



中国石油大学 (华东)
CHINA UNIVERSITY OF PETROLEUM

深度学习 Deep Learning

张琛

2024/2/28



青 岛 软 件 学 院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

机器学习基础



■ 机器学习有下面几种定义：

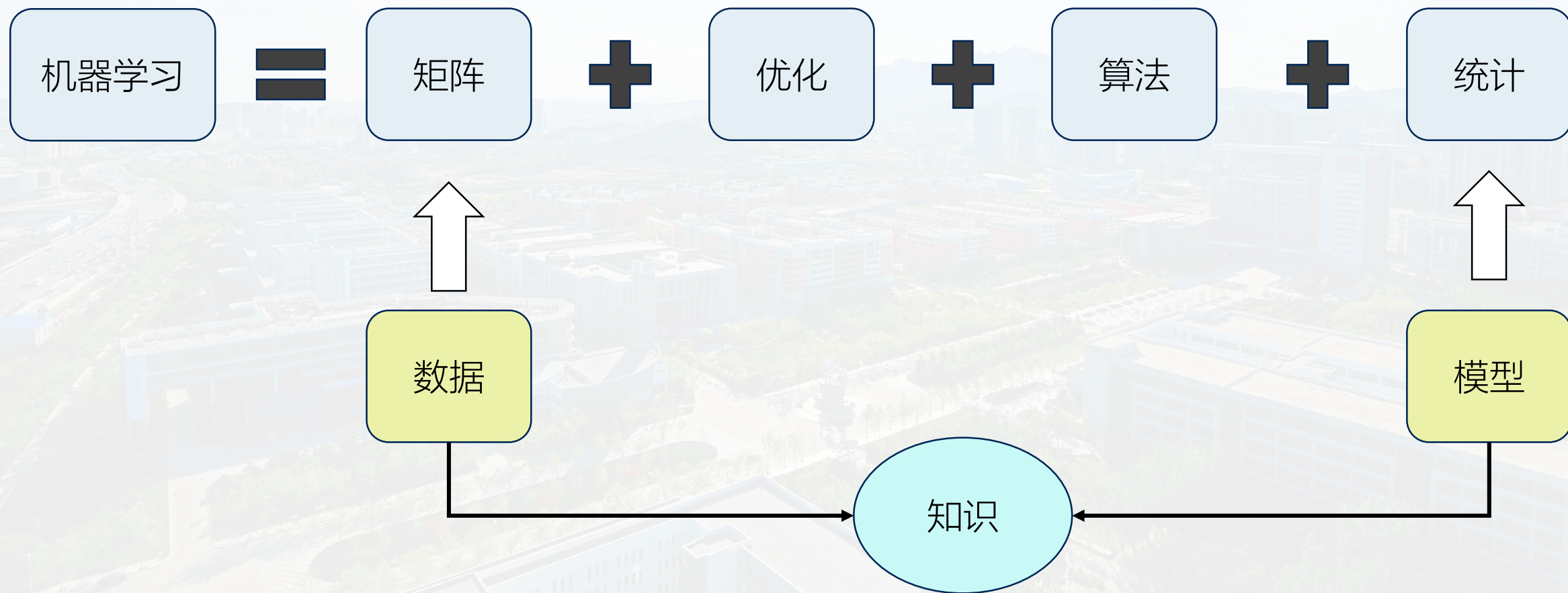
- “机器学习是一门人工智能的科学，该领域的主要研究对象是人工智能，特别是如何在经验学习中改善具体算法的性能”。
- “机器学习是对能通过经验自动改进的计算机算法的研究”。
- Michael Jordan (CMU): “一个连接计算和统计的领域，与信息论、信号处理、算法、控制理论和优化理论有联系。”
- 英文定义(Mitchell): A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .

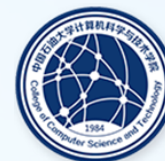


机器学习



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY





■ 机器学习的对象

- **data** : 计算机及互联网上的各种数字、文字、图像、视频、音频数据以及它们的组合。
- 数据的基本假设是同类数据具有一定的统计规律性。

■ 机器学习的目的

- 用于对数据（特别是未知数据）进行预测和分析。



机器学习



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

□ 机器学习的研究：

- 机器学习方法
- 机器学习理论（机器学习方法的有效性和效率和基本理论）
- 机器学习应用



■ 机器学习的方法

□ 分类：

- Supervised learning 监督学习
- Unsupervised learning 无监督学习
- Semi-supervised learning 半监督学习
- Reinforcement learning 强化学习

□ 监督学习：

- 训练数据 training data
- 模型 model ----- 假设空间 hypothesis
- 评价准则 evaluation criterion ----- 策略 strategy
- 算法 algorithm



- Instance (实例), feature vector (特征向量), feature space (特征空间)
- 输入实例 x 的特征向量:

$$x = [x^{(1)}, x^{(2)}, \dots, x^{(i)}, \dots, x^{(n)}]^T$$

- $x^{(i)}$ 与 x_i 不同,后者表示多个输入变量中的第 i 个

$$x_i = [x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)}]^T$$

- 训练集:

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

- 输入变量和输出变量:
 - 分类问题、回归问题、标注问题



■ 联合概率分布

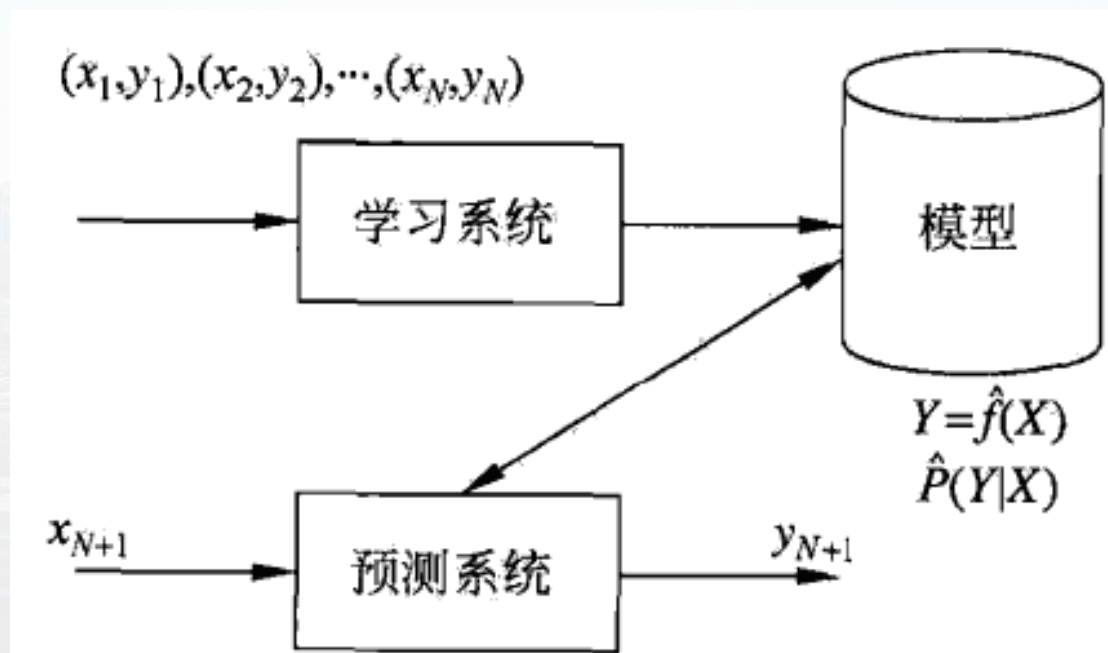
- 假设输入与输出的随机变量 X 和 Y 遵循联合概率分布 $P(X, Y)$
- $P(X, Y)$ 为分布函数或分布密度函数
- 对于学习系统来说，联合概率分布是未知的
- 训练数据和测试数据被看作是依联合概率分布 $P(X, Y)$ 独立同分布产生的。

■ 假设空间

- 监督学习目的是学习一个由输入到输出的映射，称为模型
- 模型的集合就是假设空间 (hypothesis space)
- 概率模型:条件概率分布 $P(Y|X)$, 决策函数: $Y = f(X)$



问题的形式化



$$y_{N+1} = \arg \max_{y_{N+1}} \hat{P}(y_{N+1} | x_{N+1})$$

$$y_{N+1} = \hat{f}(x_{N+1})$$



例子：回归

$$x = [x^{(1)}, x^{(2)}, \dots, x^{(i)}, \dots, x^{(n)}]^T$$

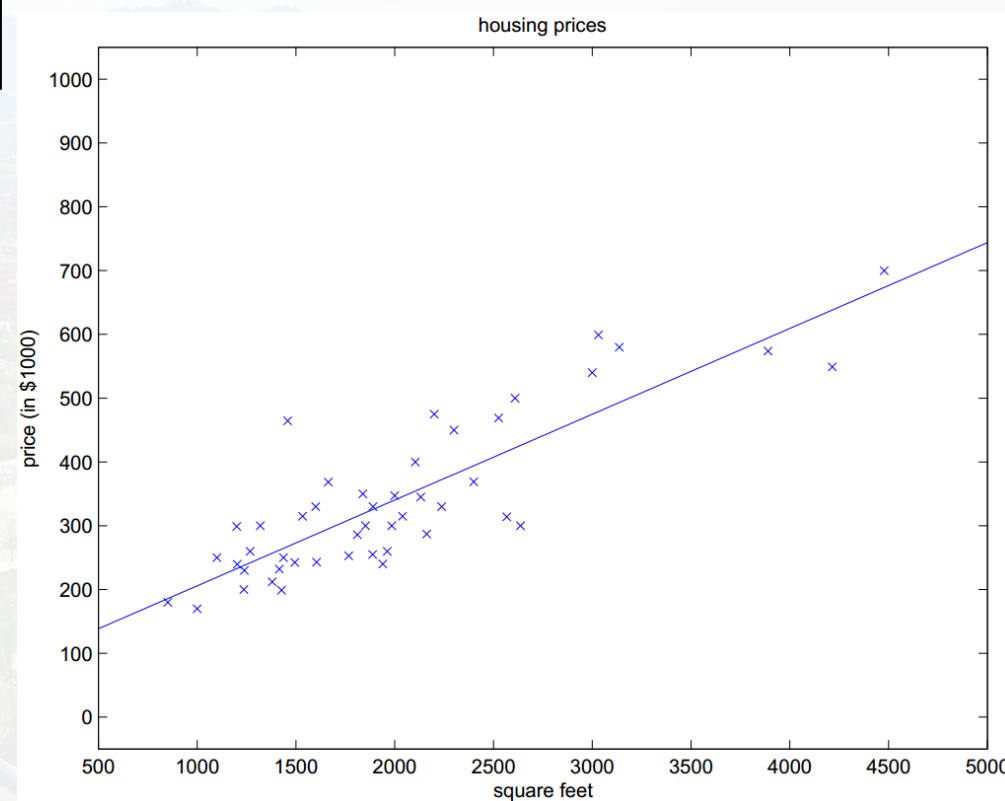
$$x_i = [x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)}]^T$$

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

$$X_{N \times (n+1)} = \begin{bmatrix} -x_1^T & - \\ \vdots & \\ -x_n^T & - \end{bmatrix}$$

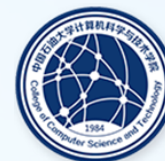
$$\begin{aligned} h_{\theta}(x) &= \theta_0 + \theta_1 x^{(1)} + \dots + \theta_n x^{(n)} \\ &= \theta^T x \end{aligned}$$

$$\begin{aligned} J(\theta) &= \frac{1}{2} \sum_{i=1}^N (h_{\theta}(x_i) - y_i)^2 \\ &= \frac{1}{2} (Y - X\theta)^T (Y - X\theta) \end{aligned}$$





例子：回归



$$X_{N \times (n+1)} = \begin{bmatrix} -\mathbf{x}_1^T & - \\ \vdots & \\ -\mathbf{x}_n^T & - \end{bmatrix}$$

$$\begin{aligned} J(\theta) &= \frac{1}{2} \sum_1^N (h_{\theta}(x_i) - y_i)^2 \\ &= \frac{1}{2} (\mathbf{X}\Theta - \mathbf{Y})^T (\mathbf{X}\Theta - \mathbf{Y}) \end{aligned}$$

$$\text{令 } \mathbf{b} = (\mathbf{X}\Theta - \mathbf{Y}) \in \mathbb{R}^{N \times 1}$$

$$\mathbf{c} = \mathbf{X}\Theta \in \mathbb{R}^{N \times 1}$$

$$\frac{\partial J(\theta)}{\partial \Theta} = \frac{\partial J(\theta)}{\partial \mathbf{b}} \frac{\partial \mathbf{b}}{\partial \mathbf{c}} \cdot \frac{\partial \mathbf{c}}{\partial \Theta}$$

$$= \frac{1}{2} \cdot \frac{\partial \mathbf{b}^2}{\partial \mathbf{b}} \frac{\partial \mathbf{b}}{\partial \mathbf{c}} \cdot \frac{\partial \mathbf{c}}{\partial \Theta}$$

$$= \mathbf{b}^T \mathbf{I}_{N \times N} \mathbf{X} = (\mathbf{X}\Theta - \mathbf{Y})^T \mathbf{X} = \mathbf{0}$$



$$((\mathbf{X}\Theta - \mathbf{Y})^T \mathbf{X})^T = \mathbf{X}^T (\mathbf{X}\Theta - \mathbf{Y}) = \mathbf{0}$$

