第六章 机器学习

主要内容：

§6.1 什么是机器学习

§6.2 机器学习的分类及学习策略

§6.3 机器学习算法

Machine Learning

6.1 What is Machine Learning

- *Learning denotes changes in the system that enable a system to do the same task more efficiently the next time.*

—— Herbert Simon,1983

赫伯特·西蒙
(Herbert Simon)

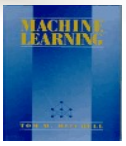
- *Learning is making useful changes in the workings of our minds.*

—— Marvin Minsky, 1986

马文·闵斯基
(Marvin Minsky)

Machine Learning

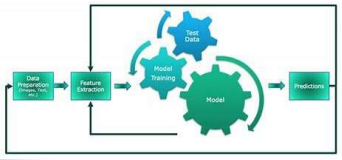
What is Machine Learning



- Machine learning <T, P, E>:
  - Computer automatically improves
    - at task T (任务)
    - according to performance metric P (性能)
    - through experience E (经验)

—— Tom Mitchell，1997

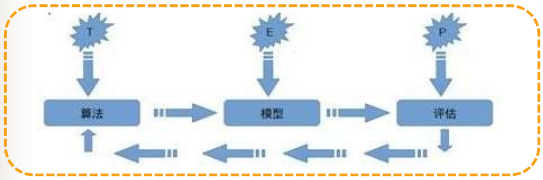
A Standard Machine Learning Pipeline



汤姆·米切尔
(Tom M. Mitchell)

T, P, E

三个关键词：任务、经验、性能



数据通过算法构建出模型并对模型进行评估，评估的性能如果达到要求就拿这个模型来测试其他的数据，如果达不到要求就调整算法来重新建立模型，再次进行评估，如此循环往复，最终获得满意的经验来处理其他的数据。

T, P, E

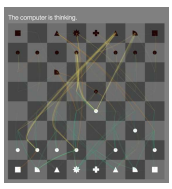
Examples of the Learning Tasks

- 下棋

T: 下棋

P: 比赛中击败对手的概率

E: 与自己对弈的训练



T, P, E

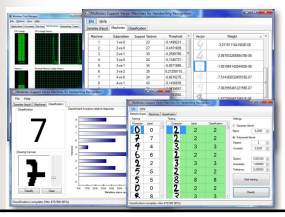
Examples of the Learning Tasks

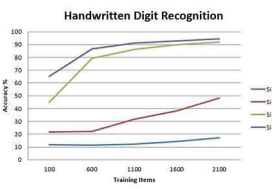
- 手写体识别

T: 识别手写文字

P: 识别的正确率

E: 已经做好的具有代表性分类的手写体数据库





T, P, E

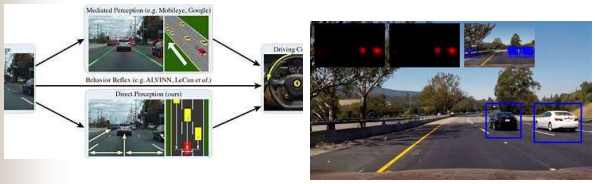
Examples of the Learning Tasks

- 自动驾驶

  T: 通过视觉传感器在高速路上自动驾驶

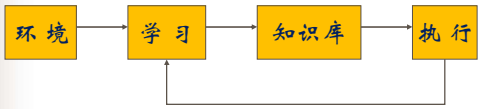
  P: 平均无差错行驶里程

  E: 观察人在驾驶过程中记录图像和驾驶指令数据库



学习系统

- 学习系统的基本结构

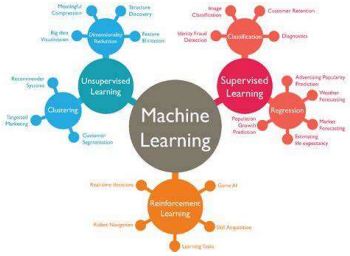


- 影响学习系统设计的要素
  - 环境：环境向系统提供信息的水平（一般化程度）和质量（正确性）
  - 知识库：表达能力，易于推理，容易修改，知识表示易于扩展。

