大数据导论
Introduction to Big Data

大数据回归预测：基础概念与算法

叶允明

计算机科学与技术学院

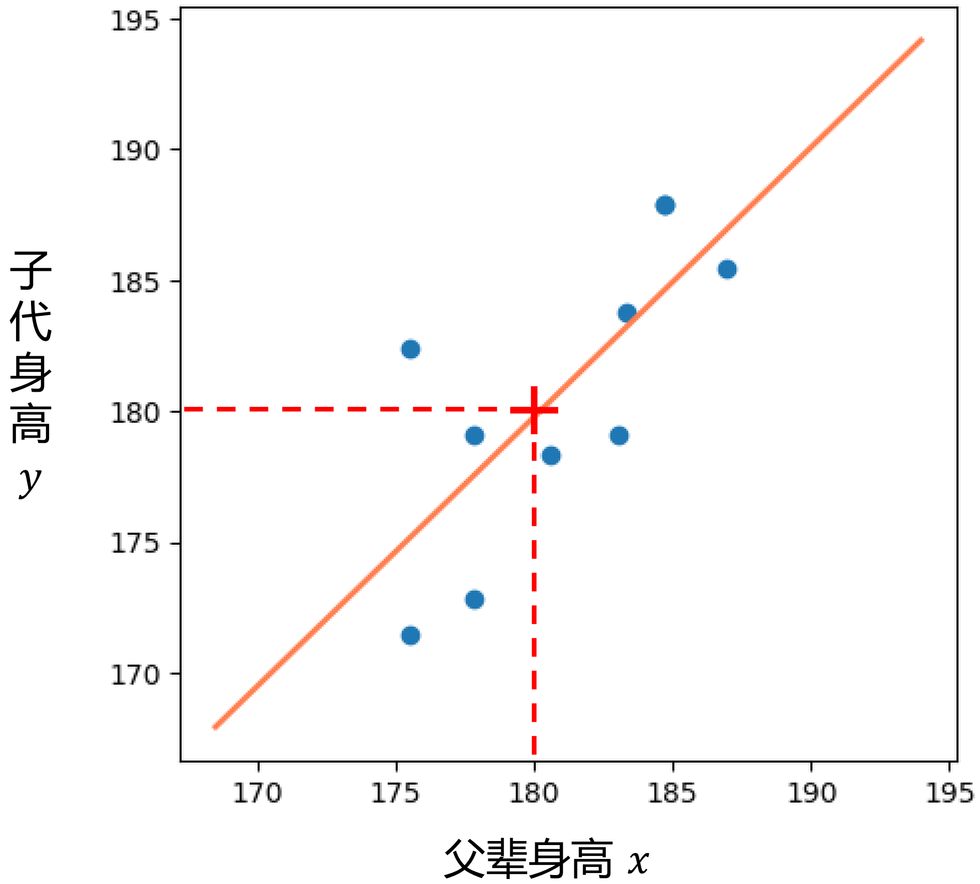
哈尔滨工业大学（深圳）

目录

- 回归预测的基本概念
- 线性回归
- 基于梯度下降法的线性回归
- 基于最小二乘法的线性回归（optional）

回归预测任务

- 回归（regression）：预测给定数据对象的目标值（与分类的类别对应）



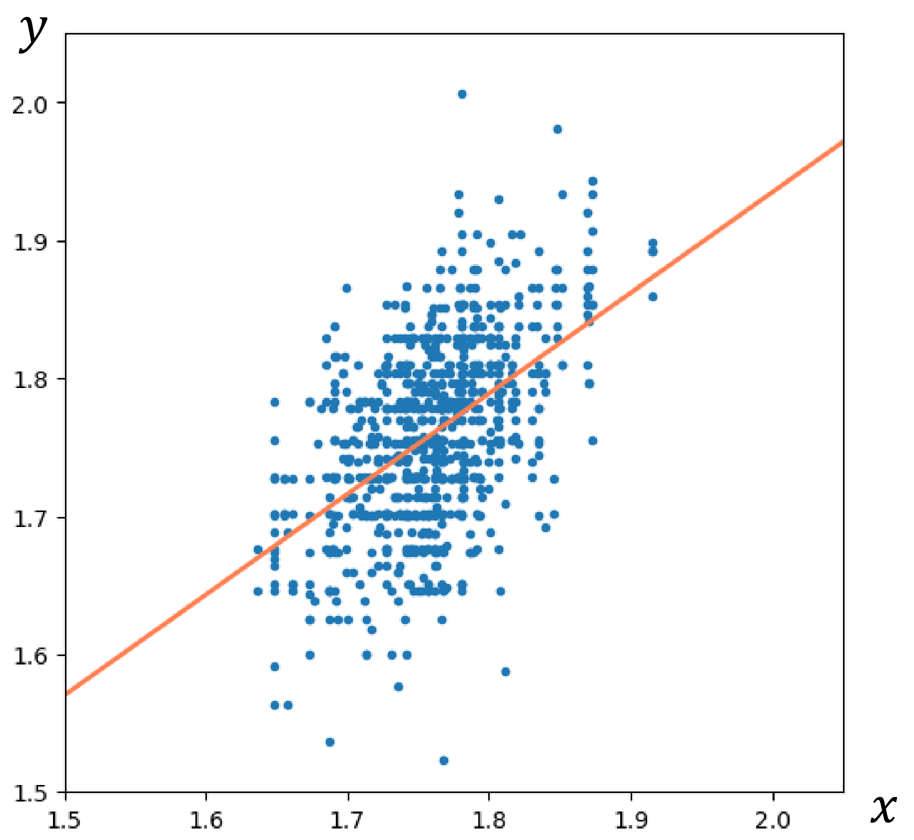
x = 180 时，y = ?

回归预测举例

- 根据父母平均身高，预测子女成年身高。 （Francis Galton，1886）
  - 观察了205对父母及每对父母的后代
  - 总的趋势:父辈的身材偏高（矮）时，
    子代的身材也偏高（矮）
  - 子代的身高 y 与父母平均身高 x 大致满足：

$$y = 0.477 + 0.729x \text{ (米)}$$

  - 回归（regression）：
    子代身高的分布不会向高矮两个极端发展，
    而是趋于“回归”到父辈身高的平均值



回归任务的定义

- 回归任务可以用一个形式化函数表示：

$$y = f(\boldsymbol{x}),$$

$$其中\ \boldsymbol{x} \in \mathbb{D}, y \in \mathbb{R}$$

  - 回归函数$f(\boldsymbol{x})$经过运算可以输出一个连续的实数值 y ，即"回归模型"

如何构造回归函数$\boldsymbol{f}(\boldsymbol{x})$呢?

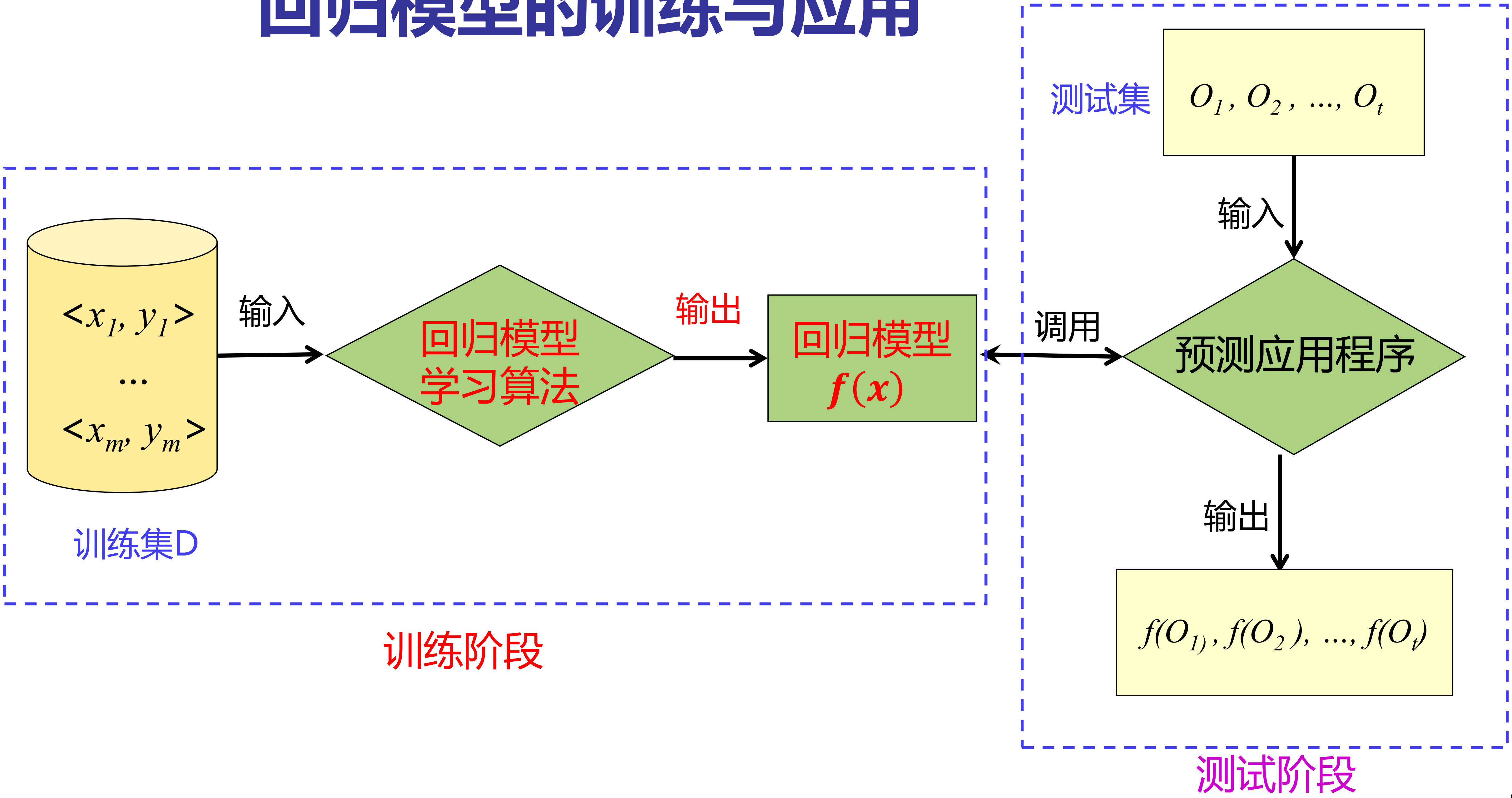
回归预测的应用领域

- 几乎每个人工智能应用领域都涉及到预测问题
  - 股票预测
  - 贷款额度估计
  - 视频预测
  - 销售业绩预测
  - 医学诊断
  - 欺诈检测
  - ……

