量不足的模型不能解决复杂任务。**
  - ✓ **容量高的模型能够解决复杂的任务，但是当其容量高于任务所需时，有可能会过拟合。**
- **So far we have only described changing a model's capacity by changing the number of input features it has (and simultaneously adding new parameters associated with those features).**

Ref. 2.3.4.3 Cross-Validation

Define KFoldXV($\mathbb{D}, A, L, k$):

Require: $\mathbb{D}$ 为给定数据集，其中元素为 $\boldsymbol{z}^{(i)}$

Require: A 为学习算法，可视为一个函数（使用数据集作为输入，输出一个学好的函数）

Require: L 为损失函数，可视为来自学好的函数 f，将样本 $\boldsymbol{z}^{(i)} \in \mathbb{D}$ 映射到 $\mathbb{R}$ 中标量的函数

Require: k 为折数

将 $\mathbb{D}$ 分为 k 个互斥子集 $\mathbb{D}_i$，它们的并集为 $\mathbb{D}$

for i **from** 1 **to** k **do**

$f_i = A(\mathbb{D} \backslash \mathbb{D}_i)$

for $\boldsymbol{z}^{(j)}$ **in** $\mathbb{D}_i$ **do**

$e_j = L(f_i, \boldsymbol{z}^{(j)})$

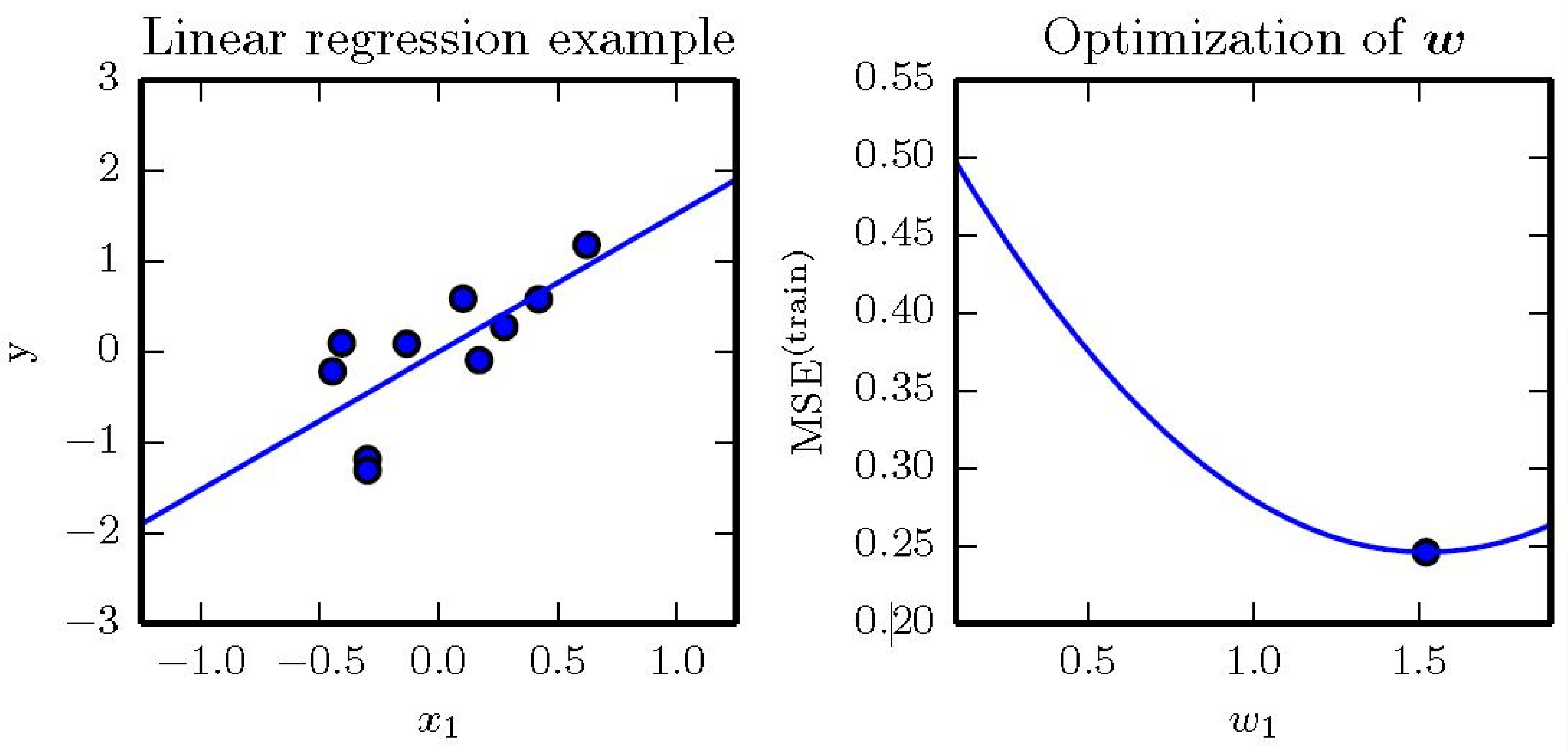
end for

end for

Return $\boldsymbol{e}$

Lab 1-2

- **Try to Write up Pseudo-code for the *Linear Regression Example***
- **Not limited to the data and form of the function**