$$\Theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$



正规方程



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

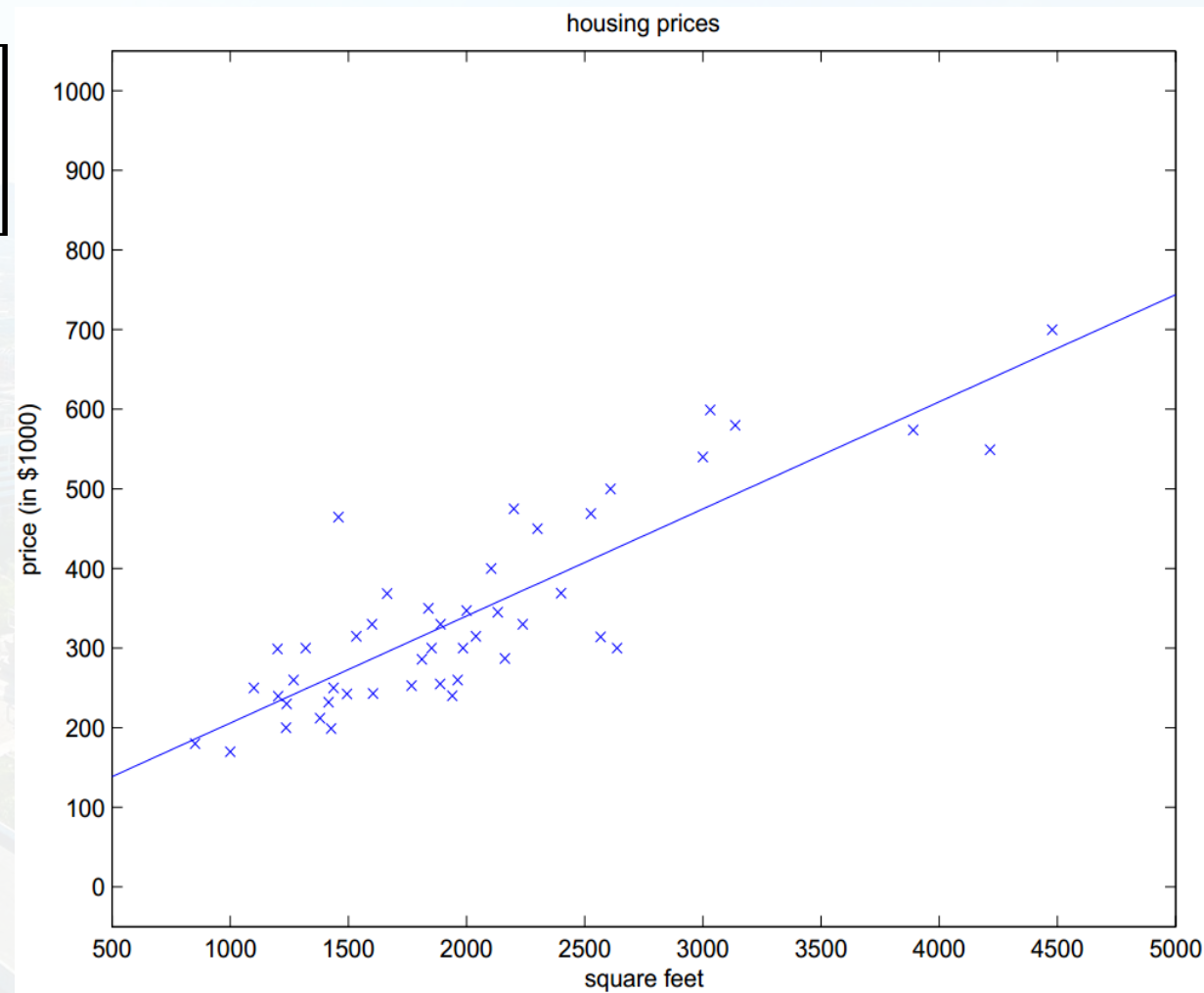
$$x = [x^{(1)}, x^{(2)}, \dots, x^{(i)}, \dots, x^{(n)}]^T$$
$$x_i = [x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)}]^T$$
$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$
$$X_{N \times (n+1)} = \begin{bmatrix} -x_1^T & - \\ \vdots & \\ -x_n^T & - \end{bmatrix}$$

$$h_{\theta}(x) = \theta^T x$$

$$J(\theta) = \frac{1}{2} (Y - X\theta)^T (Y - X\theta)$$

$$X^T X \theta = X^T Y \quad (1)$$

$$\theta = (X^T X)^{-1} X^T Y$$





梯度下降法



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

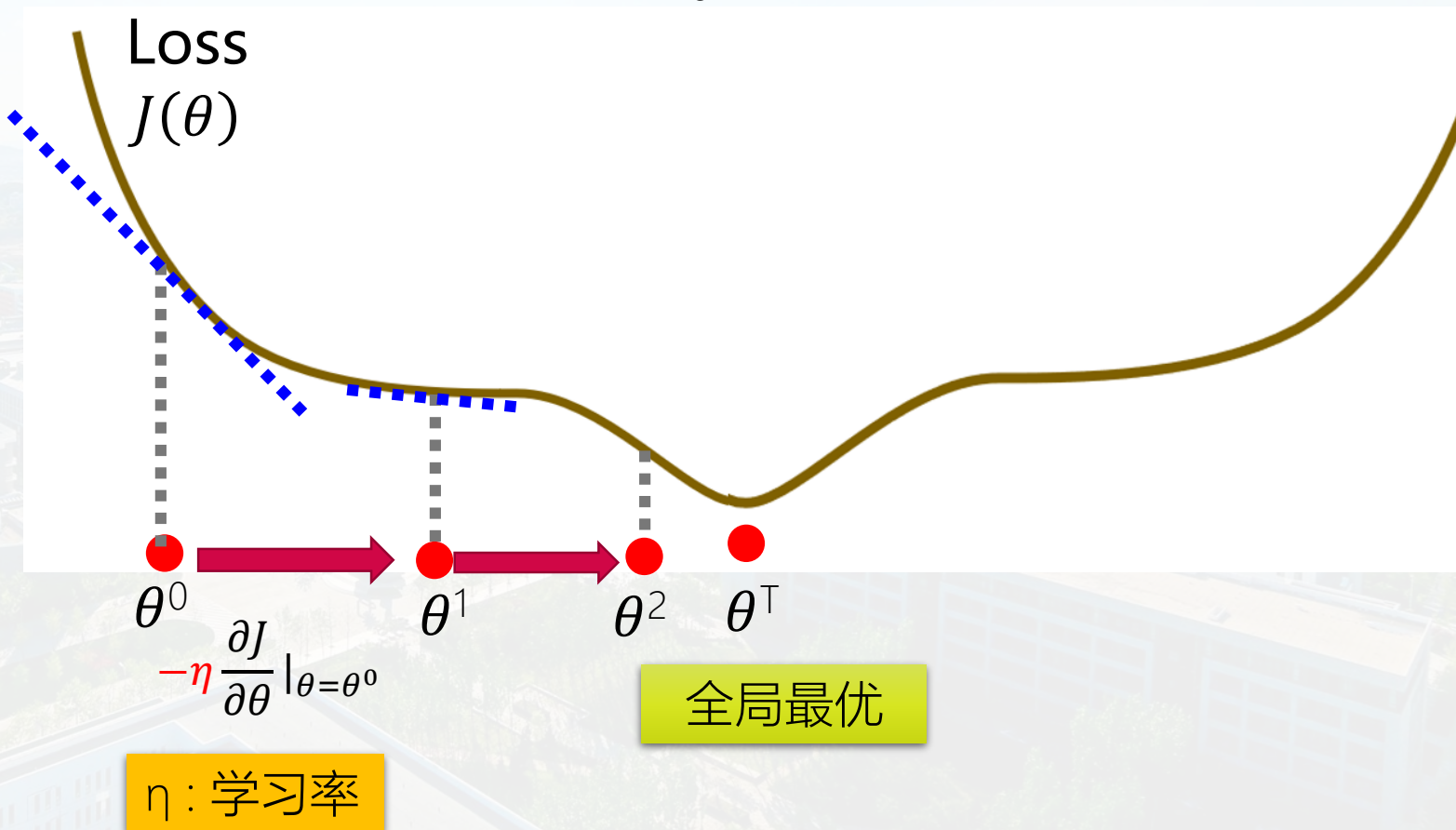
$$\theta^* = \arg \min_{\theta} J(\theta)$$

$X^T X$ 不可逆或维数特别高

$$J(\theta) = \frac{1}{2} \sum_{i=1}^N (h_{\theta}(x_i) - y_i)^2$$

$$\frac{\partial J}{\partial \theta} = \sum_{i=1}^N (h_{\theta}(x_i) - y_i) x_i$$

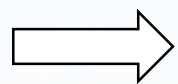
$$\theta := \theta - \eta \sum_{i=1}^N (h_{\theta}(x_i) - y_i) x_i$$





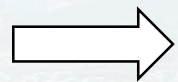
正则化-岭回归

$X^T X$ 不可逆



$X^T X + \lambda I$ 可逆

$$\theta = (X^T X)^{-1} X^T Y$$



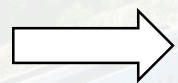
$$\theta = (X^T X + \lambda I)^{-1} X^T Y$$

$$X^T X \theta = X^T Y$$

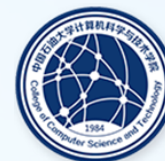


$$X^T X \theta + \lambda \theta = X^T Y$$

$$J(\theta) = \frac{1}{2} (Y - X\theta)^T (Y - X\theta)$$



$$J(\theta) = \frac{1}{2} (Y - X\theta)^T (Y - X\theta) + \frac{\lambda}{2} \theta^T \theta$$



$$y = \theta^T x + \epsilon$$

$$p(\epsilon) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$$

$$p(y|x; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \theta^T x)^2}{2\sigma^2}\right)$$

$$\log L(\theta) = N \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^N (y - \theta^T x)^2$$



将参数 θ 视为随机变量

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} = \frac{p(D|\theta)p(\theta)}{\int p(D|\theta)p(\theta) d\theta}$$

高斯先验

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

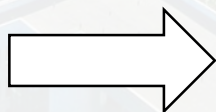
$$\theta \sim \mathcal{N}(0, \frac{1}{\alpha} I)$$

$$p(\theta|\alpha) = \frac{1}{(\frac{2\pi}{\alpha})^{(n+1)/2}} \exp\left(-\frac{1}{2/\alpha} \theta^T \theta\right)$$

$$\beta = (\sigma_y^2)^{-1}$$

$$p(y|x, \theta, \beta) \sim \mathcal{N}(\theta^T x, \beta^{-1})$$

$$p(\theta|D) \propto p(y|x, \theta, \beta) \cdot p(\theta|\alpha)$$



$$J(\theta) = \frac{1}{2} (Y - X\theta)^T (Y - X\theta) + \frac{\lambda}{2} \theta^T \theta$$

$$\lambda = \alpha/\beta$$



将参数 θ 视为随机变量

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} = \frac{p(D|\theta)p(\theta)}{\int p(D|\theta)p(\theta) d\theta}$$

拉普拉斯先验

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$\theta \sim \text{Laplace}(0, t) \quad p(\theta|t) = \frac{1}{2t} \exp\left(-\frac{1}{t}|\theta|\right)$$

$$\beta = (\sigma_y^2)^{-1} \quad p(y|x, \theta, \beta) \sim \mathcal{N}(\theta^T x, \beta^{-1})$$

$$p(\theta|D) \propto p(y|x, \theta, \beta) \cdot p(\theta|\alpha) \quad \Rightarrow \quad J(\theta) = \frac{1}{2} (Y - X\theta)^T (Y - X\theta) + \frac{\lambda}{2} |\theta|_1$$



无监督学习



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 训练集:

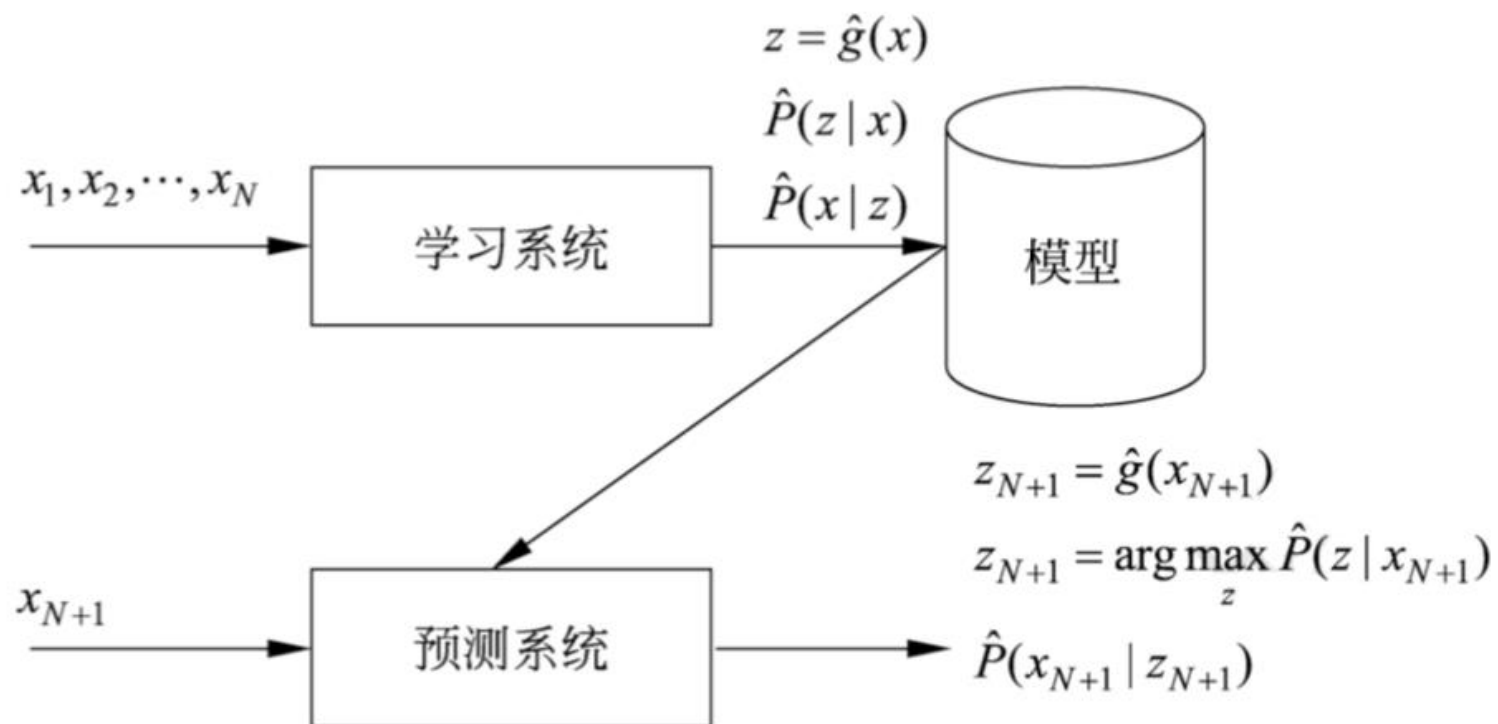
$$U = \{x_1, x_2, \dots, x_N\}$$

■ 模型函数:

