人工智能

沙瀛

信息学院
2020.3

ARTIFICIAL INTELLIGENCE

第5章 机器学习

5.1.1 学习的概念

1. 学习的心理学观点

基于脑科学和认知科学对人类学习机理的认识，心理学有两种观点：

联结论观点：学习的实质是联结的形成；

认知论观点：学习的实质是学习者头脑中认知结构的变化。

这些改变既包括行为，也包括行为潜能；既包括感知、记忆、想象、思维等内部心理过程，也包括语言、表情、动作等外部活动。

心理学的学习概念，主要有以下三个核心要点：

(1) 学习的发生以行为和行为潜能的变化为标志。

学习总会引起行为的改变，并且这种改变可以是外显的，也可以是内隐的，即“行为潜能”的改变。

(2) 学习是由经验引起的行为改变。

经验是一个人通过活动直接与客观世界相互作用的过程或在这一过程中所得到的结果。只有因经验而引起的行为改变才是学习。

(3) 学习所引起的行为变化时间比较持久。

学习对行为的影响时间一般比较长，只有当旧的学习被新的学习代替时，旧的行为变化才会消失。

5.1.1 学习的概念

2. 学习的人工智能观点

在人工智能领域，对学习的概念有多种不同的解释。其中，影响最大的观点有以下几种：

(1) 西蒙（Simon，1983）：学习就是系统中的适应性变化，这种变化使系统在重复同样工作或类似工作时，能够做得更好。

(2) 明斯基（Minsky，1985）：学习是在人们头脑里（心理内部）有用的变化。

(3) 米哈尔斯基（Michalski，1986）：学习是对经历描述的建立和修改。

这些观点虽然不尽相同，但却都包含了知识获取和能力改善这两个主要方面。其中

知识获取是指获得知识、积累经验、发现规律等。

能力改善是指改进性能、适应环境、实现自我完善等。

二者之间，知识获取是学习的核心，能力改善是学习的结果。

学习的一般性解释：

学习是一个有特定目的的知识获取和能力增长过程，其内在行为是获得知识、积累经验、发现规律等，其外部表现是改进性能、适应环境、实现自我完善等。

5.1.2 机器学习的概念

1. 什么是机器学习

一般性解释

机器学习就是让机器（计算机）来模拟和实现人类的学习功能。

学科性解释

是一门研究如何利用机器模拟或实现人类学习功能的学科。

主要研究内容

认知模拟

通过对人类学习机理的研究和模拟，从根本上解决机器学习方面存在的种种问题。

理论性分析

从理论上探索各种可能的学习方法，并建立起独立于具体应用领域的学习算法。

面向任务的研究

根据特定任务的要求，建立相应的学习系统。

5.1.2 机器学习的概念

2. 机器学习的发展过程

按机器学习的研究途径和研究目标，机器学习可划分为以下4个阶段：

(1) 神经元模型研究

20世纪50年代中期到60年代初期，也被称为机器学习的热烈时期，最具有代表性的工作是罗森勃拉特1957年提出的感知器模型。

(2) 符号概念获取

20世纪60年代中期到70年代初期。其主要研究目标是模拟人类的概念学习过程。这一阶段神经学习落入低谷，称为机器学习的冷静时期。

(3) 知识强化学习

20世纪70年代中期到80年代初期。人们开始把机器学习与各种实际应用相结合，尤其是专家系统在知识获取方面的需求，也为机器学习的复兴时期。

(4) 连接学习和混合型学习

20世纪80年代中期至21世纪初。把符号学习和连接学习结合起来的混合型学习系统研究已成为机器学习研究的一个新的热点。

(5) 大规模学习与深度学习

21世纪初以来，深度学习提出，一个以深度学习为标志的机器学习热潮比较明显；同时，随着大数据时代的到来，大规模机器学习也发展迅猛。

5.1.2 机器学习的概念

3. 机器学习系统

按机器学习系统的含义

是指能够在一定程度上实现机器学习系统。

机器学习系统的典型定义

萨利斯(Saris) 1973年的解释：

如果一个系统能够从某个过程和环境的未知特征中学到有关信息，并且能把学到的信息用于未来的估计、分类、决策和控制，以便改进系统的性能，那么它就是学习系统

史密斯(Smith) 1977年给出的解释：

如果一个系统在与环境相互作用时，能利用过去与环境作用时得到的信息，并提高其性能，那么这样的系统就是学习系统

学习系统的基本要求：

(1)具有适当的学习环境

(2)具有一定的学习能力

(3)够运用所学知识求解问题

(4)通过学习提高自身性能

5.1.2 机器学习的概念

4. 机器学习的类型

按有无导师指导

有导师指导的机器学习（有监督学习），无导师指导的机器学习（无监督学习）。其中，有监督学习是一种分类式学习方式；无监督学习是一种生成式学习方式。

按学习策略来分类

即按学习中所使用的推理方法来分，可分为记忆学习、传授学习、演绎学习、归纳学习等。归纳学习又可分为示例学习、观察发现学习等。

按应用领域分类

专家系统学习、机器人学习、自然语言理解学习等。

按对人类学习的模拟方式

符号主义学习、统计学习、连接主义学习等。其中

符号主义学习又可分为基于样例的符号学习和基于概率统计的统计学习。

连接主义机器学习 又可分为基于浅层神经网络的浅层连接学习和基于深层神经网络的深度学习。

5.1.3 学习系统的基本模型

环境

是学习系统所感知到的外界信息集合，也是学习系统的外界来源。信息的水平（一般化程度）和质量（正确性）对学习系统影响较大。

学习环节

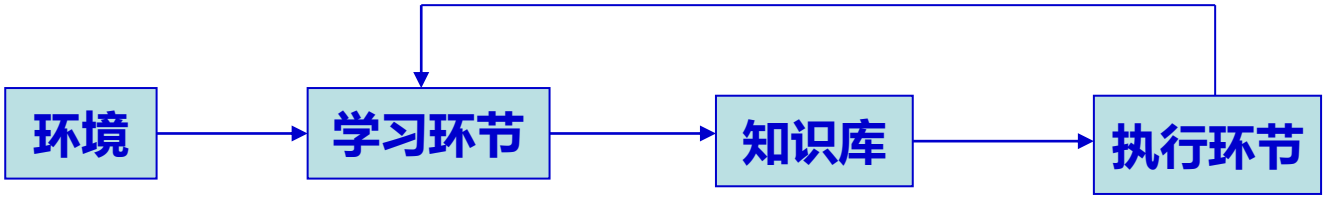
对环境提供的信息进行整理、分析归纳或类比，形成知识，并将其放入知识库。

知识库

存储经过加工后的信息（即知识）。其表示形式是否合适非常重要。

执行环节

根据知识库去执行一系列任务，并将执行结果或执行过程中获得的信息反馈给学习环节。学习环节再利用反馈信息对知识进行评价，进一步改善执行环节的行为。



第5章 机器学习

5.1 机器学习概述

5.2 记忆学习

5.3 示例学习

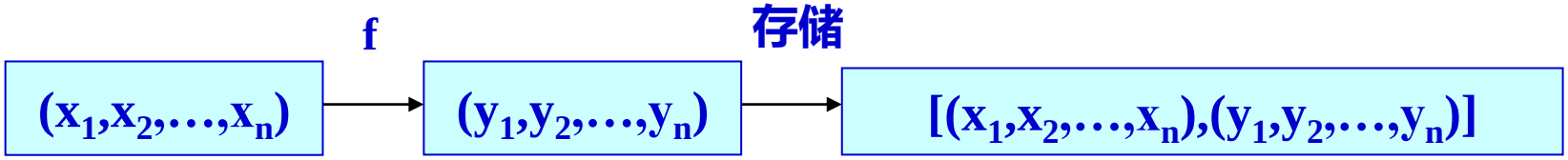
5.4 决策树学习

5.5 统计学习

5.6 集成学习

5.2 记忆学习

记忆学习也叫**死记硬背学习**，其基本过程是每当系统解决一个问题时，就系统就记住这个问题和它的解，当以后再遇到此类问题时，不必重新计算，直接找出原来的解即可使用。记忆学习的基本模型如下：



