



# 机器学习

# Machine Learning

## 复习串讲

重庆大学计算机学院 何静媛

Email: [ibm\\_hjy@cqu.edu.cn](mailto:ibm_hjy@cqu.edu.cn)



## 第一章 绪论

1、**什么是机器学习**：通过算法使得机器能从大量数据中学习规律从而对新的样本做决策。（相当于构建一个映射函数）

2、**常见的机器学习问题**：回归（线性拟合）、分类、聚类（多分类）

3、**机器学习研究的主要内容**：

在计算机上从数据中产生“模型”的算法，即“学习算法”，我们把经验数据提供给它，它就能基于这些数据产生模型；在面对新的情况时，模型会给我们提供相应的判断

4、**常见的机器学习类型**：

- (1) 监督学习（学习准则为期望风险最小化、最大似然估计）
- (2) 无监督学习（最大似然估计、最小重构错误）



根据训练数据是否拥有标记信息，将学习任务分为监督学习和无监督学习。

**监督学习**：分类和回归

线性回归、对数几率回归、决策树、支持向量机、贝叶斯分类器、神经网络

- **无监督学习**：聚类

聚类算法：原型聚类：（k均值，学习向量量化，高斯混合聚类）；密度聚类（DBSCAN）；层次聚类（AGNES）。降维。话题分析。图分析。

- 机器学习的目标是使学得模型能很好地适用于“新样本”。
- 学得模型适用于新样本的能力，称为“**泛化能力**”