$$z = g(x)$$

■ 条件概率分布:

$$P(z|x)$$





无监督学习



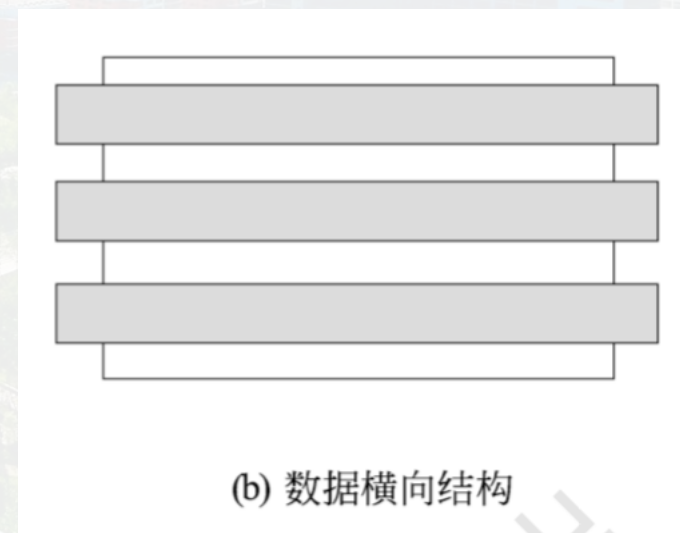
青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 使用无标注数据 $U = \{x_1, x_2, \dots, x_N\}$ 学习或训练，由特征向量组成
- 无监督学习的模型是函数 $z = g_\theta(x)$ ，条件概率分布 $P_\theta(z|x)$ ，或条件概率分布 $P_\theta(x|z)$ 。
- 假设训练数据集由 N 个样本组成，每个样本是一个 M 维向量。训练数据可以由一个矩阵表示，每一列对应一个特征，每一行对应一个样本

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NM} \end{bmatrix}$$

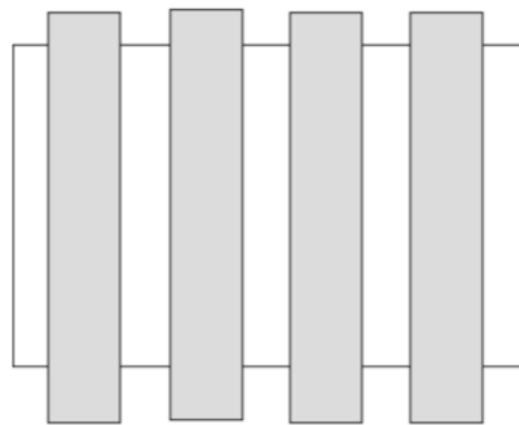


- 无监督学习的基本想法是对给定数据（矩阵数据）进行某种“压缩”，从而找到数据的潜在结构。假定损失最小的压缩得到的结果就是最本质的结构。
- 考虑发掘数据的横向结构，把相似的样本聚到同类，即对数据进行聚类





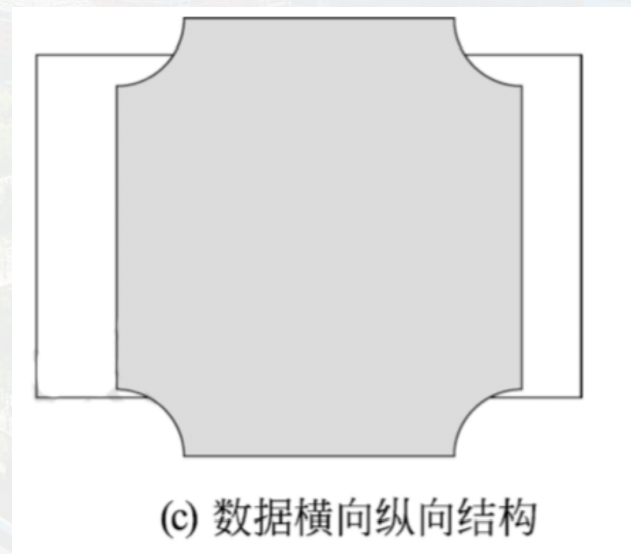
- 无监督学习的基本想法是对给定数据（矩阵数据）进行某种“压缩”，从而找到数据的潜在结构。假定损失最小的压缩得到的结果就是最本质的结构。
- 考虑发掘数据的纵向结构，把高维空间的向量转换为低维空间的向量，即对数据进行降维。



(a) 数据纵向结构



- 无监督学习的基本想法是对给定数据（矩阵数据）进行某种“压缩”，从而找到数据的潜在结构。假定损失最小的压缩得到的结果就是最本质的结构。
- 同时考虑发掘数据的纵向与横向结构，假设数据由含有隐式结构的概率模型生成得到，从数据中学习该概率模型。





- 硬聚类时，每一个样本属于某一类

$$z_i = g_{\theta}(x_i), i = 1, 2, \dots, N$$

- 软聚类时，每一个样本依概率属于每一个类

$$P_{\theta}(z_i|x_i), i = 1, 2, \dots, N$$



降维



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 降维 (dimensionality reduction) 是将训练数据中的样本 (实例) 从高维空间转换到低维空间。
- 假设样本原本存在于低维空间, 或者近似地存在于低维空间, 通过降维则可以更好地表示样本数据的结构, 即更好地表示样本之间的关系。
- 高维空间通常是高维的欧氏空间, 而低维空间是低维的欧氏空间或者流形 (manifold)。
- 从高维到低维的降维中, 要保证样本中的信息损失最小。

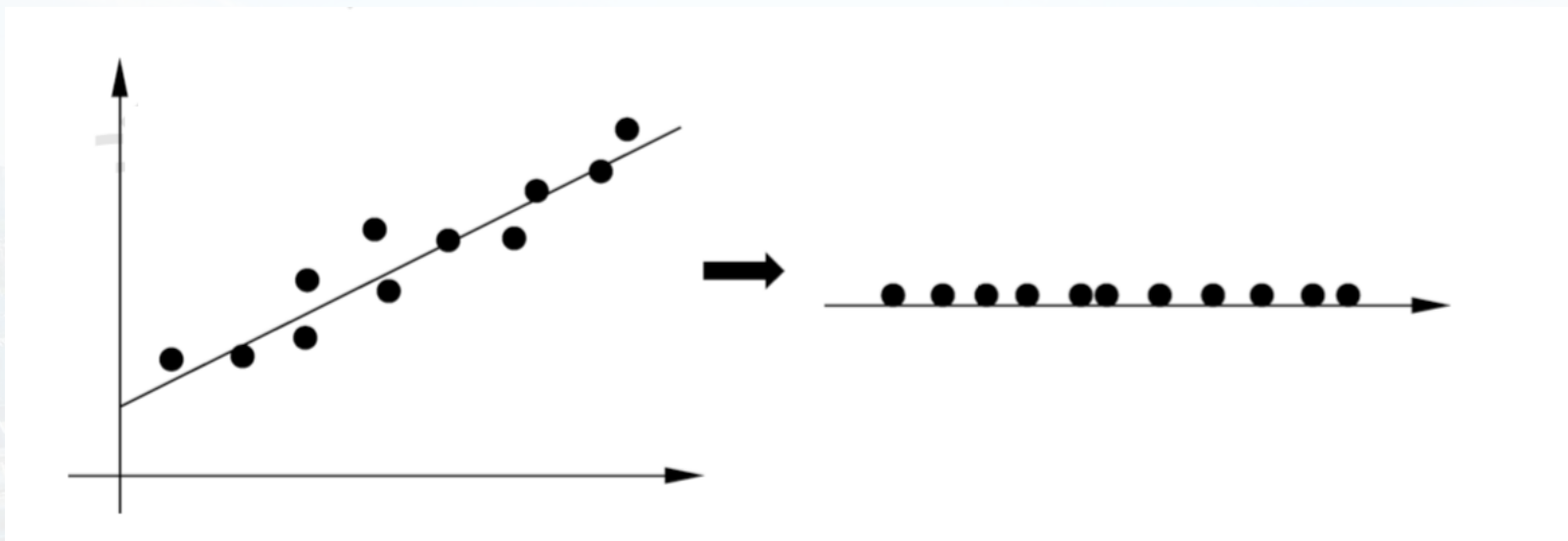


降维



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 降维有线性的降维和非线性的降维。



- 二维空间的样本存在于一条直线的附近，可以将样本从二维空间转换到一维空间。通过降维可以更好地表示样本之间的关系。



降维



- 假设输入空间是欧氏空间 $X \subseteq \mathbb{R}^d$ ，输出空间也是欧氏空间，
 $Z \subseteq \mathbb{R}^{d'}$, $d' \ll d$ ，后者的维数低于前者的维数。降维的模型是函数
$$z = g_{\theta}(x)$$
- 其中 $x \in X$ 是样本的高维向量， $z \in Z$ 是样本的低维向量， θ 是参数。
函数可以是线性函数也可以是非线性函数。
- 降维的过程就是学习降维模型的过程。降维时，每一个样本从高维向量转换为低维向量 $z_i = g_{\theta}(x_i), i = 1, 2, \dots, N$



概率模型估计



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 假设训练数据由一个概率模型生成，由训练数据学习概率模型的结构和参数。
- 概率模型的结构类型，或者说概率模型的集合事先给定，而模型的具体结构与参数从数据中自动学习。学习的目标是找到最有可能生成数据的结构和参数。
- 概率模型包括混合模型、概率图模型等。
- 概率图模型又包括有向图模型和无向图模型。

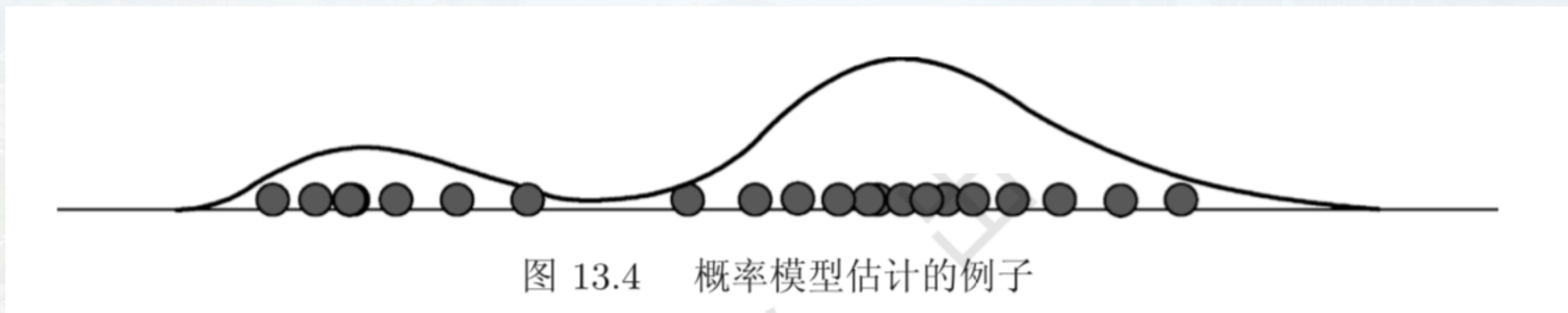


图 13.4 概率模型估计的例子

- 假设数据由高斯混合模型生成，学习的目标是估计这个模型的参数。



概率模型估计



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 概率模型表示为条件概率分布 $P_{\theta}(x|z)$
- 随机变量 x 表示观测数据，可以是连续变量也可以是离散变量
- 随机变量 z 表示隐式结构，是离散变量
- 随机变量 θ 表示参数
- 模型是混合模型时， z 表示成分的个数
- 模型是概率图模型时， z 表示图的结构



概率模型估计



青 岛 软 件 学 院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 概率模型的一种特殊情况是隐式结构不存在，即满足

$$P_{\theta}(x|z) = P_{\theta}(x)$$

- 这时条件概率分布估计变成概率分布估计，只要估计分布 $P_{\theta}(x)$ 的参数即可。
- 传统统计学概率密度估计：高斯分布参数估计。



概率模型估计

- 概率模型估计是从给定的训练数据 $U = \{x_1, x_2, \dots, x_N\}$ 中学习模型 $P_\theta(x|z)$ 的结构和参数，计算出模型相关的任意边缘分布和条件分布。
- 注意随机变量 x 是多元变量，甚至是高维多元变量
- 软聚类也可以看作是概率模型估计问题。根据贝叶斯公式

$$P(z|x) = \frac{P(z)P(x|z)}{P(x)} \propto P(z)P(x|z)$$

- 假设先验概率服从均匀分布，只需要估计条件概率分布 $P_\theta(x|z)$ 。
这样，可以通过对条件概率分布 $P_\theta(x|z)$ 的估计进行软聚类