执行函数 *f* 是记忆学习系统的核心，若将由环境得到的输入模式记为 $(x_1,x_2,\ldots,x_n)$，*f* 的作用就是要对该输入模式进行计算，得到其对应的输出模式 $(y_1,y_2,\ldots,y_m)$。即如下输入/输出模式对：

$$[(x_1,x_2,\ldots,x_n)，(y_1,y_2,\ldots,y_m)]$$

然后，由系统将这一输入/输出模式对保存到知识库中，当以后再遇见输入模式$(x_1,x_2,\ldots,x_n)$时，就可以直接从存储器中把$(y_1,y_2,\ldots,y_m)$检索出来，而不需要重新进行计算。

第5章 机器学习

5.3.1 示例学习的类型

按例子的来源分类

① 例子来源于教师的示例学习

② 例子来源于学习者本身的示例学习

学习者明确知道自己的状态，但完全不清楚所要获取的概念。

③ 例子来源于学习者以外的外部环境的示例学习

例子的产生是随机的。

按例子的类型分类

① 仅利用正例的示例学习

这种学习方法会使推出的概念的外延扩大化。

② 利用正例和反例的示例学习

这是示例学习的一种典型方式，它用正例用来产生概念，用反例用来防止概念外延的扩大。

5.3.2 示例学习的模型

示例空间

是我们向系统提供的示教例子的集合。研究问题：例子质量，搜索方法。

归纳过程

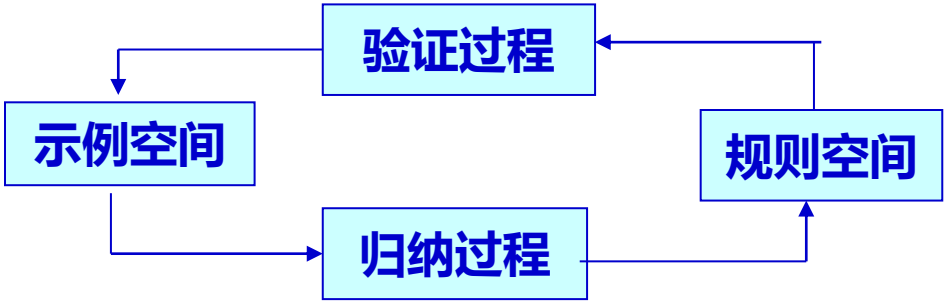
是从搜索到的示例中抽象出一般性的知识的归纳过程。归纳方法：常量转换为变量，去掉条件，增加选择，曲线拟合等。

规则空间

是事务所具有的各种规律的集合。研究问题：对空间的要求，搜索方法

验证过程

是要从示例空间中选择新的示例，对刚刚归纳出的规则做进一步的验证和修改。



5.3.3 示例学习的归纳方法

1. 把常量化为变量

把示例中的常量换成相应的变量即可得到一个一般性的规则。下面以扑克牌中同花的概念为例，进行讨论。

假设例子空间中有以下两个关于扑克牌中“同花”概念的示例：

示例1：花色(c_1，梅花) ∧ 花色(c_2，梅花) ∧ 花色(c_3，梅花) ∧ 花色(c_4，梅花)∧ 花色(c_5，梅花) → 同花(c_1, c_2, c_3, c_4, c_5)

示例2：花色(c_1，红桃) ∧ 花色(c_2，红桃) ∧ 花色(c_3，红桃) ∧ 花色(c_4，红桃)∧ 花色(c_5，红桃) → 同花(c_1, c_2, c_3, c_4, c_5)

其中，示例1表示5张梅花牌是同花，示例2表示5张红桃牌是同花。

对这两个示例，采把常量化为变量的归纳方法，只要把“梅花”和“红桃”用变量x代换，就可得到如下一般性的规则：

规则1：花色(c_1，x) ∧ 花色(c_2，x) ∧ 花色(c_3，x) ∧ 花色(c_4，x)∧ 花色(c_5，x)→ 同花(c_1, c_2, c_3, c_4, c_5)

5.3.3 示例学习的归纳方法

2. 去掉条件

该方法是要把示例中的某些无关的子条件舍去，得到一个一般性的结论

例如，有如下示例：

示例3：花色(c_1, 红桃)∧点数(c_1，2)

∧花色(c_2, 红桃)∧点数(c_2，3)

∧花色(c_3, 红桃)∧点数(c_3，4)

∧花色(c_4, 红桃)∧点数(c_4，5)

∧花色(c_5, 红桃)∧点数(c_5，6)

→ 同花(c_1, c_2, c_3, c_4, c_5)

为了学习同花的概念，除了需要把常量变为变量外，还需要把与花色无关的“点数”子条件舍去。这样也可得到上述规则1：

规则1：花色(c_1，x) ∧ 花色(c_2，x) ∧ 花色(c_3，x) ∧ 花色(c_4，x)∧ 花色(c_5，x)→ 同花(c_1, c_2, c_3, c_4, c_5)

5.3.3 示例学习的归纳方法

3. 增加选择

该方法是要在析取条件中增加一个新的析取项。它包括前件析取法和内部析取法。

前件析取法：是通过对示例的前件的析取来形成知识的。例如：

示例4：点数(c_1, J)→脸(c_1)

示例5：点数(c_1, Q)→脸(c_1)

示例6：点数(c_1, K)→脸(c_1)

将各示例的前件进行析取，就可得到所要求的规则：

规则2：点数(c_1, J)∨点数(c_1, Q)∨点数(c_1, K)→脸(c_1)

内部析取法：是在示例的表示中使用集合与集合的成员关系来形成知识的。例如，有如下关于“脸牌”的示例：

示例7：点数$c_1 \in \{J\}$→脸(c_1)

示例8：点数$c_1 \in \{Q\}$→脸(c_1)

示例9：点数$c_1 \in \{K\}$→脸(c_1)

用内部析取法，可得到如下规则：

规则3：点数(c_1)$\in \{J, Q, K\}$→脸(c_1)

5.3.3 示例学习的归纳方法

4. 曲线拟合

对数值问题的归纳可采用曲线拟合法。假设示例空间中的每个示例(x, y, z)都是输入x, y与输出z之间关系的三元组。例如，有下3个示例：

示例10：(0, 2, 7)

示例11：(6, -1, 10)

示例12：(-1, -5, -16)

用最小二乘法进行曲线拟合，可得x, y, z之间关系的规则如下：

规则4：$z=2x+3y+1$

说明：在上述前三种方法中，方法(1)是把常量转换为变量；方法(2)是去掉合取项（约束条件）；方法(3)是增加析取项。它们都是要扩大条件的适用范围。从归纳速度上看，方法(1)的归纳速度快，但容易出错；方法(2)归纳速度慢，但不容易出错。因此，在使用方法(1)时应特别小心。例如：