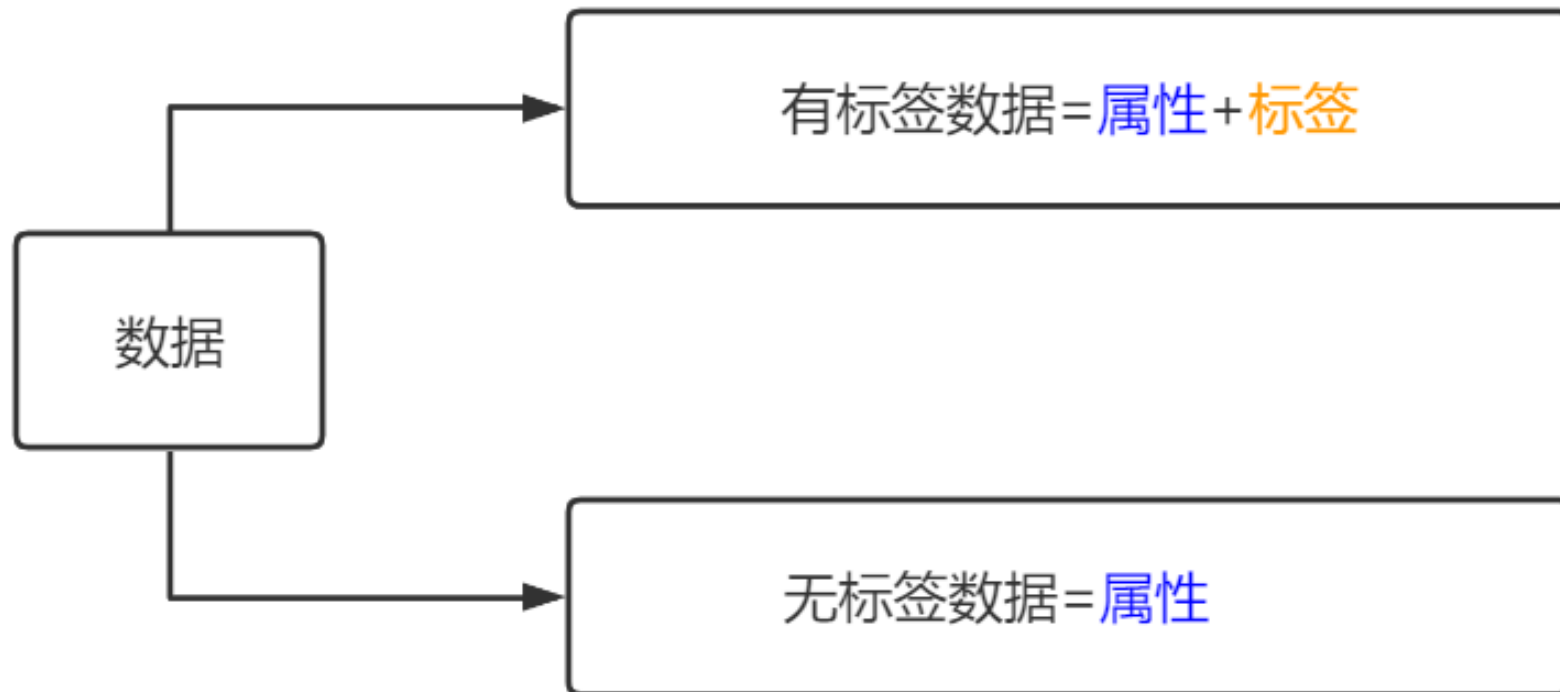


## 属性与标签

- 属性空间（又称为样本空间，输入空间）

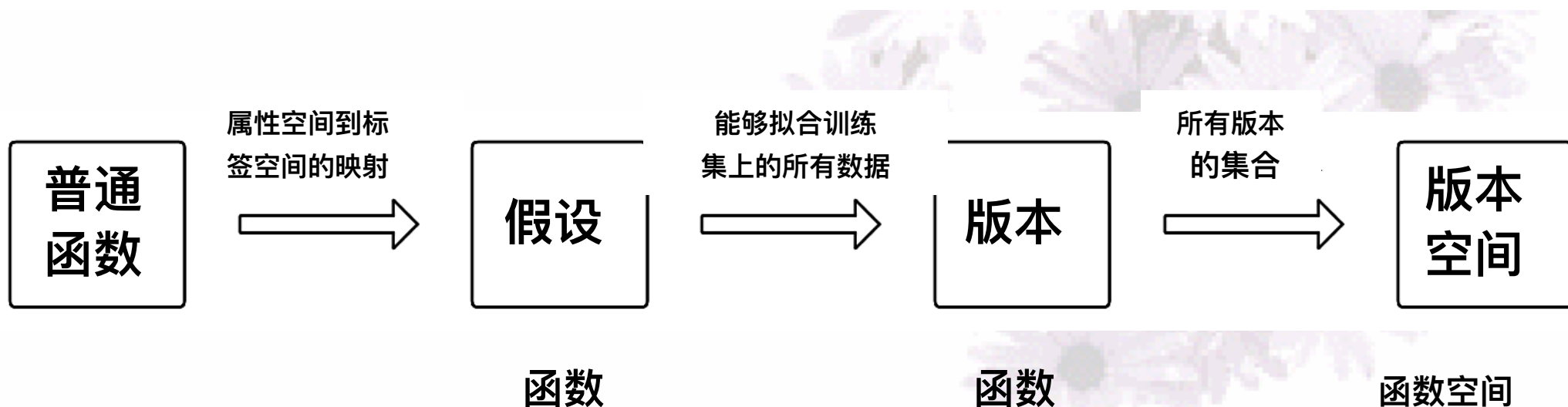
：描述数据的各个数据维度张成的空间

- 标签空间（又称为标记空间）：所有标记的集合



## 假设、版本和版本空间

- 假设：一个从有限属性空间到有限标签空间的映射。
- 版本：与训练集相符合，也就是能够将训练集上所有特征毫无错误的映射到对应标签的假设称为版本。
- 版本空间：所有版本的集合称为版本空间。



真实问题中，不一定存在版本，更别说版本空间了。但是存在许多近似满足版本条件的假设



## 第二章 模式评估与选择

### 三类误差

模型的误差分为训练误差，泛化误差和测试误差。

- 训练误差**：训练集上的训练所得模型与目标的差异
- 泛化误差**：在除去训练集的样本空间上训练所得模型与目标的差异
- 测试误差**：测试集上的训练所得模型与目标的差异

哪个指标是衡量模型性能最公平的指标？

答：泛化误差

为什么要谈测试误差？

答：因为泛化误差要遍历整个除去训练集的样本空间，现实情况下不可能（因为你连样本空间的边界都不知道，几乎所有样本空间都是不可数的。比如自动驾驶，你能列出所有可能的驾驶场景吗？）

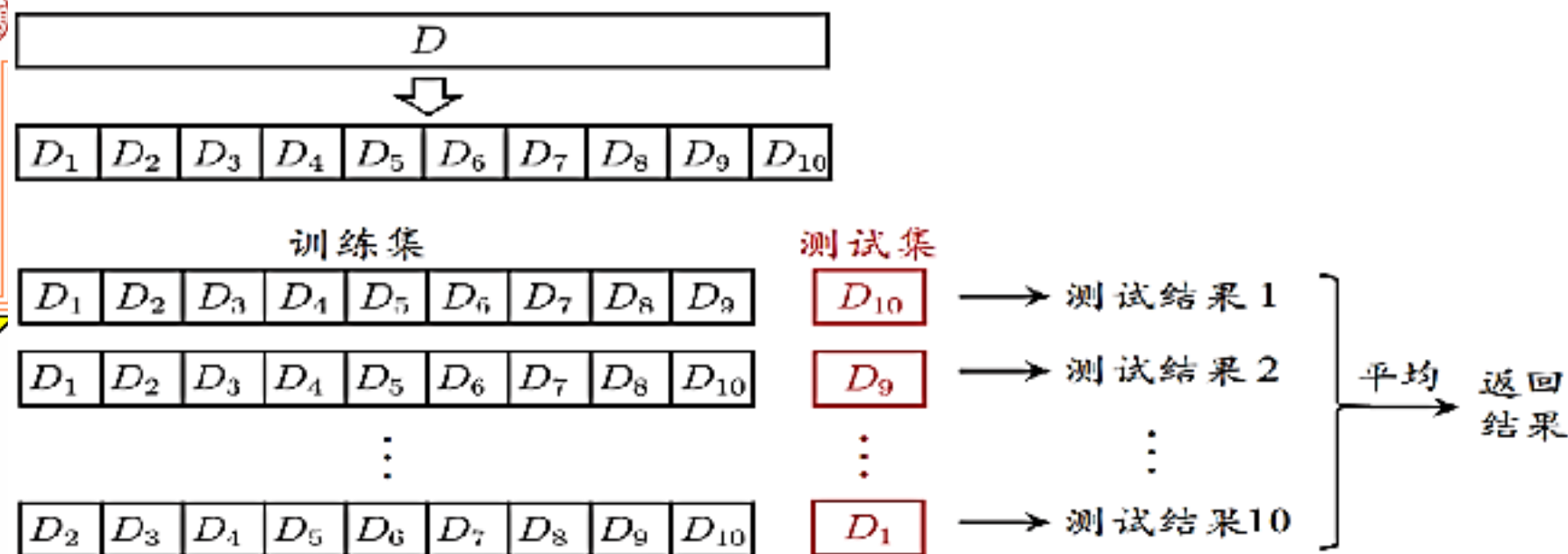




## 欠拟合与过拟合

- 欠拟合：训练误差大，测试误差大
- 过拟合：训练误差小，测试误差大

方法名称	操作流程	优点	缺点
留出法	留出 $\alpha$ 的数据作为测试集，剩下 $1-\alpha$ 作为训练集。	深度学习最常用的方法，简单易懂好操作，计算成本还小	但是结果可能可信度不算很高
K折交叉验证法	将原数据切成大小相同的K份，然后开始进行K重循环，第i次循环使用第i份的数据作为测试集，剩余K-1份数据作为训练集	这个方法的可信度比较高，还能利用seaborn画出比一般plot看起来NB很多的置信度曲线	计算量比较大，适合传统机器学习模型，不适合大型神经网络。
自助法采样	假设原数据集为D，大小为m，我们有放回地从D中依次采样m次，得到D'，将D'作为训练集，剩下的样本作为测试集	数据集较小时比较有效	改变了原数据的分布，引入了估计偏差，在数据集较大时会非常明显



10 折交叉验证示意图

### 自助法:

以自助采样法为基础, 对数据集 $D$ 有放回采样 $m$ 次得到训练集 $D'$ ,  $D, D'$ 用做测试集。

- 实际模型与预期模型都使用 $M$ 个训练样本
- 约有 $1/3$ 的样本没在训练集中出现

$$\lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m \mapsto \frac{1}{e} \approx 0.368,$$