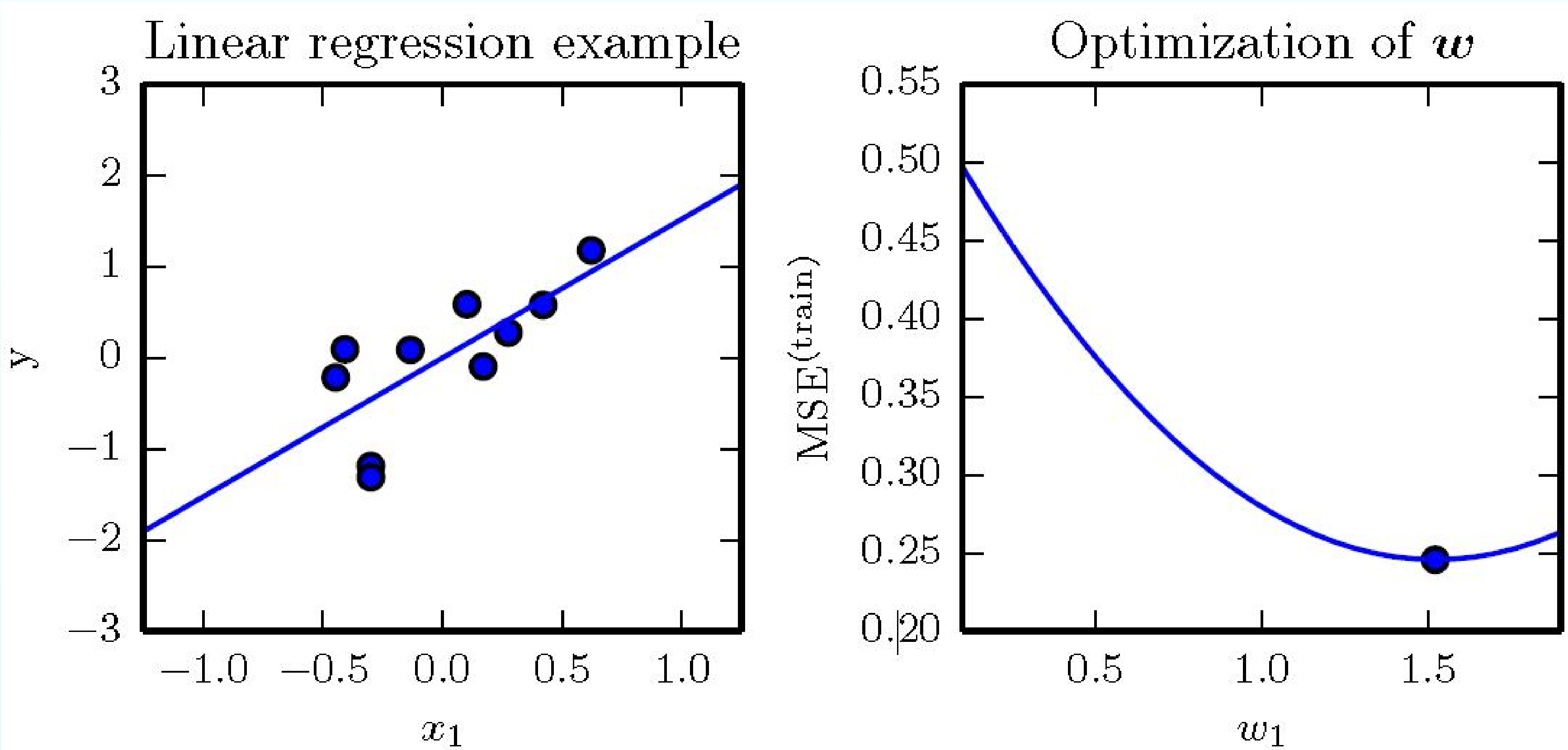


Lab 1-2-Ref

Eg. of the Linear Regression Learning

$$\Rightarrow w = \left(\boldsymbol{X}^{(\text{train})\top} \boldsymbol{X}^{(\text{train})} \right)^{-1} \boldsymbol{X}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})}$$

- 训练集包括十个数据点，每点包含一个特征，只有一个要学习的参数w_1。
- (左) 观察到线性回归学习w_1，使直线$y=w_1x$尽量接近穿过所有的训练点。
- (右) 标注点表示由学习到的w_1的值，它可以最小化训练集上的均方误差。



End of Part II