对示例4、示例5及示例6，若使用方法(1) ,则会归纳出如下的错误规则：

规则5：（错误）点数(c_1, x)→脸(c_1)

它说明，归纳过程是很容易出错的。

第5章 机器学习

5.4.1 决策树的概念

概念说明

决策树是一种由节点和边构成的用来描述分类过程的层次数据结构。

根节点：表示分类的开始

叶节点：表示一个实例的结束

中间节点：表示相应实例中的某一属性

边代表：某一属性可能的属性值

路径：从根节点到叶节点的每一条路径都代表一个具体的实例，并且同一路径上的所有属性之间为合取关系，不同路径（即一个属性的不同属性值）之间为析取关系。

决策树的分类过程：从树的根接点开始，按照给定的事例的属性值去测试对应的树枝，并依次下移，直至到达某个叶节点为止。

图5.4是一个简单的用来对鸟类进行分类的决策树。在该图中，根节点包含各种鸟类；叶节点是所能识别的各种鸟的名称；中间节点是鸟的一些属性；边是鸟的某一属性的属性值；

从根节点到叶节点的每一条路径都描述了一种鸟，它包括该种鸟的一些属性及相应的属性值。

5.4.1 决策树的概念

一棵简单的决策树

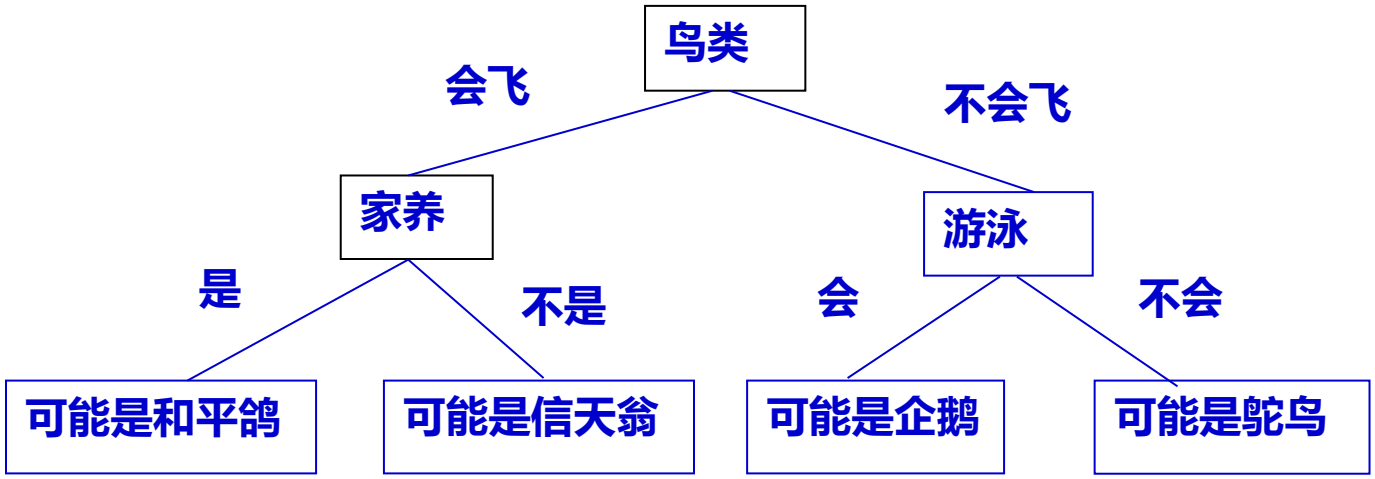


图5.4 一个简单的鸟类识别决策树

决策树还可以表示为规则的形式。上图的决策树可表示为如下规则集：

IF 鸟类会飞 AND 是家养的 THEN 可能是和平鸽

IF 鸟类会飞 AND 不是家养的 THEN 可能是信天翁

IF 鸟类不会飞 AND 会游泳 THEN 可能是企鹅

IF 鸟类不会飞 AND 不会游泳 THEN 可能是鸵鸟

决策树学习过程实际上是一个构造决策树的过程。当学习完成后，就可以利用这棵决策树对未知事物进行分类。

5.4.2 ID3算法

1. 信息熵和信息增益(1/2)

信息熵

信息熵是对信息源整体不确定性的度量。假设S为样本集，S中所有样本的类别有k种，如$y_1,y_2,\ldots,y_k$，各种类别样本在S上的概率分布分别为$P(y_1),P(y_2),\ldots,P(y_k)$，则S的信息熵可定义为：

$$E(S) = -P(y_1)\log P(y_1) - P(y_2)\log P(y_2) - \mathrm{L} - P(y_r)\log P(y_r)$$

$$= -\sum_{j=1}^{k} P(y_j)\log P(y_j)$$

其中，概率$P(y_j)$（j=1,2,…,k），实际上为yj的样本在S中所站的比例；对数可以是以各种数为底的对数，在ID3算法中，我们取以2为底的对数。

E(S)的值越小，S的不确定性越小，即其确定性越高。

5.4.2 ID3算法

1. 信息熵和信息增益(2/2)

信息增益（information gain）

是对两个信息量之间的差的度量。其讨论涉及到样本集S中样本的结构。

对S中的每一个样本，除其类别外，还有其条件属性，或简称为属性。若假设S中的样本有m个属性，其属性集为$X=\{x_1,x_2,\ldots,x_m\}$，且每个属性均有r种不同的取值，则我们可以根据属性的不同取值将样本集S划分成r个不同的子集$S_1, S_2, \ldots, S_r$。

此时，可得到由属性x_i的不同取值对样本集S进行划分后的加权信息熵

$$\mathbf{E}(S, x_i) = \sum_{t=1}^{r} \frac{|S_t|}{|S|} \times E(S_t)$$

其中，t为条件属性x_i的属性值；S_t为$x_i=t$时的样本子集；$E(S_t)$为样本子集S_t信息熵；$|S|$和$|S_t|$分别为样本集S和样本子集S_t的大小，即样本个数。

有了信息熵和加权信息熵，就可以计算信息增益。所谓信息增益就是指$E(S)$和$E(S, x_i)$之间的差，即

$$\mathbf{G}(S, x_i) = \mathbf{E}(S) - \mathbf{E}(S, x_i)$$
$$= \mathbf{E}(S) - \sum_{t=1}^{r} \frac{|S_t|}{|S|} \times E(S_t)$$

可见，信息增益所描述的是信息的确定性，其值越大，信息的确定性越高。

5.4.2 ID3算法

2. ID3算法的描述(1/2)

ID3算法的学习过程是一个以整个样本集为根节点，以信息增益最大为原则，选择条件属性进行扩展，逐步构造出决策树的过程。若假设$S=\{s_1, s_2, \cdots s_n\}$为整个样本集，$X=\{x_1, x_2, \cdots, x_m\}$为全体属性集，$Y=\{y_1, y_2, \cdots y_k\}$为样本类别。则ID3算法描述如下：

（1）初始化样本集$S=\{s_1, s_2, \cdots s_n\}$和属性集$X=\{x_1, x_2, \cdots, x_m\}$，生成仅含根节点（S, X）的初始决策树。

