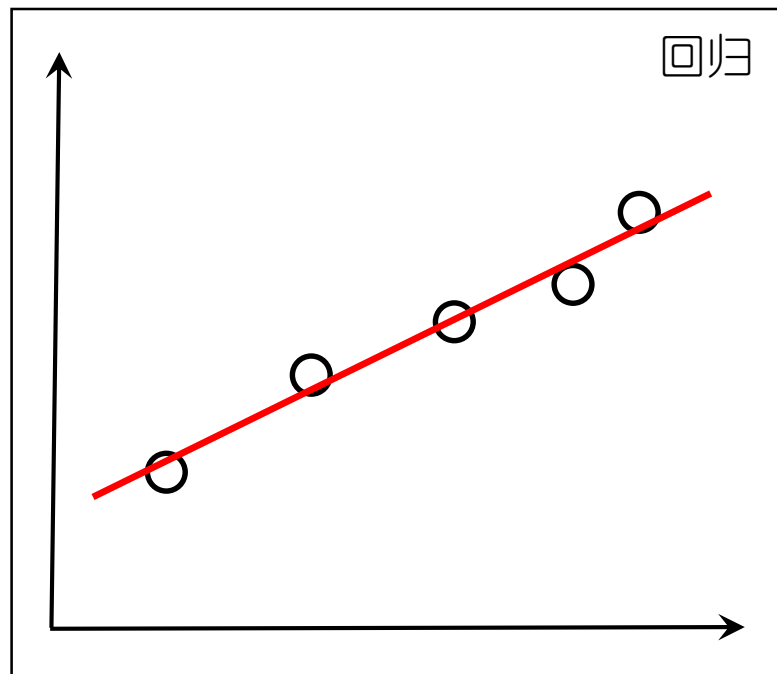
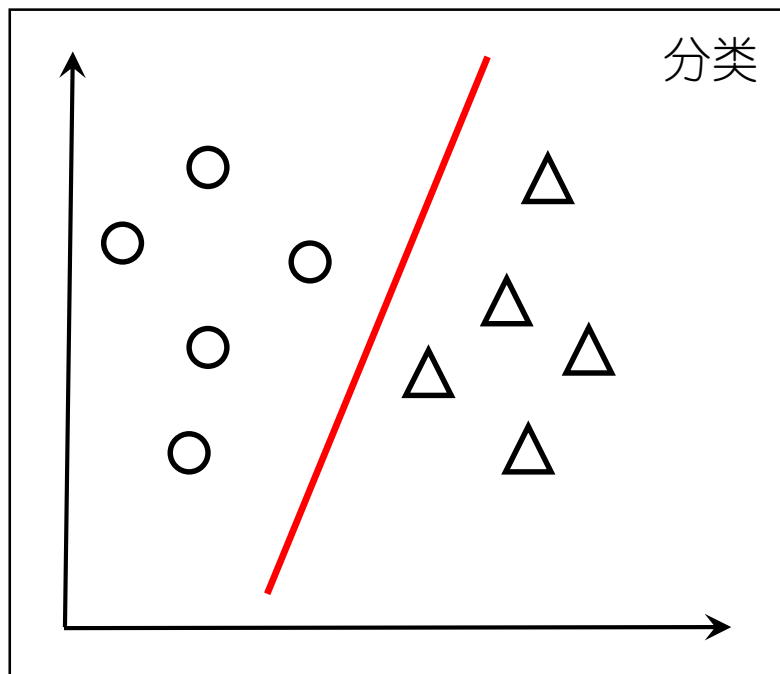


二、典型方法

线性模型



线性模型(linear model)试图学得一个通过属性的线性组合来进行预测的函数

$$f(x) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b$$

向量形式: $f(x) = w^T x + b$

简单、基本、可理解性好

线性模型的变化

对于样例 (x, y) , $y \in \mathbb{R}$, 若希望线性模型的预测值逼近真实标记, 则得到线性回归模型 $y = w^T x + b$

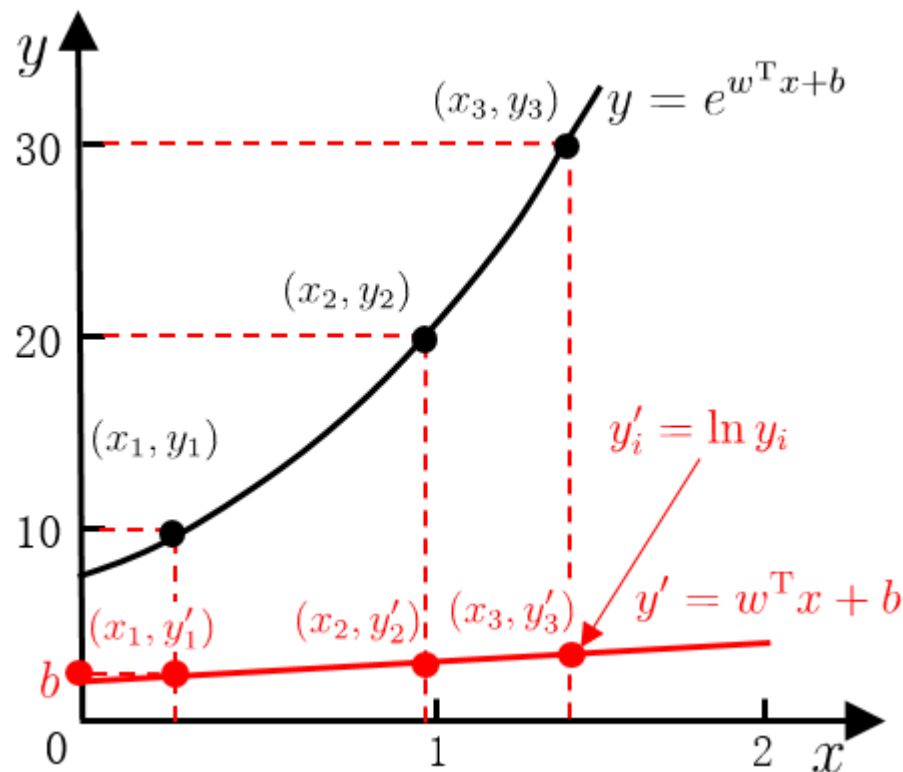
令预测值逼近 y 的衍生物 ?

