一叶收获

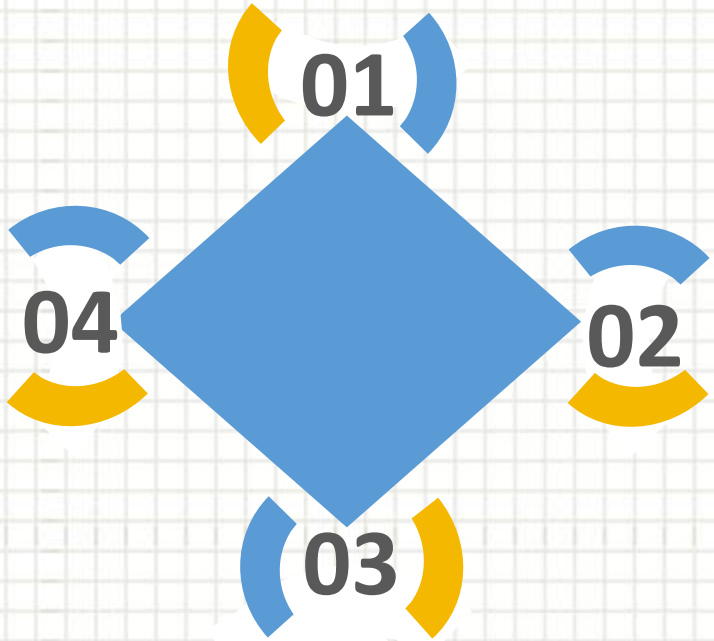
- 法线
- 属性筛选可用决策树的方法
- 累计BP法：对整个训练集计算一遍后才做参数更新
- 正则化损失的处理
- Batchsize较小些好

《机器学习》第四章

目录

决策树

基本流程&划分选择



剪枝处理

连续与缺失值

多变量决策树

基本流程 | 决策树

- 决策树（decision tree）:

决策树是基于树结构来对问题进行决策的。

- 一棵决策树包含一个根结点，若干个内部结点 和若干个叶结点。

叶结点 → 决策结果，其它结点→ 一个属性测试

- 决策树学习的目的是产生一棵泛化能力强的决策树。（处理未见示例的能力强）

划分选择　信息增益

- 信息熵（information entropy）：度量样本集合纯度最常用的一种指标。假定样本集合D中第k类样本所占比例为p_k(k = 1,2...,|y|),则D的信息熵定义为：

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k \ .$$

- 信息熵越小，D的纯度越高。

- 假定离散属性a有V个可能的取值$\{a^1,a^2,a^3,\ldots,a^V\}$,若使用a来对样本集D进行划分，会产生V个分支结点，第v个分支结点包含了D中所有属性a上取值为a^v的样本，记为D^v .
- 属性a对样本集D进行划分可获得信息增益(information gain)：

$$\mathrm{Gain}(D,a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|}\mathrm{Ent}(D^v)$$

- 信息增益越大，属性a的划分所获得的“纯度提升”越大

划分选择　信息增益

表 4.1　西瓜数据集 2.0

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

图 1

- 以属性“色泽”为例，该属性对D进行划分，可得到3个子集：D^1(色泽=青绿），D^2(色泽=乌黑），D^3（色泽=浅白）
- D^1包含编号{1,4,6,10,13,17}，其中正例p1=3/6,反例p2=3/6;
- D^2包含编号{2,3,7,8,9,15}，其中正例p1=4/6,反例p2=2/6;
- D^3包含编号{5,11,12,14,16}，其中正例p1=1/5,反例p2=4/5;
- 三个分支结点信息熵为：

$$\mathrm{Ent}(D^1) = -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1.000\,,$$

$$\mathrm{Ent}(D^2) = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = 0.918\,,$$

$$\mathrm{Ent}(D^3) = -\left(\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right) = 0.722\,,$$

- 色泽的信息增益为：

$$\begin{aligned}\mathrm{Gain}(D, \text{色泽}) &= \mathrm{Ent}(D) - \sum_{v=1}^{3}\frac{|D^v|}{|D|}\mathrm{Ent}(D^v)\\ &= 0.998 - \left(\frac{6}{17}\times 1.000 + \frac{6}{17}\times 0.918 + \frac{5}{17}\times 0.722\right)\\ &= 0.109\,.\end{aligned}$$