（2）如果节点样本集中的所有样本全都属于同一类别，则将该节点标记为叶节点，并标出该叶节点的类别。算法结束。 否则执行下一步。

（3）如果属性集为空；或者样本集中的所有样本在属性集上都取相同值，即所有样本都具有相同的属性值，则同样将该节点标记为叶节点，并根据各个类别的样本数量，按照少数服从多数的原则，将该叶节点的类别标记为样本数最多的那个类别。算法结束。否则执行下一步。

5.4.2 ID3算法

2. ID3算法的描述(2/2)

(4) 计算每个属性的信息增益，并选出信息增益最大的属性对当前决策树进行扩展。

(5) 对选定属性的每一个属性值，重复执行如下操作，直止所有属性值全部处理完为止：

① 为每一个属性值生成一个分支；并将样本集中与该分支有关的所有样本放到一起，形成该新生分支节点的样本子集；

② 若样本子集为空，则将此新生分支节点标记为叶节点，其节点类别为原样本集中最多的类别；

③ 否则，若样本子集中的所有样本均属于同一类别，则将该节点标记为叶节点，并标出该叶节点的类别。

(6) 从属性集中删除所选定的属性，得到新的属性集。

(7) 转第**(3)**步。

5.4.2 ID3算法

3. ID3算法简例(1/14)

例5.1 用ID3算法完成下述学生选课的例子

假设将决策y分为以下 3 类：

y_1：必修AI

y_2：选修AI

y_3：不修AI

做出这些决策的依据有以下3个属性：

x_1：学历层次　x_1=1 研究生，x_1=2 本科

x_2：专业类别　x_2=1 电信类，x_2=2 机电类

x_3：学习基础　x_3=1 修过AI，x_3=2 未修AI

表5.1给出了一个关于选课决策的训练例子集S。

5.4.2 ID3算法

3. ID3算法简例(2/14)

序号	属性值			决策方案
	x_1（层次）	x_2（专业）	x_3（学否）	y_j
1	1	1	1	y_3
2	1	1	2	y_1
3	1	2	1	y_3
4	1	2	2	y_2
5	2	1	1	y_3
6	2	1	2	y_2
7	2	2	1	y_3
8	2	2	2	y_3

表5.1　学生选课决策的训练例子集

该训练例子集S的大小为８。ID3算法就是依据这些训练例子，以(S,X)为根节点，按照信息熵下降最大的原则来构造决策树的。

5.4.2 ID3算法

3. ID3算法简例(3/14)

解：按照ID3算法，先初始化样本集S={1,2,3,4,5,6,7,8}和属性集$X=\{x_1,x_2,x_3\}$，生成仅含根节点(S,X)的初始决策树。其中，S中的数字为样本集中相应样本的编号。然后通过算法第(2)、(3)步，执行其第(4)步，计算根节点(S,X)关于每一个属性的信息增益，并选择具有最大信息增益的属性对根节点进行扩展。

为此，需要先计算根节点的信息熵

$$\mathbf{E}(S,X)=-\sum_{i=1}^{3}P(y_i)log_2P(y_i)$$

式中，3为样本集中样本类别的总数；概率$P(y_i)$为第i类样本在整个样本集S中所占的比例。即

$$\mathbf{P}(y_1)=\frac{1}{8},P(y_2)=\frac{2}{8},P(y_3)=\frac{5}{8}$$

即有根节点的信息熵

$$\mathbf{E}(S,X)=-\frac{1}{8}\times log_2\left(\frac{1}{8}\right)-\frac{2}{8}\times log_2\times\left(\frac{2}{8}\right)-\frac{5}{8}\times log_2\left(\frac{5}{8}\right)=1.2988$$

然后再计算根节点(S,X)关于每个属性的加权信息熵

$$\mathbf{E}\big((S,X),x_i\big)=\sum_i\frac{|S_t|}{|S|}\cdot E(S_t,X)$$

其中，t为属性x_i的属性值；S_t为x_i=t时的样本子集；|S|、$|S_t|$分别为样本集S和样本子集S_t的大小，即相应集合中的样本个数。

5.4.2 ID3算法

3. ID3算法简例(4/14)

先考虑属性x_1，

由表5.1可知，其属性值为1或者2。

当x_1=1时，t=1，有S_1={1, 2, 3, 4}

当x_1=2时，t=2，有S_2={5, 6, 7, 8}

子集S_1和S_2中的数字均为样本集中相应样本的编号，且有|S|=8|，S_1|=|S_2|=4。

由S_1可知

$$P_{s_1}(y_1)=\frac{1}{4}, P_{s_1}(y_2)=\frac{1}{4}, P_{s_1}(y_3)=\frac{2}{4}$$

则

$$E(S_1, X)=-P_{S_1}(y_1)\log_2 P_{S_1}(y_1)$$
$$-P_{S_1}(y_1)\log_2 P_{S_1}(y_2)-P_{S_1}(y_3)\log_2 P_{S_1}(y_3)$$
$$=-\frac{1}{4}\times\log\frac{1}{4}-\frac{1}{4}\times\log\frac{1}{4}-\frac{2}{4}\times\log\frac{2}{4}$$
$$=1.5$$

5.4.2 ID3算法

3. ID3算法简例(5/14)

再由S_2可知

$$P_{s_2}(y_1)=0, P_{s_2}(y_2)=\frac{1}{4}, P_{s_2}(y_3)=\frac{3}{4}$$

则

$$\begin{aligned}E(S_2,X)&=-P_{S_2}(y_1)\log_2 P_{S_2}(y_1)\\&\quad -P_{S_2}(y_1)\log_2 P_{S_2}(y_2)-P_{S_2}(y_3)\log_2 P_{S_2}(y_3)\\&=0-\frac{1}{4}\times\log\frac{1}{4}-\frac{3}{4}\times\log\frac{3}{4}\\&=0.8113\end{aligned}$$

将$E(S_1,X)$和$E(S_2,X)$代入加权信息熵公式，有

$$\begin{aligned}E((S,X),x_1)&=\frac{|S_1|}{|S|}\times E(S_1,X)+\frac{|S_2|}{|S|}\times E(S_2,X)\\&=\frac{4}{8}\times 1.5+\frac{4}{8}\times 0.8113\\&=1.1557\end{aligned}$$

5.4.2 ID3算法

3. ID3算法简例(6/14)

同样可以求得

$$E((S,X),x_2)=1.1557$$

$$E((S,X),x_3)=0.75$$

据此，可求得各属性的信息增益为：

G((S,X), x_1)= E(S,X) - E((S,X), x_1)
= 1.2988 - 1.1557 = 0.1431

G((S,X), x_2)= E(S ,X) - E((S,X), x_2)
= 1.2988 - 1.1557 = 0.1431

G((S,X), x_3)= E(S,X) - E((S,X), x_3)
= 1.2988 - 0.75 = 0.5488

显然，x_3的信息增益最大，因此应先选择x_3对根节点进行扩展。

接着执行(5)，对属x_3的所有属性值分别生成根节点(S, X)的不同分支节点。

先取x_3=1，生成根节点（S,X）的左分支节点。由于t=1，设所得节点的样本子集为S_1'，则有S_1'={1,3,5,7}。又由于该样本子集S_1中的所有样本均属于同一类别，故将该节点标记为叶节点，并标出其类别y_3。