- 从初始数据集中产生多个不同的训练集, 对集成学习有很大的好处



# 错误率&精度

对于分类任务,错误率和精度是最常用的两种性能度量:

- 错误率: 分错样本占样本总数的比率
- 精度 (正确率) : 分对样本占样本总数的比率

## 查准率&查全率

信息检索、Web搜索等场景中经常需要衡量正例被预测出来的比率或者预测出来的正例中正确的比率,此时查准率和查全率比错误率和精度更适合。

统计真实标记和预测结果的组合可以得到“混淆矩阵”:

分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	$TP$ (真正例)	$FN$ (假反例)
反例	$FP$ (假正例)	$TN$ (真反例)

查准率  
(精确率, precision)  $P = \frac{TP}{TP + FP}$

查全率  
(召回率, recall)  $R = \frac{TP}{TP + FN}$

查准率: 在预测结果中, 预测出的真实正例所占所有预测正例中的比例 (竖着来)

查全率: 在真实情况中, 预测出的真实正例所占所有真实情况中的比例 (横着来)

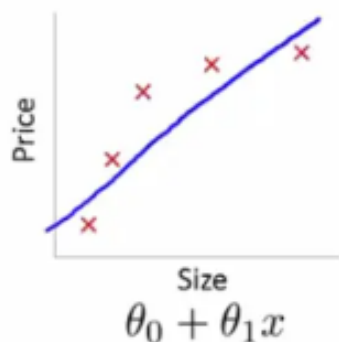


结论：对于采用MSE作为损失函数的回归任务来说，对于数据集  $D$  和已经学好的模型  $f$ ，假设  $y_D$  是  $x$  在数据集上的标记， $y$  是  $x$  的真实标记，那么标签噪声为  $\epsilon^2 = \mathbb{E}_D[(y_D - y)^2]$ ，且均值假设为0。则  $f$  在  $D$  上的期望泛化误差  $E(f; D)$  可作如下分解：

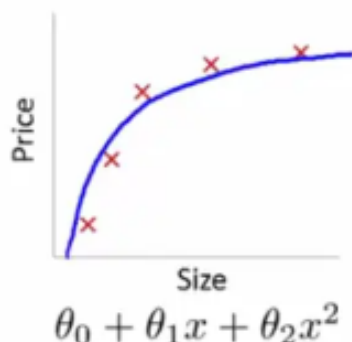
$$E(f; D) = \text{bias}^2(x) + \text{var}(x) + \epsilon^2$$

也就是说：泛化误差可分解为偏差、方差与噪音之和。

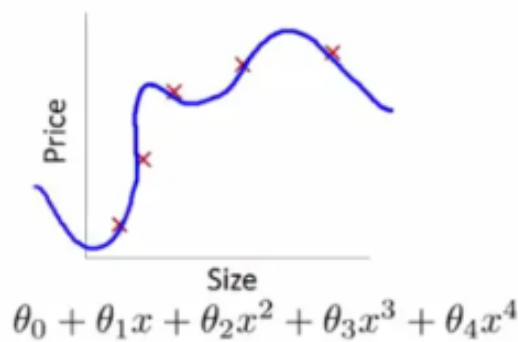
其中：  $\text{bias}^2(x) = \mathbb{E}_D[(f(x; D) - \bar{f}(x))^2]$ ， $\text{var}(x) = (\bar{f}(x) - y)^2$ ，  
 $\bar{f}(x) = \mathbb{E}_D[f(x; D)]$



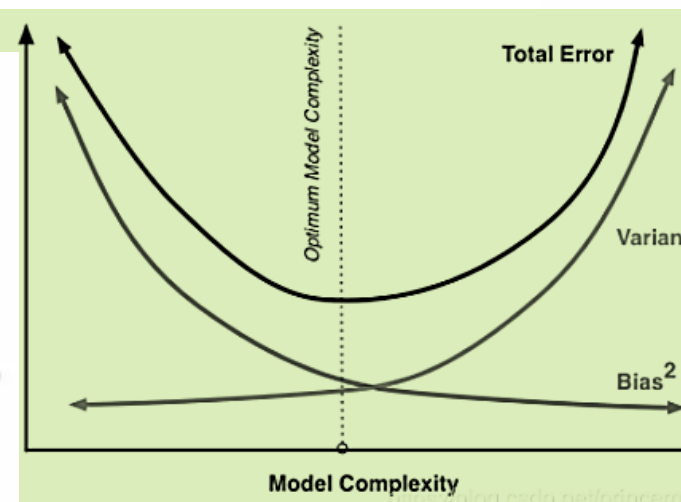
High bias  
(underfit)  
 $d=1$



"Just right"  
 $d=2$



High variance  
(overfit)  
 $d=4$





## 第三章 线性模型

### 1.基本形式

什么是线性模型？通过属性的线性组合构成预测函数，该函数就是属于线性模型

$$f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + \dots + w_d x_d + b$$

向量形式是  $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$ ，这里的  $\mathbf{w}$  和  $\mathbf{b}$  学习后即可确定模型。几种经典的线性模型有：线性回归<sup>Q</sup>、对数几率回归、线性判别分析。

### 2.线性回归

数据集  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ ，其中  $\mathbf{x}_i = (x_{i1}; x_{i2}; \dots; x_{id})$ ,  $y_i \in \mathbb{R}$