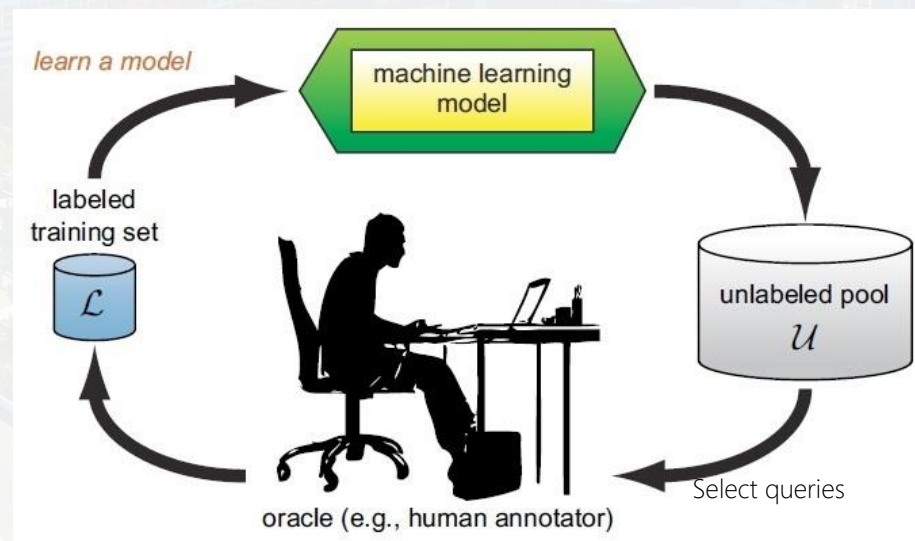


半监督学习

- 少量标注数据，大量未标注数据
- 利用未标注数据的信息，辅助标注数据，进行监督学习
- 较低成本

主动学习

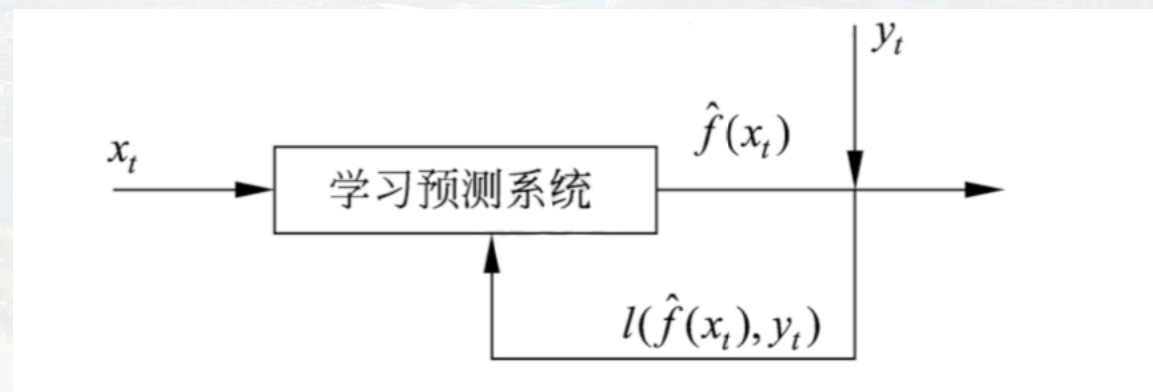
- 机器主动给出实例，教师进行标注
- 利用标注数据学习预测模型





■ 按算法分类：

□ 在线学习 (online learning)



□ 批量学习 (batch learning)



■ 按技巧分类：

□ 贝叶斯学习 (Bayesian learning)

模型估计时，估计整个后验概率分布 $P(\theta|D)$ 。如果需要给出一个模型，通常取后验概率最大的模型。

预测时，计算数据对后验概率分布的期望值：

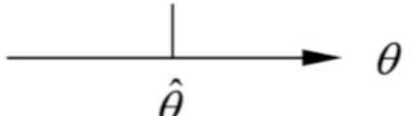
$$P(x|D) = \int P(x|\theta, D)P(\theta|D)d\theta$$

这里 x 是新样本。

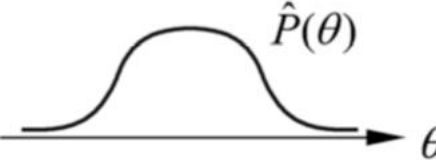


- 按技巧分类：
 - 贝叶斯学习 (Bayesian learning)

极大似然估计

$$D \longrightarrow \hat{\theta} = \arg \max_{\theta} P(D | \theta)$$


贝叶斯估计

$$D \longrightarrow \hat{P}(\theta | D) = \frac{P(\theta)P(D | \theta)}{P(D)}$$




■ 按技巧分类：

□ 核方法 (Kernel method)

- 使用核函数表示和学习非线性模型，将线性模型学习方法扩展到非线性模型的学习
- 不显式地定义输入空间到特征空间的映射，而是直接定义核函数，即映射之后在特征空间的内积
- 假设 x_1, x_2 是输入空间的任意两个实例，内积为 $\langle x_1, x_2 \rangle$ ，输入空间到特征空间的映射为 φ ，核方法在输入空间中定义核函数 $K(x_1, x_2)$ ，使其满足 $K(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle$



机器学习三要素



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

方法=模型+策略+算法

■ 模型:

□ 决策函数的集合: $\mathcal{F} = \{f|Y = f(X)\}$

□ 参数空间 $\mathcal{F} = \{f|Y = f_{\theta}(X), \theta \in R^n\}$

□ 条件概率的集合: $\mathcal{F} = \{P|P(Y|X)\}$

□ 参数空间 $\mathcal{F} = \{P|P_{\theta}(Y|X), \theta \in R^n\}$



■ 策略

- ❑ 损失函数：预测错误的程度
- ❑ 风险函数：平均意义下模型预测的好坏
- ❑ 0-1损失函数 0-1 loss function

$$L(Y, f(X)) = \begin{cases} 1, Y \neq f(X) \\ 0, Y = f(X) \end{cases}$$

- ❑ 平方损失函数 quadratic loss function

$$L(Y, f(X)) = (Y - f(X))^2$$

- ❑ 绝对损失函数 absolute loss function

$$L(Y, f(X)) = |Y - f(X)|$$



■ 策略

- 对数损失函数 logarithmic loss function 或对数似然损失函数 loglikelihood loss function

$$L(Y, P(Y|X)) = -\log P(Y|X)$$

- 损失函数的期望

$$R_{exp}(f) = E_P[L(Y, f(X))] = \int_{x \times y} L(Y, f(x)) P(x, y) dx dy$$

- 风险函数 risk function 期望损失 expected loss
- 由 $P(x, y)$ 可以直接求出 $P(x|y)$,但 $P(x, y)$ 是未知的。

给定训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$

- 经验风险 empirical risk , 经验损失 empirical loss $R_{emp}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i))$



机器学习三要素



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 策略：经验风险最小化与结构风险最小化

□ 经验风险最小化最优模型

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i))$$

- 当样本容量很小时，经验风险最小化学习的效果未必很好，会产生“过拟合over-fitting”
- 结构风险最小化 structure risk minimization，为防止过拟合提出的策略，等价于正则化（regularization），加入正则化项regularizer，或罚项 penalty term：

$$R_{srm}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f)$$



机器学习三要素



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 求最优模型就是求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f)$$



机器学习三要素



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 算法：

- 求解最优模型
- 如果最优化问题有显式的解析式，算法比较简单
- 但通常解析式不存在，就需要数值计算的方法



模型评估与模型选择



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 训练误差，训练数据集的平均损失

$$R_{emp}(\hat{f}) = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}(x_i))$$

- 测试误差，测试数据集的平均损失

$$e_{test} = \frac{1}{N'} \sum_{i=1}^{N'} L(y_i, \hat{f}(x_i))$$

- 测试误差：损失函数是0-1 损失时：

$$e_{test} = \frac{1}{N'} \sum_{i=1}^{N'} I(y_i \neq \hat{f}(x_i))$$

- 测试数据集的准确率：

$$r_{test} = \frac{1}{N'} \sum_{i=1}^{N'} I(y_i = \hat{f}(x_i))$$



模型评估与模型选择



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 过拟合与模型选择：以多项式函数拟合为例
- 假设给定训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$
- $f_M(x, w) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j$
- 经验风险最小：

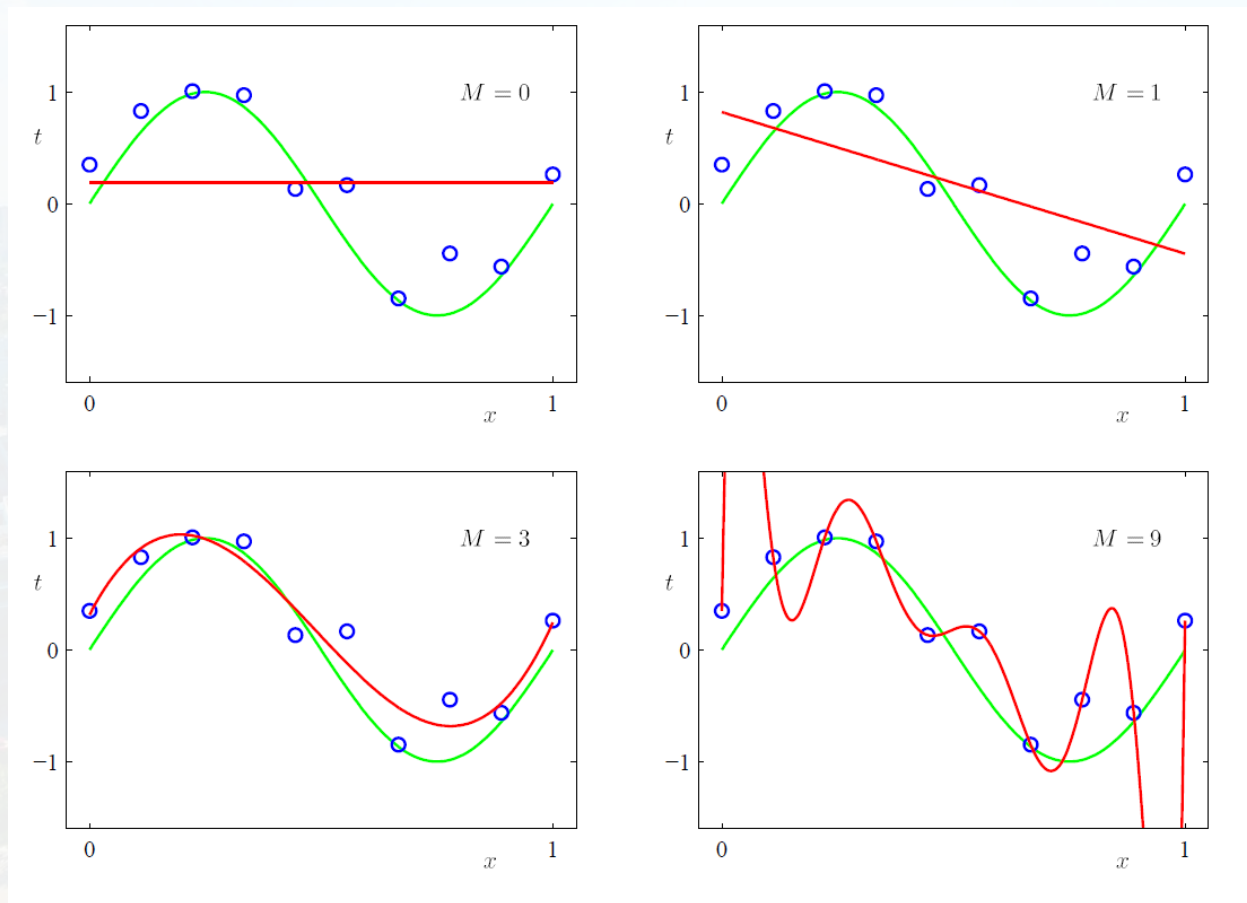
$$L(w) = \frac{1}{2} \sum_{i=1}^N (f(x_i, w) - y_i)^2 \quad L(w) = \frac{1}{2} \sum_{i=1}^N \left(\sum_{j=0}^M w_j x_i^j - y_i \right)^2 \quad w_j = \frac{\sum_{i=1}^N x_i y_i}{\sum_{i=1}^N x_i^{j+1}}, j = 0, 1, 2, \dots, M$$