完成回归任务的“两阶段”流程

- **回归模型构建（训练阶段）：即学习阶段**
  - 从目标值（target value）已知的训练数据集中学习，生成回归模型 $f(x)$
  - 回归模型可表示成线性函数、超平面、回归树等形式
- **回归模型应用（测试阶段）：**
  - 用回归模型 $f(x)$来预测新的数据对象的目标值

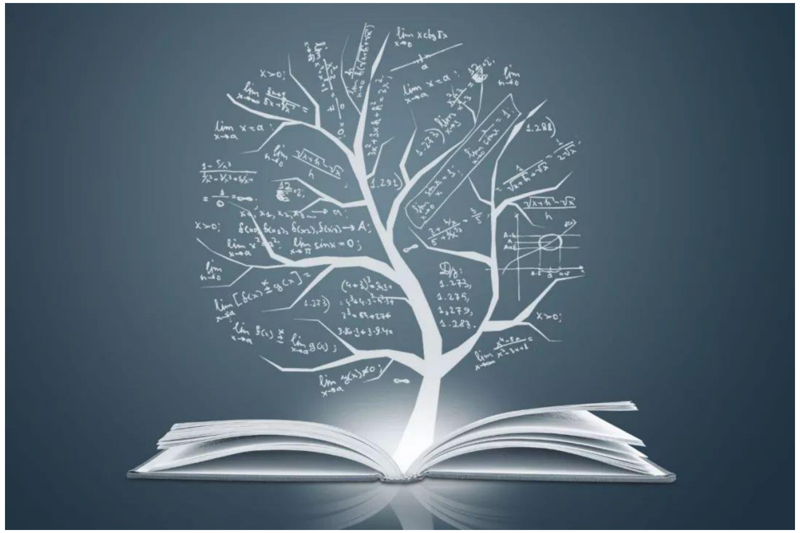
回归模型的训练与应用



测试集
$O_1, O_2, \ldots, O_t$
输入
<x_1, y_1>
…
<x_m, y_m>
输入
回归模型学习算法
输出
回归模型 $f(x)$
调用
预测应用程序
输出
$f(O_1), f(O_2), \ldots, f(O_t)$
训练集D
训练阶段
测试阶段

常用的回归方法

- 线性回归
  - 套索回归（Lasso Regression）
  - 岭回归（Ridge Regression）
  - 弹性网络回归 （ElasticNet Regression）
- 非线性回归
  - KNN回归（K Neighbors Regression）
  - 决策树回归（Decision Tree Regression）
  - 支持向量回归（Support Vector Regression，SVR）
  - 集成学习回归：随机森林、AdaBoost、XGBoost、LightGBM
  - 神经网络与深度学习



线性回归

应用案例

- 预测下一天的日均气温 $y = f(\boldsymbol{x})$, 其中 $\boldsymbol{x} \in \mathbb{D}, y \in \mathbb{R}$
  - Kaggle数据集（daily climate time series data）

日期	日均气温 (mean temp)	相对湿度 (humidity)	风速 (wind speed)	气压 (pressure)
…	…	…	…	…
2017-01-02	7.40	92.00	2.980	1017.80
2017-01-03	7.17	87.00	4.63	1018.67

$\boldsymbol{x}$

f

y

https://www.kaggle.com/sumanthvrao/daily-climate-time-series-data

线性回归模型

- 给定数据集$\mathbb{X}=\{(\boldsymbol{x_1},y_1),(\boldsymbol{x_2},y_2),\ldots,(\boldsymbol{x_n},y_n)\}$，其中，每个样本$\boldsymbol{x_i}$有$d$ 个特征：

$$\boldsymbol{x_i}=\left(x_{i,1};x_{i,2};\ldots;x_{i,d}\right)^T,\ \ y_i\in\mathbb{R}.$$

- 线性模型的目的是学习一个关于 $\boldsymbol{x}$ 的线性函数$f(\boldsymbol{x})$，来尽可能准确地预测 y

$$f(\boldsymbol{x})=\boldsymbol{w}\cdot\boldsymbol{x}+b,\quad \boldsymbol{w}=\begin{pmatrix}w_1\\w_2\\\ldots\\w_d\end{pmatrix}$$

$$y\approx f(\boldsymbol{x})$$

  - 即：$f(\boldsymbol{x})$与y之间的误差越小越好.
  - 其中$\boldsymbol{w}$与b是需要学习的参数项.

在前面的案例中，

日均气温 (mean temp)	相对湿度 (humidity)	风速 (wind speed)	气压 (pressure)
7.40	92.00	2.980	1017.80

$$\boldsymbol{x_i}=(\ x_{i,t}\ ,\ x_{i,h}\ ,\ x_{i,w}\ ,\ x_{i,p}\ )^T$$

线性回归模型的优劣评价方法

- 在求解$\boldsymbol{w}$和b之前，需要给出衡量$f(\boldsymbol{x})$与y之间误差的损失函数
- 在回归方法中，均方误差是比较常用的损失函数，其损失函数定义如下：

$$L(\boldsymbol{w}, b) = \frac{1}{2n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - y_i)^2$$

$$(\boldsymbol{x}_i, y_i)$$

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$$

单变量线性回归

- 第一步：确定模型空间
  - 直观想法：下一天的日均气温很可能与前一天的日均气温相关

回归模型

$$y = w \cdot x_t + b$$

w :相关性权重, b : 偏差

模型空间（函数集合）

$$y = 0.9 \cdot x_t + 2.0$$
$$y = 1.0 \cdot x_t + 0.1$$
$$y = 1.1 \cdot x_t + 1.8$$

…… ……

线性回归模型：$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + \mathrm{b}$

数据：$(\boldsymbol{x_i}, y_i)$

日均气温 (mean temp)	相对湿度 (humidity)	风速 (wind speed)	气压 (pressure)
7.40	92.00	2.980	1017.80

$$\boldsymbol{x_i} = (x_{i,t}, x_{i,h}, x_{i,w}, x_{i,p})^T$$

损失函数

- 第二步：模型优劣的评价标准

$$L(w,b)=\sum_{i=1}^{n}\Big[\underbrace{y_i-\big(\underbrace{w\cdot x_{i,t}+b}_{\text{预测值}}\big)}_{\text{预测误差}}\Big]^2$$

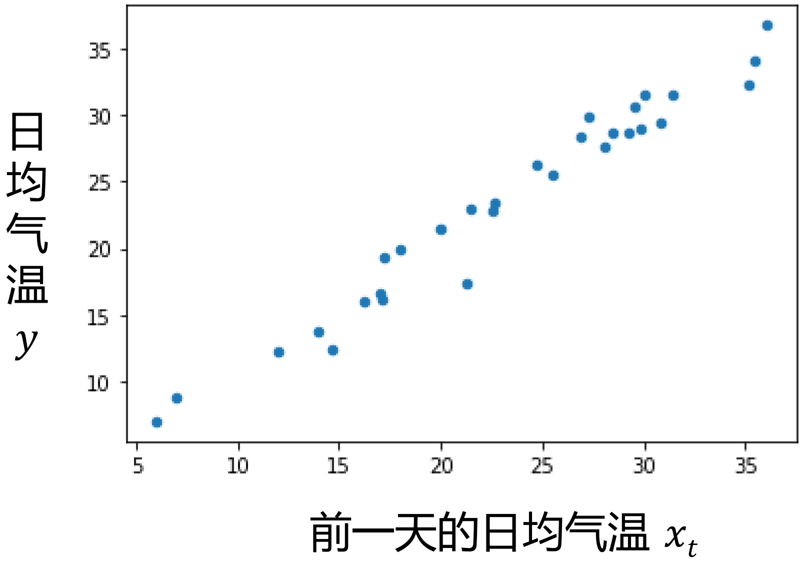
模型空间（函数集合）

$y = 0.9 \cdot x_t + 2.0$
$y = 1.0 \cdot x_t + 0.1$
$y = 1.1 \cdot x_t - 1.8$
…… ……

- 为了便于计算，随机选取了三十天的数据作为训练集(单位，℃)

$(x_{1,t}\ ,\ y_1)$
$(x_{2,t}\ ,\ y_2)$
⋮
$(x_{30,t}, y_{30})$

散点图 ⟹

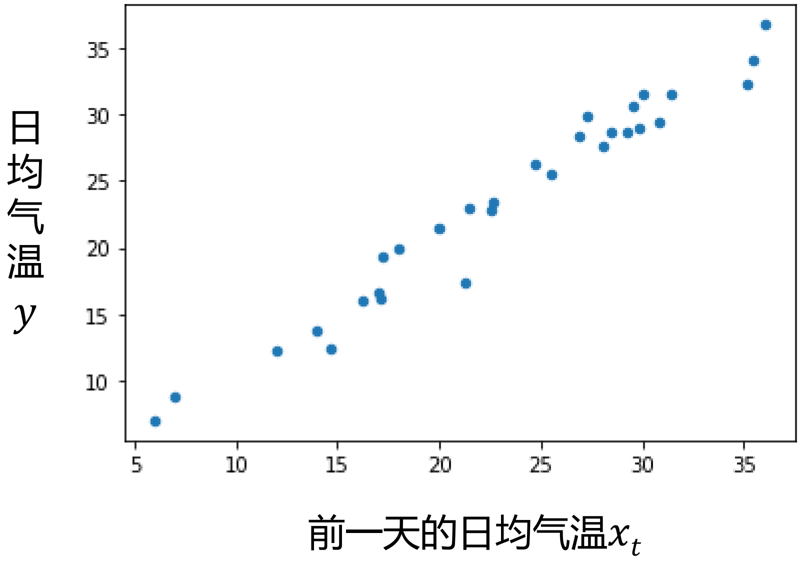


损失函数

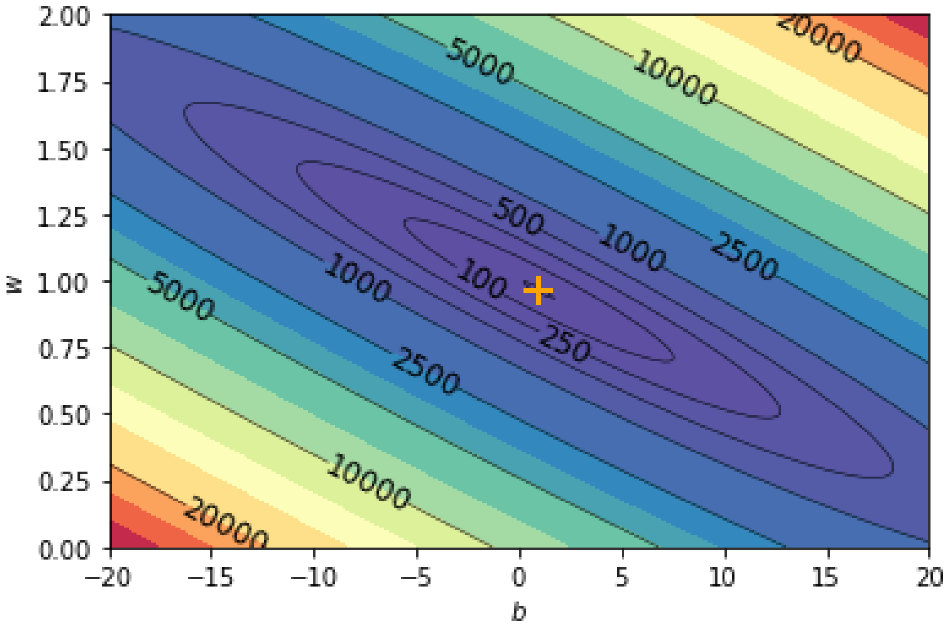
- 损失函数等值线图

$$L(w,b)=\sum_{i=1}^{n}\left(y_i-w\cdot x_{i,t}-b\right)^2$$

如何寻找最小值点?