划分选择　信息增益

- 类似的，我们可以计算出其他属性的信息增益：

$$\text{Gain}(D, \text{根蒂}) = 0.143; \quad \text{Gain}(D, \text{敲声}) = 0.141;$$
$$\text{Gain}(D, \text{纹理}) = 0.381; \quad \text{Gain}(D, \text{脐部}) = 0.289;$$
$$\text{Gain}(D, \text{触感}) = 0.006.$$

- 由于“纹理”的信息增益最大，因此被选为划分属性：

图 2

- 以分支结点（“纹理=清晰”）为例，属性集合包含{色泽，根蒂，敲声，脐部，触感}，计算出各属性的信息增益：

$$\text{Gain}(D^1, \text{色泽}) = 0.043; \quad \text{Gain}(D^1, \text{根蒂}) = 0.458;$$
$$\text{Gain}(D^1, \text{敲声}) = 0.331; \quad \text{Gain}(D^1, \text{脐部}) = 0.458;$$
$$\text{Gain}(D^1, \text{触感}) = 0.458.$$

- 由于“根蒂”，“脐部”，“触感”的信息增益最大，则可任选其中之一作为划分属性，最终可得：

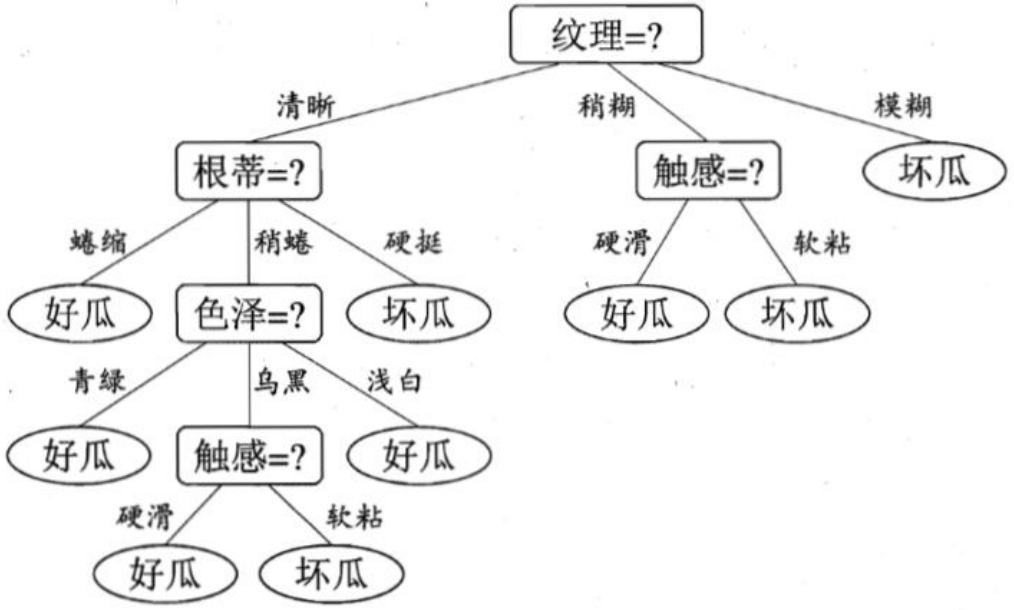


图 3

划分选择

增益率

- 信息增益准则对取值数目较多的属性有所偏好，
- 增益率则对取值数目较少的属性有所偏好
- 增益率（gain ratio)：

$$\text{Gain\_ratio}(D,a)=\frac{\text{Gain}(D,a)}{\text{IV}(a)},$$

- 其中，属性a的固有值(intrinsic value)：

$$\text{IV}(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

- 注意：先从候选划分属性中信息增益高于平均水平的属性，再从中选择增益率最高的。

基尼指数

- 数据集D的纯度可用基尼值来度量：

$$\begin{aligned}\text{Gini}(D)&=\sum_{k=1}^{|\mathcal{Y}|}\sum_{k'\neq k}p_k p_{k'}\\&=1-\sum_{k=1}^{|\mathcal{Y}|}p_k^2.\end{aligned}$$