5.4.2 ID3算法

3. ID3算法简例(7/14)

再取x_3=2，生成根节点（S,X）的右分支节点。由于t=2，设所得节点的样本子集为S_2'，则有S_2'={ 2,4,6,8 }。

显然该样本子集非空，且其中的样本并非同一类别，故算法第(5)步全部完成。

执行(6)，从属性集X={x_1, x_2, x_3}中删除本轮扩展所选定的属性x_3，得到新的属性集X_1={ x_1，x_2}。至此，根节点（S,X）的扩展过程完成，所得到的当前部分决策树如图5.5所示。

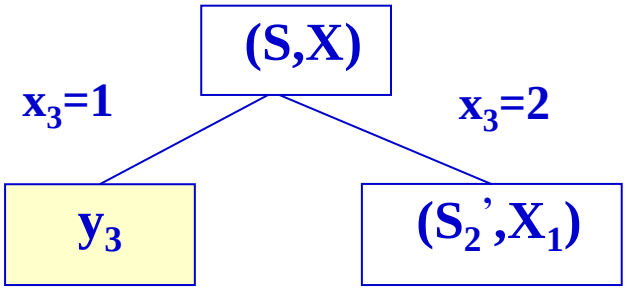


图5.5 扩展根节点后的部分决策树

然后返回算法第(3)步，进入下一轮扩展过程。

5.4.2 ID3算法

3. ID3算法简例(8/14)

显然，(3)的条件不满足，接着执行算法第(4)步。

计算节点(S_2', X_1)下各属性的信息增益，并选择具有最大信息增益的属性对决策树进行扩展。其过程如下：

先计算节点(S_2'，X_1)的信息熵：

$$E(S_2', X_1) = -\frac{1}{4}log_2\frac{1}{4} - \frac{1}{4}log_2\frac{1}{4} - \frac{2}{4}log_2\frac{2}{4} = 1.5$$

然后再分别计算节点(S_2'，X_1)关于属性x_1和x_2的信息熵、加权信息熵和信息增益。

先考虑属性x_1，对x_1的不同属性值：

当取x_1=1时，有t=1，设所得样本子集为S_{21}'，则$S_{21}' = \{2, 4\}$，

当取x_1=2时，有t=2，设所得样本子集为S_{22}'，则$S_{22}' = \{6, 8\}$，

其中S_{21}'和S_{22}'中的数字也为样本集S中各样本的序号，且有$|S_2'| = 4$，$|S_{21}'| = |S_{22}'| = 2$。

5.4.2 ID3算法

3. ID3算法简例(9/14)

若取x_1=1，不同类别样本在S'_{21}上的概率分别为

$$P_{S'_{21}}(y_1)=\frac{1}{2},\ P_{S'_{21}}(y_2)=\frac{1}{2},\ P_{S'_{21}}(y_3)=0$$

故有节点(S_2'，X_1)关于属性x_1=1的信息熵

$$\mathbf{E}(S'_{21},X_1)=-\frac{1}{2}log_2\frac{1}{2}-\frac{1}{2}log_2\frac{1}{2}=1$$

若取x_1=2，不同类别样本在S'_{22}上的概率分别为

$$P_{S'_{22}}(y_1)=0,\ P_{S'_{22}}(y_2)=\frac{1}{2},\ P_{S'_{22}}(y_3)=\frac{1}{2}$$

故有节点(S_2'，X_1)关于属性x_1=2的信息熵

$$\mathbf{E}(S'_{22},X_1)=-\frac{1}{2}log_2\frac{1}{2}-\frac{1}{2}log_2\frac{1}{2}=1$$

由$\mathbf{E}(S'_{21},X_1)$和$\mathbf{E}(S'_{22},X_1)$可求出x_1的加权信息熵

$$E((S'_2,X_1),x_1)=\frac{|S'_{21}|}{|S'_2|}E(S'_{21},X_1)+\frac{|S'_{22}|}{|S'_2|}E(S'_{22},X_1)=\frac{2}{4}\times 1+\frac{2}{4}\times 1=1$$

5.4.2 ID3算法

3. ID3算法简例(10/14)

进而求出x_1的信息增益

$$G((S_2',X_1),x_1)=E(S_2',X_1)-E((S_2',X_1),x_1)=1.5-1=0.5$$

然后再考虑属性x_2，对x_2的不同属性值：

当取x_2=1时，有t=1，设所得样本子集为S_{21}''，则$S_{21}''=\{2,6\}$；

当取x_2=2时，有t=2，设所得样本子集为S_{22}''，则$S_{22}''=\{4,8\}$。

其中S_{21}''和S_{22}''中的数字也同样为样本集S中各样本的序号，且有$|S_2|=4$，$|S_{21}''|=|S_{22}''|=2$。

先取x_2=1，不同类别样本在子集S_{21}''上的概率分别为

$$P_{S_{21}''}(y_1)=\frac{1}{2},\qquad P_{S_{21}''}(y_2)=\frac{1}{2},\qquad P_{S_{21}''}(y_3)=0$$

故有节点$(S_2'，X_1)$关于属性x_2=1的信息熵

$$E(S_{21}'',X_1)=-\frac{1}{2}log_2\frac{1}{2}-\frac{1}{2}log_2\frac{1}{2}=1$$

5.4.2 ID3算法

3. ID3算法简例(11/14)

再取x_2=2，不同类别样本在样本子集S_{22}'上的概率分别为

$$P_{S_2''}(y_1)=0,\ P_{S_2''}(y_2)=0,\ P_{S_2''}(y_3)=1$$

故有节点(S_2'，X_1)关于属性x_2=2的信息熵

$$\mathrm{E}(S_{22}'',X_1)=-log_2 1=0$$

由$\mathrm{E}(S_{21}'',X_1)$和$\mathrm{E}(S_{22}'',X_1)$可求出x_2的加权信息熵

$$\mathrm{E}((S_2',X_1),x_2)=\frac{2}{4}\mathrm{E}(S_{21}'')+\frac{2}{4}\mathrm{E}(S_{22}'')=\frac{2}{4}\times 1+\frac{2}{4}\times 0=0.5$$

故有x_2的信息增益

$$\mathrm{G}((S_2',X_1),x_2)=\mathrm{E}(S_2',X_1)-\mathrm{E}((S_2',X_1),x_2)=1.5-0.5=1$$

可见，x_2的信息增益大于x_1的信息增益，因此应先扩展属性x_2。

接着执行算法第(5)步，对属性x_2的所有取值分别生成节点(S_2'，X_1) 的不同分支节点。当x_2=1时，生成其左子节点；当x_2=2时，生成其右子节点。

5.4.2 ID3算法

3. ID3算法简例(12/14)

接着执行算法第(6)步，从当前属性集$X_1=\{x_1, x_2\}$中删除本轮扩展所选定的属性x_2，得到新的属性集$X_2=\{\ x_1\ \}$。当前的部分决策树如图5.6所示。