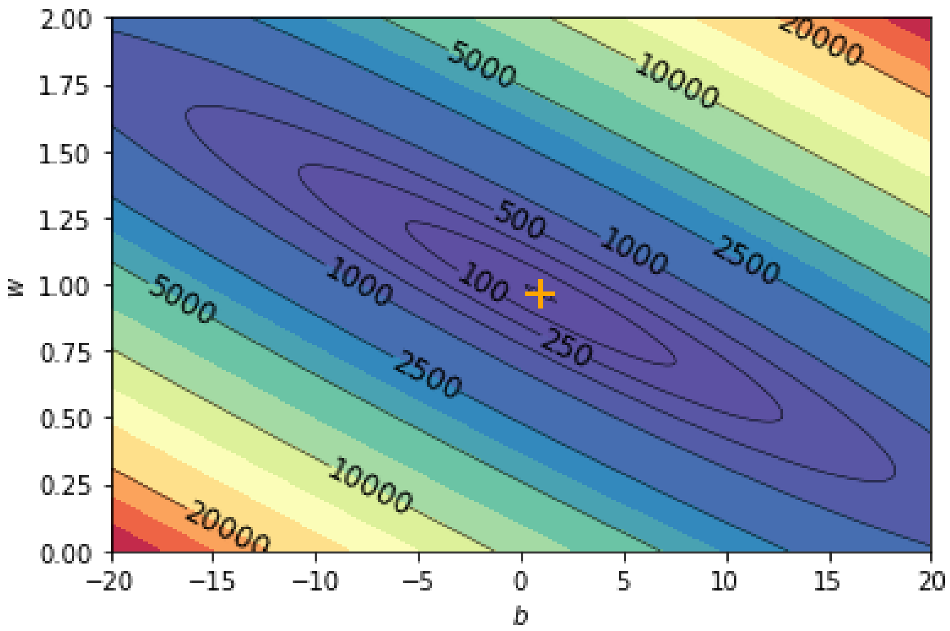
模型空间



最优化算法

- 第三步：找到“最优”模型
- 寻找一组最优的参数 (w^*, b^*)，能够最小化损失函数，即

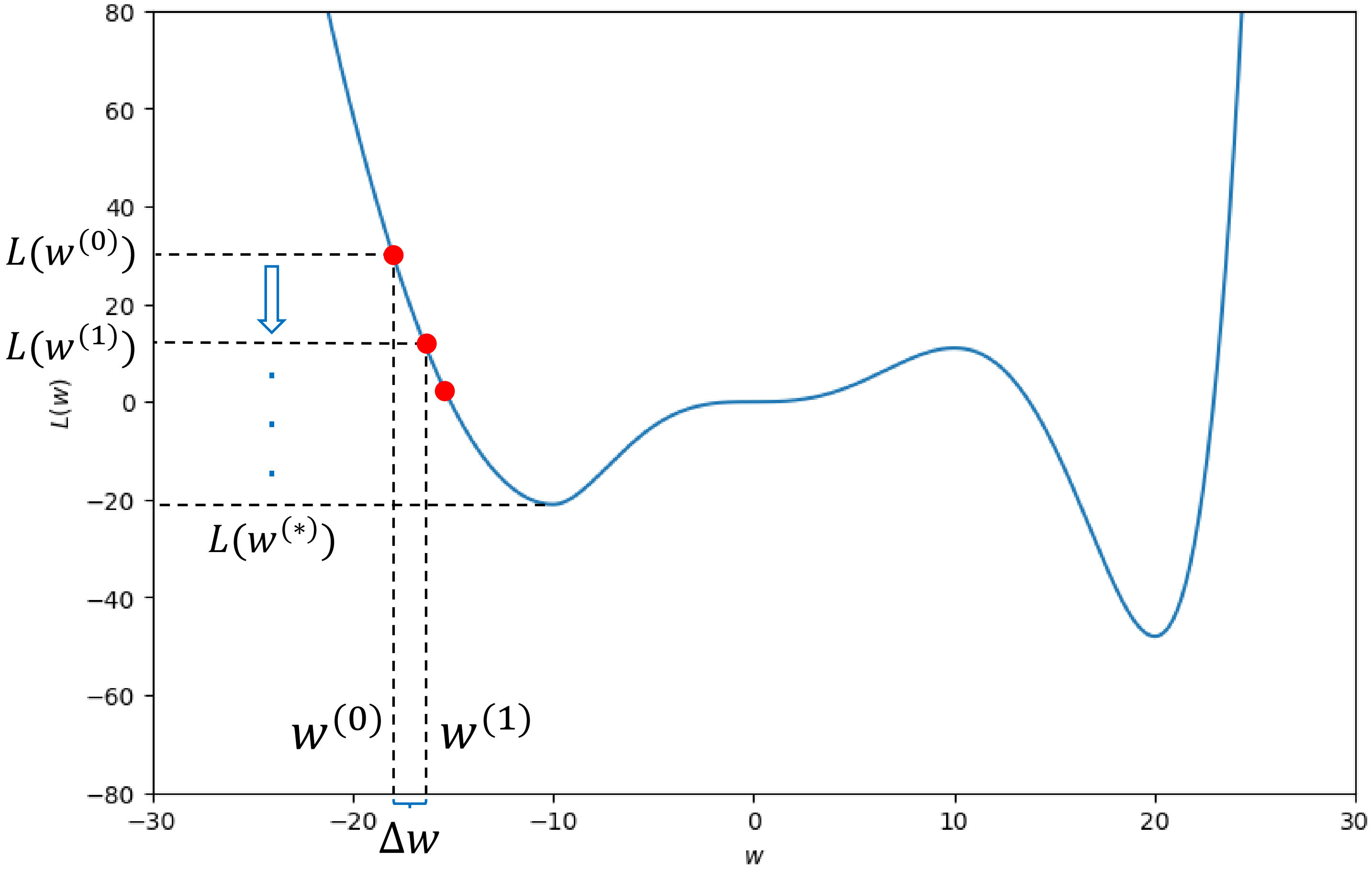
$$w^*, b^* = \arg\min_{b,w} L(w, b)$$
$$= \arg\min_{b,w} \sum_{i=1}^{n} \left[y_i - (w \cdot x_{i,t} + b)\right]^2$$



- 常用算法：
  - 梯度下降法
  - 最小二乘法（小规模数据）

基于梯度下降法的线性回归模型

梯度下降法的基本思想



80
60
40
20
0
−20
−40
−60
−80
$L(w)$
$L(w^{(0)})$
$L(w^{(1)})$
$L(w^{(*)})$
$w^{(0)}$
$w^{(1)}$
Δw
−30
−20
−10
0
10
20
30
w

梯度下降法的数学原理(一元函数)

- 一元函数泰勒公式
  - 如果一元函数 $L(w)$ 在点 $w^{(0)}$ 的某一邻域内可导，则有

$$L(w) = L\left(w^{(0)}\right) + L'\left(w^{(0)}\right)\left(w - w^{(0)}\right) + o\left(w - w^{(0)}\right)$$

- 如果变化量 $\Delta w = w - w^{(0)} = -\eta L'\left(w^{(0)}\right)$，学习率 η为一个较小的正数，则

"负梯度方向"

$$\begin{aligned} L(w) &\approx L\left(w^{(0)}\right) - L'\left(w^{(0)}\right) \cdot \eta L'\left(w^{(0)}\right) \\ &= L\left(w^{(0)}\right) - \eta\left(L'\left(w^{(0)}\right)\right)^2 \\ &< L\left(w^{(0)}\right) \end{aligned}$$

附注：$L'\left(w^{(0)}\right) = \frac{dL}{dw}\Big|_{w=w^{(0)}}$

梯度下降法的数学原理(多元函数)

- 对于线性回归模型 $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b, \boldsymbol{w} = \begin{pmatrix} w_1 \\ w_2 \\ \cdots \\ w_d \end{pmatrix}$，损失函数$L(\boldsymbol{w}, b)$为多元函数
- 为了简洁起见，将待求解参数 $\boldsymbol{w}$ 和 b 表示为 W， $W = \begin{pmatrix} b \\ \boldsymbol{w} \end{pmatrix}$
- 将一元函数泰勒公式 $L(w) = L(w^{(0)}) + L'(w^{(0)})(w - w^{(0)}) + o(w - w^{(0)})$ 推广，可以得到

多元函数泰勒公式：

- 如果多元函数 $L(W)$ 在点 $W^{(0)}$ 的某一邻域内有一阶连续偏导数，则有

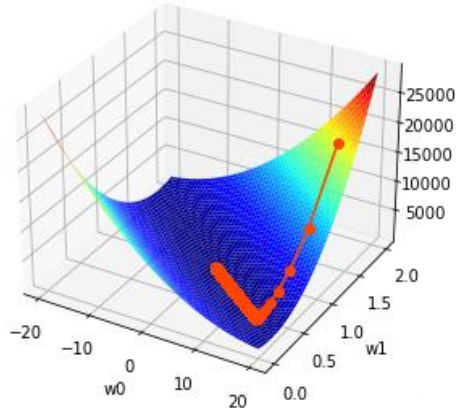
$$L(W) = L(W^{(0)}) + \nabla L(W^{(0)})^T (W - W^{(0)}) + o(W - W^{(0)}),$$

其中 ∇ 为梯度算子（nabla operator)

附注：$\nabla L(W^{(0)}) = \left(\frac{\partial L}{\partial b}|_{W=W^{(0)}}, \frac{\partial L}{\partial w_1}|_{W=W^{(0)}}, \frac{\partial L}{\partial w_2}|_{W=W^{(0)}}, \dots, \frac{\partial L}{\partial w_d}|_{W=W^{(0)}}\right)^T$

梯度下降法的数学原理(多元函数)

- 多元函数 $L(W)$ 在点 $W^{(0)}$ 处的梯度 $\nabla L(W^{(0)})$ 为一个向量：
  - 梯度的方向与取得最大方向导数的方向一致，
  - 梯度的模为方向导数的最大值。