若令 $\ln y = w^T x + b$

则得到对数线性回归

(log-linear regression)

实际是在用 $e^{w^T x + b}$ 逼近 y



广义(generalized)线性模型

一般形式: $y = \underline{g^{-1}}(w^T x + b)$



单调可微的 联系函数 (link function)

令 $g(\cdot) = \ln(\cdot)$ 则得到 对数线性回归

$$\ln y = w^T x + b$$

...

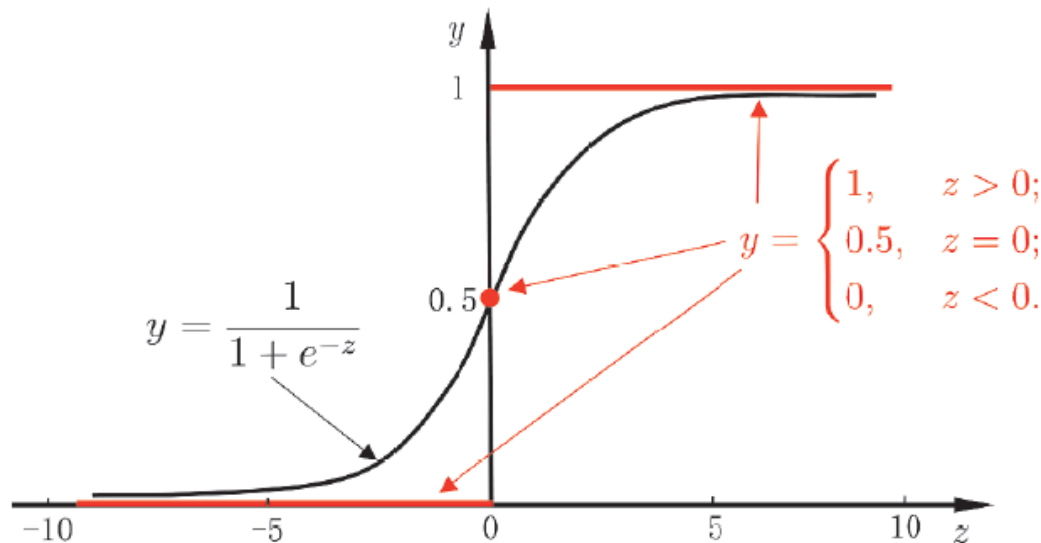
二分类任务

线性回归模型产生的实值输出 $z = \mathbf{w}^T \mathbf{x} + b$
期望输出 $y \in \{0, 1\}$

} 找 z 和 y 的联系函数

理想的“单位阶跃函数”
(unit-step function)

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



性质不好,
需找“替代函数”
(surrogate function)

常用
单调可微、任意阶可导

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

对数几率函数
(logistic function)
简称“对率函数”

对率回归

以对率函数为联系函数：

$$y = \frac{1}{1 + e^{-z}} \quad \text{变为} \quad y = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$

即：

$$\ln \frac{y}{1 - y} = \mathbf{w}^T \mathbf{x} + b$$

“对数几率”

(log odds, 亦称 logit)

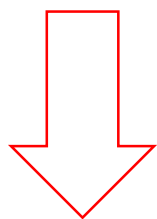
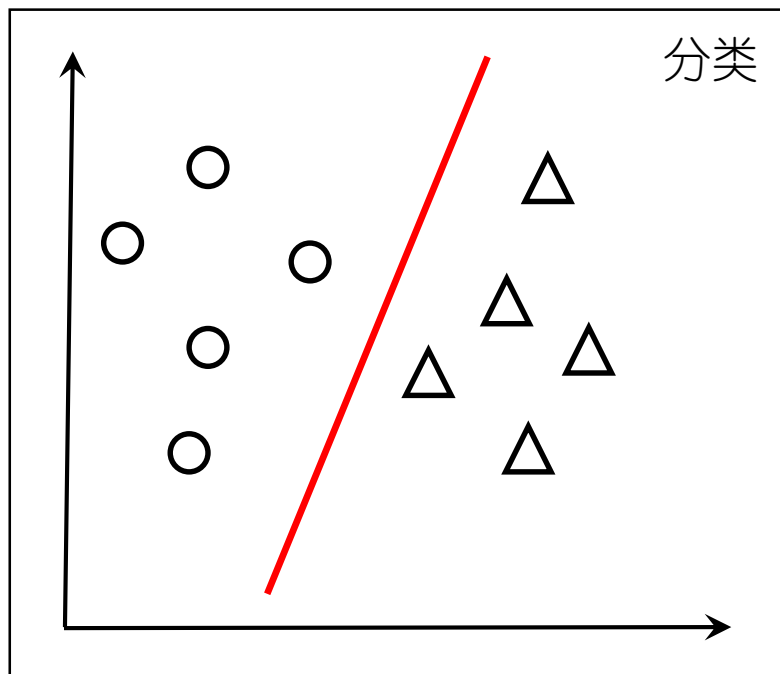
几率(odds), 反映了 \mathbf{x} 作为正例的相对可能性

“对数几率回归” (logistic regression)
简称 “对率回归”

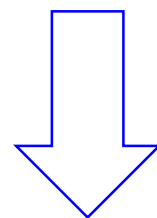
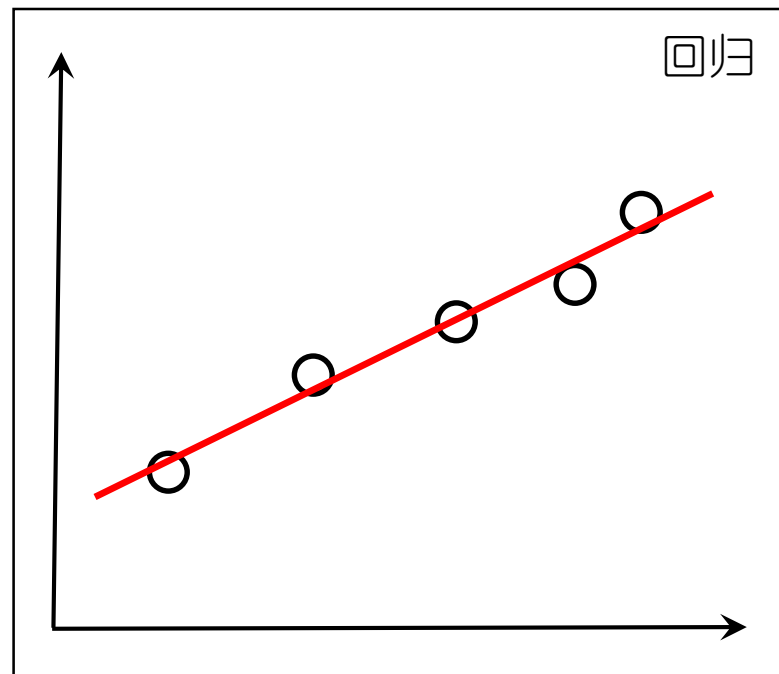
- 无需事先假设数据分布
- 可得到 “类别” 的近似概率预测
- 可直接应用现有数值优化算法求取最优解