机器学习分类

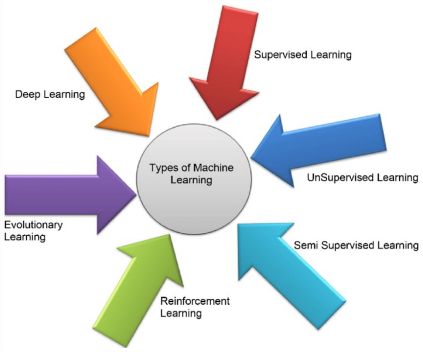
6.2 机器学习的分类及学习策略

1. 按学习能力分类
   - 有监督学习
   - 无监督学习
   - 强化学习



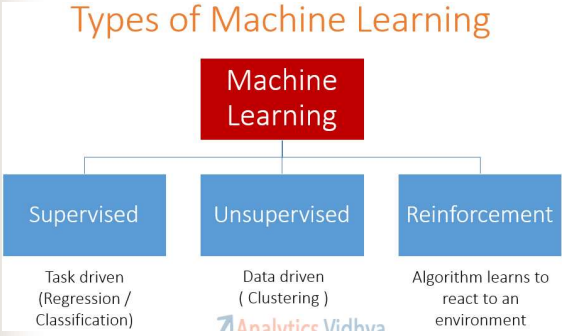
机器学习分类

按学习能力分类（更细致的划分）



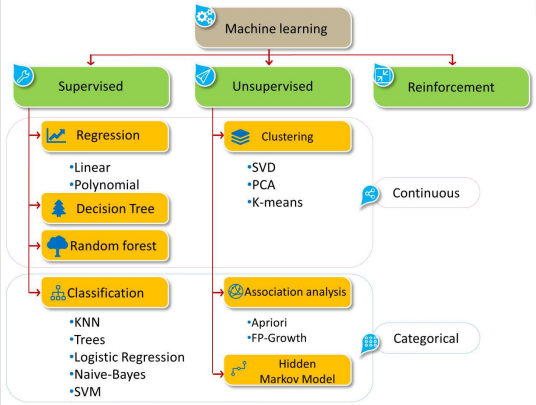
机器学习分类

机器学习方法比较



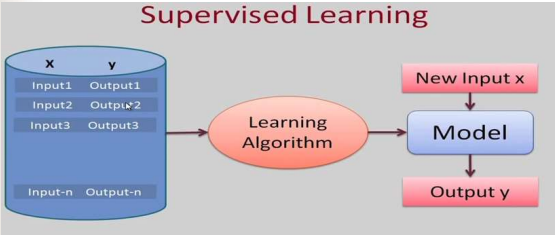
机器学习分类

机器学习方法中的算法

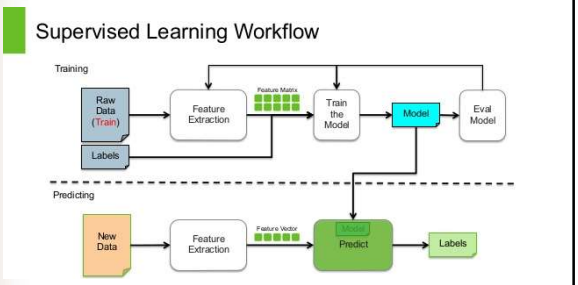


机器学习按学习能力的分类 – 有监督学习

从给定的训练数据集中学习一个模型，当新的数据到来时，可以根据这个模型预测结果；

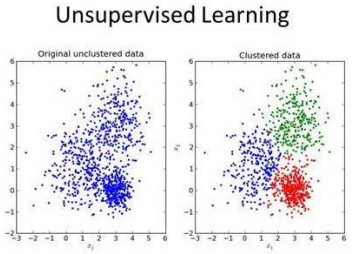


监督式学习建立一个学习过程，将预测结果与“测试数据”的实际结果进行比较，不断调整预测模型，直到模型的预测结果达到一个预期的准确率。



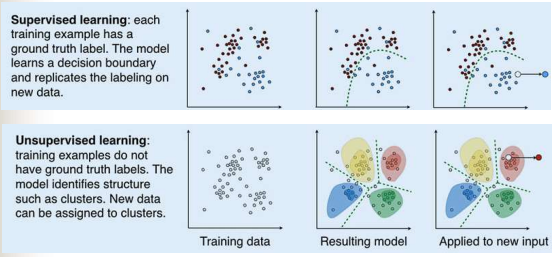
机器学习按学习能力的分类 – 无监督学习

在没有类别信息情况下，通过对所研究对象的大量样本的数据分析实现对样本分类的一种数据处理方法

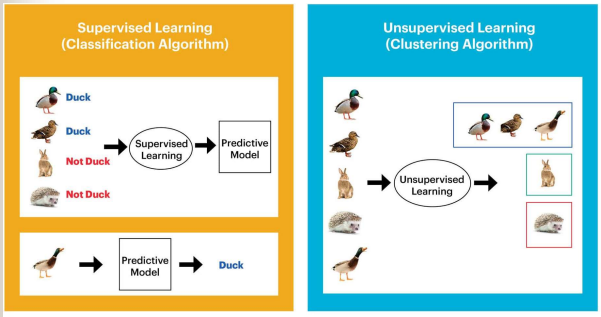


监督学习&无监督学习的区别：

训练集目标是否被标记。他们都有训练集，且都有输入和输出。

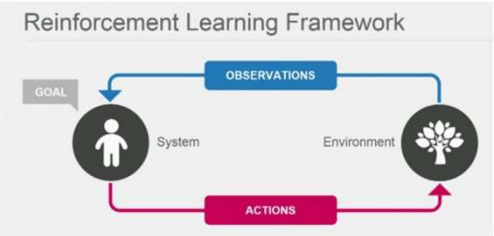


监督学习&无监督学习的区别：



机器学习按学习能力的分类 – 强化学习

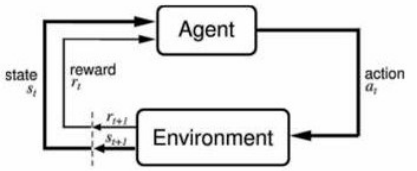
每个动作都会对环境有所影响，学习对象根据观察到的周围环境的反馈来做出判断和学习 (Learn through trial and error from interaction with an environment)



Basic Idea

Reinforcement Learning

- Basic idea:
  - Receive feedback in the form of rewards
  - Agent's utility is defined by the reward function
  - Must learn to act so as to maximize expected rewards



分类

2. 按学习方法分类

- 机械式学习（Rote learning）
- 归纳学习 （Induction learning）
- 类比学习
- 解释学习

机械学习

机器学习按学习方法的分类 – 机械学习

- 机械学习（Rote learning）
  - 简单的学习方法
  - 就是记忆，即把新的知识存储起来，供需要检索时调用
  - 不需计算和推理

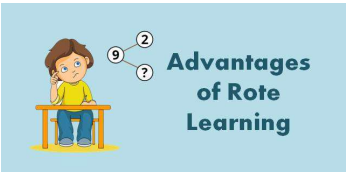
$(x_1,x_2,\ldots,x_n)$ → f → $(y_1,y_2,\ldots,y_n)$ → 存储 → $[(x_1,x_2,\ldots,x_n),(y_1,y_2,\ldots,y_n)]$

输入模式　执行函数　输出模式　　输入输出模式对

机械式学习的模型

主要问题

- 机械学习的主要问题
  - 存储信息：采用好的存储方式，使检索速度快；
  - 环境的稳定性与存储信息的适用性：机械学习系统须保证所保存的信息适应于外界环境变化的需要；
  - 存储与计算之间的权衡：不降低系统的效率。

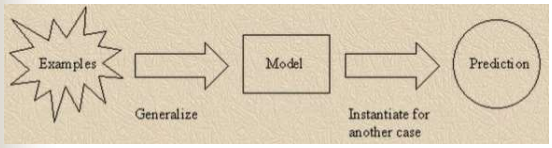


Induction

机器学习按学习方法的分类 – 归纳学习

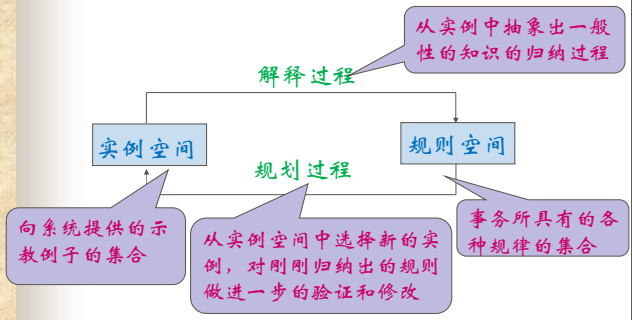
Induction Learning

- Induction is the process of reaching a general conclusion from specific examples



Induction

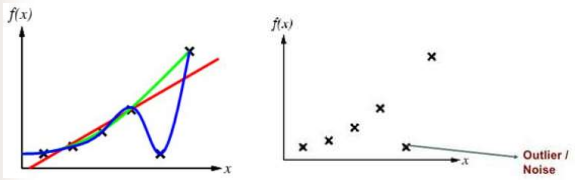
- 实例归纳学习的模式



Induction

Induction Learning

- Given examples of a function $(x, f(x))$
- Predict function $f(x)$ for new examples x



曲线拟合

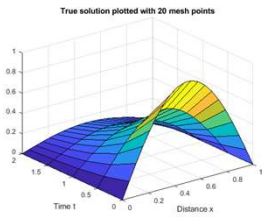
- Curve or Mesh Fitting

Given examples: (x, t, z)

Example1: (1,0,0), Example2: (0.5,0,0.8),

Example3: (0,1,0) …

If x =, t =, then z =?