- 根据公式 $L(W) = L(W^{(0)}) + \nabla L(W^{(0)})^T (W - W^{(0)}) + o(W - W^{(0)})$,

如果变化量 $\Delta W = W - W^{(0)} = -\eta \nabla L(W^{(0)})$，学习率 η 为一个较小的正数，则

"负梯度方向"

⇩

函数值下降

$$
\begin{aligned}
L(W) &\approx L(W^{(0)}) - \eta \nabla L(W^{(0)})^T \nabla L(W^{(0)}) \\
&= L(W^{(0)}) - \eta \left\| \nabla L(W^{(0)}) \right\|^2 \\
&< L(W^{(0)})
\end{aligned}
$$

梯度下降法的算法流程（单个参数）

- 以一个仅有单个参数 w 的光滑的损失函数 $L(w)$ 为例，梯度下降法：

随机选取初始值 $w^{(0)}$

$$w^{(1)} \leftarrow w^{(0)} - \eta \frac{dL}{dw}|_{w=w^{(0)}}$$

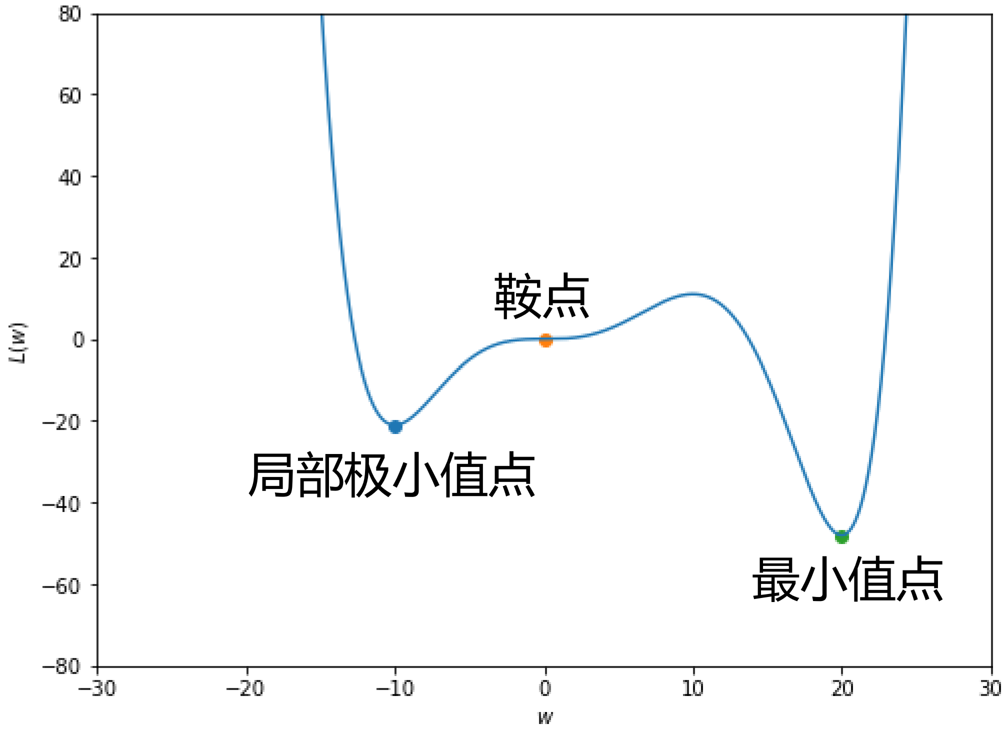
$$w^{(2)} \leftarrow w^{(1)} - \eta \frac{dL}{dw}|_{w=w^{(1)}}$$

…… ……

$$w^{(j+1)} \leftarrow w^{(j)} - \eta \frac{dL}{dw}|_{w=w^{(j)}}$$

…… ……

直到 $|w^{(n+1)} - w^{(n)}| < \varepsilon$, ε为终止条件



梯度下降法的算法流程（两个参数）

- 案例的损失函数$L(w,b)$含有两个参数 $$L(w,b)=\sum_{i=1}^{N}\left(y_i-w\cdot x_{i,t}-b\right)^2$$
  - （随机）选取两个初始值 $b^{(0)}$, $w^{(0)}$
  - 计算 $\frac{\partial L}{\partial w}|_{w=w^{(0)},b=b^{(0)}}$，$\frac{\partial L}{\partial b}|_{w=w^{(0)},b=b^{(0)}}$，更新 b,w $$\nabla L(b,w)=\begin{pmatrix}\frac{\partial L}{\partial b}\\ \frac{\partial L}{\partial w}\end{pmatrix}$$

$$w^{(1)}\leftarrow w^{(0)}-\eta\frac{\partial L}{\partial w}|_{w=w^{(0)},b=b^{(0)}}\quad,\quad b^{(1)}\leftarrow b^{(0)}-\eta\frac{\partial L}{\partial b}|_{w=w^{(0)},b=b^{(0)}}$$

  - 计算 $\frac{\partial L}{\partial w}|_{w=w^{(1)},b=b^{(1)}}$，$\frac{\partial L}{\partial b}|_{w=w^{(1)},b=b^{(1)}}$，更新 b,w

$$w^{(2)}\leftarrow w^{(1)}-\eta\frac{\partial L}{\partial w}|_{w=w^{(1)},b=b^{(1)}}\quad,\quad b^{(2)}\leftarrow b^{(1)}-\eta\frac{\partial L}{\partial b}|_{w=w^{(1)},b=b^{(1)}}$$

$\dots\dots$

梯度下降法的应用示例

- 计算梯度

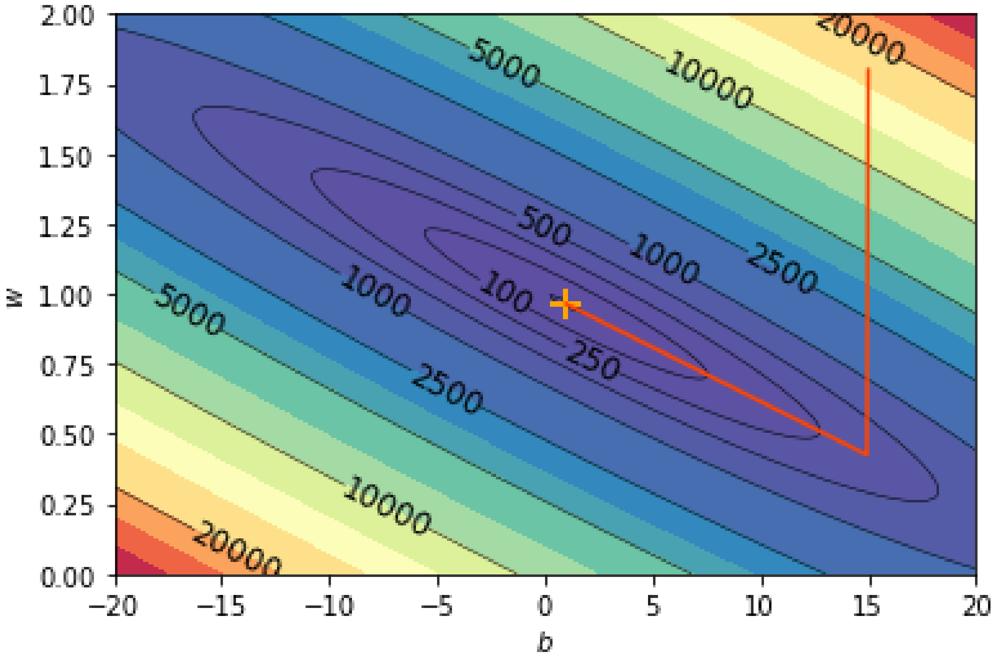
$$L(w,b)=\sum_{i=1}^{n}\left(y_i\,\underline{-\,w\cdot x_{i,t}}\,\underline{-\,b}\right)^2$$

$$\frac{\partial L}{\partial w}=\sum_{i=1}^{n}2\left(y_i-w\cdot x_{i,t}-b\right)\left(-x_{i,t}\right)$$

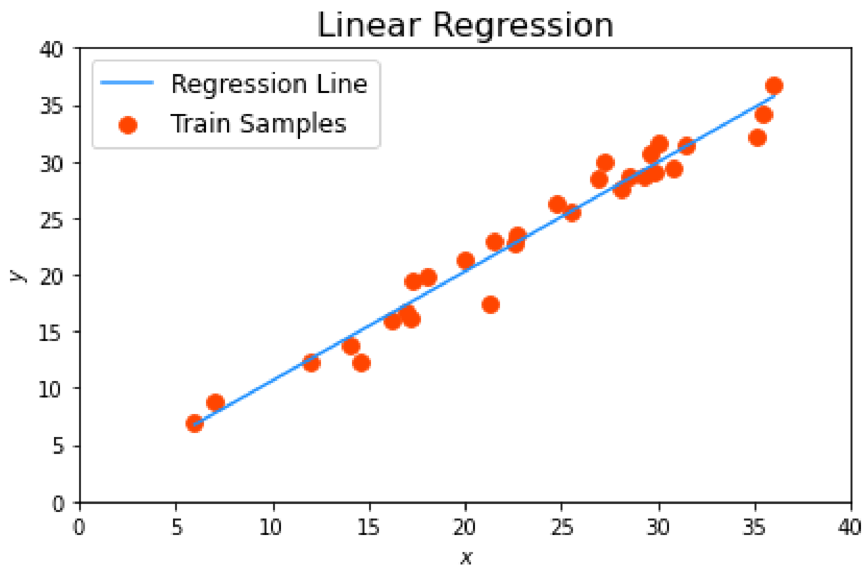
$$\frac{\partial L}{\partial b}=\sum_{i=1}^{n}2\left(y_i-w\cdot x_{i,t}-b\right)(-1)$$

梯度下降法示例

- 沿梯度下降的方向寻找极小值



$w^* = 0.965, b^* = 0.980$

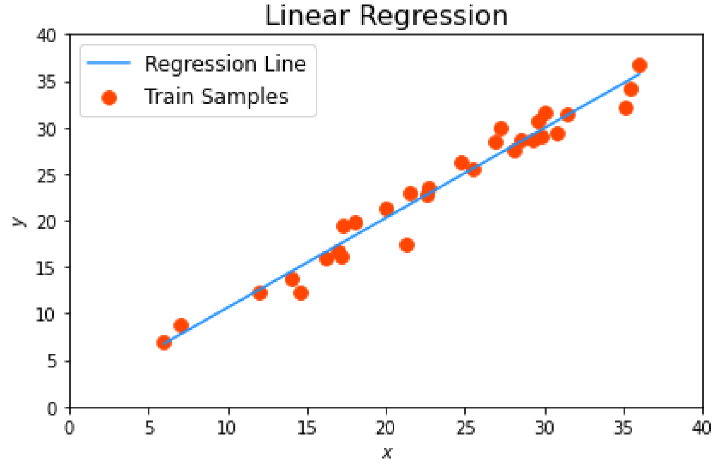


$y = 0.965x_t + 0.980$

测试模型效果（单变量线性回归）

- 训练集均方误差

$$\frac{1}{30}\sum_{i=1}^{30}\left(y_i - w^* \cdot x_{i,t} - b^*\right)^2 = 2.134$$