注意：它是
分类学习算法！

线性模型做“分类”



如何“直接”做分类？



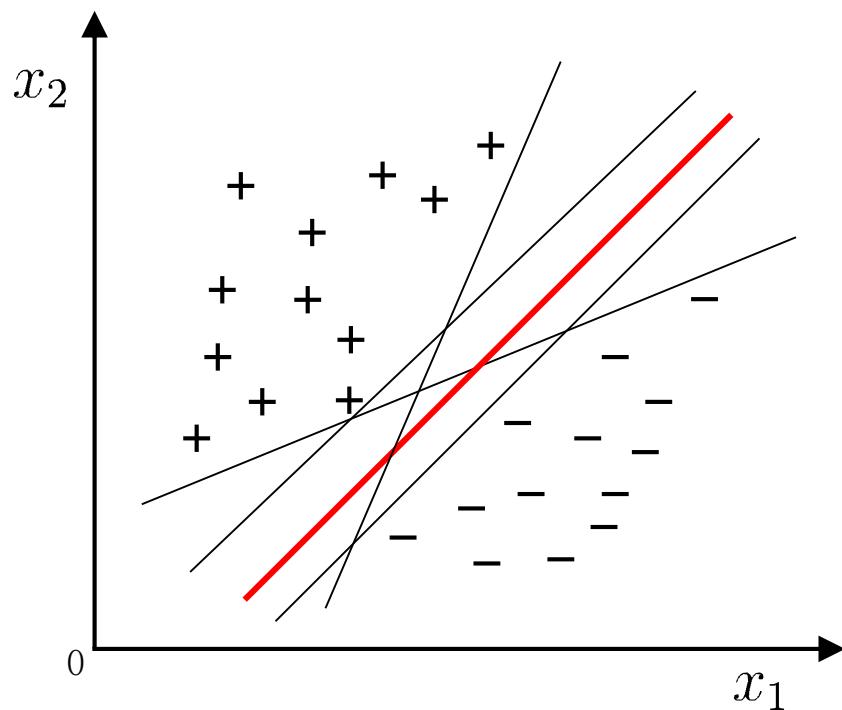
广义线性模型；
通过“联系函数”

例如，对率回归

支持向量机

线性分类器

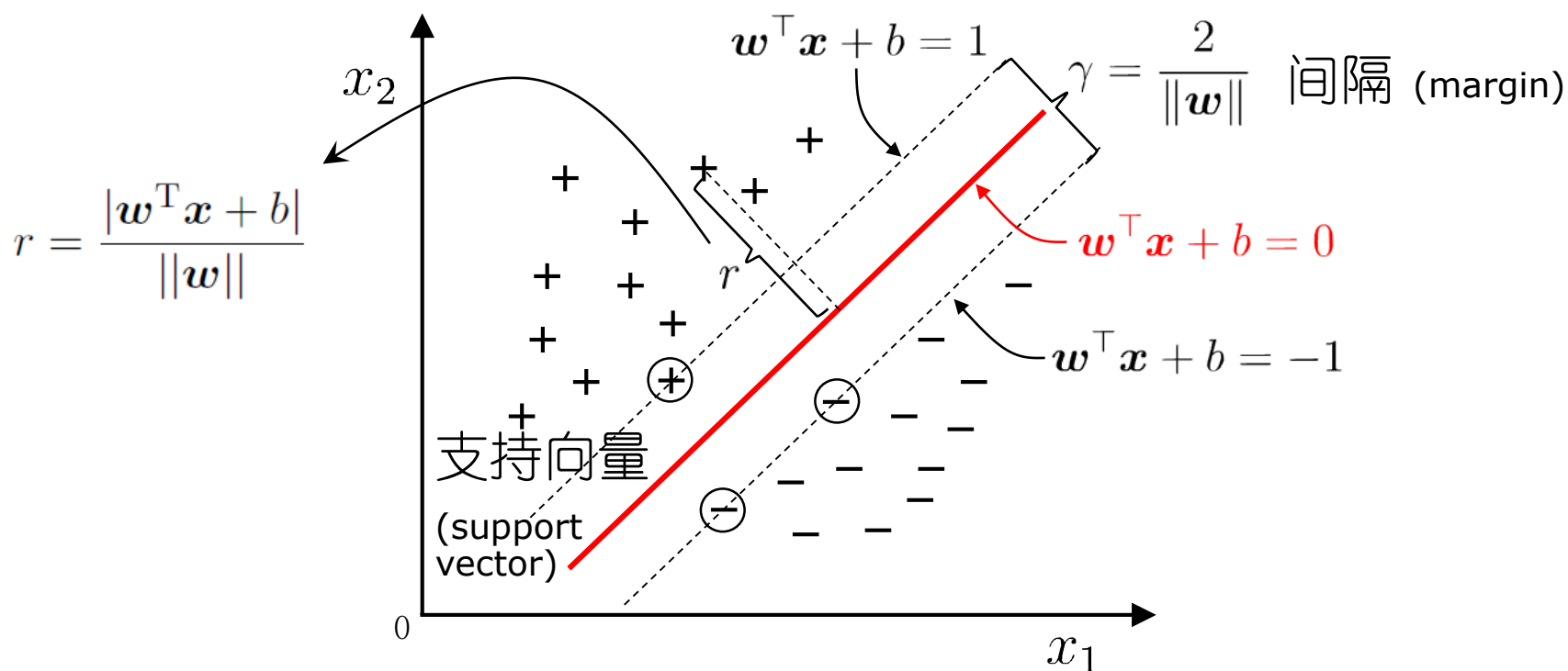
将训练样本分开的超平面可能有很多，哪一个更好呢？



“正中间”的：鲁棒性最好，泛化能力最强

间隔与支持向量

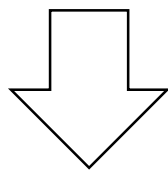
超平面方程: $w^T x + b = 0$



支持向量机基本型

最大间隔：寻找参数 \mathbf{w} 和 b ，使得 γ 最大

$$\begin{aligned} \arg \max_{\mathbf{w}, b} \quad & \frac{2}{\|\mathbf{w}\|} \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$