Example

举例：学习“同花”概念

实例空间：(2, 梅花)， (3, 梅花)， (5, 梅花)

(J, 梅花)， (K, 梅花)

规则空间：描述一手牌的全部谓词表达式

如：SUIT(花色)，RANK(点数)，常量梅花，方块，A，1，2…

SUIT(c1，x)∧SUIT(c2，x)∧SUIT(c3，x)∧SUIT(c4，x)

->同花(c1, c2, c3, c4)

Example

- 归纳推理的方法
  - 变量代换常量

实例1：SUIT(c1,梅花)∧SUIT(c2,梅花)∧SUIT(c3,梅花)∧SUIT(c4，梅花)

实例2：SUIT(c1,红桃)∧SUIT(c2,红桃)∧SUIT(c3,红桃)∧SUIT(c4,红桃)

用变量x代替常量：

规则1：SUIT(c1,x)∧SUIT(c2,x)∧SUIT(c3,x)∧SUIT(c4,x) ->同花(c1,c2,c3,c4)

舍弃条件

- 舍弃条件

示例：SUIT(c1,红桃) ∧ RANK(c1,2) ∧ SUIT(c2,红桃) ∧ RANK(c2,4) ∧ SUIT(c3,红桃) ∧ RANK(c3,6) ∧ SUIT(c4,红桃) ∧ RANK(c4,7) -> 同花(c1,c2,c3,c4)

舍去点数，用x代替红桃

规则1：SUIT(c1,x)∧SUIT(c2,x)∧SUIT(c3,x)∧ SUIT(c4,x) -> 同花(c1,c2,c3,c4)

增加选择项

- 增加选择项
  - 示例1：RANK(c1,J)->FACE(c1)
  - 示例2：RANK(c1,Q)->FACE(c1)
  - 示例3：RANK(c1,K)->FACE(c1)

规则2：RANK(c1,J)∨ RANK(c1,Q)∨RANK(c1,K) ->FACE(c1)

Example

示例1：某天下雨，且自行车在路上出了毛病要修理，所以他上班迟到。

示例2：某天没下雨，但交通堵塞，所以他上班迟到。

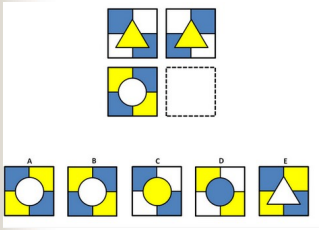
通过归纳归结，得：

如果自行车在路上出了毛病要修理，或者交通堵塞，则他有可能上班迟到

类比学习

机器学习按学习方法的分类 – 类比学习

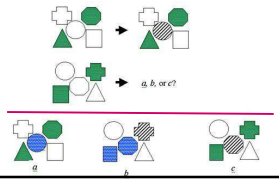
- 通过类比，即通过对相似事物加以比较所进行的一种学习



类比学习过程

类比推理过程：

- 回忆与联想：找出当前情况的相似情况
- 选择：选择最相似的情况及相关知识
- 建立对应关系：建立相似元素之间的映射
- 转换：求解问题或产生新的知识



类比学习方式

类比学习方式

类比学习是利用二个不同域（源域、目标域）中的知识相似性，可以通过类比，从源域的知识推导出目标域的相应知识，从而实现学习。

- E.g. 一个从未开过Truck的司机，只要他有开Car的知识就可完成开Truck的任务。
- E.g. 若把某个人比喻为消防车，则可通过观察消防车的行为，推断出这个人的性格。

神经学习

机器学习按学习方法的分类 – 神经学习

神经学习

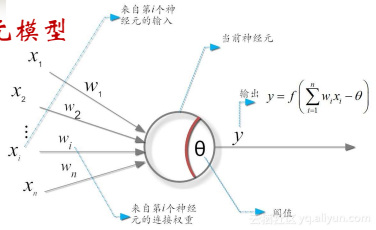
是对人脑神经系统学习机理的一种模拟

按学习规则分类：

- Hebb学习
- 反向传播 (BP)网络学习
- Hopfield学习

神经元模型

一个简单的神经元模型



权值调整公式

$w_{ij}(t+1)= w_{ij}(t)+\eta\,(d_i - y_i)x_j$

w_{ij}：i到j的权值　y_i：i的实际输出

η：学习率　d_i：i的期望输出

纠错学习

■ 纠错学习

纠错学习的基本思想：利用神经网络的期望输出与实际输出之间的偏差作为连接权值调整的参考，并最终减少这种偏差。纠错学习是一种有监督的学习过程。

$$w_{ij}(t+1) = w_{ij}(t) + \eta[d_j(t) - y_j(t)]x_i(t)$$

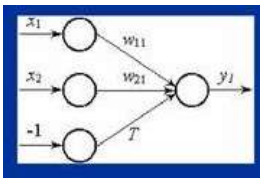
$y_j(t)$为神经元j的实际输出

$d_j(t)$为神经元j的希望输出

Hebb Learning

■ Hebb Learning Rule

Supervised learning：如果某一神经元同另一直接与它连接的神经元同时处于兴奋状态，那么这两个神经元之间的连接强度将得到加强，反之应该减弱



Hebb Learning

■ Hebb Learning Rule

Hebb学习对连接权值的调整可表示为：

$$w_{ij}(t+1) = w_{ij}(t) + \eta[x_i(t)x_j(t)]$$

$w_{ij}(t+1)$表示新的权值；η取正值，称为学习因子，

$x_i(t)$、$x_j(t)$表示t时刻第i个和第j个神经元的状态。

Hebb Learning

■ Hebb Learning Rule

Algorithm : Hebb net (supervised) learning algorithm

- Step0 . initialize weights and bias
- Step1 . per each training sample s:t do steps 2 – 4
- Step2 . Set activations of input units :
  xi = si
- STEP3 . Set activations of output units :
  y = t
- Step4. update weight and bias :
  wi (new) = wi(old) + xi. y
  b (new) = b (old) + y