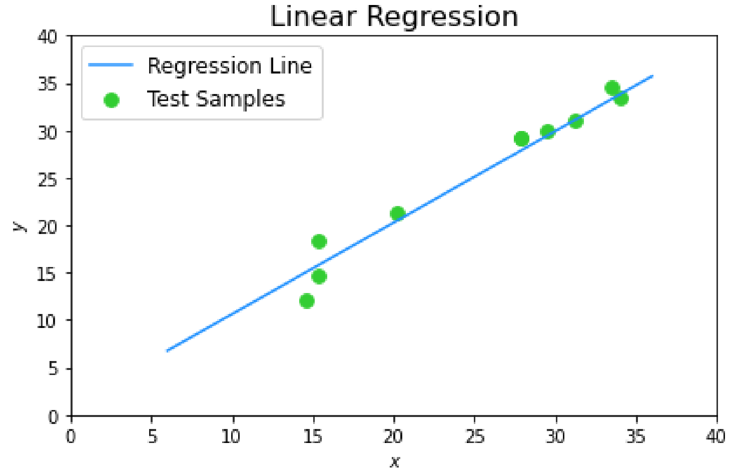
$$w^* = 0.965, b^* = 0.980$$

- 在测试数据中随机选取十天的数据作为测试集，测试模型的泛化性能
  - 测试集均方误差

$$\frac{1}{10}\sum_{i=1}^{10}\left(y_i - w^* \cdot x_{i,t} - b^*\right)^2 = 2.294$$

  - 测试集平均误差

$$\frac{1}{10}\sum_{i=1}^{10}\left|y_i - w^* \cdot x_{i,t} - b^*\right| = 1.229$$



改进模型：增加二次项特征

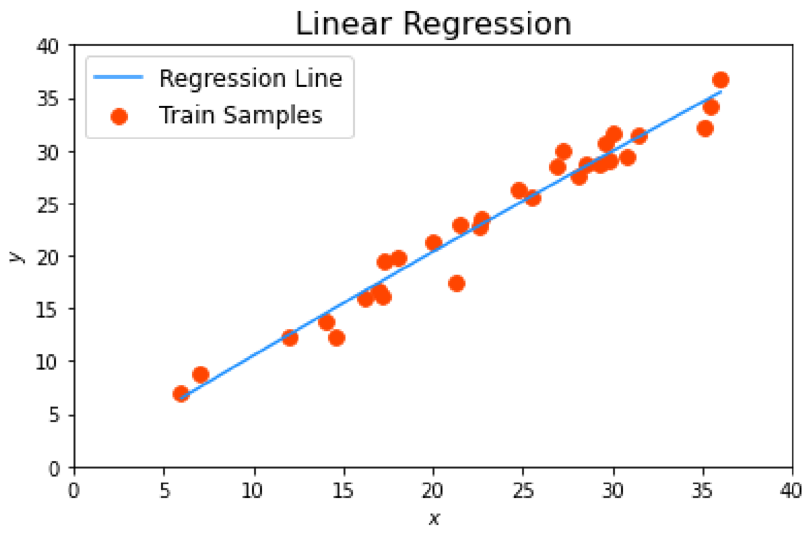
- 回归模型：

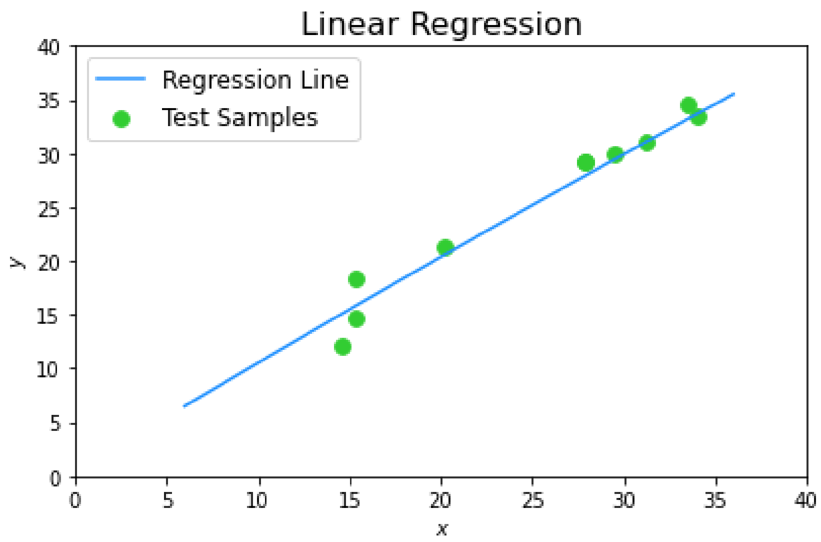
$$y = w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

- 计算模型参数，可得

$$w_2^* = -1.50 \times 10^{-3}, w_1^* = 1.030, b^* = 0.361$$

- 模型误差：
  - 训练集均方误差：2.123<2.134
  - 测试集均方误差：2.278<2.294





改进模型：增加三次项特征

- 回归模型：

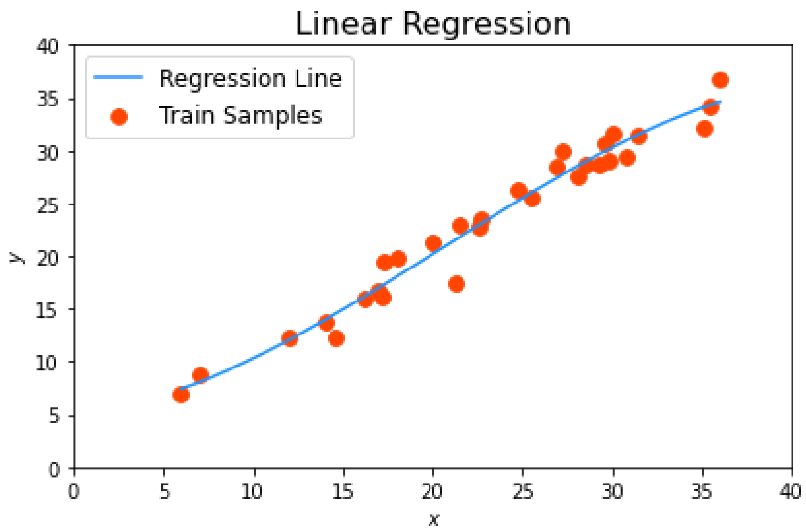
$$y = w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

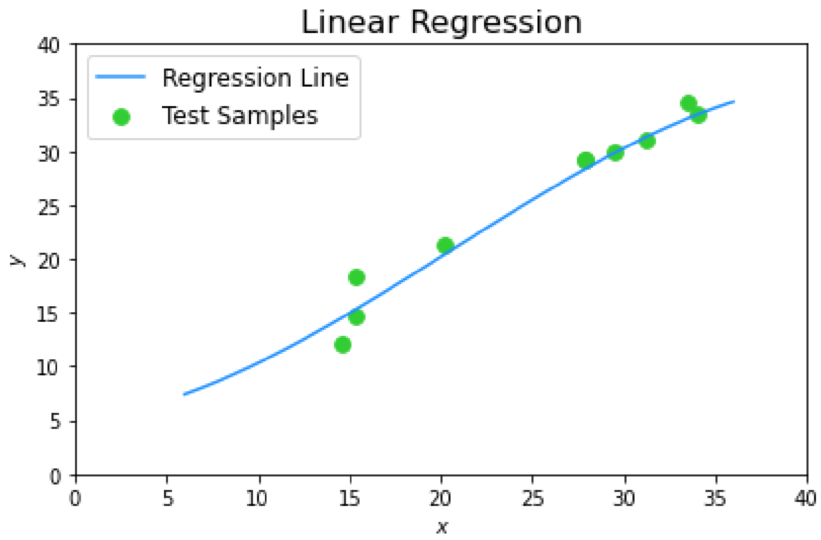
- 计算模型参数，可得

$$w_3^* = -7.43 \times 10^{-4}, w_2^* = 0.046,$$

$$w_1^* = 0.136, b^* = 5.123$$

- 模型误差：
  - 训练集均方误差：1.913<2.123
  - 测试集均方误差：2.042<2.278





改进模型：增加四次项特征

- 回归模型：

$$y = w_4 \cdot x_t^4 + w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

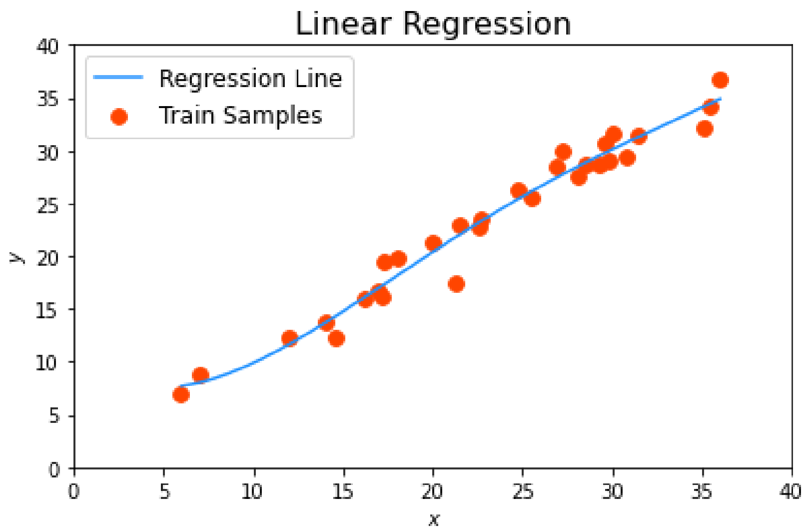
- 计算模型参数，可得

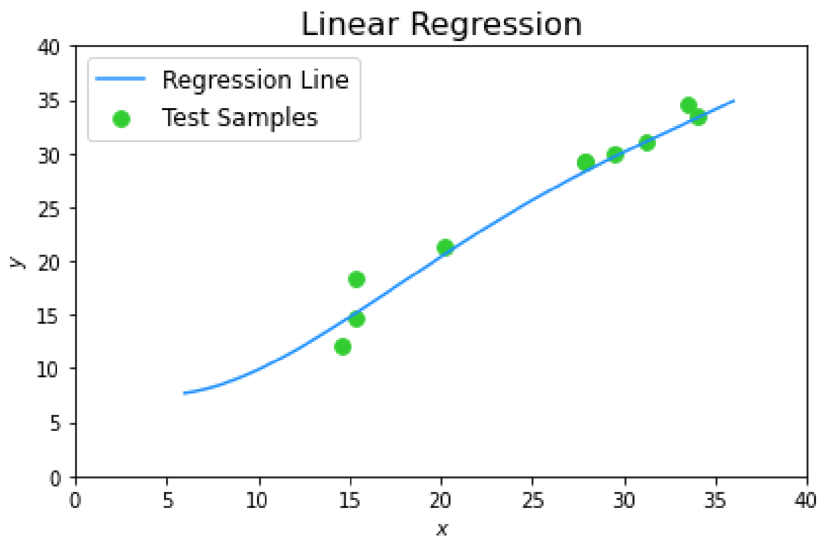
$$w_4^* = 4.75 \times 10^{-5}, w_3^* = -4.83 \times 10^{-3},$$

$$w_2^* = 0.167, w_1^* = -1.290, b^* = 10.43$$

- 模型误差：
  - 训练集均方误差：1.878<1.913
  - 测试集均方误差：2.053>2.042

过拟合





改进模型：增加五次项特征

- 回归模型：

$$y = w_5 \cdot x_t^5 + w_4 \cdot x_t^4 + w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