$$\begin{aligned} \arg \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$

凸二次规划问题

对偶问题

- 引入拉格朗日乘子 $\alpha_i \geq 0$ 得到拉格朗日函数

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$$

- 令 $L(\mathbf{w}, b, \boldsymbol{\alpha})$ 对 \mathbf{w} 和 b 的偏导为零可得

$$\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i, \quad 0 = \sum_{i=1}^m \alpha_i y_i$$

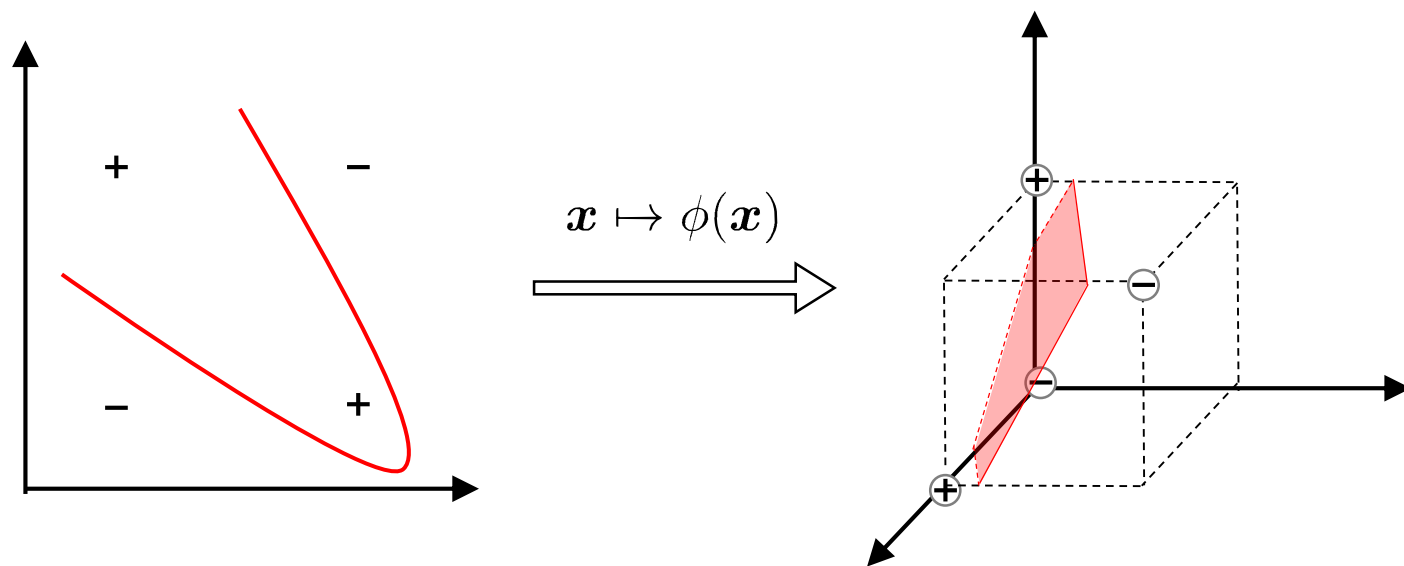
- 回代可得

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, 2, \dots, m \end{aligned}$$

特征空间映射

若不存在一个能正确划分两类样本的超平面, 怎么办?

将样本从原始空间映射到一个更高维的特征空间, 使样本在这个特征空间内线性可分



如果原始空间是有限维(属性数有限), 那么一定存在一个高维特征空间使样本可分

在特征空间中

设样本 \mathbf{x} 映射后的向量为 $\phi(\mathbf{x})$, 划分超平面为 $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$

原始问题

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$

对偶问题

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, 2, \dots, m \end{aligned}$$

预测

$$f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_{i=1}^m \alpha_i y_i \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}) + b$$

只以内积
形式出现

核函数 (kernel function)