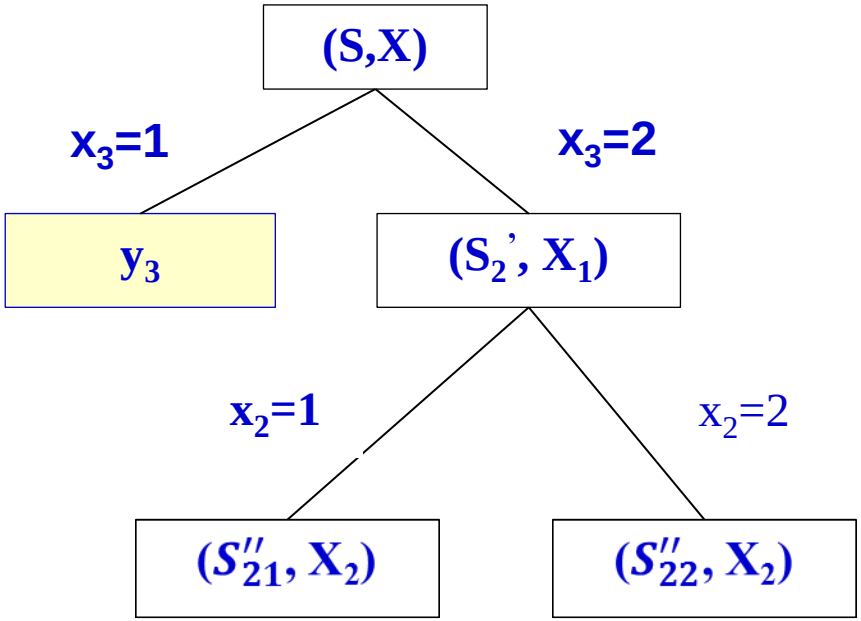


图5.6 扩展根节点(S_2', X_1)后的部分决策树

5.4.2 ID3算法

3. ID3算法简例(13/14)

接着返回算法第(3)步，进入下一轮扩展过程。由于第(3)步中的条件都不满足，故执行第(4)步。由于此时属性集X中只有x_1，无须再进行属性选择，直接执行算法第(5)步，对属性x_1的所有取值，依次完成对各非叶节点的扩展，并将所有新生分支节点标记为叶节点。

然后执行算法第(6)步，此时从$X_2=\{ x_1 \}$中删除属性x_1，当前属性集为空。

然后返回算法第(3)步，此时因属性集为空，算法结束。

图5.7为最终所得完整决策树，

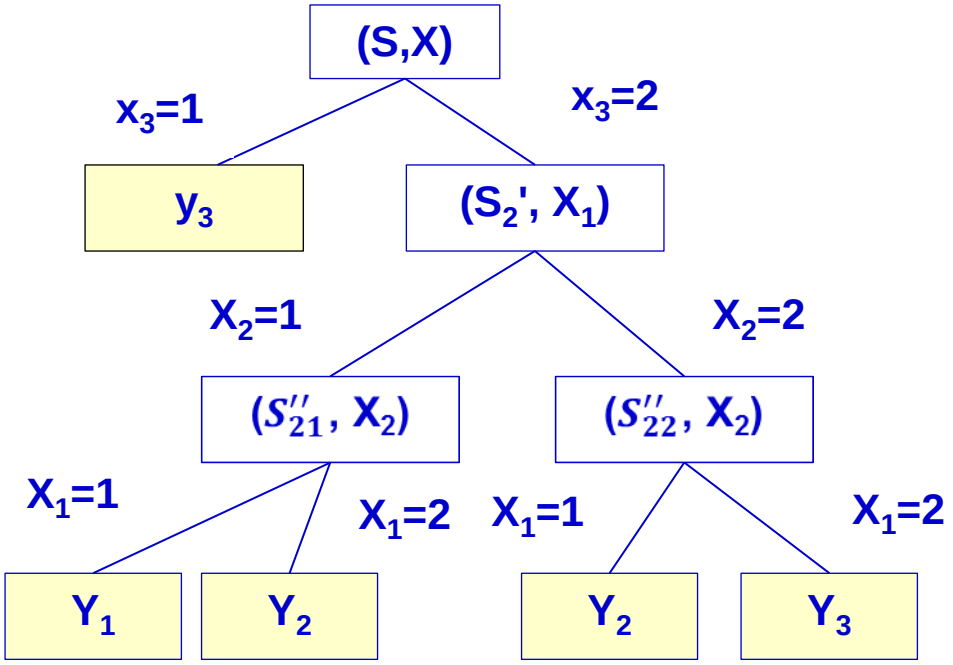


图5.7 最终得到的完整决策树

5.4.2 ID3算法

3. ID3算法简例(14/14)

上述该决策树的含义如图5. 8所示。其中。从根节点到每个叶节点的路径都代表了一条知识。

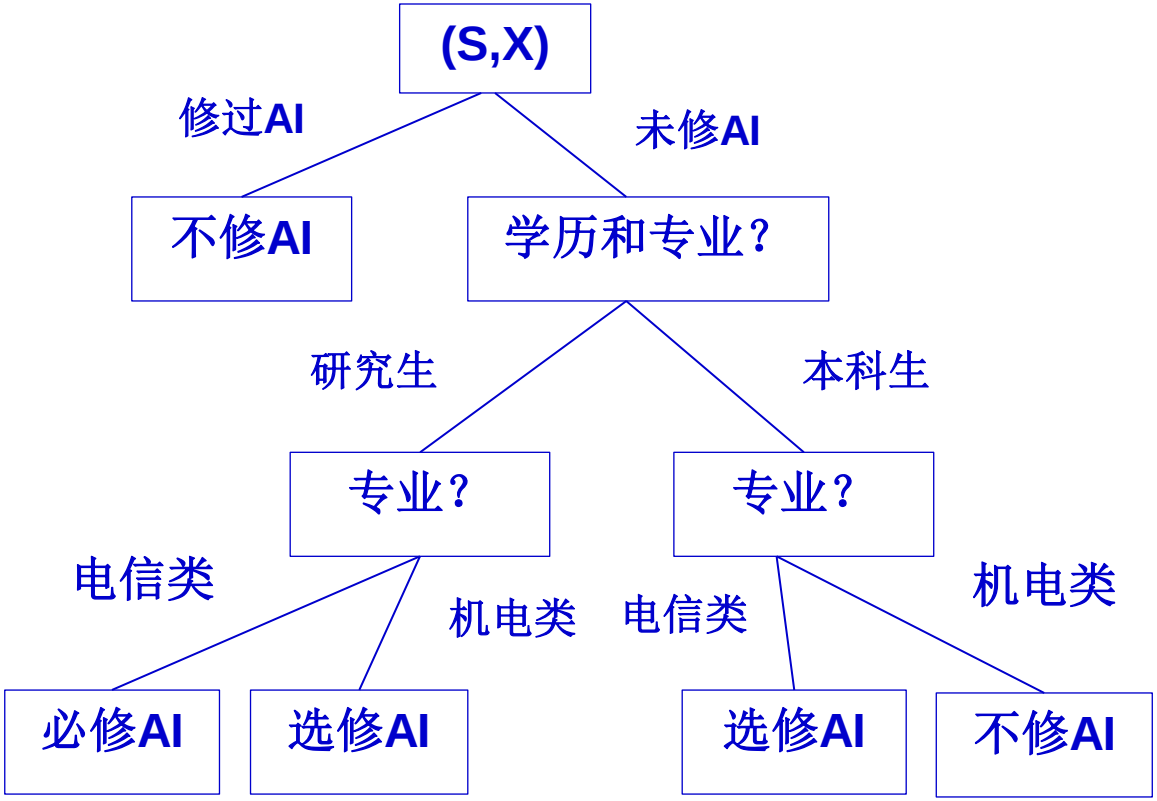


图5.8 最终所得完整决策树的含义

第5章 机器学习

5.5.1 小样本统计学习理论

1. 期望风险和经验风险(1/2)