一元线性回归即

$$f(x_i) = wx_i + b, \text{ 使得 } f(x_i) \simeq y_i$$

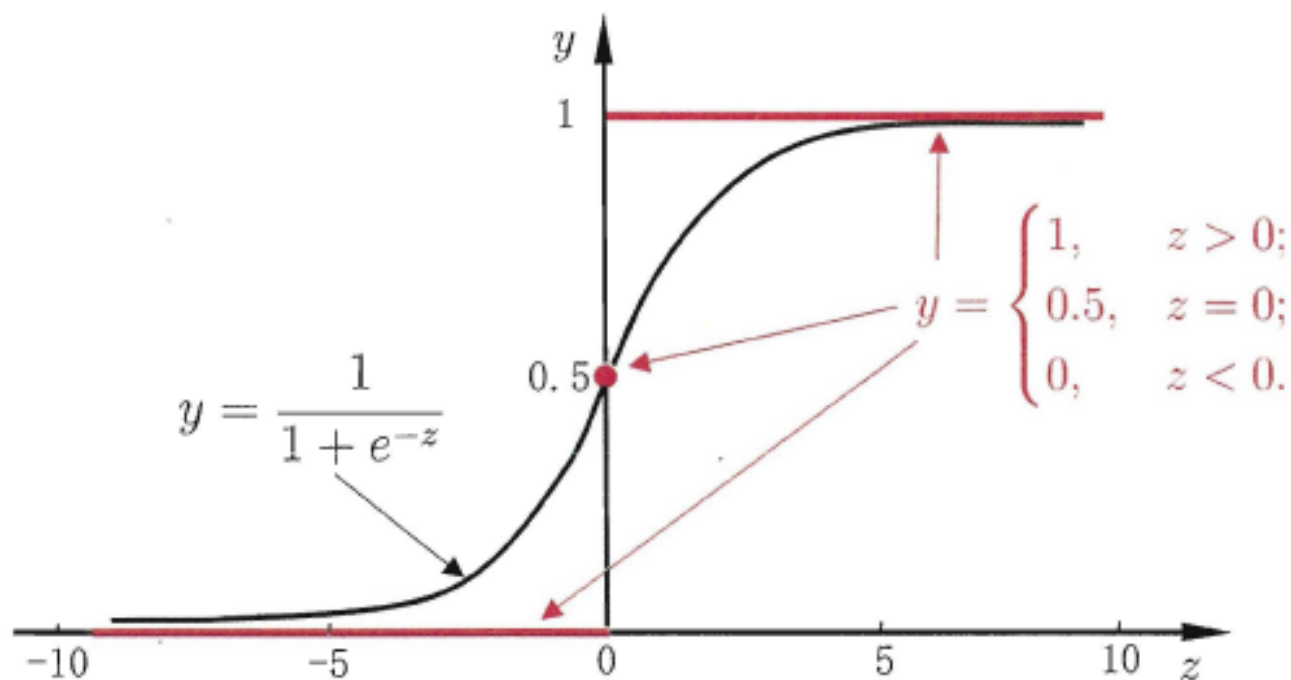
模型确定，如何去选择  $w$  和  $b$ ，这里选择均方误差作为损失函数，均方误差是回归任务中最常用的性能度量，目标即均方误差最小化，即

$$(w^*, b^*) = \arg \min_{w, b} \sum_{i=1}^m (f(x_i) - y_i)^2$$



### 3、线性几率回归

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



$$y = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}} \rightarrow \ln \frac{y}{1 - y} = \mathbf{w}^T \mathbf{x} + b .$$



若将 $y$ 看做样本为正例的概率， $(1-y)$ 看做样本为反例的概率，则上式实际上使用线性回归模型的预测结果逼近真实标记的对数几率。因此这个模型称为“对数几率回归” (logistic regression)，也有一些书籍称之为“逻辑回归”。下面使用最大似然估计的方法来计算 $w$ 和 $b$ 两个参数的取值，下面只列出求解的思路，不列出具体的计算过程。

$$p(y=1 | x) = 1 - p(y=0 | x)$$

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

$$p(y=1 | x) = \frac{e^{w^T x + b}}{1 + e^{w^T x + b}} \quad \text{正例}$$

$$p(y=0 | x) = \frac{1}{1 + e^{w^T x + b}} \quad \text{负例}$$

$$\ell(w, b) = \sum_{i=1}^m \ln p(y_i | x_i; w, b)$$

对数变乘为加

→ 最大化熵然

即所有样本出现真实值的概率乘积最大

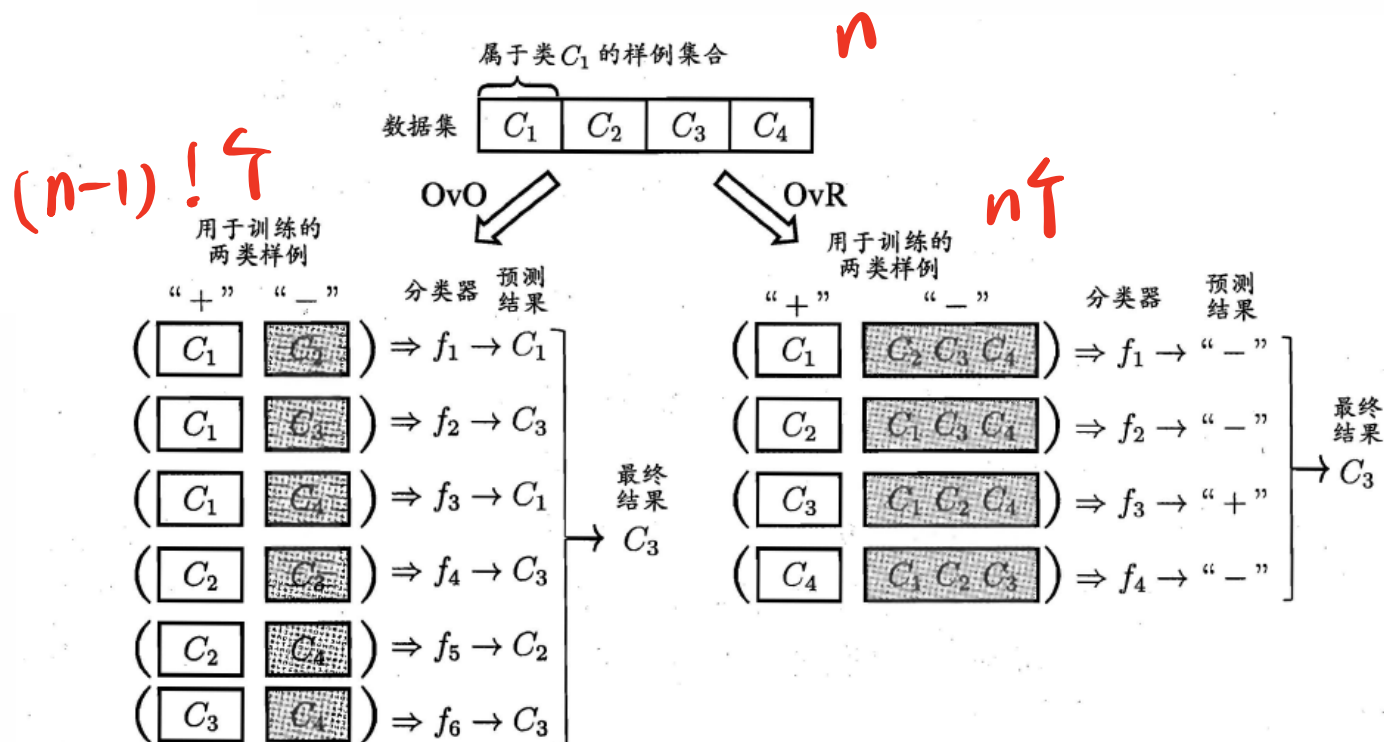


## 4、多分类学习

[?] OvO:

[?] OvM:

[?] MvM:



	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	海明距离	欧氏距离
$C_1 \rightarrow$	-1	+1	-1	+1	+1	3	$2\sqrt{3}$
$C_2 \rightarrow$	+1	-1	-1	+1	-1	4	4
$C_3 \rightarrow$	-1	+1	+1	-1	+1	1	2
$C_4 \rightarrow$	-1	-1	+1	+1	-1	2	$2\sqrt{2}$
测试示例 $\rightarrow$	-1	-1	+1	-1	+1		

(a) 二元 ECOC 码

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	海明距离	欧氏距离
$C_1 \rightarrow$	-1	-1	+1	+1	-1	+1	+1	4	4
$C_2 \rightarrow$	-1				+1	-1		2	2
$C_3 \rightarrow$	+1	+1	-1	-1	-1	+1	-1	5	$2\sqrt{5}$
$C_4 \rightarrow$	-1	+1		+1	-1		+1	3	$\sqrt{10}$
测试示例 $\rightarrow$	-1	+1	+1	-1	+1	-1	+1		

(b) 三元 ECOC 码





## 5、类别不平衡问题

### 类别不平衡 (class-imbalance)

) 就是指分类问题中不同类别的训练样本相差悬殊的情况, 例如正例有900个, 而反例只有100个, 这个时候我们就需要进行相应的处理来平衡这个问题。常见的做法有三种:

[?] 在训练样本较多的类别中进行“欠采样”(undersampling), 比如从正例中采出100个。

[?] 在训练样本较少的类别中进行“过采样”(oversampling), 例如通过对反例中的数据进行插值, 来产生额外的反例。

[?]

直接基于原数据集进行学习, 对预测值进行“再缩放”处理。其中再缩放也是代价敏感学习的基础。

$$\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+} \rightarrow \frac{\text{cost}(+ \rightarrow -)}{\text{cost}(- \rightarrow +)} \quad \text{即代价敏感}$$



## 第4章- 决策树

### 4、信息增益 (ID3算法为代表)

- (1) **信息熵**: 用于度量样本集合“纯度”, 其值越小, 则样本集合的纯度越高, 当其等于零时, 达到最大
- (2) **信息增益**: 对可取值数目较多的属性有所偏好, 其值最大时, 则被选为划分属性。  
缺点时可能会把“编号”也作为一个属性
- (3) **增益率** (C4.5算法使用): 中和或平衡掉只使用信息增益对属性数目多的偏好, 缺点是会取值数目少的属性有所偏好。则可以先从候选划分属性中找出信息增益高于平均水平的属性, 再从中选取增益率最高的
- (4) **基尼指数** (衡量纯度的另一种方法): 反映了从  $D$  中随机抽取两个样例, 其类别标记不一致的概率, 值越小, 则数据集的纯度越高。在候选属性集合中, 选取那个使划分后基尼指数最小的属性, 在CART算法中使用



## 5、划分选择 vs. 剪枝 对泛化性能的影响

- (1) 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小
- (2) 剪枝方法和程度对决策树泛化性能的影响更为显著
- (3) 剪枝是决策树对付“过拟合”的主要手段

**预剪枝：**提前终止某些分支的生长

**后剪枝：**生成一棵完全树，再“回头”剪枝

剪枝过程中需用测试集来评估剪枝前后决策树的优劣，划分后精度增大才允许划分

仅有一层划分的决策树，称为“决策树桩”