基本思路：设计核函数

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

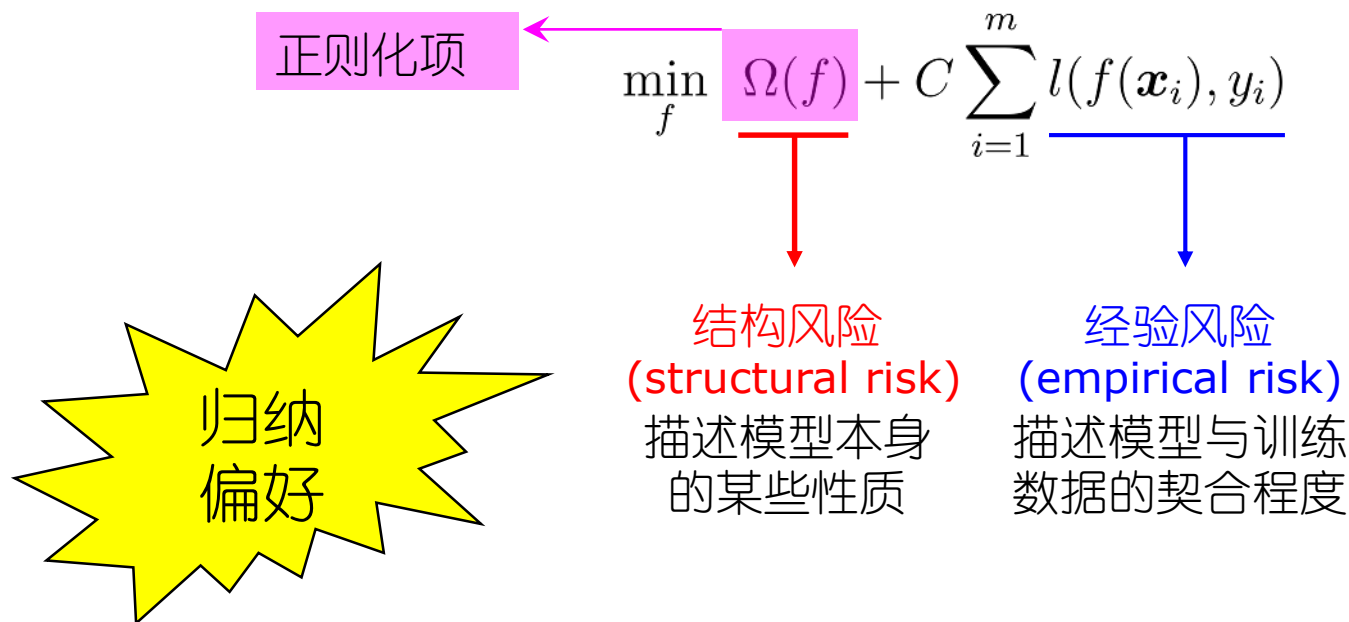
绕过显式考虑特征映射、以及计算高维内积的困难

Mercer 定理：若一个对称函数所对应的核矩阵**半正定**，则它就能作为核函数来使用

“核函数选择”成为决定支持向量机性能的关键！

正则化 (regularization)

统计学习模型（例如 SVM）的更一般形式



□ 正则化可理解为“罚函数法”

通过对不希望的结果施以惩罚，使得优化过程趋向于希望目标

□ 从贝叶斯估计的角度，则可认为是提供了模型的先验概率

决策树

基本流程

策略：“分而治之” (divide-and-conquer)

自根至叶的递归过程

在每个中间结点寻找一个“划分” (split or test) 属性

三种停止条件：

- (1) 当前结点包含的样本全属于同一类别，无需划分；
- (2) 当前属性集为空，或是所有样本在所有属性上取值相同，无法划分；
- (3) 当前结点包含的样本集合为空，不能划分。

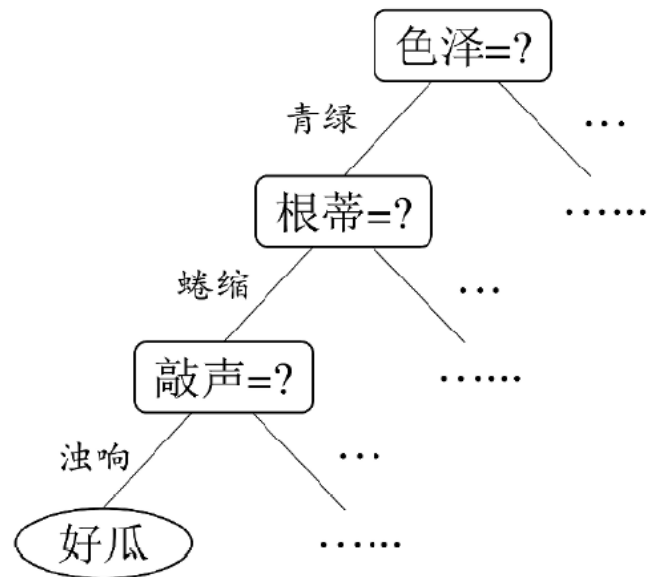


图 4.1 西瓜问题的一棵决策树

划分选择 vs. 剪枝

研究表明：划分选择的各种准则虽然对决策树的尺寸有较大影响，但对泛化性能的影响很有限

例如信息增益与基尼指数产生的结果，仅在约 2% 的情况下不同

剪枝方法和程度对决策树泛化性能的影响更为显著

在数据带噪时甚至可能将泛化性能提升 25%

Why?

剪枝 (pruning) 是决策树对付“过拟合”的主要手段！

缺失值

现实应用中，经常会遇到属性值“缺失”(missing)现象

仅使用无缺失的样例？ → 对数据的极大浪费

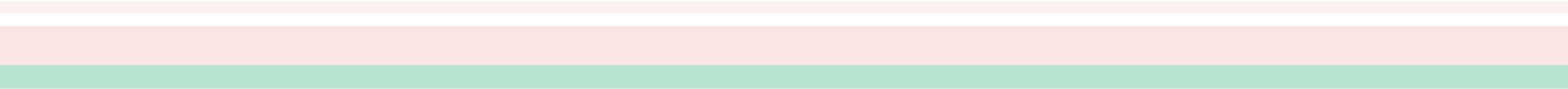
使用带缺失值的样例，需解决：

Q1：如何进行划分属性选择？

Q2：给定划分属性，若样本在该属性上的值缺失，如何进行划分？

基本思路：样本赋权，权重划分

神经网络

The bottom of the slide features three horizontal bars of equal width. The top bar is light red, the middle bar is white, and the bottom bar is light green.

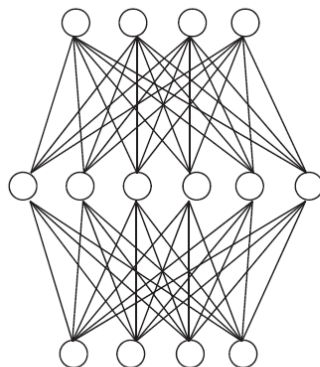
神经网络

神经网络是一个学科，本课程仅讨论它与机器学习的交集

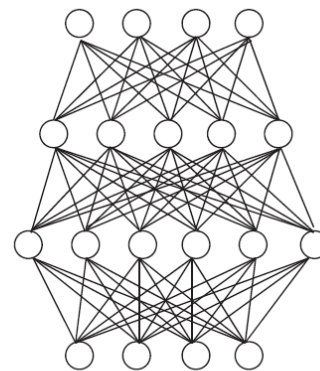
多层网络：包含隐层的网络

前馈网络：神经元之间不存在同层连接也不存在跨层连接

隐层和输出层神经元亦称“功能单元” (functional unit)