Hebb Learning

■ Hebb Learning Rule

e.g. Training an AND gate

x_1	x_2	t
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

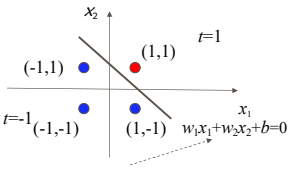


Figure out the function, confirm the value of w_1, w_2 , b

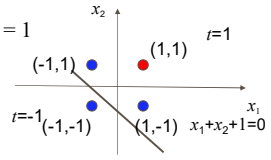
Hebb Learning

step 1 Set initial

$w_1 = 0, w_2 = 0, b = 0$

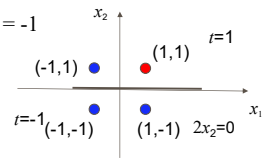
step 2 Input $(x_1,x_2) = (1,1)$, $t = 1$

$$\begin{cases} w_1 = 0+1\cdot 1 = 1 \\ w_2 = 0+1\cdot 1 = 1 \\ b = 0+1 = 1 \end{cases}$$



step 3 Input $(x_1,x_2) = (1,-1)$, $t = -1$

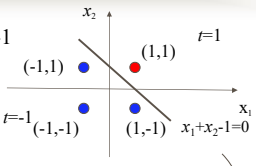
$$\begin{cases} w_1 = 1+1\cdot(-1) = 0 \\ w_2 = 1+(-1)\cdot(-1) = 2 \\ b = 1+(-1) = 0 \end{cases}$$



Hebb Learning

step 4 Input $(x_1,x_2) = (-1,1)$, $t = -1$

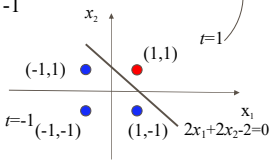
$$\begin{cases} w_1 = 0 + -1 \cdot (-1) = 1 \\ w_2 = 2 + 1 \cdot (-1) = 1 \\ b = 0 + (-1) = -1 \end{cases}$$



AND Gate

step 5 Input $(x_1,x_2) = (-1,-1)$, $t = -1$

$$\begin{cases} w_1 = 1 + (-1) \cdot (-1) = 2 \\ w_2 = 1 + (-1) \cdot (-1) = 2 \\ b = -1 + (-1) = -2 \end{cases}$$

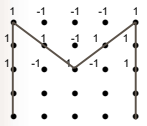


Hebb Learning

Application: Design a Neural Network to recognize "M" and "L".

1. Represent "M" and "L" as vector

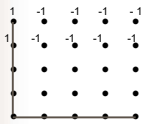
(1) "M"



M = (1 -1 -1 -1 1, 1 1 -1 1 1, 1 -1 1 -1 1, 1 -1 -1 -1 1, 1 -1 -1 -1 1)

desired output = target = $t = 1$

(2) "L"

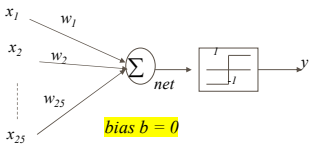


L = (1 -1 -1 -1 -1, 1 -1 -1 -1 -1, 1 -1 -1 -1 -1,1 -1 -1 -1 -1, 1 1 1 1 1)

desired output = target = $t = -1$

Hebb Learning

2. Use Hebb Learning Rule to Train the Neural Network



Hebb learning rule:

(1) Set initial $w_i = 0$ i=1~25

(2) Input $x = M$ 且 $t = 1$

$\mathbf{W} = 0 + \mathbf{X} \cdot 1$

$\therefore \mathbf{W} = M$

(3) Input $x = L$ 且 $t = -1$

$\mathbf{W} = M + \mathbf{X} \cdot (-1)$

$\therefore \mathbf{W} = M - L$

$W = M\text{-}L$ = (0 0 0 0 2,0 2 0 0 2, 0 0 2 0 2,0 0 0 0 2,0 -2 -2 -2 0)

Hebb Learning

$W = M\text{-}L$ = (0 0 0 0 2,0 2 0 0 2, 0 0 2 0 2,0 0 0 0 2,0 -2 -2 -2 0)

3. Recognition

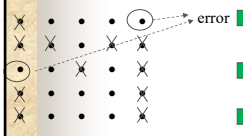
if input $X = M$=(1 -1 -1 -1 1, 1 1 -1 1 1, 1 -1 1 -1 1, 1 -1 -1 -1 1, 1 -1 -1 -1 1)

$net = \mathrm{W^T X} = 20 > 0 \Rightarrow y = 1$

if input $X = L$=(1 -1 -1 -1 -1, 1 -1 -1 -1 -1, 1 -1 -1 -1 -1,1 -1 -1 -1 -1, 1 1 1 1 1)

$net = \mathrm{W^T X} = -20 < 0 \Rightarrow y = -1$

if there is a not perfect "*M*"



X = (1 -1 -1 -1 -1, 1 1 -1 1 1, -1 -1 1 -1 1,1 -1 -1 -1 1, 1 -1 -1 -1 1)

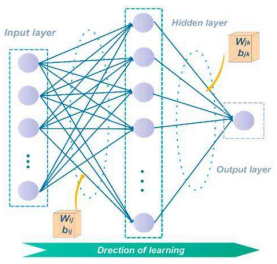
$net = \mathrm{W^T X} = 16 > 0 \Rightarrow y = 1$

The result still "M"

多层神经网络

多层神经网络

多层网络可以表示任意函数，但要构造高效的学习算法较困难，因为隐层神经元的期望输出不易给出。



Backpropagation

反向传播算法（Backpropagation Learning）

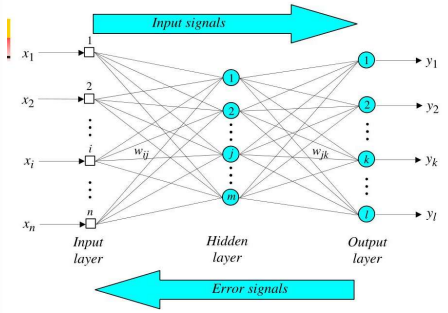
- 有监督学习
- 核心思想：
  - 将输出误差通过隐含层向输入层反传。
  - 利用传播公式，沿着减小误差的方向调整网络，连接权值和阈值的过程。
- 学习过程