模型评估与模型选择



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

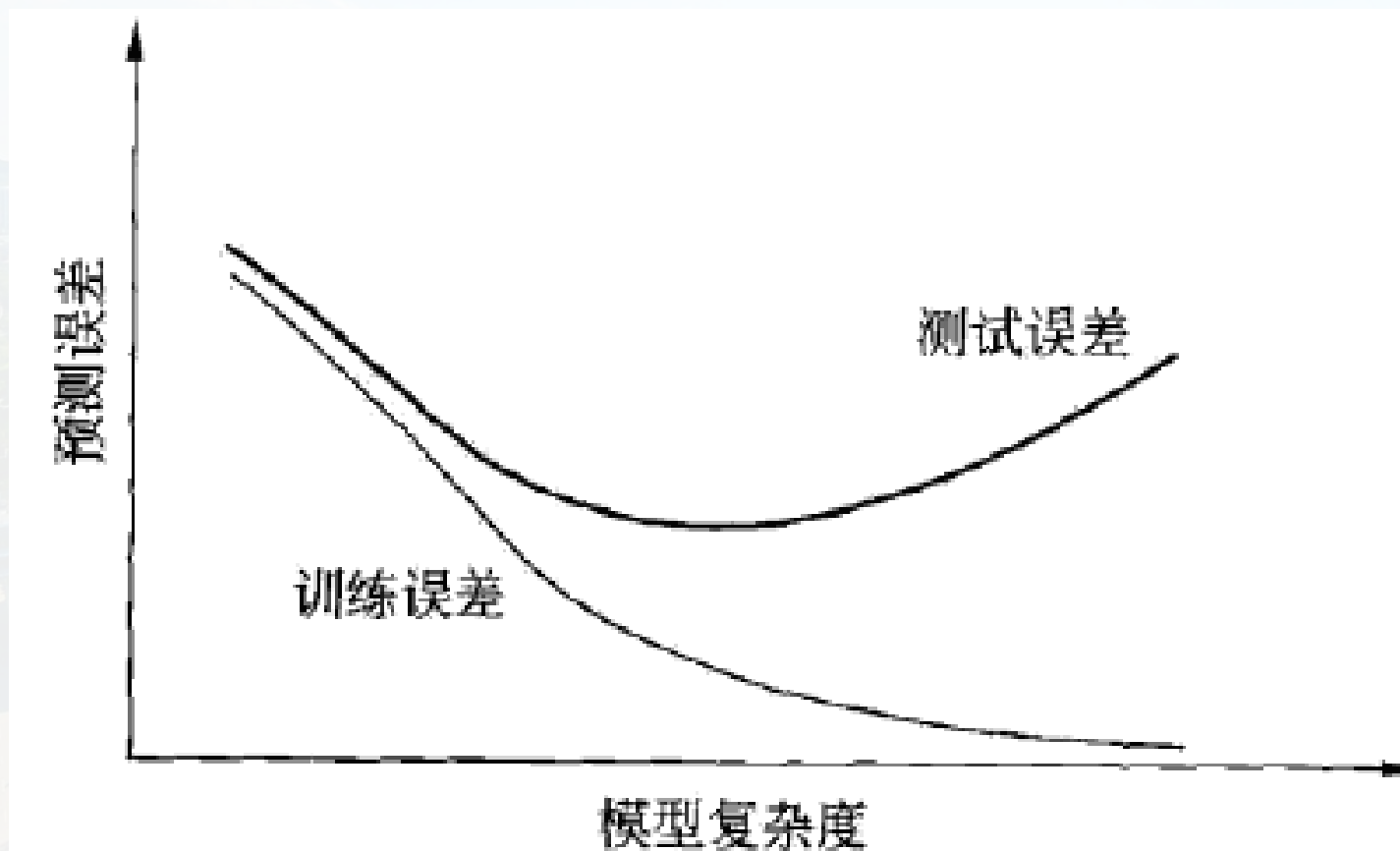




模型评估与模型选择



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

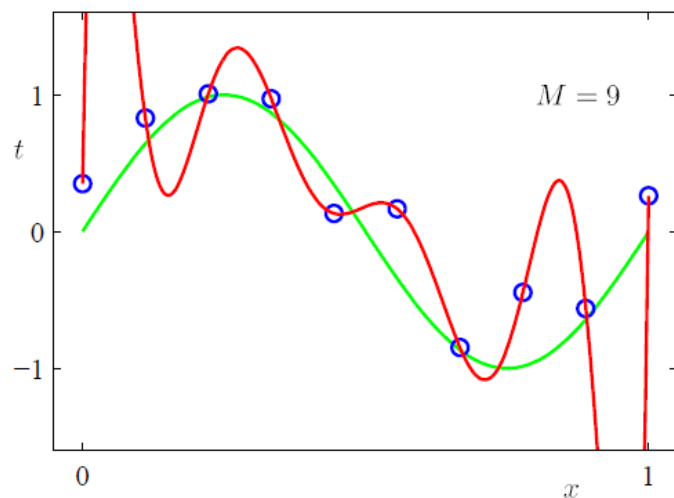




模型评估与模型选择



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY



	$M = 0$	$M = 1$	$M = 3$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

随着多项式M阶的增加，系数的大小也会增加！

$$L(w) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i; w))^2 + \frac{\lambda}{2} \|w\|^2$$

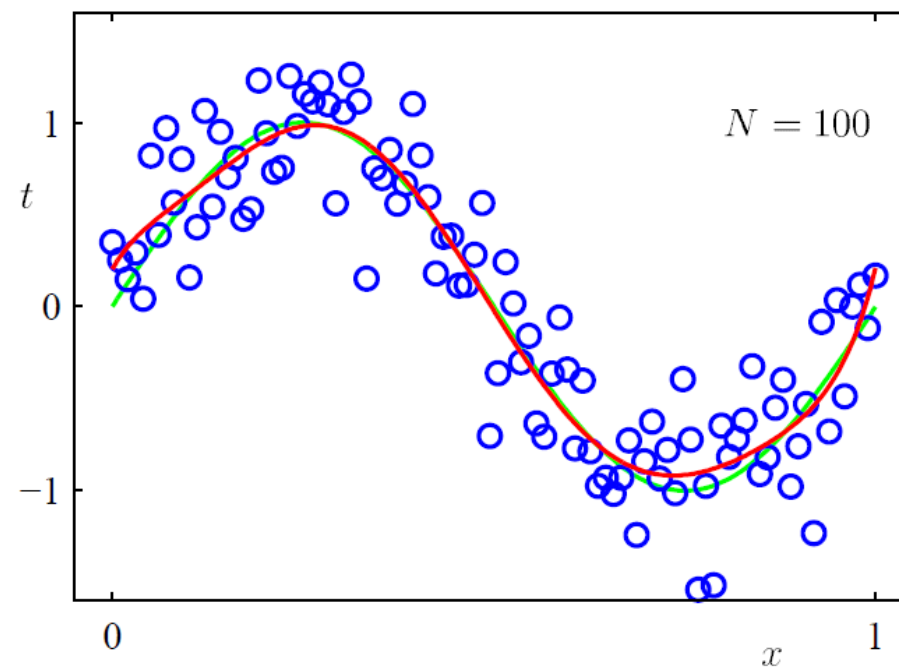
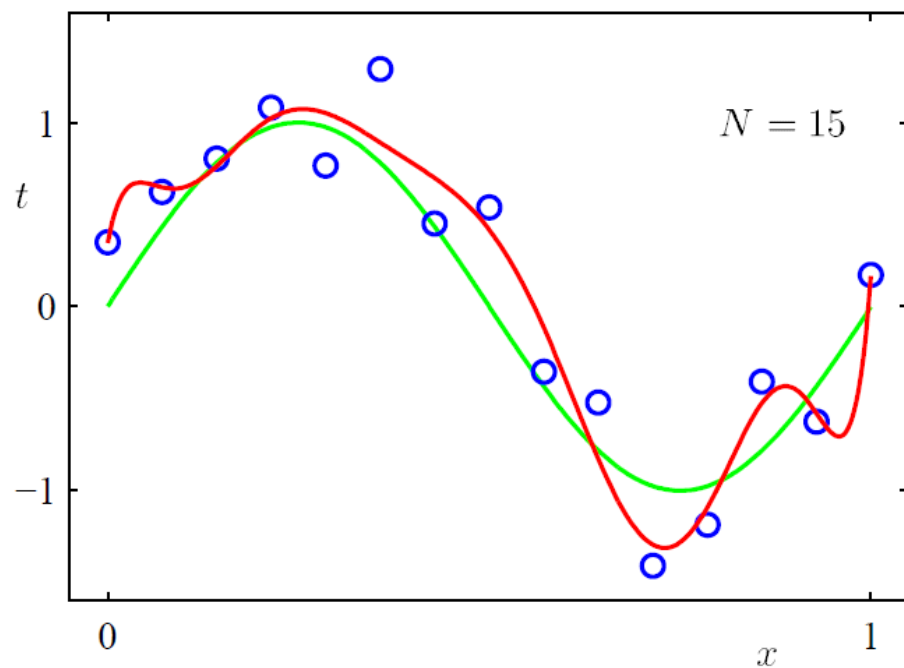
对大的系数进行惩罚



控制过拟合:数据集大小



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY





偏差与方差



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

通过实验可以估计学习算法的泛化性能，而“偏差-方差分解”可以用来帮助解释泛化性能。偏差-方差分解试图对学习算法期望的泛化错误率进行拆解。

对测试样本 x ，令 y_D 为 x 在数据集中的标记， y 为 x 的真实标记， $f(x; D)$ 为训练集 D 上学得模型 f 在 x 上的预测输出。以回归任务为例：学习算法的期望预期为：

$$\bar{f}(x) = \mathbb{E}_D[f(x; D)]$$

使用样本数目相同的不同训练集产生的方差为

$$\text{var}(x) = \mathbb{E}_D \left[\left(f(x; D) - \bar{f}(x) \right)^2 \right]$$

噪声为

$$\varepsilon^2 = \mathbb{E}_D[(y_D - y)^2]$$



期望输出与真实标记的差别称为偏差，即 $bias^2(x) = (\bar{f}(x) - y)^2$

为便于讨论，假定噪声期望为0，也即 $\mathbb{E}_D[y_D - y] = 0$ ，对泛化误差分解

$$\begin{aligned} E(f; D) &= \mathbb{E}_D \left[(f(x; D) - y_D)^2 \right] \\ &= \mathbb{E}_D \left[(f(x; D) - \bar{f}(x) + \bar{f}(x) - y_D)^2 \right] \\ &= \mathbb{E}_D \left[(f(x; D) - \bar{f}(x))^2 \right] + \mathbb{E}_D \left[(\bar{f}(x) - y_D)^2 \right] \\ &\quad + \mathbb{E}_D \left[2(f(x; D) - \bar{f}(x))(\bar{f}(x) - y_D) \right] \\ &= \mathbb{E}_D \left[(f(x; D) - \bar{f}(x))^2 \right] + \mathbb{E}_D \left[(\bar{f}(x) - y_D)^2 \right] \end{aligned}$$



偏差与方差



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

$$\begin{aligned} &= \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right] + \mathbb{E}_D \left[(\bar{f}(\mathbf{x}) - y + y - y_D)^2 \right] \\ &= \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right] + \mathbb{E}_D \left[(\bar{f}(\mathbf{x}) - y)^2 \right] + \mathbb{E}_D \left[(y - y_D)^2 \right] \\ &\quad + 2\mathbb{E}_D \left[(\bar{f}(\mathbf{x}) - y)(y - y_D) \right] \end{aligned}$$

又由假设中噪声期望为0，可得

$$E(f; D) = \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right] + (\bar{f}(\mathbf{x}) - y)^2 + \mathbb{E}_D \left[(y_D - y)^2 \right]$$

于是： $E(f; D) = bias^2(\mathbf{x}) + var(\mathbf{x}) + \varepsilon^2$

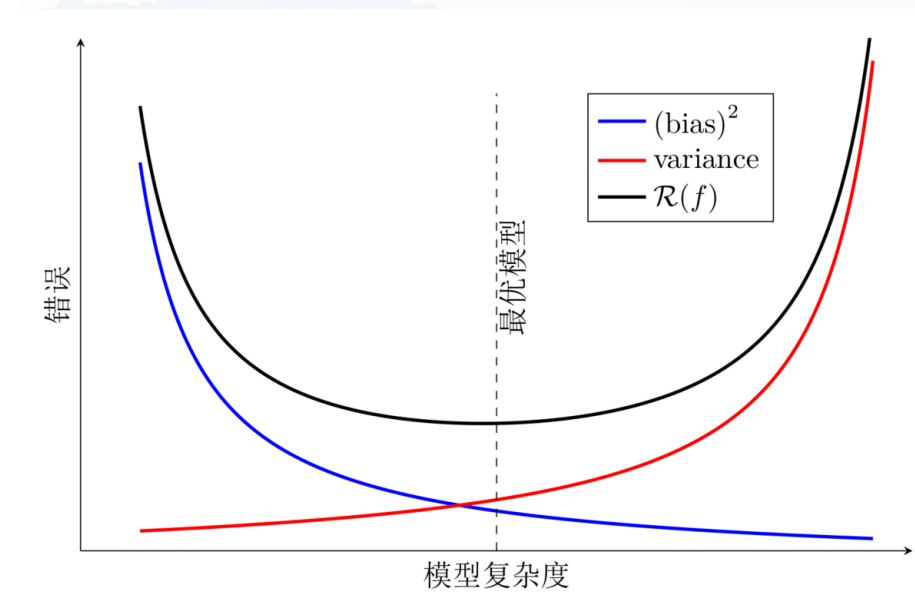
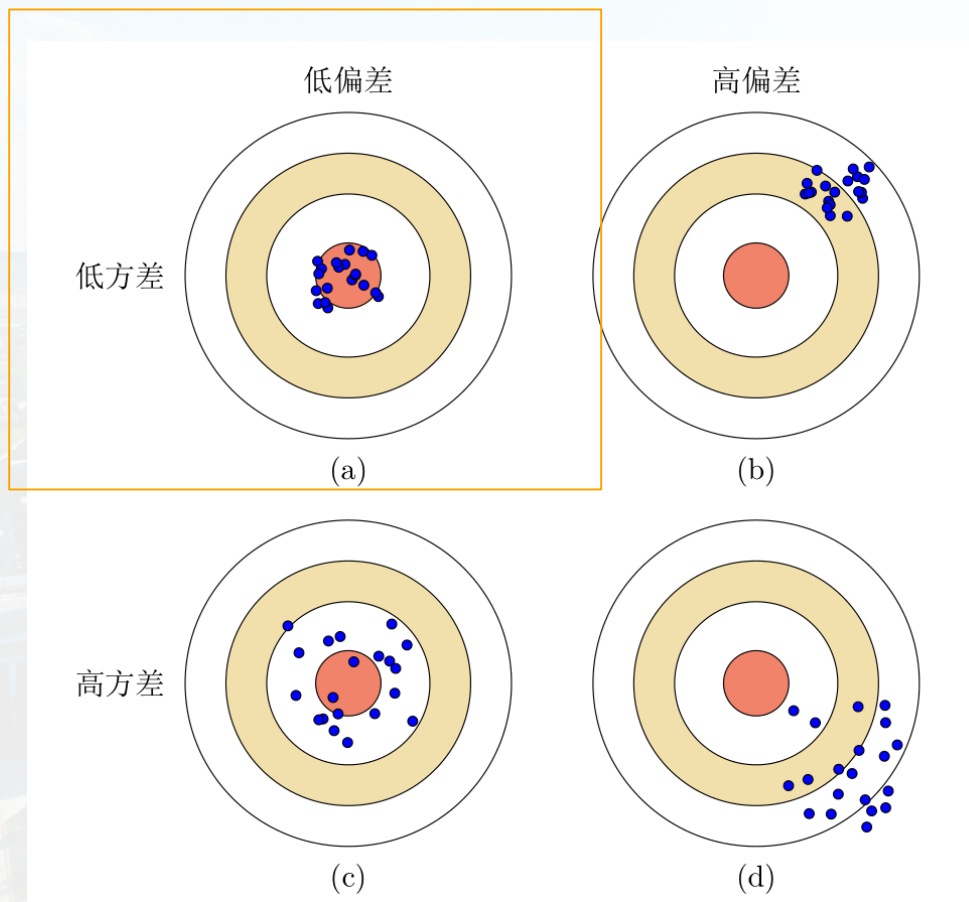
也即泛化误差可分解为偏差、方差与噪声之和。