(a) 单隐层前馈网络



(b) 双隐层前馈网络

只需一个包含足够多神经元的隐层，多层前馈神经网络就能以任意精度逼近任意复杂度的连续函数 [Hornik et al., 1989]

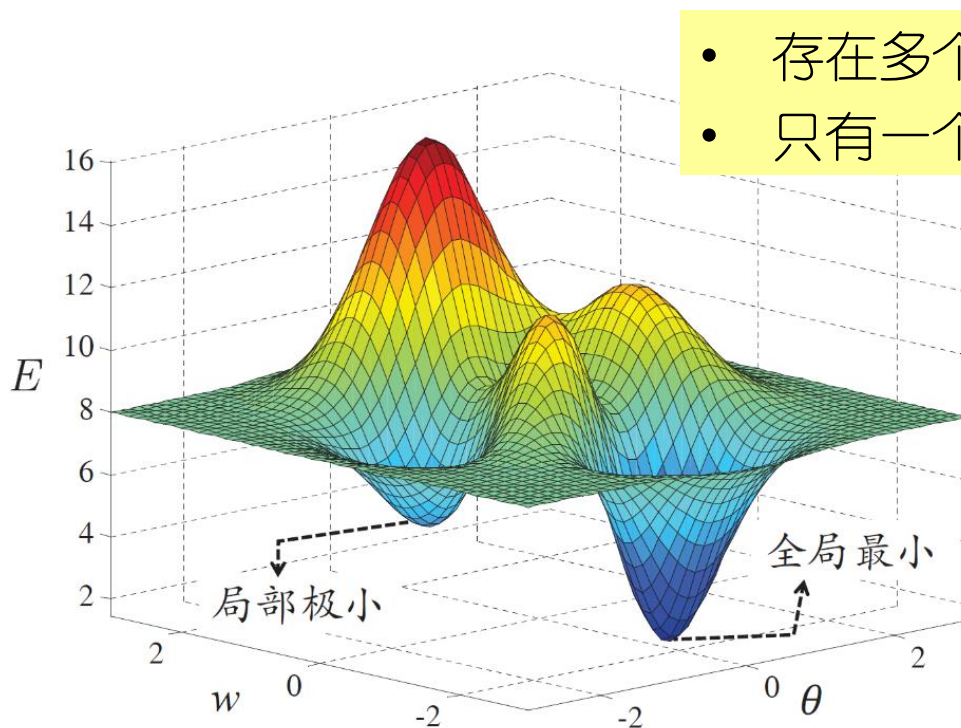
BP (BackPropagation; 误差逆传播) 算法：

最成功、最常用的神经网络算法，可被用于多种任务（不仅限于分类）

全局最小 vs. 局部极小

学习过程可看作一个参数寻优过程：

在参数空间中，寻找一组最优参数使得误差最小



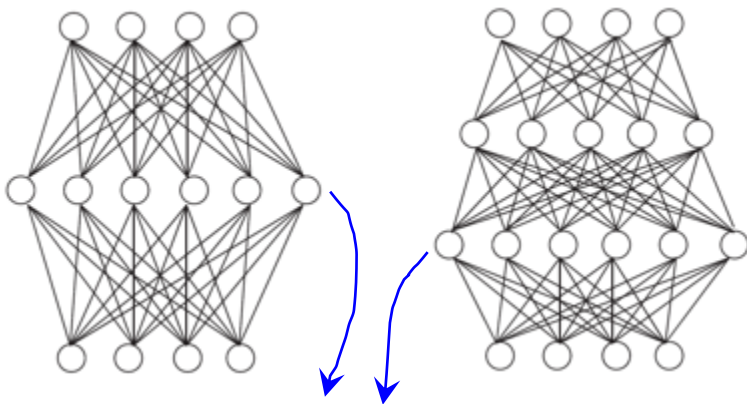
- 存在多个“局部极小”
- 只有一个“全局最小”

“跳出”局部极小的常见策略：

- ✓ 不同的初始参数
- ✓ 模拟退火
- ✓ 随机扰动
- ✓ 遗传算法
- ✓

深度神经网络

Traditional, single or double hidden layers



e.g., ImageNet winners:

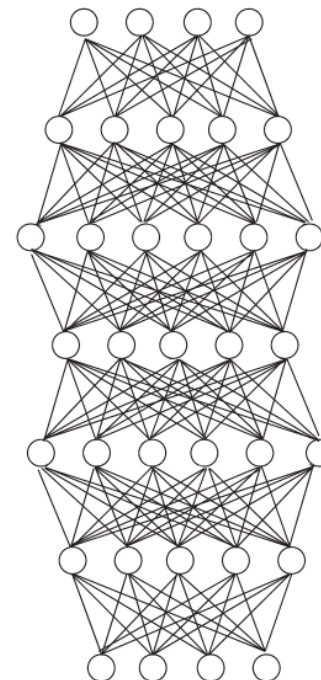
2012: 8 layer

2015: 152 layer

2016: 1207 layer

deep

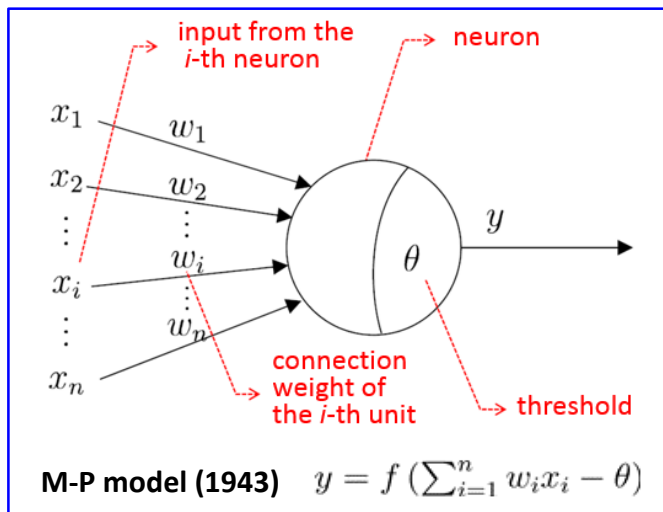
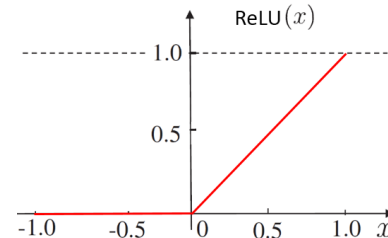
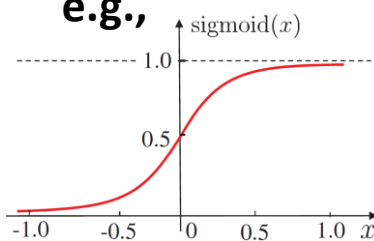
Many layers