信号的正向传播，误差的反向传播。

三层BP网络

三层BP网络

Three-layer back-propagation neural network



修正公式

BP算法中权值的修正公式

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

$$\delta_j = o_j(1-o_j)(t_j - o_j) \quad \text{if } j \text{ is an output unit}$$

$$\delta_j = o_j(1-o_j)\sum_k \delta_k w_{kj} \quad \text{if } j \text{ is a hidden unit}$$

- where η is a constant called the learning rate

t_j is the correct (expected-teacher) output for unit j

o_j is the computed (current) output for unit j

δ_j is the error measure for unit j

Example

- 首先计算输出层单元的误差，并用该误差调整输出层的权值

Current output: o_j=0.2

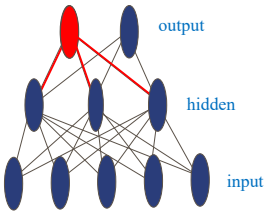
Correct output: t_j=1.0

Error $\delta_j = o_j(1-o_j)(t_j-o_j)$

0.2(1–0.2)(1–0.2)=0.128

Update weights into j

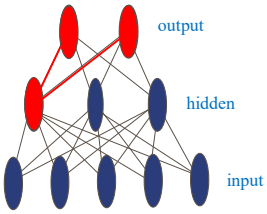
$$\Delta w_{ji} = \eta \delta_j x_{ji}$$



Example

- 接着根据输出层的误差计算隐层单元的误差

$$\delta_j = o_j(1-o_j)\sum_k \delta_k w_{kj}$$



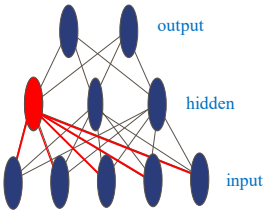
Example

- 最后根据隐层单元的误差调整下层的权值

$$\delta_j = o_j(1-o_j)\sum_k \delta_k w_{kj}$$

Update weights into j

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$



BP算法步骤

BP算法步骤

(1)初始化权值及阈值为小的随机数

(2)给出输入x_0，$x_1 \cdots x_{n-1}$及期望输出t_0，t_1，t_{n-1}

(3)逐层计算输出

(4)修正权值

(5)重复（3）～（5），直到对所有样本权值不变

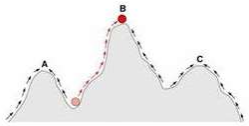
BP算法局限性

反向传播算法缺点

- 收敛性和局部极值
- 过渡拟合

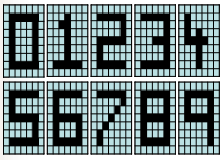
Disadvantage of Backprop

- Easy to get stuck in local optima



Example

E.g. 设计一个三层BP网络对数字0至9进行分类。



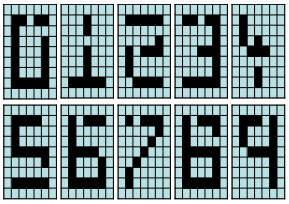
数字用9×7的网格表示，灰色像素代表0，黑色像素代表1。将每个网格表示为0，1的长位串。位映射由左上角开始向下直到网格的整个一列，然后重复其他列。

选择BP网络结构为63-6-9。9×7个输入结点，对应上述网格的映射。9个输出结点对应10种分类。

学习步长为0.3。训练600个周期，如果输出结点的值大于0.9，则取为ON，如果输出结点的值小于0.1，则取为OFF。

Example

当训练成功后，对如图所示测试数据进行测试。测试数据都有一个或者多个位丢失。



结果表明：除了8以外，所有被测的数字都能够被正确地识别。

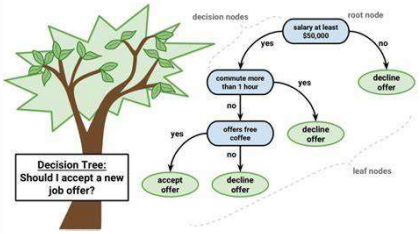
对于数字8，神经网络的第6个结点的输出值为0.53，第8个结点的输出值为0.41，表明第8个样本是模糊的，可能是数字6，也可能是数字8，但也不完全确信是两者之一。

决策树学习

机器学习按学习方法的分类 – 决策树学习

什么是决策树

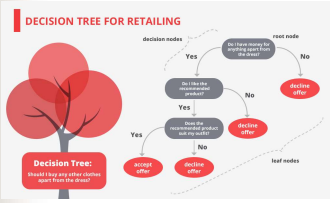
一种树形结构，通过做出一系列决策（选择）来对数据进行划分。



什么是决策树

■ 什么是决策树

从根节点开始，测试待分类项中对应的特征属性，并按照其值选择输出分支，直到叶子节点，将叶子节点的存放的类别作为决策结果。

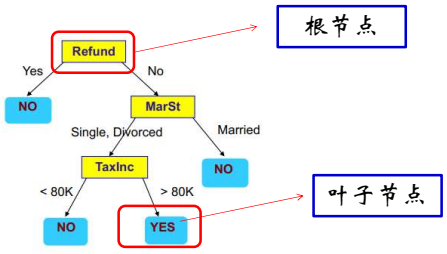


- 根节点-分类开始
- 叶节点-分类结果

什么是决策树

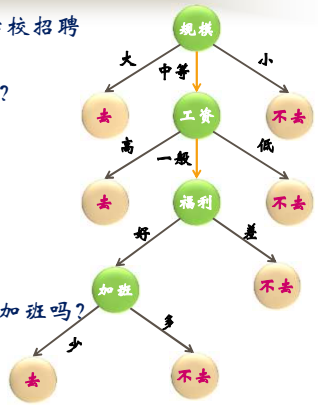
什么是决策树

从训练数据中学习得出一个类似于流程图的树型结构



Example

- 学生甲：XX公司来学校招聘，去看吗？
- 学生乙：是大公司吗？
- 甲：中等
- 乙：工资高吗？
- 甲：一般
- 乙：有五险一金吗？
- 甲：有
- 乙：活累不累，老要加班吗？
- 甲：据说还好
- 乙：那就去看看吧