- Gini（D)越小，数据集D的纯度越高。
- 属性a的基尼指数定义：

$$\text{Gini\_index}(D,a)=\sum_{v=1}^{V}\frac{|D^v|}{|D|}\text{Gini}(D^v)$$

- 在候选属性集合A中，使得划分后的基尼指数最小的属性作为最优划分属性

剪枝处理　预剪枝

- 剪枝(pruning)：决策树学习算法通过主动去掉一些分支来降低过拟合的主要手段（预剪枝/后剪枝）
- 预剪枝(prepruning)：在决策树生成过程中，对每个结点在划分前进行估计，若当前结点划分不能带来决策树泛化性的提升，则停止划分并将当前结点标记为叶结点。

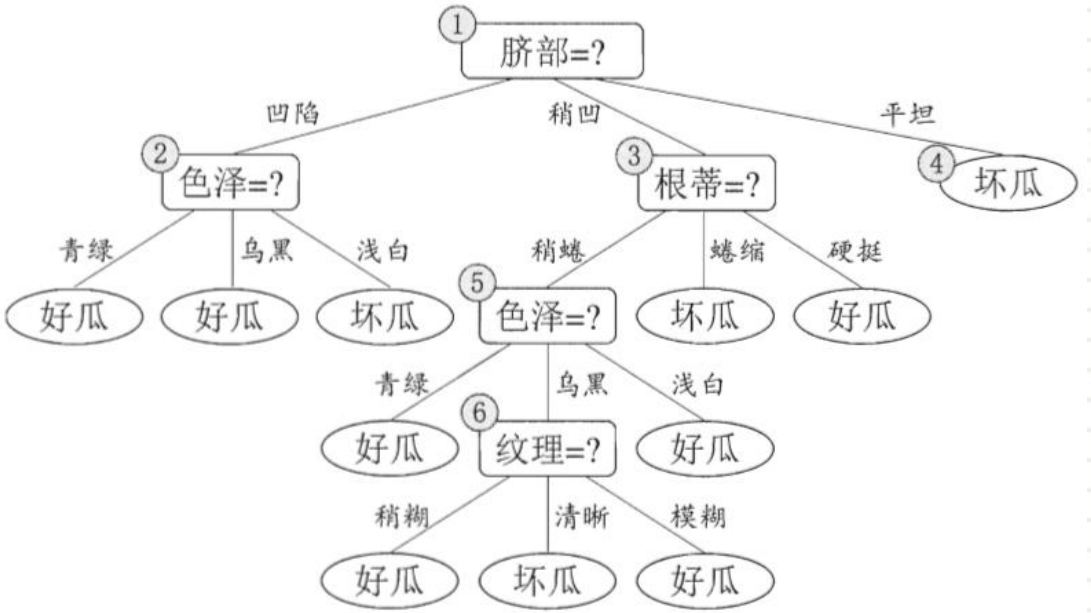


图4：未剪枝决策树

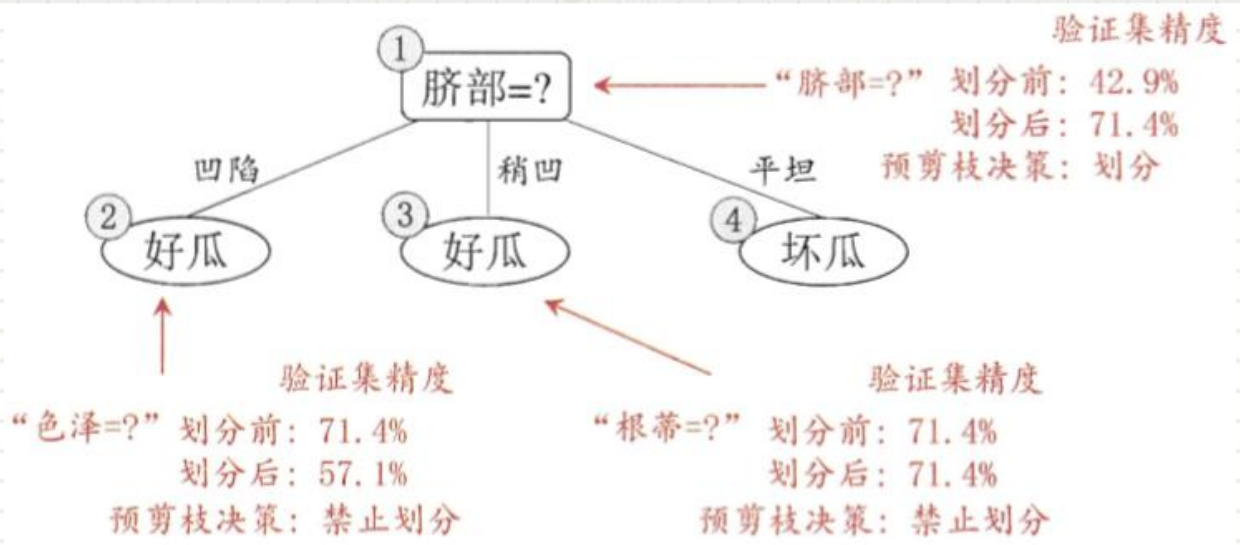


图5：预剪枝决策树

- 决策树桩(decision stump):仅有一层划分的决策树。

剪枝处理　后剪枝

- 后剪枝(postpruning)：先从训练集生成一棵完整的决策树，然后自底向上地对非叶结点进行考察，若将该结点对应的子树替换成叶结点能带来决策树泛化性能的提升，则将该子树替换为叶结点。