模型选择：偏差与方差



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY



集成模型：有效的降低方差的方法



正则化与交叉验证



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 正则化一般形式:

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f)$$

■ 回归问题中:

$$L(w) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i; w))^2 + \frac{\lambda}{2} \|w\|^2$$

$$L(w) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i; w))^2 + \lambda \|w\|_1$$



正则化与交叉验证



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 交叉验证:

- 训练集 training set: 用于训练模型
- 验证集 validation set: 用于模型选择
- 测试集 test set: 用于最终对学习方法的评估

$$J(\theta) = \frac{1}{2} (Y - X\theta)^T (Y - X\theta) + \frac{\lambda}{2} |\theta|_1$$

- 简单交叉验证 (70%, 30%)
- S折交叉验证
- 留一交叉验证

Training Set

Train	Train	Val
Train	Val	Train
Val	Train	Train

Model 1 Model 2 Model 3

Avg Err



泛化能力 generalization ability



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 泛化误差 generalization error

$$R_{exp}(\hat{f}) = E_P[L(Y, \hat{f}(X))] = \int_{\mathcal{X} \times \mathcal{Y}} L(Y, \hat{f}(x)) P(x, y) dx dy$$

■ 泛化误差上界

- 比较学习方法的泛化能力-----比较泛化误差上界
- 性质：样本容量增加，泛化误差趋于0，假设空间容量越大，泛化误差越大

■ 二分类问题 $X \in \mathbb{R}^n, Y \in \{-1, +1\}$

■ 期望风险和经验风险 $R(f) = \mathbb{E}[L(Y, f(X))]$ $\hat{R}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i))$



- PAC: **Probably Approximately Correct**
- 根据大数定律，当训练集大小 $|D|$ 趋向无穷大时，泛化错误趋向于0，即经验风险趋近于期望风险。

$$\lim_{|D| \rightarrow \infty} [R(f) - \hat{R}(f)] = 0$$

- PAC学习：泛化误差上界，二分类问题，当假设空间是有限个函数的集合对任意一个函数 f ，至少以概率 $1 - \delta$ ，以下不等式成立：

$$P([R(f) - \hat{R}(f)] \leq \epsilon) \geq 1 - \delta$$

近似正确， $0 < \epsilon < 0.5$

可能， $0 < \delta < 0.5$



样本复杂度



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 如果固定 ϵ, δ ，可以反过来计算出样本复杂度为

$$n(\epsilon, \delta) \geq \frac{1}{2\epsilon^2} \left(\ln|\mathcal{F}| + \ln \frac{2}{\delta} \right)$$

- 其中 $|\mathcal{F}|$ 为假设空间的大小，可以用Rademacher复杂性或VC维来衡量。
- PAC学习理论可以帮助分析一个机器学习方法在什么条件下可以学习到一个近似正确的分类器。
- 如果希望模型的假设空间越大，泛化错误越小，其需要的样本数量越多。



生成模型与判别模型



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 监督学习的目的就是学习一个模型：
- 决策函数： $Y = f(X)$
- 条件概率分布： $P(Y|X)$
- 生成方法Generative approach 对应生成模型：generative model,

$$P(Y|X) = \frac{P(X, Y)}{P(X)}$$

- 朴素贝叶斯法和隐马尔科夫模型



生成模型与判别模型



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 判别方法由数据直接学习决策函数 $f(X)$ 或条件概率分布 $P(Y|X)$ 作为预测的模型，即判别模型
- Discriminative approach对应discriminative model

$$Y = f(X)$$

$$P(Y|X)$$

- K近邻法、感知机、决策树、logistic回归模型、最大熵模型、支持向量机、提升方法和条件随机场。



生成模型与判别模型



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 各自优缺点：

- 生成方法：可还原出联合概率分布 $P(X, Y)$ ，而判别方法不能。生成方法的收敛速度更快，当样本容量增加的时候，学到的模型可以更快地收敛于真实模型；当存在隐变量时，仍可以使用生成方法，而判别方法则不能用。
- 判别方法：直接学习到条件概率或决策函数，直接进行预测，往往学习的**准确率更高**；由于直接学习 $Y = f(X)$ 或 $P(Y|X)$ ，可对数据进行各种程度上的抽象、定义特征并使用特征，因此可以简化学习过程。

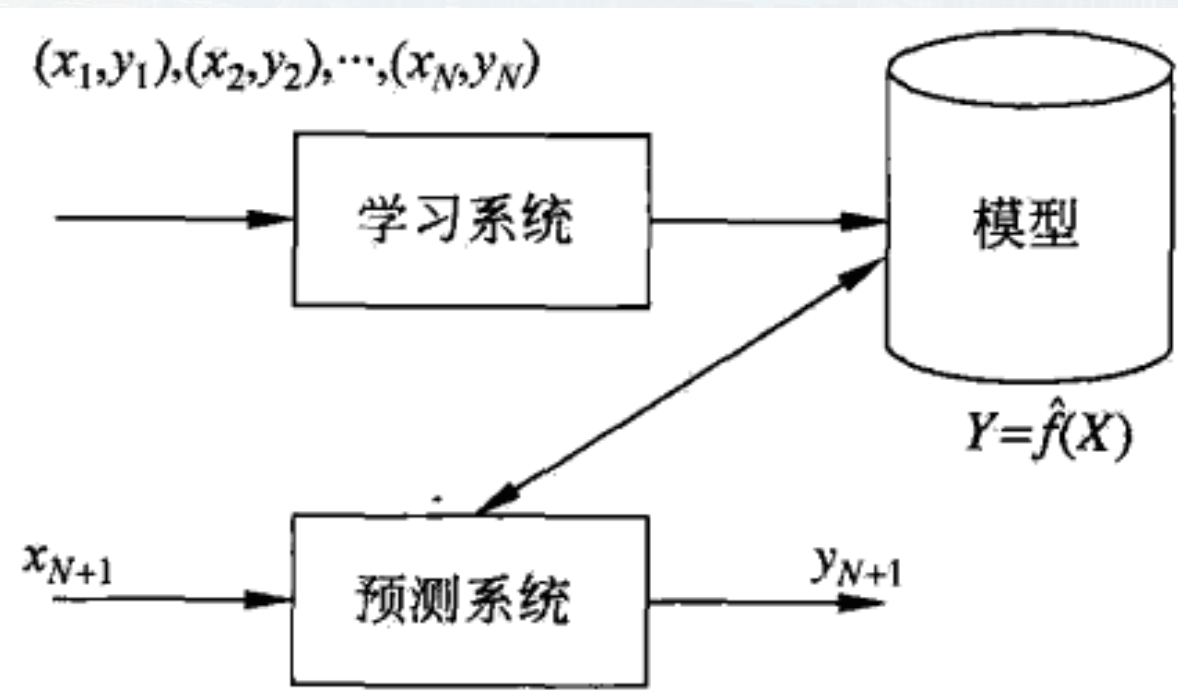


- 回归模型是表示从输入变量到输出变量之间映射的函数.回归问题的学习等价于函数拟合。

- 学习和预测两个阶段

- 训练集:

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$





回归问题



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 回归学习最常用的损失函数是平方损失函数，在此情况下，回归问题可以由著名的最小二乘法(least squares)求解。
- 股价预测



回归



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 回归：广义线性模型 (generalized linear model)
- 分类：根据因变量的不同
 - 连续：线性回归
 - 二项分布：logistic回归
 - poisson分布：poisson回归
 - 负二项分布：负二项回归



标注问题

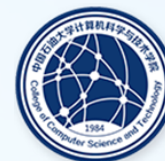


青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 标注: tagging, 结构预测: structure prediction
- 输入: 观测序列, 输出: 标记序列或状态序列
- 学习和标注两个过程
- 训练集: $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$
- 观测序列: $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})^T, i = 1, 2, \dots, N$
- 输出标记序列: $y_i = (y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(n)})^T$
- 模型: 条件概率分布 $P(Y^{(1)}, Y^{(2)}, \dots, Y^{(n)} | X^{(1)}, X^{(2)}, \dots, X^{(n)})$



标注问题

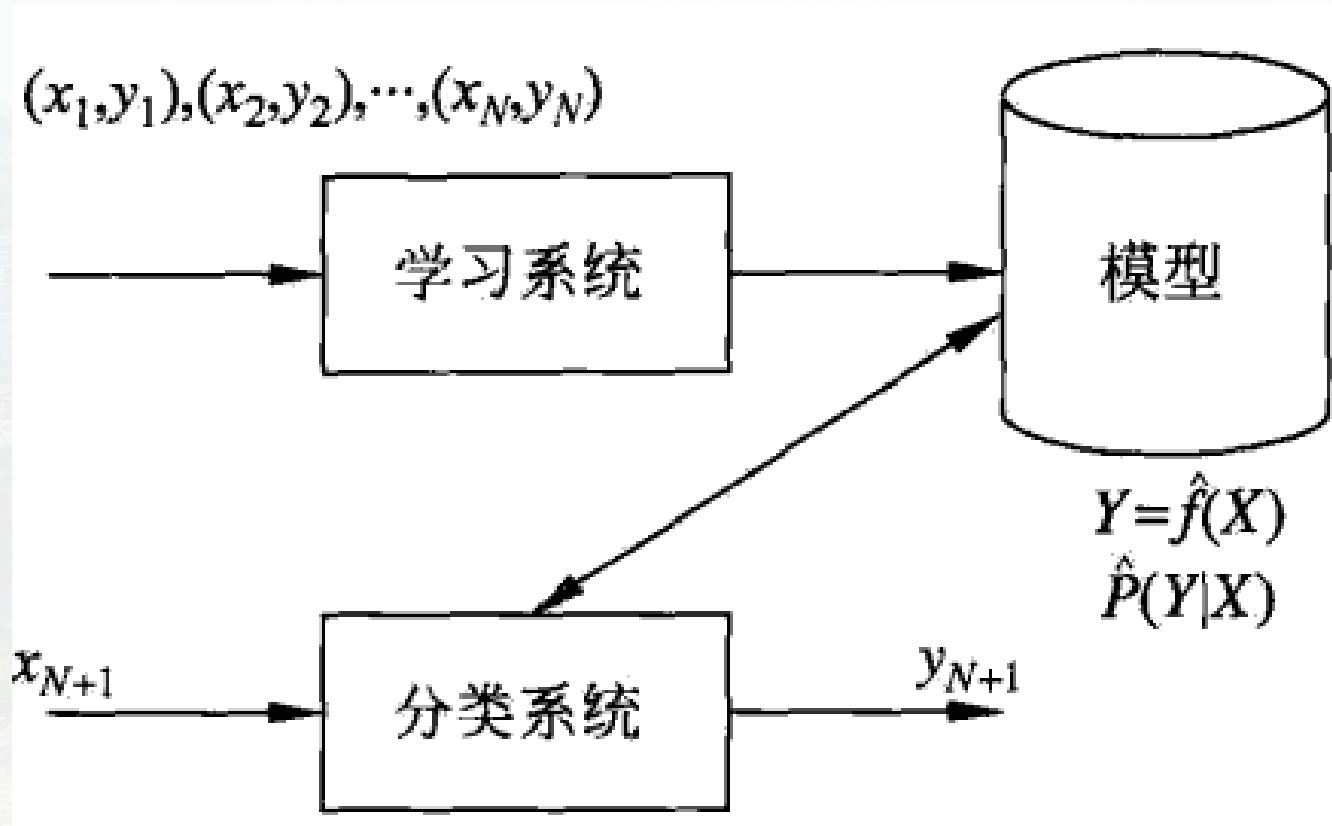


青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 例子：
- 标记表示名词短语的“开始”、“结束”或“其他”（分别以B, E, O表示）
- 输入：At Microsoft Research, we have an insatiable curiosity and the desire to create new technology that will help define the computing experience.
- 输出：At/O Microsoft/B Research/E, we/O have/O an/O insatiable/O curiosity/E and/O the/O desire/BE to/O create/O new/B technology/E that/O will/O help/O define/O the/O computing/B experience/E.