## 6、预剪枝 vs. 后剪枝





## 6、预剪枝 vs. 后剪枝

### (1) 时间开销:

预剪枝: 训练时间开销降低, 测试时间开销降低

后剪枝: 训练时间开销增加很多, 测试时间开销降低

### (2) 过/欠拟合风险:

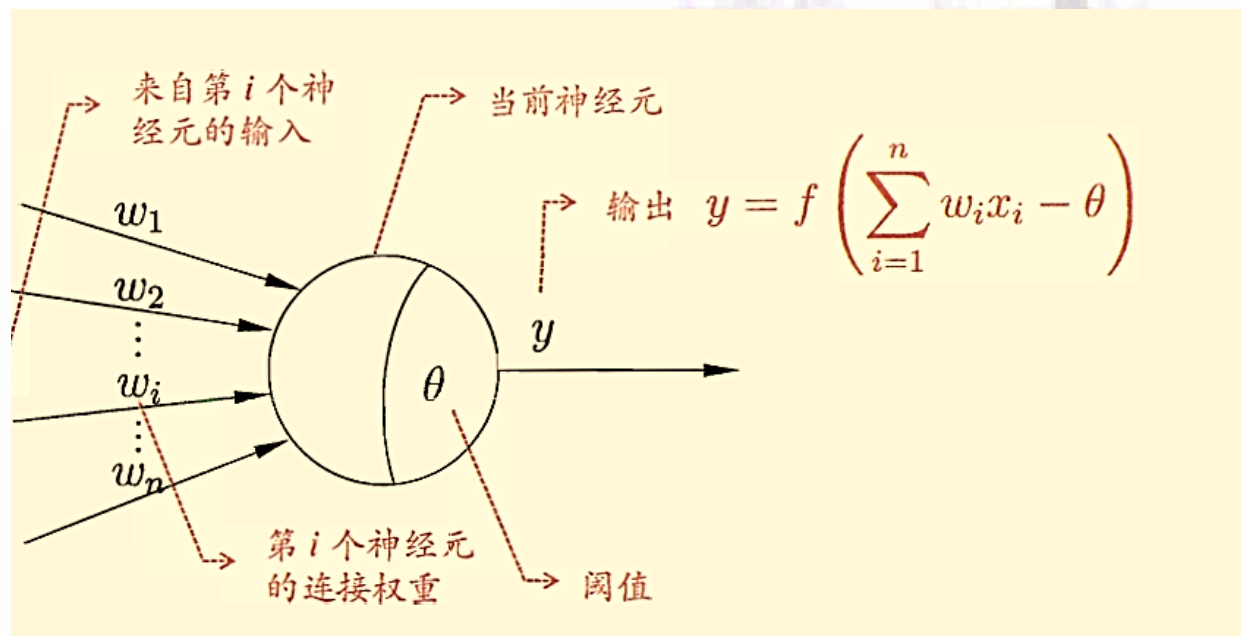
预剪枝: 过拟合风险降低, 欠拟合风险增加

后剪枝: 过拟合风险降低, 欠拟合风险很小, 基本不变

泛化性能: 后剪枝通常优于预剪枝



## 第五章 神经网络



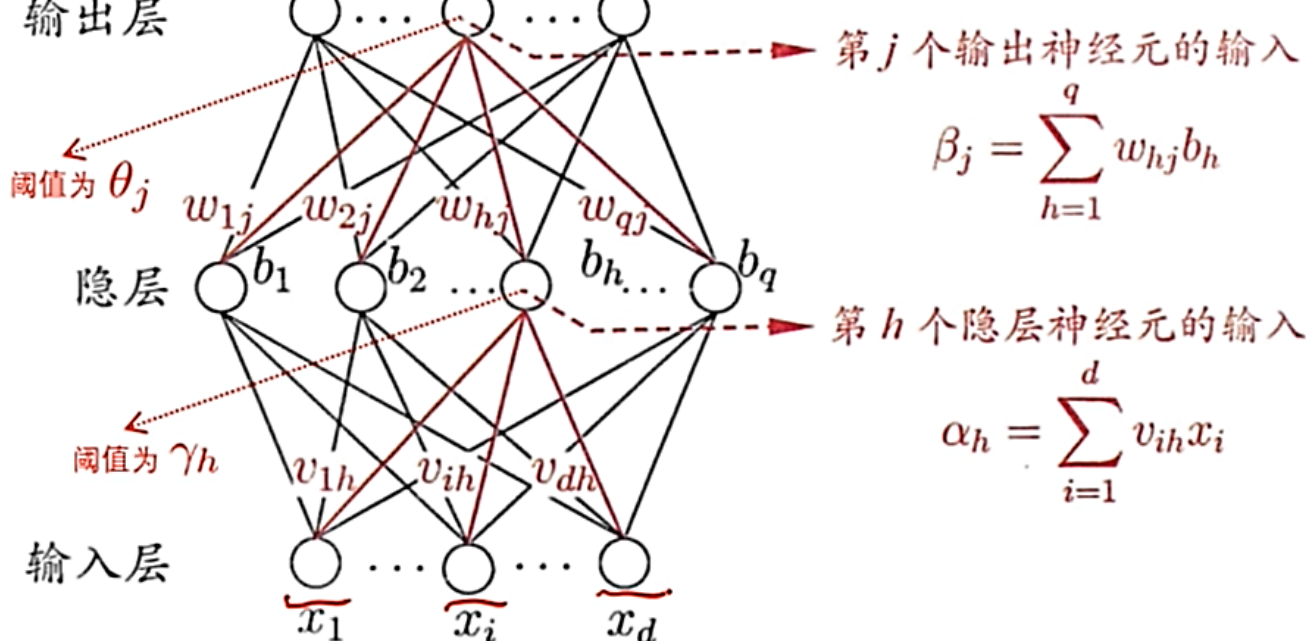
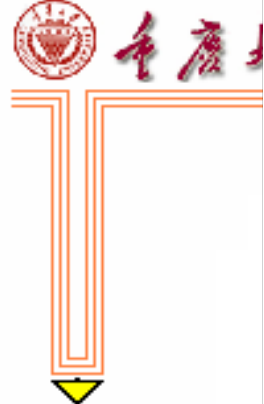
单个M-P神经元：感知机（sgn作为激活函数）、对数几率回归（sigmoid作激活函数）

多个M-P神经元：神经网络

感知机模型：激活函数为sgn（阶跃函数）的神经元

$$y = \text{sgn}(\mathbf{w}^T \mathbf{x} - \theta) = \begin{cases} 1, & \mathbf{w}^T \mathbf{x} - \theta \geq 0 \\ 0, & \mathbf{w}^T \mathbf{x} - \theta < 0 \end{cases}$$





$$\beta_j = \sum_{h=1}^q w_{hj} b_h$$

$$\alpha_h = \sum_{i=1}^d v_{ih} x_i$$

## 5、误差逆传播算法 (BP)

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）

(1) **标准BP算法**：每次针对单个训练样例更新权值与阈值；参数更新频繁，不同样例可能抵消，需要多次迭代

(2) **累积 BP 算法**：其优化目标是最小化整个训练集上的累计误差；读取整个训练集一遍才对参数进行更新，参数更新频率较低

累计误差下降到一定程度后，进一步下降会非常缓慢，这时标准BP算法往往会获得较好的解，尤其当训练集非常大时效果更明显。





BP算法的更新规则是基于每个样本的预测值与真实类标的均方误差来进行权值调节，即BP算法每次更新只针对于单个样例。

需要注意的是：BP算法的最终目标是要最小化整个训练集D上的累积误差，即：

$$E = \frac{1}{m} \sum_{k=1}^m E_k$$

前面提到，BP神经网络强大的学习能力常常容易造成过拟合问题，有以下两种策略来缓解BP网络的过拟合问题：

- 早停：将数据分为训练集与测试集，训练集用于学习，测试集用于评估性能，若在训练过程中，训练集的累积误差降低，而测试集的累积误差升高，则停止训练。
- 引入正则化 (regularization)：基本思想是在累积误差函数中增加一个用于描述网络复杂度的部分，例如所有权值与阈值的平方和， $E_k$  表示第k个训练样例上的误差， $w_i$  表示连接权和阈值，则误差目标函数可表示为：

$$E = \lambda \frac{1}{m} \sum_{k=1}^m E_k + (1 - \lambda) \sum_i w_i^2$$



## 第六章 支持向量机

### 1、间隔与支持向量 (SVM) :

- (1) 支持向量机在文本分类任务中显示出卓越性能, 成为了机器学习的主流技术 (统计学习)
- (2) 线性模型: 在样本空间中寻找一个超平面, 将不同类别的样本分开.
- (3) 将训练样本分开的超平面可能有很多, 应选择“正中间”, 容忍性好, 鲁棒性高, 泛化能力最强的
- (4) SVM想要的就是找到各类样本点到超平面的距离最远, 也就是找到最大间隔超平面 (支撑向量), 任意超平面可以用线性方程来描述