Example

课堂练习:训练样例如下，构造一个邮件阅读决策树

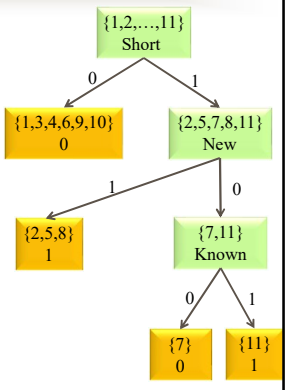
序号	Known	New	Short	Home	Reads
1	1	1	0	1	0
2	0	1	1	0	1
3	0	0	0	0	0
4	1	0	0	1	0
5	1	1	1	1	1
6	1	0	0	0	0
7	0	0	1	0	0
8	0	1	1	0	1
9	1	0	0	1	0
10	1	1	0	0	0
11	1	0	1	1	1

Example

举例:邮件阅读

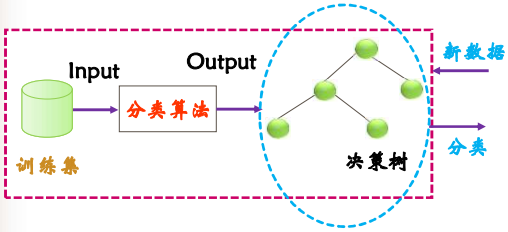
训练样例

序号	Known	New	Short	Home	Reads
1	1	1	0	1	0
2	0	1	1	0	1
3	0	0	0	0	0
4	1	0	0	1	0
5	1	1	1	1	1
6	1	0	0	0	0
7	0	0	1	0	0
8	0	1	1	0	1
9	1	0	0	1	0
10	1	1	0	0	0
11	1	0	1	1	1



学习过程

- 决策树学习过程
  - 有监督学习
  - 给定训练样例，包含正例和反例
  - 从训练样例创建决策树



学习策略

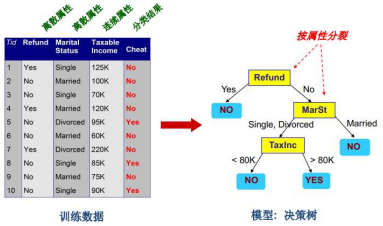
决策树学习策略

- 基本思想：分两个步骤
  - 树的生成
    - 一开始数据都在根节点
    - 递归的进行数据分片
  - 树的修剪：去掉一些可能是噪音或者异常的数据
- 决策树使用：对未知数据进行分割
  - 按照决策树上采用的分割属性逐层往下，直到叶子节点

划分属性

决策树学习的关键是选择最优划分属性

我们希望决策树的分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”越来越高，可以高效地从根结点到达叶结点，得到决策结果。



决策树的生成

决策树的生成

在当前状态下选择哪个属性作为分类依据。根据不同的目标函数，建立决策树主要有以下三种算法度量结点“纯度”的指标：

信息增益—ID3

增益比率—C4.5

基尼指数—CART

算法	支持模型	树结构	特征选择	连续值处理	缺失值处理	剪枝	特征属性多次使用
ID3	分类	多叉树	信息增益	不支持	不支持	不支持	不支持
C4.5	分类	多叉树	信息增益率	支持	支持	支持	不支持
CART	分类、回归	二叉树	基尼系数、均方差	支持	支持	支持	支持

ID3 Algorithm

Decision Tree Generation-ID3 Algorithm

ID3 Algorithm

- Is the algorithm to construct a decision tree
- Using *Entropy* to generate the *Information Gain*
- The best value then be selected

信息熵

信息熵

一条信息的信息量和它的不确定性有着直接的关系

公式定义：

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

变量的不确定性越大，熵也就越大，需要的信息量也越大，纯度就越小。

ID3 Algorithm

Decision Tree Generation-ID3 Algorithm

Entropy

- The complete formula for entropy is

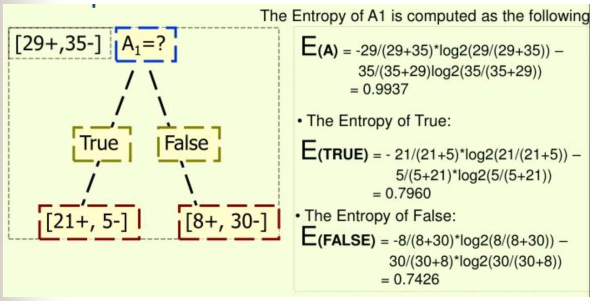
$$E(S) = -(p_+)*\log2(p_+) - (p_-)*\log2(p_-)$$

- where p_+ is the proportion of positive samples
- where p_- is the proportion of negative samples
- where S is the sample of attributions

ID3 Algorithm

Decision Tree Generation-ID3 算法

Example $E(S) = -(p_+)*\log2(p_+) - (p_-)*\log2(p_-)$



ID3 Algorithm

Decision Tree Generation-ID3 算法

Information Gain

Gain (Sample, Attributes) or Gain (S, A) is expected reduction in entropy due to sorting S on attribute A

$$Gain(S, A) = Entropy(S) - \sum v \in values(A)\ |Sv|/|S|\ Entropy(Sv)$$

So, for the previous example, the Information Gain is calculated:

$$G(A1) = E_{(A1)} - \frac{(21+5)}{(29+35)} * E_{(TRUE)} - \frac{(8+30)}{(29+35)} * E_{(FALSE)}$$

$$= E_{(A1)} - \frac{26}{64} * E_{(TRUE)} - \frac{38}{64} * E_{(FALSE)}$$

$$= 0.9937 - \frac{26}{64} * 0.796 - \frac{38}{64} * 0.7426 = 0.5465$$

Example

- The complete example:

Consider the following table

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

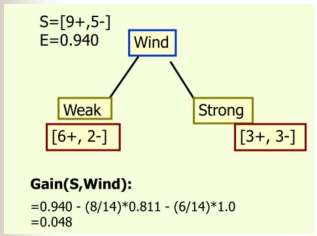
Example

Example

- Calculating the *Information Gains* for each of the weather attributes:
  - For the Wind
  - For the Humidity
  - For the Outlook