分类问题





分类问题

■ 二分类评价指标

- TP true positive
- FN false negative
- FP false positive
- TN true negative

准确率(ACC): $acc = \frac{TP+TN}{TP+FN+FP+TN}$

• 精确率

$$P = \frac{TP}{TP+FP}$$

• 召回率

$$R = \frac{TP}{TP+FN}$$

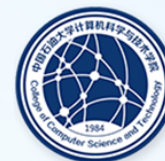
• F_1 值

$$\frac{2}{F_1} = \frac{1}{P} + \frac{1}{R}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}$$



二分类任务



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 预测值与输出标记

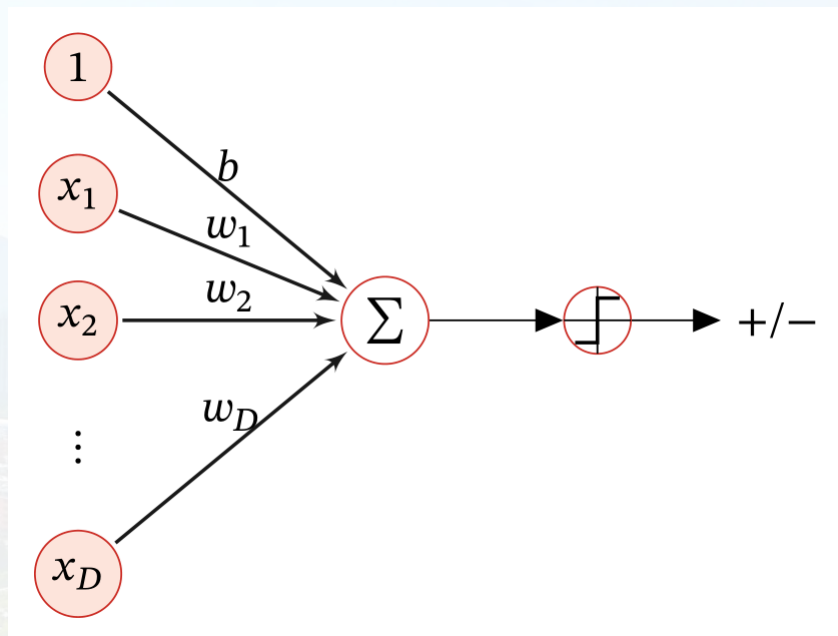
$$z = w^T x + b \triangleq w^T x \quad y \in \{0, 1\}$$

- 寻找函数将分类标记与线性回归模型输出联系起来

- 最理想的函数——单位阶跃函数

$$y = \begin{cases} 0, & z < 0 \\ 0.5, & z = 0 \\ 1, & z > 0 \end{cases}$$

- 预测值大于零就判为正例，小于零就判为反例，预测值为临界值零则可任意判别





二分类任务



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 单位阶跃函数缺点

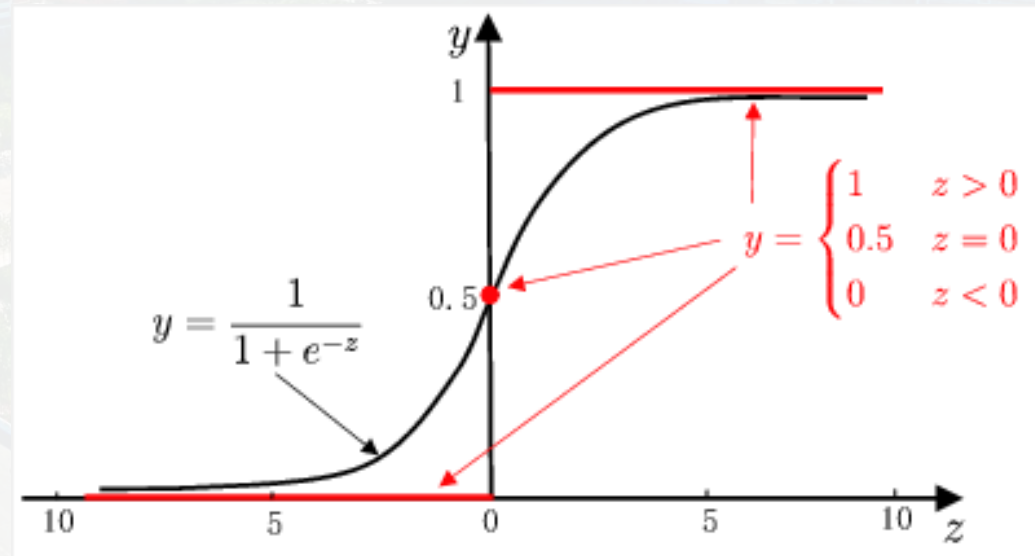
- 不连续

■ 替代函数——对数几率函数 (logistic function)

- 单调可微、任意阶可导

$$y = \frac{1}{1 + e^{-z}}$$

单位阶跃函数与对数几率函数的比较





对数几率回归

■ 运用对数几率函数

$$y = \frac{1}{1+e^{-z}} \text{ 变为 } y = \frac{1}{1+e^{-w^T x}}$$

$$p(y = 1|x) = \sigma(w^T x)$$

■ 对数几率 (log odds)

□ 样本作为正例的相对可能性的对数

$$= \frac{1}{1 + \exp(-w^T x)}$$

$$\log \frac{y}{1-y}$$

■ 对数几率回归优点

- 无需事先假设数据分布
- 可得到“类别”的近似概率预测
- 可直接应用现有数值优化算法求取最优解



对数几率回归 - 极大似然法



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 对数几率

$$\log \frac{p(y = 1|x)}{p(y = 0|x)} = w^T x$$

$$p(y = 1|x) = \frac{1}{1 + \exp(-w^T x)} = \frac{\exp(w^T x)}{1 + \exp(w^T x)}$$

$$p(y = 0|x) = \frac{\exp(-w^T x)}{1 + \exp(-w^T x)} = \frac{1}{1 + \exp(w^T x)}$$



似然函数



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- logistic分类器是由一组权值系数组成的，最关键的问题就是如何获取这组权值，通过极大似然函数估计获得，并且 $Y \sim f(x; w)$
- **似然函数**是统计模型中参数的函数。给定输出 x 时，关于参数 θ 的似然函数 $L(\theta|x)$ （在数值上）等于给定参数 θ 后变量 X 的概率： $L(\theta|x) = P(X = x|\theta)$
- 似然函数的重要性不是它的取值，而是当参数变化时概率密度函数到底是变大还是变小。
- 极大似然函数：似然函数取得最大值表示相应的参数能够使得统计模型最为合理



- 那么对于上述 N 个观测事件，设

$$P(Y = 1|x) = \pi(x), P(Y = 0|x) = 1 - \pi(x)$$

$$\begin{aligned}\pi(x) &= \frac{1}{1 + \exp(-w^T x)} \\ &= \frac{\exp(w^T x)}{1 + \exp(w^T x)}\end{aligned}$$

- 其联合概率密度函数，即似然函数为：

$$\prod_{i=1}^N [\pi(x_i)]^{y_i} [1 - \pi(x_i)]^{1-y_i}$$

- 目标：求出使这一似然函数的值最大的参数估， $w_0(b), w_1, w_2, \dots, w_n$ ，使得 $L(w)$ 取得 最大值。
- 对 $L(w)$ 取对数：



模型参数估计



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 对数似然函数

$$L(w) = \sum_{i=1}^N [y_i \log \pi(x_i) + (1 - y_i) \log(1 - \pi(x_i))]$$

$$= \sum_{i=1}^N \left[y_i \log \frac{\pi(x_i)}{1 - \pi(x_i)} + \log(1 - \pi(x_i)) \right]$$

$$= \sum_{i=1}^N [y_i(w^T x_i) - \log(1 + \exp(w^T x_i))]$$

- 对 $L(w)$ 求极大值，得到 w 的估计值。
- 通常采用梯度下降法及拟牛顿法，学到的模型：

$$p(y = 1|x) = \frac{\exp(\hat{w}^T x)}{1 + \exp(\hat{w}^T x)} \quad p(y = 0|x) = \frac{1}{1 + \exp(\hat{w}^T x)}$$

$$\pi(x) = \frac{1}{1 + \exp(-w^T x)} \\ = \frac{\exp(w^T x)}{1 + \exp(w^T x)}$$

$$1 - \pi(x) = \frac{1}{1 + \exp(w^T x)}$$



模型参数估计

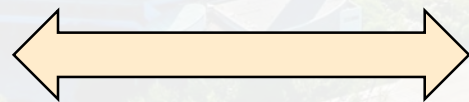
■ 对 w 求偏导:

$$\frac{\partial L(w)}{\partial w} = \frac{\partial \sum_{i=1}^N [y_i(w^T x_i) - \log(1 + \exp(w^T x_i))]}{\partial w}$$

$$= \sum_{i=1}^N \left(y_i x_i^T - \frac{\exp(w^T x_i) x_i^T}{1 + \exp(w^T x_i)} \right)$$

$$= \sum_{i=1}^N (y_i - \hat{y}_i) x_i^T$$

\uparrow
 $h_w(x_i)$



线性回归损失函数对参数求偏导

$$\frac{\partial J}{\partial \theta} = \sum_{i=1}^N (h_{\theta}(x_i) - y_i) x_i^T$$



模型参数估计

■ 对 W 求偏导:

$$\frac{\partial L(W)}{\partial W}$$

$$= \frac{\partial \sum_{i=1}^N [y_i (w^T x_i) - \log(1 + \exp(w^T x_i))]}{\partial W}$$

$$= \frac{\partial (Y^T XW - \text{sum}(g(XW)))}{\partial W}$$

$$= Y^T X - \frac{\partial \text{sum}(g(XW))}{\partial g(XW)} \cdot \frac{\partial g(XW)}{\partial XW} \cdot \frac{\partial XW}{\partial W}$$

$$g(z) = \begin{bmatrix} g(z_1) \\ \vdots \\ g(z_N) \end{bmatrix} = \begin{bmatrix} \log(1 + \exp(z_1)) \\ \vdots \\ \log(1 + \exp(z_N)) \end{bmatrix}$$



模型参数估计



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

$$g(z) = \begin{bmatrix} g(z_1) \\ \vdots \\ g(z_N) \end{bmatrix} = \begin{bmatrix} \log(1 + \exp(z_1)) \\ \vdots \\ \log(1 + \exp(z_N)) \end{bmatrix}$$

$$\frac{\partial g(z)}{\partial z} = \begin{bmatrix} \frac{\partial g(z_1)}{z_1} & \frac{\partial g(z_1)}{z_2} & \dots & \frac{\partial g(z_1)}{z_N} \\ \frac{\partial g(z_2)}{z_1} & \frac{\partial g(z_2)}{z_2} & \dots & \frac{\partial g(z_2)}{z_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g(z_N)}{z_1} & \frac{\partial g(z_N)}{z_2} & \dots & \frac{\partial g(z_N)}{z_N} \end{bmatrix} = \begin{bmatrix} \frac{\partial g(z_1)}{z_1} & 0 & \dots & 0 \\ 0 & \frac{\partial g(z_2)}{z_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial g(z_N)}{z_N} \end{bmatrix}$$



模型参数估计



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

$$g(z) = \begin{bmatrix} g(z_1) \\ \vdots \\ g(z_N) \end{bmatrix} = \begin{bmatrix} \log(1 + \exp(z_1)) \\ \vdots \\ \log(1 + \exp(z_N)) \end{bmatrix}$$

$$\frac{\partial g(z)}{\partial z} = \begin{bmatrix} \frac{\partial g(z_1)}{z_1} & 0 & \dots & 0 \\ 0 & \frac{\partial g(z_2)}{z_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial g(z_N)}{z_N} \end{bmatrix} = \begin{bmatrix} \frac{\exp(z_1)}{1 + \exp(z_1)} & 0 & \dots & 0 \\ 0 & \frac{\exp(z_2)}{1 + \exp(z_2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\exp(z_N)}{1 + \exp(z_N)} \end{bmatrix}$$



模型参数估计

$$g(z) = \begin{bmatrix} g(z_1) \\ \vdots \\ g(z_N) \end{bmatrix} = \begin{bmatrix} \log(1 + \exp(z_1)) \\ \vdots \\ \log(1 + \exp(z_N)) \end{bmatrix}$$

$$\frac{\partial g(z)}{\partial z} = \begin{bmatrix} \frac{\exp(z_1)}{1 + \exp(z_1)} & 0 & \dots & 0 \\ 0 & \frac{\exp(z_2)}{1 + \exp(z_2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\exp(z_N)}{1 + \exp(z_N)} \end{bmatrix} = \begin{bmatrix} \sigma(z_1) & 0 & \dots & 0 \\ 0 & \sigma(z_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma(z_N) \end{bmatrix}$$



模型参数估计



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 对 W 求偏导:

$$\frac{\partial \text{sum}(g(XW))}{\partial g(XW)} \cdot \frac{\partial g(XW)}{\partial XW} \cdot \frac{\partial XW}{\partial W}$$

$$= 1^T \cdot \begin{bmatrix} \sigma((XW)_1) & 0 & \dots & 0 \\ 0 & \sigma((XW)_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma((XW)_N) \end{bmatrix} \cdot X$$
$$= \sigma(XW) \cdot X = \hat{Y}^T X$$



模型参数估计



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 对 W 求偏导:

$$\frac{\partial L(w)}{\partial W}$$

$$= Y^T X - \hat{Y}^T X = (Y^T - \hat{Y}^T) X$$



多项logistic回归

- 设Y的取值集合为

$$Y \in \{1, 2, \dots, K\}$$

$$\log \frac{P(Y=i|x)}{P(Y=K|x)} = w_i^T x$$

- 多项logistic回归模型

$$P(Y = k|x) = \frac{\exp(w_k^T x)}{1 + \sum_{k=1}^{K-1} \exp(w_k^T x)}, k = 1, 2, \dots, K-1$$

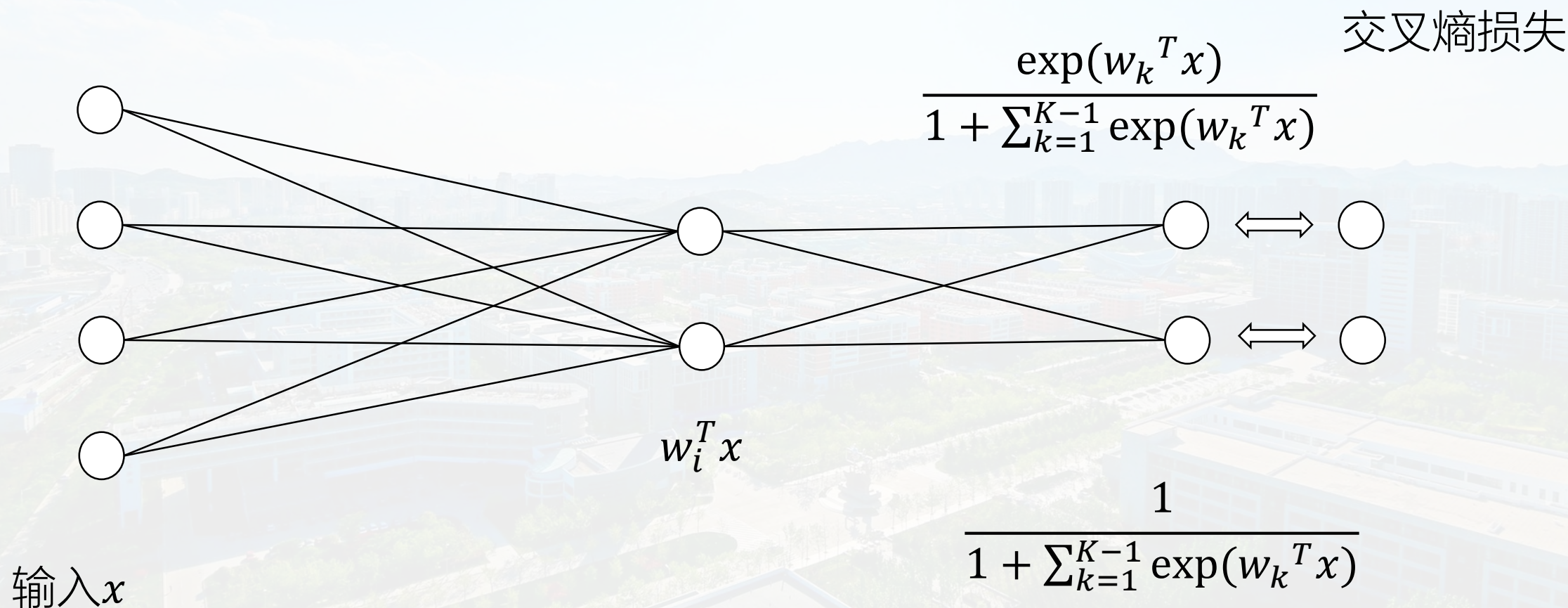
$$P(Y = K|x) = \frac{1}{1 + \sum_{k=1}^{K-1} \exp(w_k^T x)}$$