Trained by
Backpropagation (BP)
or variant

f : continuous, differentiable

e.g.,



为何深？

提升模型复杂度 → 提升学习能力

- 增加隐层神经元数目（模型宽度）
- 增加隐层数目（模型深度）

增加隐层数目比增加隐层神经元数目更有效

不仅增加了拥有激活函数的神经元数，还增加了激活函数嵌套的层数

提升模型复杂度 → 增加过拟合风险；
增加计算开销

- 过拟合风险：使用大量训练数据
- 计算开销：使用强力计算设备

误差梯度在多隐层内传播时，往往会发散而不能收敛到稳定状态，因此，难以直接用经典BP算法训练

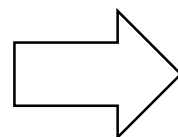
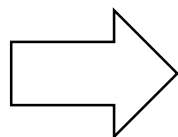
使用若干诀窍 (trick)

深度学习最重要的作用：表示学习

传统做法：



Feature Engineering
(特征工程)

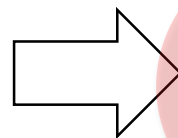


学习
分类

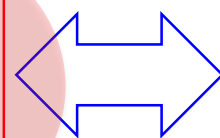
深度学习：



Representation learning
(表示学习)



关键



所谓
end-to-end
Learning
(端到端学习)

学习
分类

深度学习最重要的作用：表示学习

传统做法：

Feature Engineering

(特征工程)

深度学习何处适用？

数据的“初始表示”（例如，图像的“像素”）
与解决任务所需的“合适表示”相距甚远



初始

关键

所谓
end-to-end
Learning
(端到端学习)

刀大

贝叶斯分类器

贝叶斯决策论 (Bayesian decision theory)

概率框架下实施决策的基本理论

给定 N 个类别, 令 λ_{ij} 代表将第 j 类样本误分类为第 i 类所产生的损失, 则基于后验概率将样本 \mathbf{x} 分到第 i 类的条件风险为:

$$R(c_i | \mathbf{x}) = \sum_{j=1}^N \lambda_{ij} P(c_j | \mathbf{x})$$

贝叶斯判定准则 (Bayes decision rule):

$$h^*(\mathbf{x}) = \arg \min_{c \in \mathcal{Y}} R(c | \mathbf{x})$$

- h^* 称为贝叶斯最优分类器 (Bayes optimal classifier), 其总体风险称为贝叶斯风险 (Bayes risk)
- 反映了学习性能的理论上限

贝叶斯分类器

从这个角度来看，机器学习所要实现的是基于有限的训练样本尽可能准确地估计出后验概率

$$P(c | \mathbf{x})$$

贝叶斯公式?

$$P(c | \mathbf{x}) = \frac{P(c) P(\mathbf{x} | c)}{P(\mathbf{x})}$$

联合概率的计算将遭遇：

- 组合爆炸
- 样本稀疏
-

无法直接求解，需要设计有效/高效的算法

→ 机器学习

(考虑计算复杂度、样本复杂度……的数据分析)



Thomas Bayes
(1701?-1761)

判别式 vs. 生成式

$P(c | \mathbf{x})$ 在现实中通常难以直接获得

两种基本策略：

判别式 (discriminative) 模型

思路：直接对 $P(c | \mathbf{x})$ 建模

代表：

- 决策树
- BP 神经网络
- SVM

生成式 (generative) 模型

思路：先对联合概率分布 $P(\mathbf{x}, c)$ 建模，再由此获得 $P(c | \mathbf{x})$

$$P(c | \mathbf{x}) = \frac{P(\mathbf{x}, c)}{P(\mathbf{x})}$$

代表：贝叶斯分类器

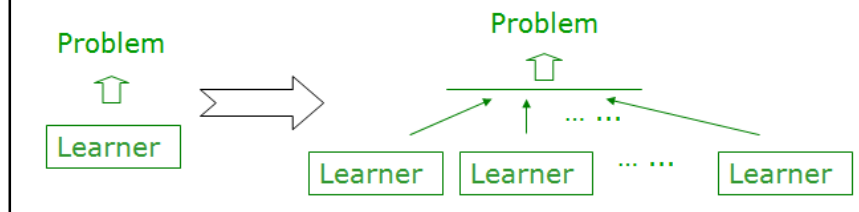
注意：贝叶斯分类器 \neq 贝叶斯学习
(Bayesian learning)

集成学习

集成学习

集成学习 (Ensemble Learning):

利用多个学习器解决问题



在现实任务中展现出极好的性能

- ❑ KDDCup'07: 1st place for "... Decision Forests and ..."
- ❑ KDDCup'08: 1st place of Challenge1 for a method using Bagging; 1st place of Challenge2 for "... Using an Ensemble Method "
- ❑ KDDCup'09: 1st place of Fast Track for "Ensemble ... "; 2nd place of Fast Track for "... bagging ... boosting tree models ...", 1st place of Slow Track for "Boosting ... "; 2nd place of Slow Track for "Stochastic Gradient Boosting"
- ❑ KDDCup'10: 1st place for "... Classifier ensembling"; 2nd place for "... Gradient Boosting machines ... "