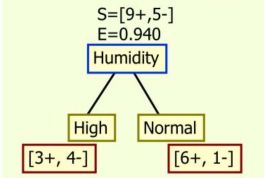
Example

- For the Wind



Gain(S,Wind):
=0.940 - (8/14)*0.811 - (6/14)*1.0
=0.048

- For the Humidity

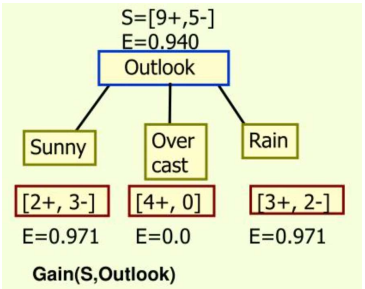


Gain(S,Humidity)
=0.940-(7/14)*0.985 – (7/14)*0.592
=0.151

$$Gain(S,A) = Entropy(S) - \sum v \in values(A)\ |Sv|/|S|\ Entropy(Sv)$$

Example

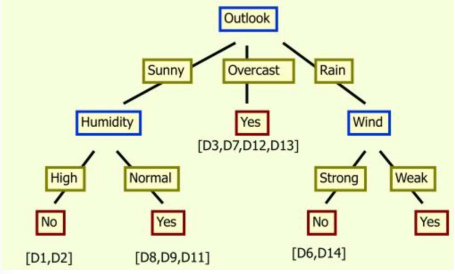
- For the Outlook



Gain(S,Outlook)
=0.940-(5/14)*0.971 -(4/14)*0.0 – (5/14)*0.0971
=0.247

Example

- Complete tree
  - Then here is the complete tree:



Example

Example

年龄	收入	学生	信用	买了电脑
<30	高	否	一般	否
<30	高	否	好	否
30-40	高	否	一般	是
>40	中等	否	一般	是
>40	低	是	一般	是
>40	低	是	好	否
30-40	低	是	好	是
<30	中	否	一般	否
<30	低	是	一般	是
>40	中	是	一般	是
<30	中	是	好	是
30-40	中	否	好	是
30-40	高	是	一般	是
>40	中	否	好	否

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

$|D|=14$
$|C_{1,D}|=5$
$|C_{2,D}|=9$

$$Info(D) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14} = 0.940$$

Example

Example $Info_A(D) = \sum_{j=1}^{v}\left[\left(\frac{|D_j|}{|D|}\right) * Info(D_j)\right]$

年龄	收入	学生	信用	买了电脑
<30	高	否	一般	否
<30	高	否	好	否
30-40	高	否	一般	是
>40	中等	否	一般	是
>40	低	是	一般	是
>40	低	是	好	否
30-40	低	是	好	是
<30	中	否	一般	否
<30	低	是	一般	是
>40	中	是	一般	是
<30	中	是	好	是
30-40	中	否	好	是
30-40	高	是	一般	是
>40	中	否	好	否

年龄<30的有5个, 其中3个为“否”
年龄30-40的有4个, 其中0个为“否”
年龄>40的有5个, 其中2个为“否”

$$Info_{年龄}(D) = \frac{5}{14}(-\frac{3}{5}\log\frac{3}{5}-\frac{2}{5}\log\frac{2}{5}) + \frac{4}{14}(-\frac{4}{4}\log\frac{4}{4}-\frac{0}{4}\log\frac{0}{4}) + \frac{5}{14}(-\frac{2}{5}\log\frac{2}{5}-\frac{3}{5}\log\frac{3}{5}) = 0.694$$

Gain(年龄)
= Info(D) - Info$_{年龄}$(D)
= 0.940 - 0.694 = 0.246

Example

Example $Info_A(D) = \sum_{j=1}^{v}\left[\left(\frac{|D_j|}{|D|}\right) * Info(D_j)\right]$

年龄	收入	学生	信用	买了电脑
<30	高	否	一般	否
<30	高	否	好	否
30-40	高	否	一般	是
>40	中	否	一般	是
>40	低	是	一般	是
>40	低	是	好	否
30-40	低	是	好	是
<30	中	否	一般	否
<30	低	是	一般	是
>40	中	是	一般	是
<30	中	是	好	是
30-40	中	否	好	是
30-40	高	是	一般	是
>40	中	否	好	否

收入=高的有4个, 其中2个为“否”
收入=中的有6个, 其中2个为“否”
收入=低的有4个, 其中1个为“否”

$$Info_{收入}(D) = \frac{4}{14}(-\frac{2}{4}\log\frac{2}{4}-\frac{2}{4}\log\frac{2}{4}) + \frac{6}{14}(-\frac{2}{6}\log\frac{2}{6}-\frac{4}{6}\log\frac{4}{6}) + \frac{4}{14}(-\frac{1}{4}\log\frac{1}{4}-\frac{3}{4}\log\frac{3}{4}) = 0.911$$

Gain(收入)
= Info(D) - Info$_{收入}$(D)
= 0.940 - 0.911 = 0.029

Example

Example $Info_A(D) = \sum_{j=1}^{v}\left[\left(\frac{|D_j|}{|D|}\right) * Info(D_j)\right]$

年龄	收入	学生	信用	买了电脑
<30	高	否	一般	否
<30	高	否	好	否
30-40	高	否	一般	是
>40	中	否	一般	是
>40	低	是	一般	是
>40	低	是	好	否
30-40	低	是	好	是
<30	中	否	一般	否
<30	低	是	一般	是
>40	中	是	一般	是
<30	中	是	好	是
30-40	中	否	好	是
30-40	高	是	一般	是
>40	中	否	好	否

是学生的有7个, 其中1个为“否”
不是学生的有7个, 其中4个为“否”

$$Info_{学生}(D) = \frac{7}{14}(-\frac{1}{7}\log\frac{1}{7}-\frac{6}{7}\log\frac{6}{7}) + \frac{7}{14}(-\frac{4}{7}\log\frac{4}{7}-\frac{3}{7}\log\frac{3}{7}) = 0.788$$

Gain(学生)
= Info(D) - Info$_{学生}$(D)
= 0.940 - 0.788 = 0.152

Example

Example $Info_A(D) = \sum_{j=1}^{v}\left[\left(\frac{|D_j|}{|D|}\right) * Info(D_j)\right]$

年龄	收入	学生	信用	买了电脑
<30	高	否	一般	否
<30	高	否	好	否
30-40	高	否	一般	是
>40	中	否	一般	是
>40	低	是	一般	是
>40	低	是	好	否
30-40	低	是	好	是
<30	中	否	一般	否
<30	低	是	一般	是
>40	中	是	一般	是
<30	中	是	好	是
30-40	中	否	好	是
30-40	高	是	一般	是
>40	中	否	好	否

信用好的有6个, 其中3个为“否”
信用一般的有8个, 其中2个为“否”

$$Info_{信用}(D) = \frac{6}{14}(-\frac{3}{6}\log\frac{3}{6}-\frac{3}{6}\log\frac{3}{6}) + \frac{8}{14}(-\frac{2}{8}\log\frac{2}{8}-\frac{6}{8}\log\frac{6}{8}) = 0.892$$