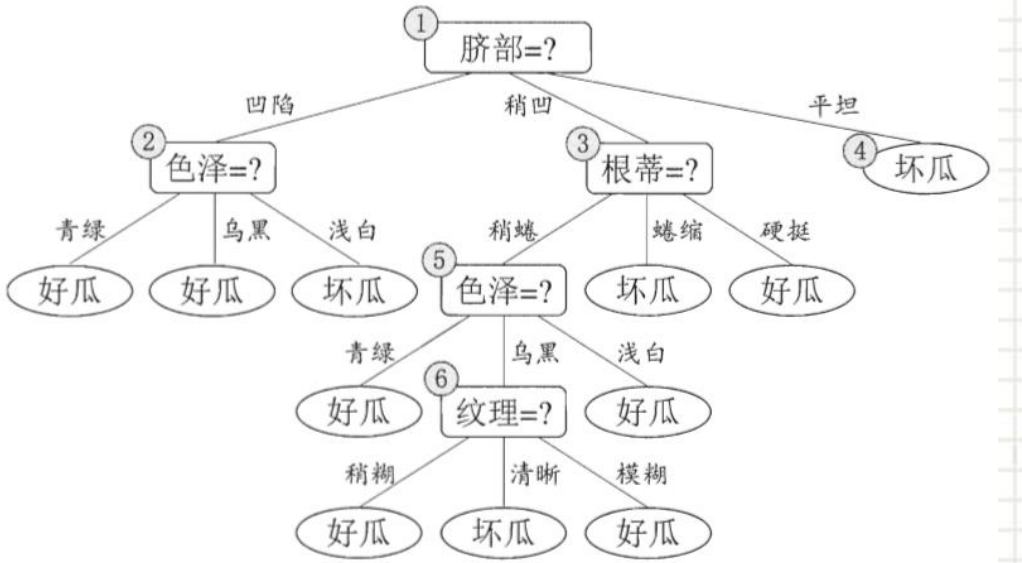


图6：未剪枝决策树

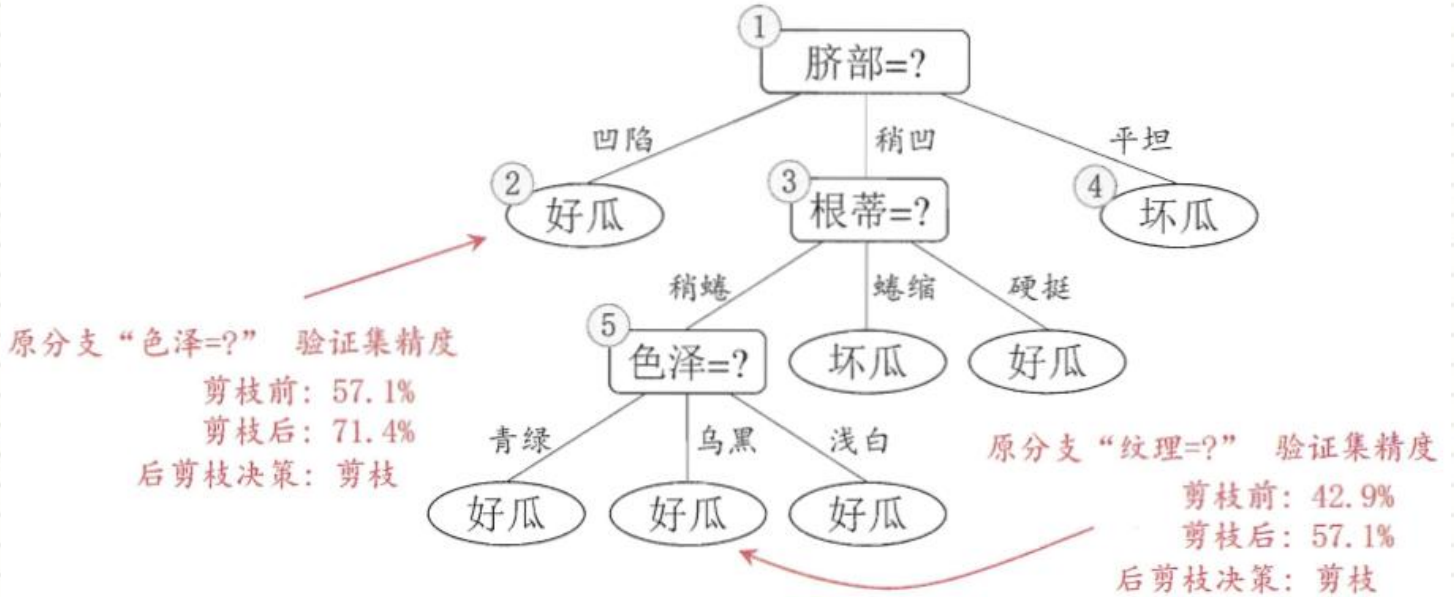


图7：后剪枝决策树

- 一般情况下，后剪枝决策树的泛化性能往往优先于预剪枝决策树，但训练时间开销要大很多。

连续与缺失值 连续值处理

- 采用二分法(bi-partition)对连续属性进行处理。
- 假定连续属性a在样本集D中出现n个不同的取值，从小到大排序，记为$\{a^1, a^2, \ldots, a^n\}$,则该属性的候选划分点集合为：

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leqslant i \leqslant n-1 \right\}$$

- 样本集D基于划分t二分后的信息增益：

$$\begin{aligned} \mathrm{Gain}(D,a) &= \max_{t \in T_a} \mathrm{Gain}(D,a,t) \\ &= \max_{t \in T_a} \mathrm{Ent}(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} \mathrm{Ent}(D_t^\lambda) , \end{aligned}$$

- 因此，我们选择使Gain(D,a，t)最大化的划分点。

连续与缺失值 缺失值处理

- **问题1：如何在属性值缺失的情况下进行划分属性选择？**
- 另D’表示D中属性a上没有缺失值的样本子集，根据D’来判断属性a的优劣。
- 则D’的信息熵：

$$\mathrm{Ent}(\tilde{D}) = -\sum_{k=1}^{|\mathcal{Y}|} \tilde{p}_k \log_2 \tilde{p}_k$$

- D的信息增益：

$$\begin{aligned}\mathrm{Gain}(D,a) &= \rho \times \mathrm{Gain}(\tilde{D},a) \\ &= \rho \times \left(\mathrm{Ent}\left(\tilde{D}\right) - \sum_{v=1}^{V} \tilde{r}_v \,\mathrm{Ent}\left(\tilde{D}^v\right)\right)\end{aligned}$$