- ❑ KDDCup'11: 1st place of Track 1 for "A Linear Ensemble ... "; 2nd place of Track 1 for "Collaborative filtering Ensemble", 1st place of Track 2 for "Ensemble ..."; 2nd place of Track 2 for "Linear combination of ..."
- ❑ KDDCup'12: 1st place of Track 1 for "Combining... Additive Forest..."; 1st place of Track 2 for "A Two-stage Ensemble of..."
- ❑ KDDCup'13: 1st place of Track 1 for "Weighted Average Ensemble"; 2nd place of Track 1 for "Gradient Boosting Machine"; 1st place of Track 2 for "Ensemble the Predictions"
- ❑ KDDCup'14: 1st place for "ensemble of GBM, ExtraTrees, Random Forest..." and "the weighted average"; 2nd place for "use both R and Python GBMs"; 3rd place for "gradient boosting machines... random forests" and "the weighted average of..."
- ❑ KDDCup'15: 1st place for "Three-Stage Ensemble and Feature Engineering for MOOC Dropout Prediction"
- ❑ KDDCup'16: 1st place for "Gradient Boosting Decision Tree"; 2nd place for "Ensemble of Different Models for Final Prediction"

近十年 KDDCup 获胜者，以及Netflix、Kaggle 等诸多数据分析竞赛的获胜者，几乎无一例外使用了集成学习

想获胜，用集成

很多成功的集成学习方法

■ 序列化方法

- **AdaBoost** [Freund & Schapire, JCSS97]
- GradientBoost [Friedman, AnnStat01]
- LPBoost [Demiriz, Bennett, Shawe-Taylor, MLJ06]
-

■ 并行化方法

- **Bagging** [Breiman, MLJ96]
- Random Forest [Breiman, MLJ01]
- Random Subspace [Ho, TPAMI98]
-

“多样性” (diversity) 是关键

误差-分歧分解 (error-ambiguity decomposition):

$$E = \bar{E} - \bar{A}$$

Diagram illustrating the error-ambiguity decomposition:

- E (Ensemble error) is represented by a black box.
- \bar{E} (Ave. error of individuals) is represented by a green box.
- \bar{A} (Ave. “ambiguity” of individuals) is represented by a red box.

Arrows point from the boxes to their respective labels:

- Black arrow from E to *Ensemble error*.
- Green arrow from \bar{E} to *Ave. error of individuals*.
- Red arrow from \bar{A} to *Ave. “ambiguity” of individuals*.

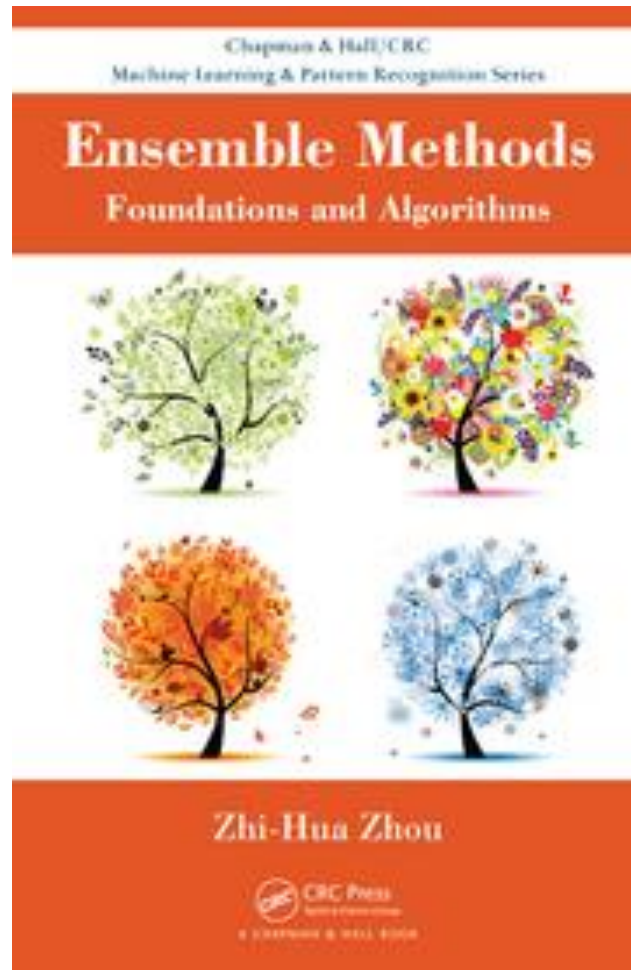
(“ambiguity” later called “diversity”)

The more **accurate** and **diverse** the individual learners,
the better the ensemble

However,

- the “ambiguity” does not have an operable definition
- The error-ambiguity decomposition is derivable only for regression setting with squared loss

更多关于集成学习的内容，可参考：



Z.-H. Zhou.
Ensemble Methods:
Foundations and Algorithms,
Boca Raton, FL: Chapman &
Hall/CRC, Jun. 2012.
(ISBN 978-1-439-830031)

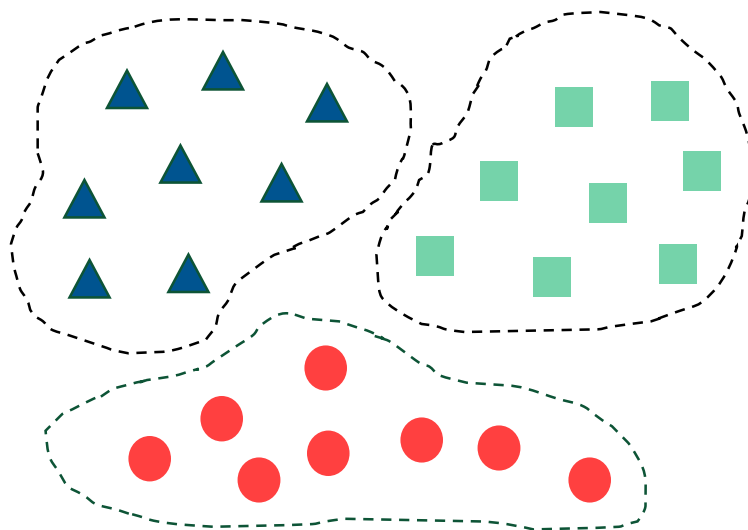
聚类

聚类 (Clustering)

在“无监督学习”任务中研究最多、应用最广

目标：将数据样本划分为若干个通常不相交的“簇”(cluster)

既可以作为一个单独过程（用于找寻数据内在的分布结构）
也可作为分类等其他学习任务的前驱过程



必须记住



聚类的“好坏”不存在绝对标准

**the goodness of clustering depends on
the opinion of the user**

聚类也许是机器学习中“新算法”出现最多、最快的领域
总能找到一个“标准”，使以往算法对它无能为力

前往下一站.....