小样本统计学习理论一种以有限样本统计学理论为基础进行统计学习的理论。其核心是结构风险最小化原理，涉及的概念主要包括经验风险、期望风险和VC维等。

期望风险

统计学习是要根据给定的训练样本，求出用联合概率分布函数P(x, y)表示的输入变量集x和输出变量集y之间未知的依赖关系，并使期望风险最小。

设有n个独立且同分布（即具有相同概率分布）的训练样本

(x_1, y_1)，(x_2, y_2)，…，(x_n, y_n)

在一个函数集{ f(x, w) }中求出一个最优函数$f(x, w_0)$，使得系统用该函数对依赖关系进行估计时期望风险

$$R(w) = \int L(y, f(x, w))dP(x, y) \qquad (5.1)$$

为最小。

其中，w为函数的广义参数；f (x,w)为学习函数集（或预测函数集），它可以表示任何函数集，用于从x预测y，目的是通过对训练样本的学习得到一个最优函数$f (x,w_0)$；L(y, f(x, w))为损失函数，表示因预测失误而产生的损失，该函数的具体表示形式与学习问题的类型有关。

5.5.1 小样本统计学习理论

1. 期望风险和经验风险(2/2)

经验风险

对上述期望风险函数，由于其中的概率分布函数P(x, y)未知，因此无法直接对其进行计算。常用的方法是利用经验风险函数

$$R_{emp}(w)=\frac{1}{n}\sum_{i=1}^{n}L(y_i, f(x_i, w)) \qquad (5.2)$$

对期望风险进行估计。

经验风险则是用样本损失的算术平均值进行计算的。

统计学习的目标就是要设计学习算法，使得该经验风险最小化。这一原理也称为**经验风险最小化原理**。

5.5.1 小样本统计学习理论

2. VC维(1/3)

VC维是小样本统计学习理论中的又一个重要概念，用于描述构成学习模型的函数集合的容量及学习能力。通常，其VC维越大、容量越大、学习能力越强。

由于VC维是通过“打散”操作定义的，下面先讨论打散操作。

(1) 打散操作

样本集的打散（shatter）操作可描述如下：

假设X为样本空间，S是X的一个子集，H是由指示性学习函数所构成的指示函数集。对一个样本集S，若其大小为h，则它应该有2^h种划分，假设S中的每一种划分都能被H中的某个指示函数将其分为两类，则称函数集H能够打散样本集S。

所谓指示性学习函数是指其值只能取0或1的学习函数。

5.5.1 小样本统计学习理论

2. VC维(2/3)

例5.2 对二维实空间R^2，假设给定的样本集S为R^2中的不共线的3个数据点，每个数据点有两种状态，指示函数集H为有向直线的集合，求H是否可以打散S?

解： S中不共线的3个数据点可构成2^3种不同的点集，如图5.9所示。在该图中，可以看出，每一点集中的数据点，都能被H中的一条有向直线按其状态分为两类。即位于有向直线正方向一侧的数据点为一类，而位于有向直线负方向一侧的数据点为另一类。因此，我们说H能够打散S。

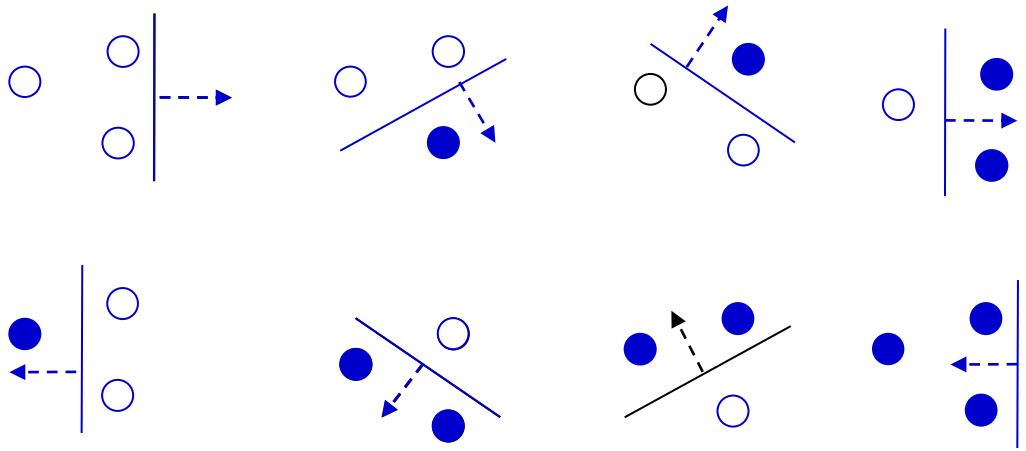


图5.9 在R^2中被H打散的3个数据点

5.5.1 小样本统计学习理论

2. VC维(3/3)

(2) VC维的确定

VC维用来表示指示函数集H能够打散一个样本集S的能力，其值定义为能被H打散的X的最大有限子集的大小。若样本空间X的任意有限大的子集都可以被H打散，则其VC为∝。

例如，对前面给出的例6.10，指示函数集H中的有向直线能够将大小为3的X的子集S打散，但却不能打散4个点，如图5.10，因此该H的VC维至少为3。

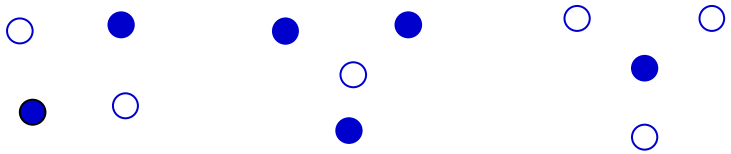


图5.10 在R^2中不能被H打散的4个点

可见，在R^2中，由有向直线构成的指示函数集H，所能打散的R^2的最大子集为3，因此H的VC维为3.

需要指出的是，目前还没有一套关于任意H的VC维的计算理论，只是对一些特殊空间，才知道其VC维。例如，对n维空间，知道其VC维为n+1。

5.5.1 小样本统计学习理论

3. 结构风险最小化原理

统计学习理论研究表明，对线性可分问题可有如下结论：期望风险与经验风险之间至少以概率（1-η）满足如下量化关系：

$$R(w) \le R_{emp}(w) + \sqrt{\frac{h(\ln(2n/h)+1) - \ln(\eta/4)}{n}} \qquad (5.3)$$

其中，h为VC维，n为样本数，η为满足0≤η≤1的参数。可以看出，期望风险由两部分所组成：一是基于样本的经验风险，即训练误差；二是置信范围，即期望风险与经验风险差值的上确界。

后者反映了结构复杂度所带来的风险，它和VC维h及训练样本数有关。若定义

$$\Phi(h/n) = \sqrt{\frac{h(\ln(2n/h)+1) - \ln(\eta/4)}{n}} \qquad (5.4)$$

则（5.3）式可简单地表示为

$$R(w) \le R_{emp}(w) + \Phi(h/n) \qquad (5.5)$$

可见，当训练样本有限时，VC维越高，经验风险和期望风险的差别就会越大。即对统计学习，不仅要使经验风险最小化，还要降低VC维，以缩小置信范围，进而使期望风险最小化。

据此，结构风险最小化原理可描述如下：同时降低经验风险和置信范围（即VC维），使期望风险最小化。

第5章 机器学习

5.6.1 集成学习概述

1. 集成学习的基本概念

集成学习是指为解决同一问题，先训练出一系列个体学习器（或称弱学习器），然后再根据某种规则把这些个体学习器的学习结果整合到一起，得到比单个个体学习器更好的学习效果。集成学习的基本结构如图5.11所示。