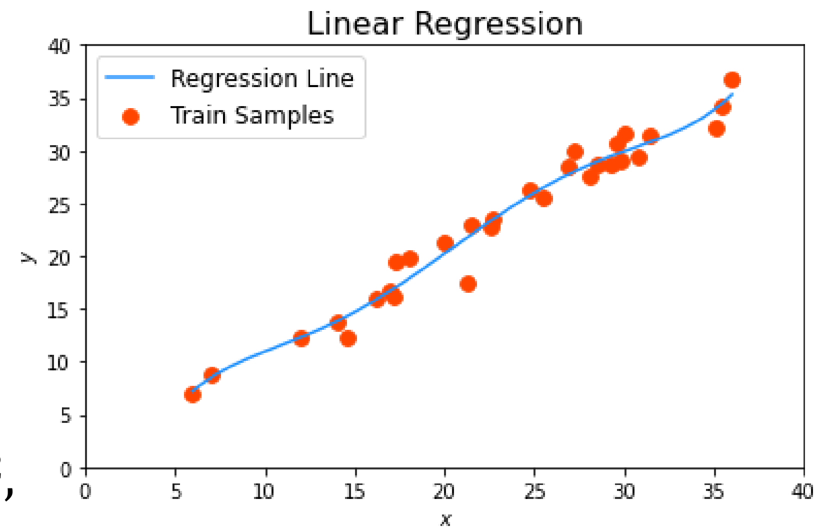
- 计算模型参数，可得

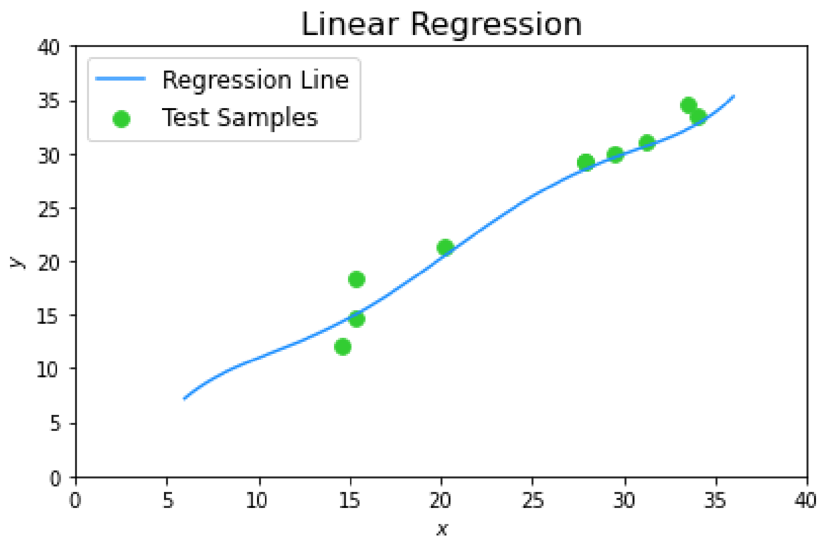
$$w_5^* = 1.184 \times 10^{-5}, w_4^* = -1.22 \times 10^{-3}, w_3^* = 4.64 \times 10^{-2},$$

$$w_2^* = -0.080, w_1^* = 6.948, b^* = -14.37$$

- 模型误差：
  - 训练集均方误差：1.797<1.878
  - 测试集均方误差：2.396>2.053

过拟合





过拟合

- 复杂的模型可以更好地拟合训练数据
- 但未必会在测试数据上获得更好的效果

$$y = w \cdot x_t + b$$

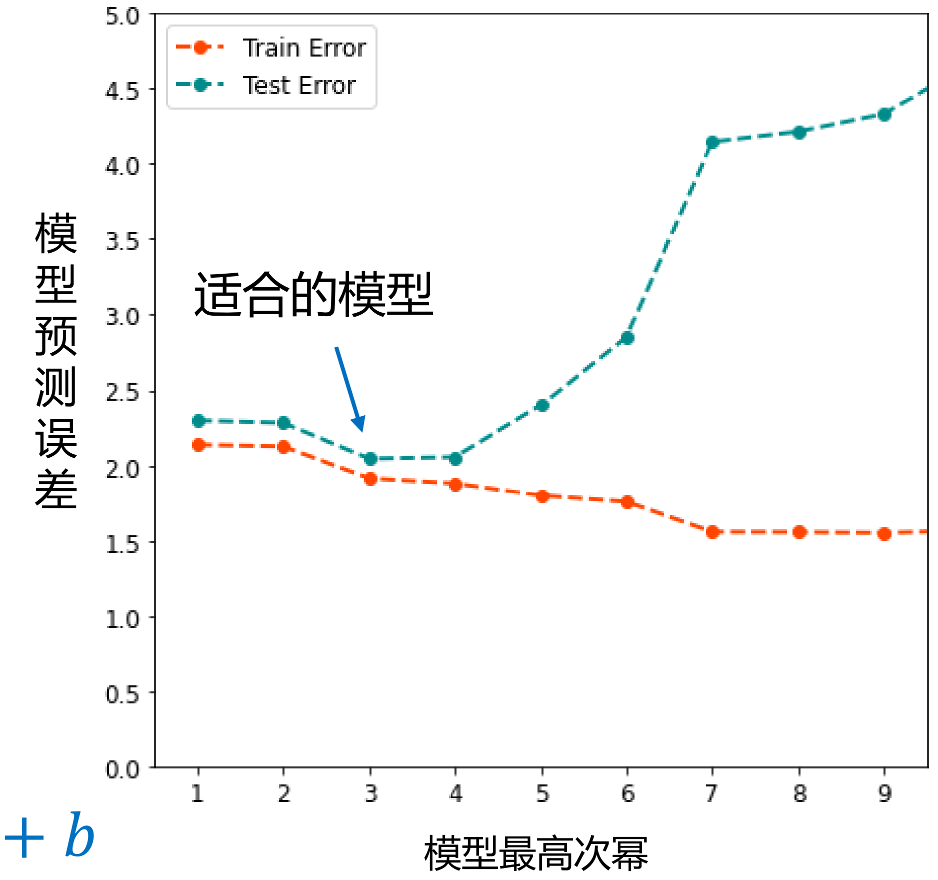
$$y = w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

$$y = w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

$$y = w_4 \cdot x_t^4 + w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

$$y = w_5 \cdot x_t^5 + w_4 \cdot x_t^4 + w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

…… ……



多变量线性回归

- 回到第一步：确定模型空间
  - 除前一天的日均气温外，考虑与前一天的相对湿度、风速、气压是否相关

$$y = w_1 \cdot x_t + w_2 \cdot x_t^2 + w_3 \cdot x_t^3 + w_4 \cdot x_h + w_5 \cdot x_w + w_6 \cdot x_p + b$$

线性回归模型：$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + \mathrm{b}$

数据：$(\boldsymbol{x}_i, y_i)$

日均气温 (mean temp)	相对湿度 (humidity)	风速 (wind speed)	气压 (pressure)
7.40	92.00	2.980	1017.80

$$\boldsymbol{x}_i = (x_{i,t}, x_{i,h}, x_{i,w}, x_{i,p})^T$$

增加其他特征变量

- 回归模型：

$$y = w_1 \cdot x_t + w_2 \cdot x_t^2 + w_3 \cdot x_t^3$$
$$+ w_4 \cdot x_h + w_5 \cdot x_w + w_6 \cdot x_p + b$$

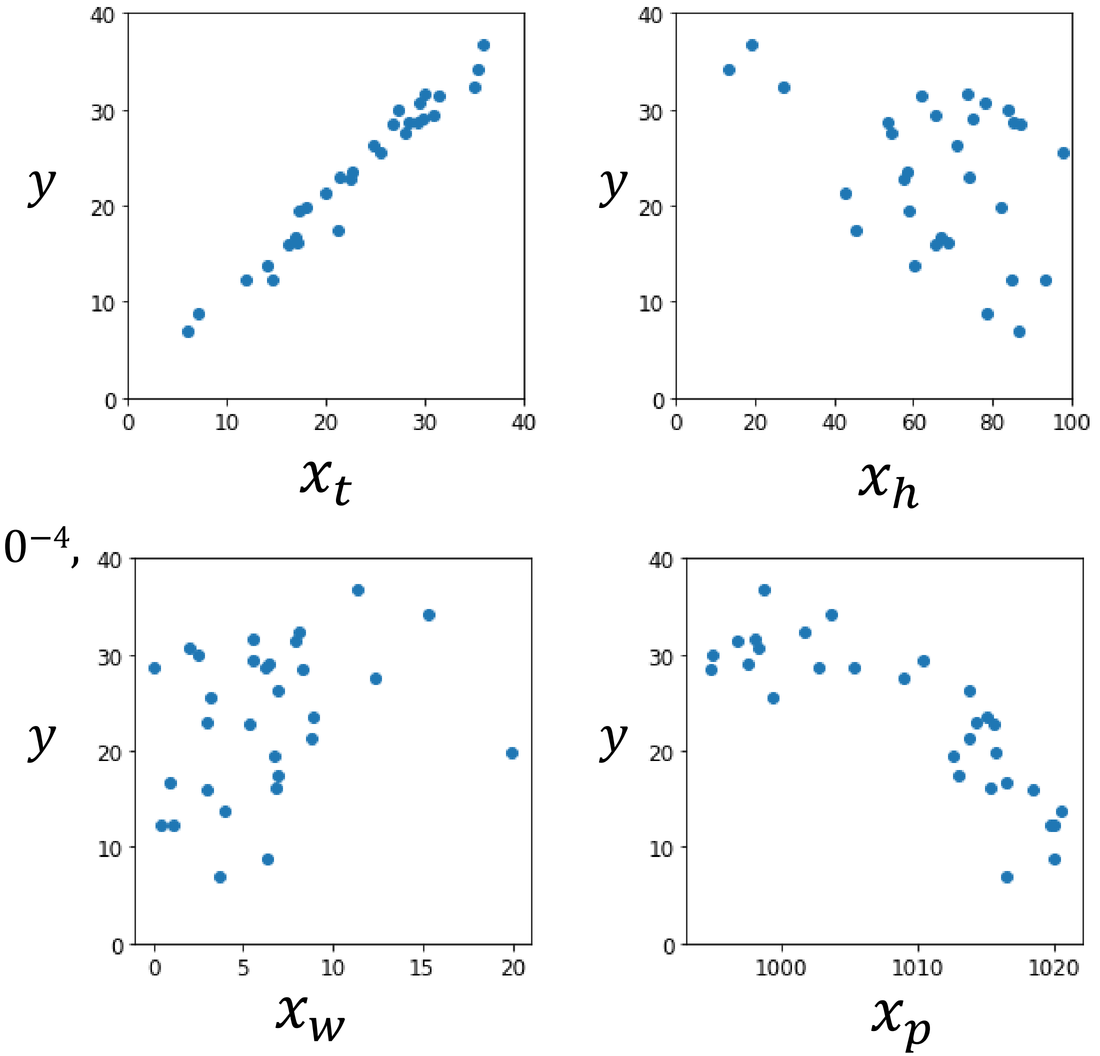
- 计算模型参数，可得

$w_6^* = -0.011, w_5^* = 0.010, w_4^* = 1.18 \times 10^{-2}, w_3^* = -2.58 \times 10^{-4},$