Gain(信用)
= Info(D) - Info$_{信用}$(D)
= 0.940 - 0.892 = 0.048

Example

Example

“年龄”属性具体最高信息增益，成为分裂属性

年龄

<30

收入	学生	信用	买了电脑
高	否	一般	否
高	否	好	否
中	否	一般	否
低	是	一般	是
中	是	好	是

30-40

收入	学生	信用	买了电脑
中	否	一般	是
低	是	一般	是
低	是	好	否
中	是	一般	是
中	否	好	否

>40

收入	学生	信用	买了电脑
高	否	一般	是
低	是	好	是
中	否	好	是
高	是	一般	是

Example

Example

收入	学生	信用	买了电脑
高	否	一般	否
高	否	好	否
中	否	一般	否
低	是	一般	是
中	是	好	是

Info$_{收入}$(D)
= 2/5 * (-2/2 * log2/2 - 0/2 * log0/2)
+ 2/5 * (-1/2 * log1/2 - 1/2 * log1/2)
+ 1/5 * (-1/1 * log1/1 - 0/1 * log0/1)
= 0.400

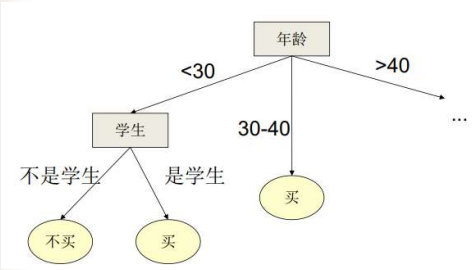
Info$_{学生}$(D)
= 3/5 * (-3/3 * log3/3 - 0/3 * log0/3)
+ 2/5 * (-2/2 * log2/2 - 0/2 * log0/2)
= 0

Info$_{信用}$(D)
= 3/5 * (-2/3 * log2/3 - 1/3 * log1/3)
+ 2/5 * (-1/2 * log1/2 - 1/2 * log1/2)
= 0.951

“学生”属性具体最高信息增益，成为分裂属性

Example

Example



Example

E.g. 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：

y1：必修AI

y2：选修AI

y3：不修AI

做出这些决策的依据有以下3个属性：

x1：学历层次　x1=1研究生，x1=2本科

x2：专业类别　x2=1电信类，x2=2机电类

x3：学习基础　x3=1修过AI，x3=2未修AI

下页的表给出了一个关于选课决策的训练例子集S。

Example

序号	属性值			决策方案
	x_1学历层次	x_2专业类别	x_3学习基础	y_i
1	1研究生	1电信类	1修过AI	y_3不修AI
2	1研究生	1电信类	2未修AI	y_1必修AI
3	1研究生	2机电类	1修过AI	y_3不修AI
4	1研究生	2机电类	2未修AI	y_2选修AI
5	2本科	1电信类	1修过AI	y_3不修AI
6	2本科	1电信类	2未修AI	y_2选修AI
7	2本科	2机电类	1修过AI	y_3不修AI
8	2本科	2机电类	2未修AI	y_3不修AI

Example

- 建立包含所有训练样例的根节点，计算其信息熵

$$Entropy(S)=\sum_{i=1}^{3}(-p_i log_2 p_i)$$
$$=-\left(\frac{1}{8}\right)log_2\left(\frac{1}{8}\right)-\left(\frac{2}{8}\right)log_2\left(\frac{2}{8}\right)-\left(\frac{5}{8}\right)log_2\left(\frac{5}{8}\right)=1.2988$$

计算S关于每个属性的期望熵

$$Entropy(S/x_i)=\sum_t\frac{|S_t|}{|S|}Entropy(S_i)$$

其中，t为属性x_i的属性值，S_t为$x_i=t$时的例子集，$|S|$和$|S_i|$分别是例子集S和S_i的大小。

$$Entropy(S/x_i)=\sum_t\frac{|S_t|}{|S|}Entropy(S_i)$$

Example

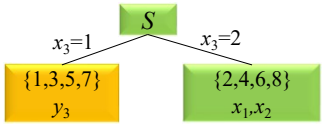
计算S关于属性（学历）的期望熵

$x_1=1: S_1=\{1,2,3,4\}$　　$x_2=2: S_2=\{5,6,7,8\}$

$$Entropy(S_1)=-\left(\frac{1}{4}\right)log_2\left(\frac{1}{4}\right)-\left(\frac{1}{4}\right)log_2\left(\frac{1}{4}\right)-\left(\frac{2}{4}\right)log_2\left(\frac{2}{4}\right)$$
$$=1.5$$
$$Entropy(S_2)=-\left(\frac{1}{4}\right)log_2\left(\frac{1}{4}\right)-\left(\frac{3}{4}\right)log_2\left(\frac{3}{4}\right)=0.8113$$
$$Entropy(S/x_1)=\left(\frac{4}{8}\right)*1.5+\left(\frac{4}{8}\right)*0.8113=1.1557$$

Example

- 类似，计算S关于属性“专业”和“学历基础”的期望熵
- 显然，期望熵越小，信息增益越大，应选择属性x_3对根节点进行扩展

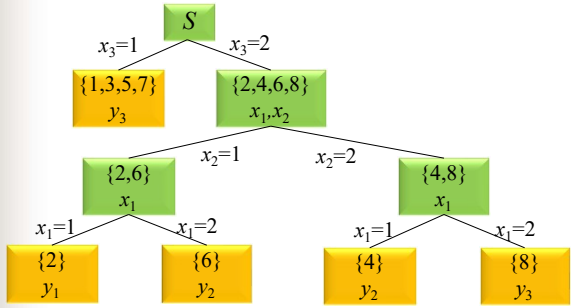


用属性x_3分类后的部分决策树

- 左边的节点只包含一种信息，无需再分
- 右边的节点在新的分类集中重复上述过程，计算属性x_1和x_2的期望熵均为1。任选一个，本例先选择x_2

Example

最终决策树



决策树优缺点

决策树优缺点

优点：

- 推理过程容易理解，计算简单，可解释性强。
- 比较适合处理有缺失属性的样本。
- 可自动忽略目标变量没有贡献的属性变量，也为判断属性变量的重要性，减少变量的数目提供参考。
- 作为一些更有用的算法的基石。例如：随机森林

决策树优缺点

决策树优缺点

缺点：

- 容易造成过拟合，需要采用剪枝操作。
- 忽略了数据之间的相关性。
- 对于各类别样本数量不一致的数据，信息增益偏向于那些更多数值的特征。

Thanks!