## 2、核函数

基本想法：不显式地设计核映射，而是设计核函数。即 $x_i$ 与 $x_j$ 在特征空间的内积等于它们在原始样本空间中通过函数 计算的结果。有了这样的函数，就不必直接去计算高维甚至无穷维特征空间中的内积

(1) 在不知道特征映射的形式时，我们并不知道什么样的核函数是合适的。若核函数选择不合适，则意味着将样本映射到了一个不合适的特征空间，很可能导致性能不佳

(2) 常用核函数：线性核、多项式核、高斯核、拉普拉斯核、Sigmoid核

(3) 核函数选择成为svm（支持向量机）的最大变数

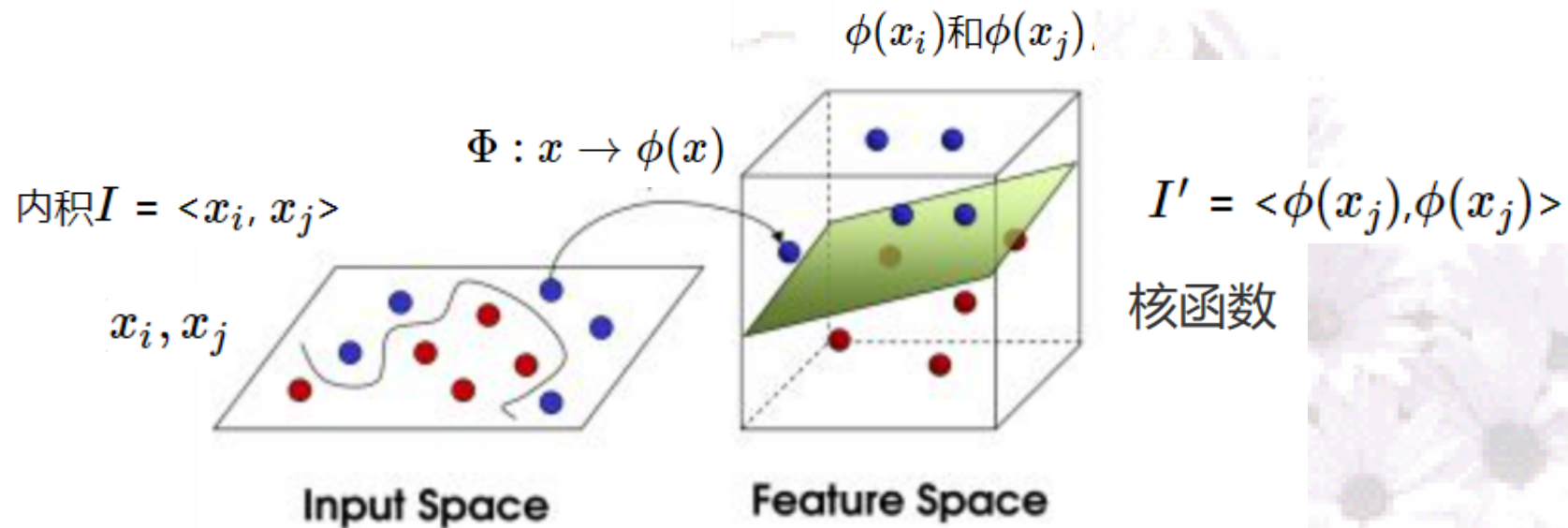
经验：文本数据使用线性核，情况不明使用高斯核

核函数的性质：

1 核函数的线性组合仍为核函数

2 核函数的直积仍为核函数

设 $k(x_1, x_2)$ 为核函数，则对于任意函数 $g$ ， $g(x_1)k(x_1, x_2)g(x_2)$ 仍为核函数



- 核函数的基本作用就是接受两个低维空间里的向量，能够计算出经过某个变换后在高维空间里的向量内积值。

能不能在原始空间找到一个函数  $K(x_i, x_j)$  使得  $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$  呢？



## 第7章 贝叶斯分类器

在机器学习中，对于有监督学习可以将其分为两类模型：**判别式模型**和**生成式模型**。

假设我们有训练数据 $(X, Y)$ ,  $X$ 是属性集合,  $Y$ 是类别标记。这时来了一个新的样本样本  $x$ ，我们想要预测它的类别 $y$ 。

我们最终的目的是求得**最大的条件概率**  $P(y|x)$  作为新样本的分类。

判别式模型是对条件概率建模，学习不同类别之间的最优边界，无法反映训练数据本身的特性，能力有限，其只能告诉我们分类的类别。



决策树



神经网络



支持向量机

贝叶斯分类器是生

成式模型






朴素贝叶斯分类器采用了“属性条件独立性假设”