$w_2^* = 1.39 \times 10^{-2}, w_1^* = 0.667,\ b^* = 109.5$

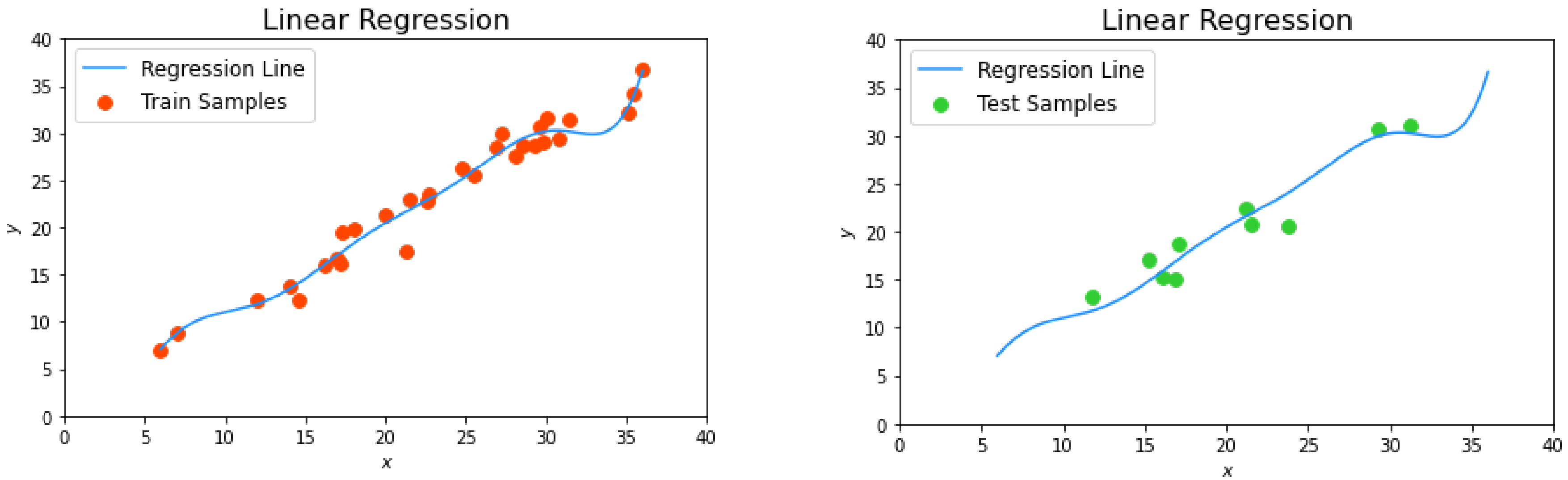
- 模型误差：
  - 训练集均方误差：1.553<1.913
  - 测试集均方误差：2.278>2.042

过拟合



正则化的基本思想

- 奥卡姆剃刀(Occam's razor)原理
  - 选择能够很好地解释已知数据但更简单的模型。
- 简单的函数更为平滑，也就不容易发生过拟合的问题。



$$y = w_{10} \cdot x_t^{10} + w_9 \cdot x_t^9 + \cdots + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

正则化

- 对于有 d 个特征的线性回归模型 $y=\sum_{i=1}^{d} w_j x_j + b$
- 损失函数为

$$L(w,b)=\sum_{i=1}^{n}\left(y_i-\sum_{i=1}^{d} w_j x_{i,j}-b\right)^2$$

- 加入正则化项的损失函数

λ为超参数，λ值越大，模型抗扰动能力越强

  - L1正则化

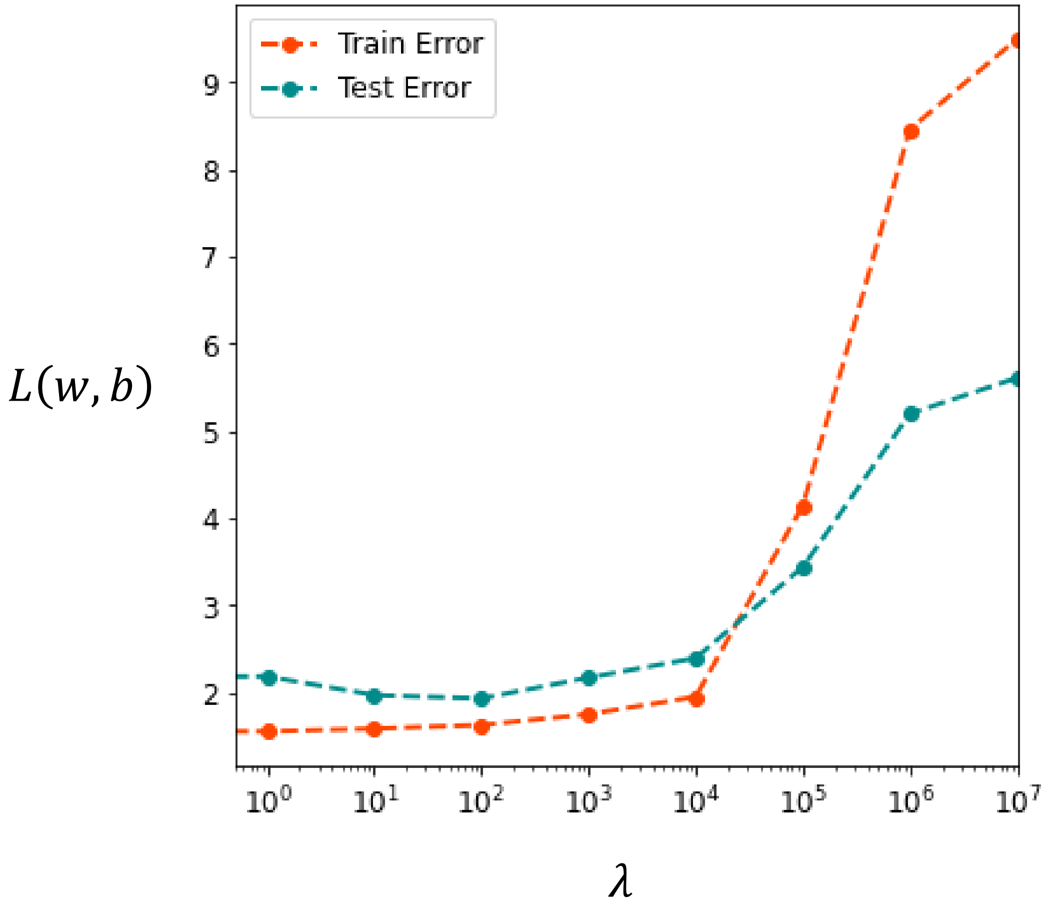
$$L(w,b)=\sum_{i=1}^{n}\left(y_i-\sum_{i=1}^{d} w_j x_{i,j}-b\right)^2+\boxed{\lambda\sum_{j=1}^{d}\left|w_j\right|}$$

  - L2正则化

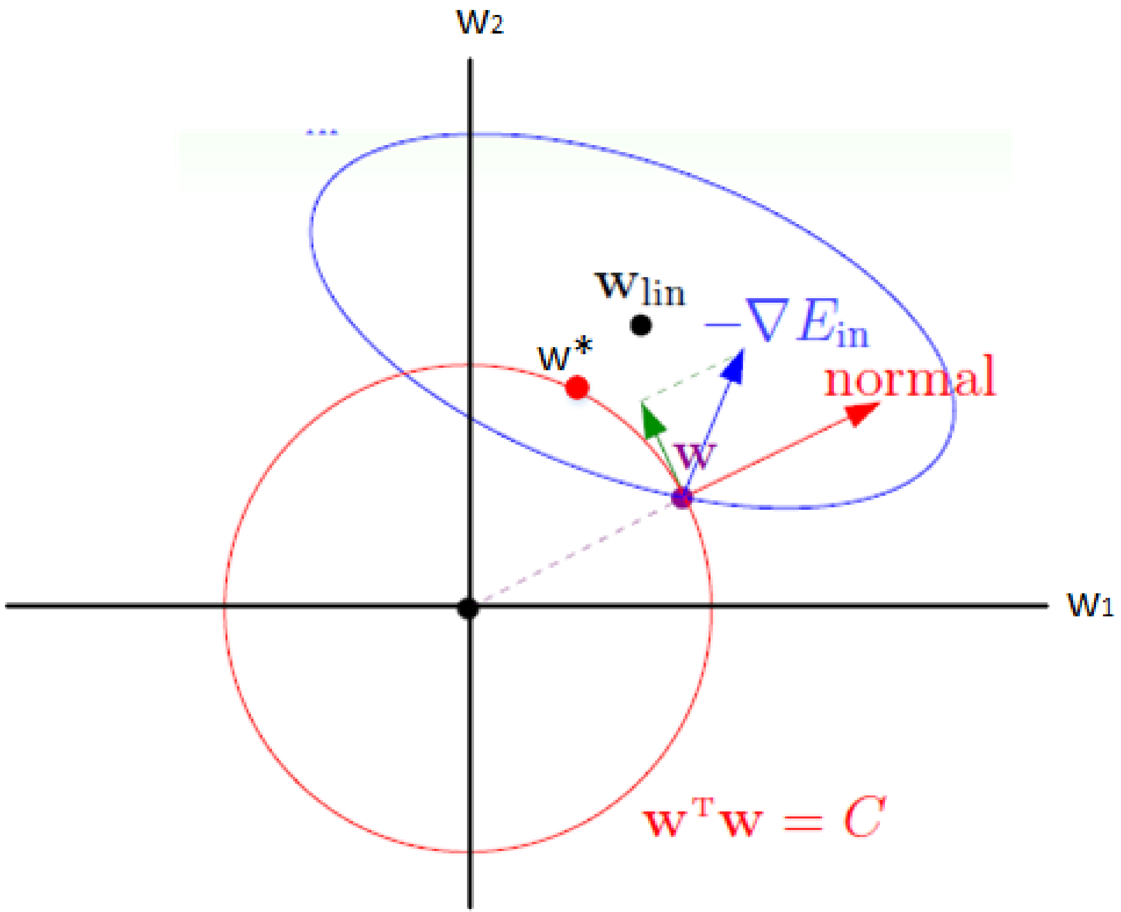
$$L(w,b)=\sum_{i=1}^{n}\left(y_i-\sum_{i=1}^{d} w_j x_{i,j}-b\right)^2+\boxed{\lambda\sum_{j=1}^{d}\left|w_j\right|^2}$$