多项logistic回归



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY





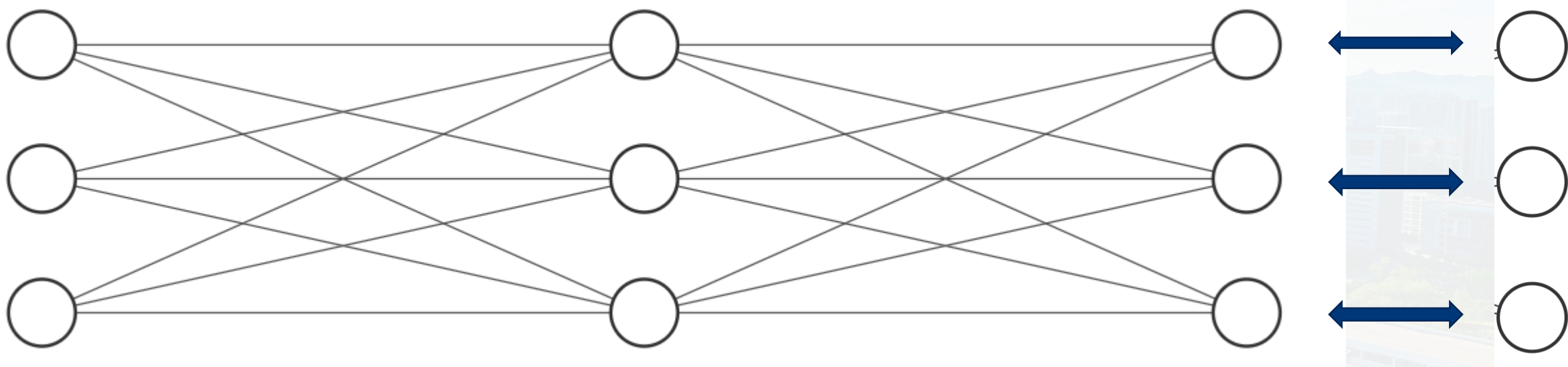
Softmax回归



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

$$\frac{\exp(w_k^T x)}{\sum_{k=1}^K \exp(w_k^T x)}$$

交叉熵损失



$$w_i^T x$$

输入 x



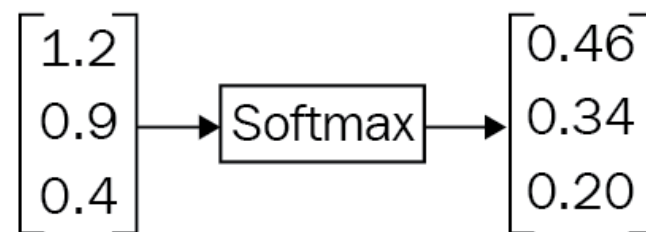
Softmax回归

■ 多分类问题

$$y = \arg \max_{c=1}^C f_c(\mathbf{x}; \mathbf{w}_c)$$

■ Softmax函数

$$\text{softmax}(x_k) = \frac{\exp(w_k^T x)}{\sum_{k=1}^K \exp(w_k^T x)}$$

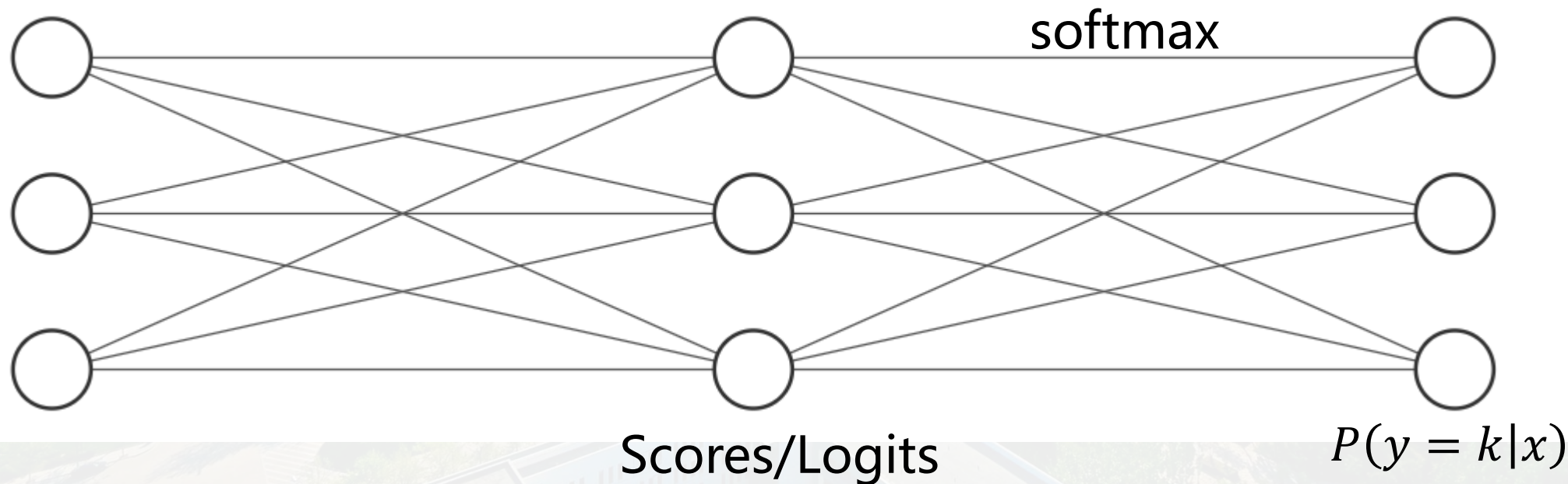




Softmax回归

- 利用softmax函数，目标类别 $y = c$ 的条件概率为：

$$P(y = k|x) = \text{softmax}(\exp(w_k^T x)) = \frac{\exp(w_k^T x)}{\sum_{k=1}^K \exp(w_k^T x)}$$





Softmax回归

KL散度

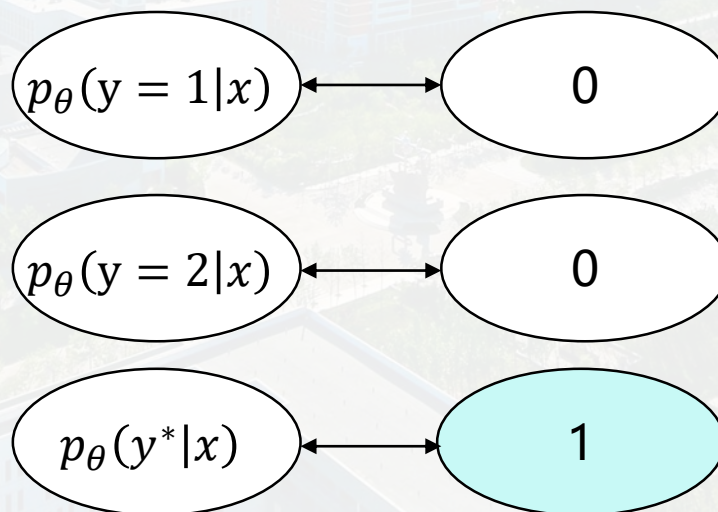
$$D_{kl}(p_r(y|x)||p_\theta(y|x)) = \sum_{y=1}^c p_r(y|x) \log \frac{p_r(y|x)}{p_\theta(y|x)}$$

$$\propto - \sum_{y=1}^c p_r(y|x) \log p_\theta(y|x)$$

交叉熵损失

y^* 为 x 的真实标签

$$= -\log p_\theta(y^*|x) \quad \text{负对数似然}$$





参数学习



青 岛 软 件 学 院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 模型：Softmax回归

$$\hat{y} = \frac{\exp(W^T x)}{1^T \cdot \exp(W^T x)}$$

$$W = \begin{bmatrix} -w_1^T & - \\ \vdots & \\ -w_k^T & - \end{bmatrix}$$

■ 学习准则：交叉熵

$$J(W) = -\frac{1}{N} \sum_{i=1}^N y_i^T \log \hat{y}_i$$

■ 优化：梯度下降 $\frac{\partial J(W)}{\partial W} = -\frac{1}{N} \sum_{i=1}^N [y_i - \hat{y}_i] x_i^T$



Softmax求导



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

$$\begin{aligned}\frac{\partial \text{softmax}(x)}{\partial x} &= \frac{\partial \left(\frac{\exp(x)}{1^T \exp(x)} \right)}{\partial x} \\&= \frac{\partial \left(\frac{\exp(x)}{1^T \exp(x)} \right)}{\partial \exp(x)} \cdot \frac{\partial \exp(x)}{\partial x} \\&= \left(\frac{I}{1^T \exp(x)} - \frac{\exp(x)}{(1^T \exp(x))^2} \cdot \frac{\partial 1^T \exp(x)}{\partial \exp(x)} \right) \cdot \text{diag}(\exp(x)) \\&= \text{diag} \left(\frac{\exp(x)}{1^T \exp(x)} \right) - \frac{\exp(x)}{(1^T \exp(x))^2} \cdot 1^T \text{diag}(\exp(x)) \\&= \text{diag}(\text{softmax}(x)) - \frac{\exp(x) \exp(x)^T}{(1^T \exp(x))^2} \\&= \text{diag}(\text{softmax}(x)) - \text{softmax}(x) \text{softmax}(x)^T\end{aligned}$$



求偏导



- 若 $y = \text{softmax}(z)$, 则 $\frac{\partial y}{\partial z} = \text{diag}(y) - yy^T$
- 若 $z = W^T x = [w_1^T x, w_2^T x, \dots, w_K^T x]^T$, 则

$$\frac{\partial z}{\partial W_k} = \begin{bmatrix} \frac{\partial w_1^T x}{\partial W_k} \\ \vdots \\ \frac{\partial w_k^T x}{\partial W_k} \\ \vdots \\ \frac{\partial w_K^T x}{\partial W_k} \end{bmatrix} = \begin{bmatrix} 0^T \\ \vdots \\ x^T \\ \vdots \\ 0^T \end{bmatrix} \triangleq \mathbb{M}_k(x)$$



求偏导



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

$$\frac{\partial J(W)_i}{\partial W_k} = -\frac{\partial(y_i^T \log \hat{y}_i)}{\partial W_k} = -\frac{\partial(y_i^T \log \hat{y}_i)}{\partial \log \hat{y}_i} \cdot \frac{\partial \log \hat{y}_i}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z} \cdot \frac{\partial z}{\partial W_k}$$

$$\blacksquare = -y_i^T (\text{diag}(\hat{y}_i))^{-1} (\text{diag}(\hat{y}_i) - \hat{y}_i \hat{y}_i^T) \mathbb{M}_k(x)$$

$$\blacksquare = -y_i^T (I - \mathbf{1} \cdot \hat{y}_i^T) \mathbb{M}_k(x)$$

$$\blacksquare = -(y_i^T - \text{sum}(y_i) \hat{y}_i^T) \mathbb{M}_k(x)$$

$$\blacksquare = -(y_i^T - \hat{y}_i^T) \begin{bmatrix} 0^T \\ \vdots \\ x_i^T \\ \vdots \\ 0^T \end{bmatrix} = -[y_i - \hat{y}_i]_k x_i^T \quad \longrightarrow \quad \frac{\partial J(W)_i}{\partial W} = -[y_i - \hat{y}_i] x_i^T$$



青 岛 软 件 学 院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

常用的定理



- 没有免费午餐定理 (No Free Lunch Theorem, NFL)
 - 对于基于迭代的最优化算法，不存在某种算法对所有问题（有限的搜索空间内）都有效。如果一个算法对某些问题有效，那么它一定在另外一些问题上比纯随机搜索算法更差。





常用的定理



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 丑小鸭定理(Ugly Duckling Theorem)

- 丑小鸭与白天鹅之间的区别和两只白天鹅之间的区别一样大.





常用的定理



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

■ 奥卡姆剃刀原理(Occam's Razor)

- 如无必要，勿增实体





归纳偏置(Inductive Bias)



青岛软件学院
QINGDAO INSTITUTE OF SOFTWARE
计算机科学与技术学院
COLLEGE OF COMPUTER SCIENCE AND TECHNOLOGY

- 很多学习算法经常会对学习的问题做一些假设，这些假设就称为**归纳偏置**。
 - 在最近邻分类器中，我们会假设在特征空间中，一个小的局部区域中的大部分样本都同属一类。
 - 在朴素贝叶斯分类器中，我们会假设每个特征的条件概率是互相独立的。
 - 在支持向量机中，我们假设间隔最大化。
 - 归纳偏置在贝叶斯学习中也经常称为**先验** (Prior) 。

THANKS

谢谢大家

中国石油大学 (华东)
CHINA UNIVERSITY OF PETROLEUM

汇报人 张琛

青岛软件学院、计算机科学与技术学院

2024/2/28