贝叶斯决策论考虑如何基于这些概率和误判损失来选择最优的类别标记。

贝叶斯判定准则：为最小化总体风险，只需在每个样本上选择那个能使条件风险 $R(c|x)$ 最小的类别标记。


$$P(c|x) = P(c) \prod_{i=1}^d P(x_i|c)$$

## EM算法

EM算法提供一种近似计算含有隐变量概率模型的极大似然估计方法。

求解：如果最后函数是收敛的，则不能保证收敛得到全局极大值，能够收敛到局部大值。





- 求  $x=(2,S)^T$  的类标记  $Y$ ;

$$P(C|x) = \frac{P(C)}{P(x)} \cdot \prod P(x_i|C)$$

$$P(C|x) = P(C) \cdot \prod_{i=1}^n P(x_i|C)$$

① 计算类先验概率 (样本中正负所占比例)

$$P(C=1) = \frac{9}{15}, \quad P(C=-1) = \frac{6}{15}$$

② 计算条件概率

$$P(x_1=1|C=1) = \frac{2}{9}$$

$$P(x_1=2|C=1) = \frac{3}{9}$$

$$P(x_1=3|C=1) = \frac{4}{9}$$

$$P(x_2=S|C=1) = \frac{1}{9}$$

$$P(x_2=M|C=1) = \frac{4}{9}$$

$$P(x_2=L|C=1) = \frac{4}{9}$$

$$P(x_1=1|C=-1) = \frac{3}{6}$$

$$P(x_1=2|C=-1) = \frac{2}{6}$$

$$P(x_1=3|C=-1) = \frac{1}{6}$$

$$P(x_2=S|C=-1) = \frac{3}{6}$$

$$P(x_2=M|C=-1) = \frac{2}{6}$$

$$P(x_2=L|C=-1) = \frac{1}{6}$$

③ 计算后验概率 (先验概率  $\times$  条件概率)

$$P(C=1|x) = P(C=1) \times P(x_1=2|C=1) \times P(x_2=S|C=1)$$

$$= \frac{9}{15} \times \frac{3}{9} \times \frac{1}{9} = \frac{1}{45}$$

$$P(C=-1|x) = P(C=-1) \times P(x_1=2|C=-1) \times P(x_2=S|C=-1)$$

$$= \frac{6}{15} \times \frac{2}{6} \times \left(\frac{3}{6}\right) = \frac{1}{15}$$

$$\therefore \frac{1}{15} > \frac{1}{45}$$

$x=(2,S)^T$  的类标记  $Y$  为 -1

$$\textcircled{1} P(C=1) = \frac{9}{15}$$

$$P(C=-1) = \frac{6}{15}$$

$$\textcircled{2} P(x_1=2|C=1) = \frac{3}{9}$$

$$P(x_2=S|C=1) = \frac{1}{9}$$

$$P(x_1=2|C=-1) = \frac{2}{6}$$

$$P(x_2=S|C=-1) = \frac{3}{6}$$

$$\therefore P(C=1|x_1=2, x_2=S)$$

$$= \frac{P(C=1)}{P(x)} \cdot P(x_1=2, x_2=S|C=1)$$

$$= \frac{P(C=1)}{P(x)} \cdot P(x_1=2|C=1) \cdot P(x_2=S|C=1) = \frac{1}{P(x)} \cdot \frac{9}{15} \cdot \frac{3}{9} \cdot \frac{1}{9}$$

$x_1$	$x_2$	$Y$
1	S	-1
1	M	-1
1	M	1
1	S	1
1	S	-1
2	S	-1
2	M	-1
2	M	1
2	L	1
2	L	1
3	L	1
3	M	1
3	M	1
3	L	1
3	L	-1



## 第8章 集成学习

$$= \frac{1}{p(x)} \cdot \frac{1}{45}$$

同理:  $p(c=-1 | x_1=2, x_2=5)$

$$= \frac{1}{p(x)} \cdot \frac{1}{15}$$

集成学习的结果通过投票法产生。即少数服从多数

个体学习器应“**好而不同**”，即个体学习器要有一定的“**准确性**”，即学习器不能太坏，并且要有“**多样性**”。

目前集成学习主要分为两大类：

一类是以 **boosting**、**Adaboost** 等算法为代表的，个体学习器间存在强依赖关系，必须串行生成的 **序列化** 方法，它试图不断增强单个学习器的学习能力。

一类是以 **bagging**、“**随机森林**” (Random Forest) 等算法为代表的，个体学习器之间不存在强依赖关系、可同时生成的 **并行化** 方法。



### Boosting的工作流程

- (1) 先从初始训练集训练出一个基学习器；
- (2) 根据基学习器的表现对训练样本分布进行调整，使得先前基学习器分错的训练样本在后续得到更多关注，然后再基于调整后的样本分布来训练下一个基学习器；
- (3) 重复 (2)，直到基学习器数目达到指定值T，最终将这T个基学习器进行加权组合。

从偏差-方差分解的角度看，Boosting 主要关注降低 **偏差**



## Bagging的工作流程

(1) Bagging的基本流程:

通过自助采样法采样出T个含m个训练样本的采样集, 然后基于每个采样集训练出一个基学习器, 再将这些基学习器进行组合。

(2) Bagging采用自助采样法 包外估计

(3) 从偏差-方差分解的角度看, Bagging 主要关注降低 方差

(4) Bagging对分类任务采用: 简单投票法

Bagging对回归任务采用: 简单平均法

随机森林 (RF) 是Bagging的一个扩展变体

RF在 以决策树为基学习器 构建Bagging集成的基础上, 进一步在决策树的训练过程中引入了 随机属性选择。

随机森林多样性体现在: 采样随机性; 属性选择随机性。



## 第9章 聚类

监督学习和无监督学习（非监督学习）解决的是两类工作。

对于监督学习，它是通过(feature,label)对来学习/训练模型，然后基于模型预测label；

对于无监督学习，它只通过feature本身来学习，发现数据的结构和隐藏的label  
(在未知中探索数据内在结构)。

### K均值聚类算法

#### 算法流程：

1. 指定需要划分的簇的个数K。
2. 随机选择K个数据对象作为初始聚类中心，这些数据对象不一定是已有的样本点。
3. 逐一计算其他各个数据点到这K个初始聚类中心的距离，把数据对象划分到距离它最近的那个中心所在的簇中。
4. 调整新的簇的聚类中心。
5. 循环执行步骤三和步骤四，观察聚类中心是否收敛（位置基本不发生变化），如果收敛或达到最大迭代次数则终止循环过程。

备注：距离的计算也可以使用曼哈顿距离、闵可夫斯基距离等（泛化的欧氏距离）。



多样性增强的4种方法：

数据样本扰动 — 自助采样法

输入属性扰动 — 随机子空间算法

输出表示扰动

算法参数扰动

假设基分类器错误率相互独立，则错误率最终会趋于0

结合策略：

平均法：（回归任务）

简单平均法：适用于性能差不多的

加权平均法：适用于性能相差较大的

投票法：（分类任务）