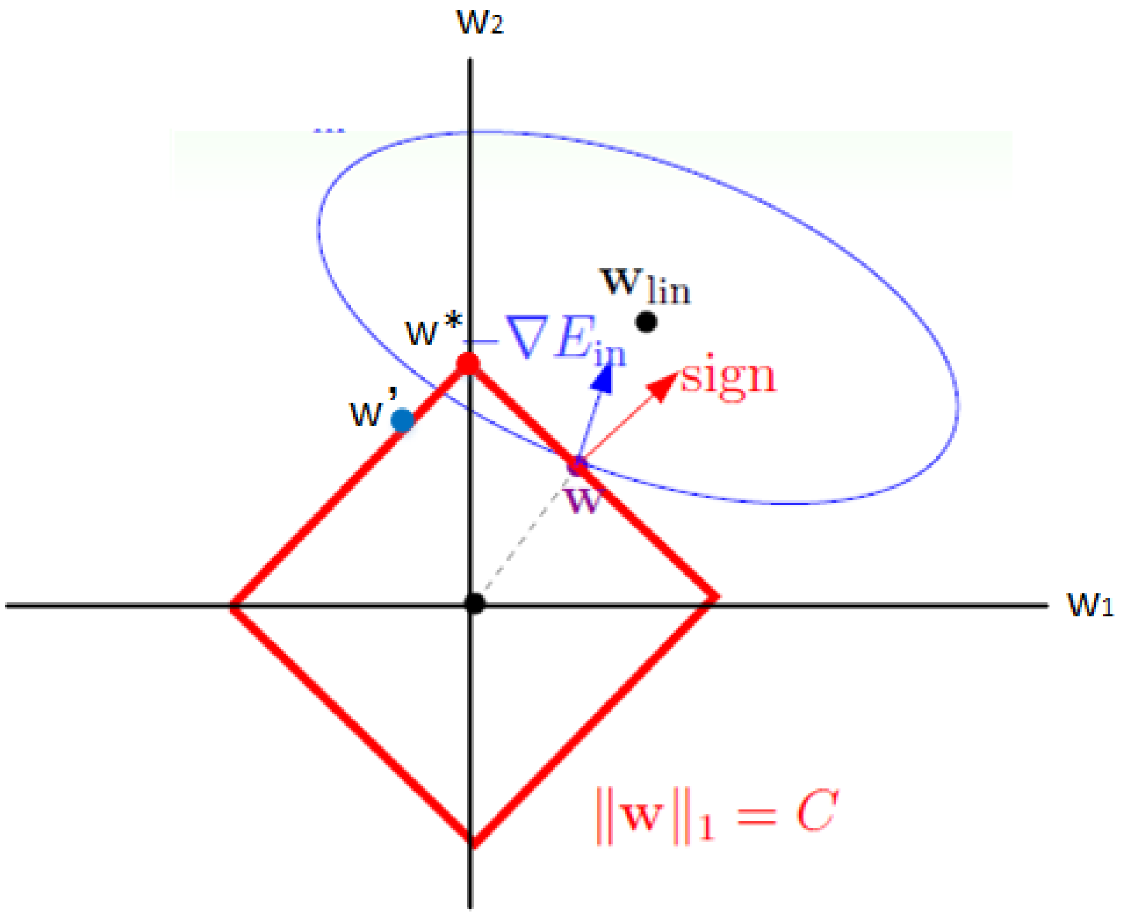
正则化

- 对于模型： $y = w_1 \cdot x_t + w_2 \cdot x_t^2 + w_3 \cdot x_t^3 + w_4 \cdot x_h + w_5 \cdot x_w + w_6 \cdot x_p + b$
- 损失函数中加入 L2 正则化项
  - λ = 0 时:
    - ✓ 训练集均方误差：1.553
    - ✓ 测试集均方误差：2.278
  - λ = 100 时:
    - ✓ 训练集均方误差：1.627
    - ✓ 测试集均方误差：1.932



L1、L2正则化方法的特点分析





基于最小二乘法的线性回归模型

基于最小二乘（Least Square）法的线性回归模型

- 将数据样本用 $n \times (d+1)$ 大小的矩阵 $\boldsymbol{X}$ 表示，每一行是一个样本，每一列是样本的某一个特征，得到

$$\boldsymbol{x_i} = (x_{i,1}; x_{i,2}; \dots; x_{i,d})$$

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ 1 & x_{2,1} & \dots & x_{2,d} \\ 1 & \dots & \dots & \dots \\ 1 & x_{n,1} & \dots & x_{n,d} \end{pmatrix}$$

$$f(\boldsymbol{x}) = \boldsymbol{wx} + b$$

- 其中第一列恒置为1，是为了在进行向量乘法与常数项 b 对应

损失函数的向量形式

$$f(\boldsymbol{x}) = \boldsymbol{w}\boldsymbol{x} + b$$

- 数据集的目标值序列也可写成向量形式 $\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}$
- 为了简洁起见，将待求解参数$\boldsymbol{w}$和b合并为$\boldsymbol{W} = \begin{pmatrix} b \\ \boldsymbol{w} \end{pmatrix}$
- 于是，线性回归的损失函数可重写如下：

$$L(\boldsymbol{W}) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{f}(\boldsymbol{x}_i) - y_i)^2 = \frac{1}{2}(\boldsymbol{X}\boldsymbol{W} - \boldsymbol{y})^T(\boldsymbol{X}\boldsymbol{W} - \boldsymbol{y})$$

最小二乘法的算法流程

- 我们的目标是寻找一组最优的参数$\boldsymbol{W}^* = \begin{pmatrix} b^* \\ \boldsymbol{w}^* \end{pmatrix}$，能够最小化损失函数：

$$\underset{\boldsymbol{W}}{\operatorname{argmin}}\, L(\boldsymbol{W}) \qquad L(\boldsymbol{W}) = \frac{1}{2}(\boldsymbol{XW} - \boldsymbol{y})^T(\boldsymbol{XW} - \boldsymbol{y})$$

- 当$X^T X$可逆时，对其求导并求驻点可以直接求得$\boldsymbol{W}$的最优解：

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{W}} &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{W}}(\boldsymbol{W}^T X^T X\boldsymbol{W} - \boldsymbol{W}^T X^T \boldsymbol{y} - \boldsymbol{y}^T X\boldsymbol{W} + \boldsymbol{y}^T \boldsymbol{y}) \\ &= \frac{1}{2}(2X^T X\boldsymbol{W} - X^T\boldsymbol{y} - X^T\boldsymbol{y}) \\ &= X^T X\boldsymbol{W} - X^T\boldsymbol{y}\end{aligned}$$

向量求导规则可以参考:
https://www.cnblogs.com/pinard/p/10773942.html

最小二乘法的算法流程

- 之前得到

$$\frac{\partial L}{\partial \boldsymbol{W}} = \frac{1}{2}\frac{\partial}{\partial \boldsymbol{W}}(\boldsymbol{W}^T X^T X \boldsymbol{W} - \boldsymbol{W}^T X^T \boldsymbol{y} - \boldsymbol{y}^T X \boldsymbol{W} + \boldsymbol{y}^T \boldsymbol{y})$$
$$= \frac{1}{2}(2X^T X \boldsymbol{W} - X^T \boldsymbol{y} - X^T \boldsymbol{y})$$
$$= X^T X \boldsymbol{W} - X^T \boldsymbol{y}$$

- 令

$$X^T X \boldsymbol{W} - X^T \boldsymbol{y} = 0$$

可以得到$\boldsymbol{W}$的解析解：

$$\boldsymbol{W} = (X^T X)^{-1} X^T \boldsymbol{y}$$

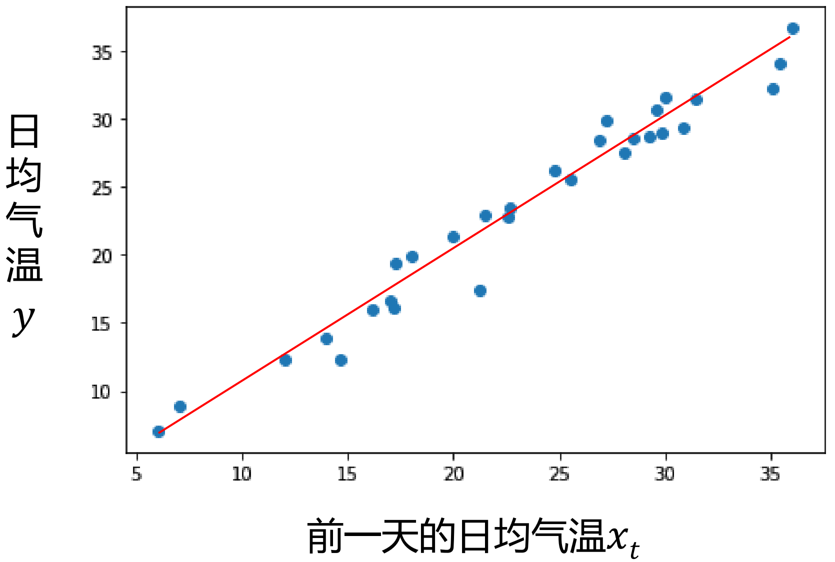
最小二乘法的应用示例

- 单变量线性回归模型： $y = w \cdot x_t + b$
- 数据表示为：

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{1,t} \\ 1 & x_{2,t} \\ \cdots & \cdots \\ 1 & x_{30,t} \end{pmatrix} \qquad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_{30} \end{pmatrix} \qquad \boldsymbol{W} = \begin{pmatrix} b \\ w \end{pmatrix}$$

- 可得

$$\boldsymbol{W} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \begin{pmatrix} 0.980 \\ 0.965 \end{pmatrix}$$



最小二乘法的问题

- 当这n个自变量不是互相独立，而是存在着一些线性关系时，此时 $X^T X$不可逆，此时得到的解为病态解，不能作为学习到的最优参数

- 当$X^T X$不可逆时，可以采用梯度下降（Gradient Descent）方法求解