集成学习包括两大基本问题，一个是个体学习器的构造，另一个是个体学习器的合成。

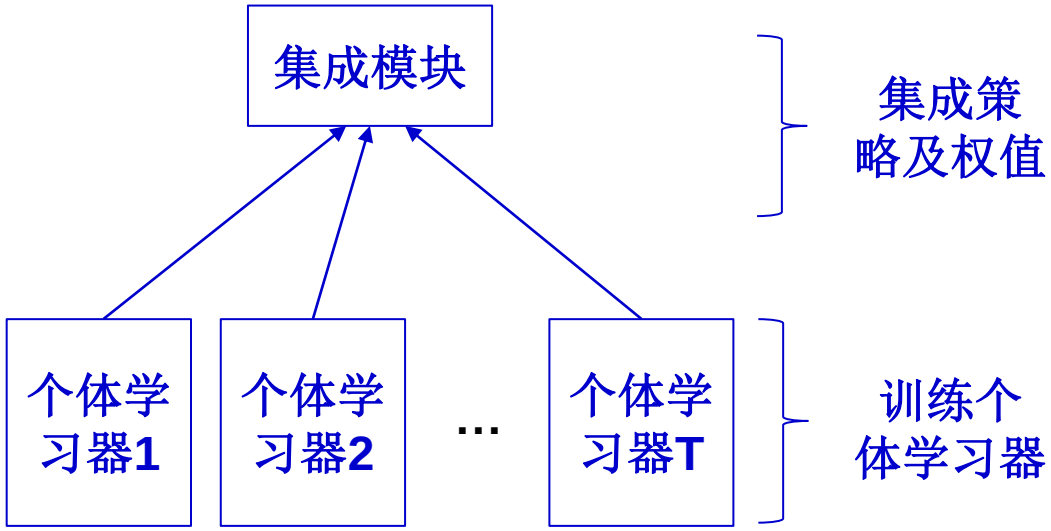


图5.11 集成学习的基本结构

5.6.1 集成学习概述

1. 集成学习的基本概念

集成学习的两种方式

（1）同质集成

要求构造个体学习器时使用相同类型的学习方法，构造出来的多个个体学习器为同质学习器。

所谓同质，是指同一类型，例如要使用决策树都为决策树。

这种采用相同学习方法构造个体学习器的集成学习称为狭义集成学习，其个体学习器称为基学习器，所用的学习算法称为基学习算法。

（2）异质集成

不要求构造个体学习器时使用同一类型的学习方法，而是可以异质。

所谓异质，是指不同类型，例如可以同时使用决策树和神经网络去构造个体学习器。

这种集成学习又称为广义集成学习，构造个体学习器所用学习算法不再称基学习算法，构造出来的个体学习器也不再称基学习器，而直接称其为个体学习器。

5.6.1 集成学习概述

2. 集成学习的产生与发展

集成学习的思想最早可追溯到1990年。当年，汉森（Hanson）和萨拉蒙（Salamon）通过对神经网络集成的研究，提出了神经网络集成的概念，并证明当单个神经网络的精度高于50%时，如果按投票的方式把它们集成到一起，则可以明显提高学习系统的泛化能力。

所谓学习系统的泛化能力是指机器学习算法对新鲜样本的适应能力。

直到1996年，弗罗因德（Freund）和史皮尔（Schapire）提出了著名的AdaBoost（adaptive Boost）算法，布雷曼（Breiman）提出了著名的Bagging算法。

上述两大集成学习算法的提出，奠定了集成学习的理论基础，形成了集成学习的基本构架，标志着集成学习的真正形成。

此后的一些研究，主要是对这两大算法的扩展和改进，理论上没有出现大的突破性进展。

5.6.1 集成学习概述

3. 集成学习的基本类型

根据个体学习器生成方式的不同，以及个体学习器之间依赖关系的不同，集成学习可分为Boosting方法和Bagging方法两大基本类。

Boosting方法的基本思想是从初始训练集开始，先为每个训练样本平均分配初始权重，并训练出弱学习器1；然后通过提高错误率高的训练样本的权重，降低错误率低的训练样本的权重，得到训练样本的新的权重分布，并在在该权重分布上训练出弱学习器2；依此逐轮迭代，直至达到最大迭代轮数，最后再将训练出来的这些弱学习器合成到一起，形成最终的强学习器。其典型代表是AdaBoost算法和提升树(boosting tree)算法。

Bagging方法则不同，其基本思想是在给定初始训练集和弱学习算法的前提下，每轮迭代都使用可重采样的随机抽样方法式从初始训练集产生出本轮的训练子集，并利用选定的弱学习算法训练出本轮迭代的弱学习器，依此逐轮迭代，直至达到最大迭代轮数，最后再按照某种合成方式将这这些训练出来的弱学习器合成到一起，形成最终的强学习器。其典型代表包括bagging算法和随机森林（Random Forest）算法等。

5.6.1 集成学习概述

4. 弱学习器的合成方式(1/2)

弱学习器的合成方式是指当利用集成学习方法训练出所需要的全部弱学习器后，将这些弱学习器集成到一起，形成一个强学习器的方式。常用的合成方式包括代数合成法、投票法。

（1）代数合成法

这种方法是通过代数表达式对诸弱学习器进行合成，获得表达式最大支持的结果作为合成结果。常用的代数合成法包括平均法、加权平均法等。

若假设D为训练集，X为训练集的输入向量，Y为训练集的输出向量；T为训练过程需要迭代的总轮数；t=1, 2, …, T，为训练过程的当前迭代轮数；J为分类的问题的最大类别个数，j=1, 2, …, J，为弱学习器的分类结果；$h_{t,j}(X)$是弱学习器t得出分类结果j的判断，若分类结果为j，则$h_{t,j}(X)=1$，否则$h_{t,j}(X)=0$。

① 平均法为

$$H_{final}(X) = arg\,max_j \frac{1}{T}\sum_{t=1}^{T} h_{t,j}(X)$$

②加权平均法为

$$H_{final}(X) = arg\,max_j \frac{1}{T}\sum_{t=1}^{T} w_t h_{t,j}(X)$$

其中，w_t为为第t个弱学习器的权重。

5.6.1 集成学习概述

4. 弱学习器的合成方式(2/2)

（2）投票法

投票法的基本思想是对训练出来的所有弱分类器，按照某种投票原则进行投票表决。常用的投票方法有相对多数投票、加权投票法等。

所谓相对多数投票法就是我们平常所说的少数服从多数。即在T个弱学习器中，选择对样本X预测结果中得票最多的弱学习器的学习结果作为强学习器的学习结果。所谓加权投票法，与加权平均法一样，需要对每个弱学习器的票数再乘以其自身的权重，然后再对加权票数求和。

以简单的多数投票法为例，有：

$$\mathbf{H}(\boldsymbol{X}) = \arg \max_{y \in Y} \sum_{t=1}^{T} \boldsymbol{I}(\boldsymbol{h_t}(\boldsymbol{x}) = \boldsymbol{y})$$

其中，符号I为取真值运算。当弱学习器$h_t(x)$的输出与训练样本中x对应的输出y相同时，$I(h_t(x)=y)$取1；否则，$I(h_t(x)=y)$取0。

本节结束！