无缺失值样本所占的比例

- **问题2：若样本在属性上的值缺失，如何对样本进行划分？**
- 让同一个样本以不同的概率（权重）划分到不同的子结点中去。

多变量决策树

- 决策树形成的分类边界由若干个与坐标轴平行的分段组成。（分类边界的每一段都与坐标轴平行）

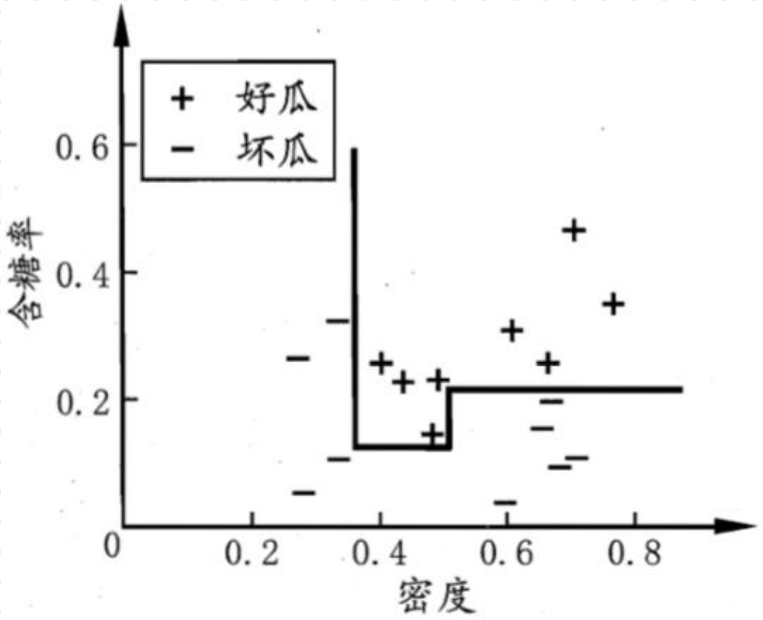


图8

- 当进行大量的属性测试时，“多变量决策树”能实现“斜划分”甚至更复杂划分的决策树。

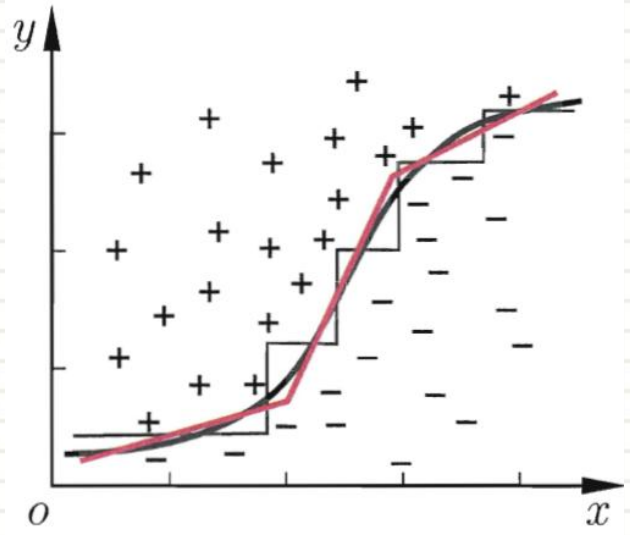


图9

- 在多变量决策树中，不是为每个非叶结点寻找一个最优的划分属性，而是建立一个合适的线性分类器

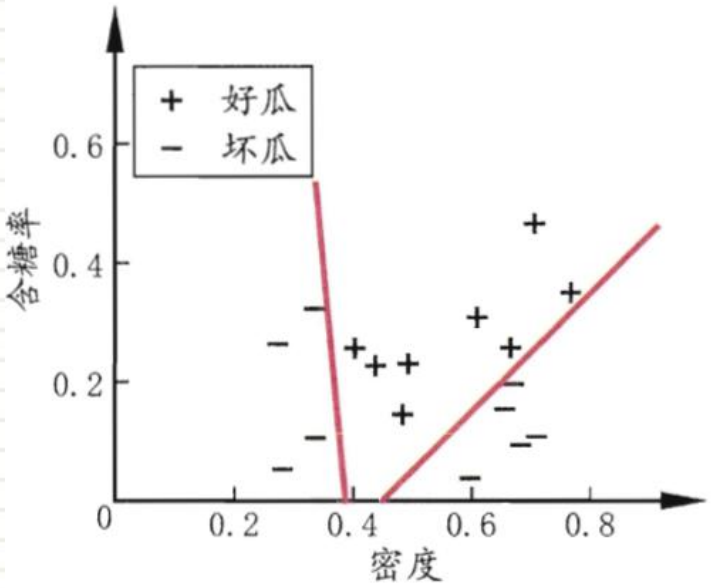
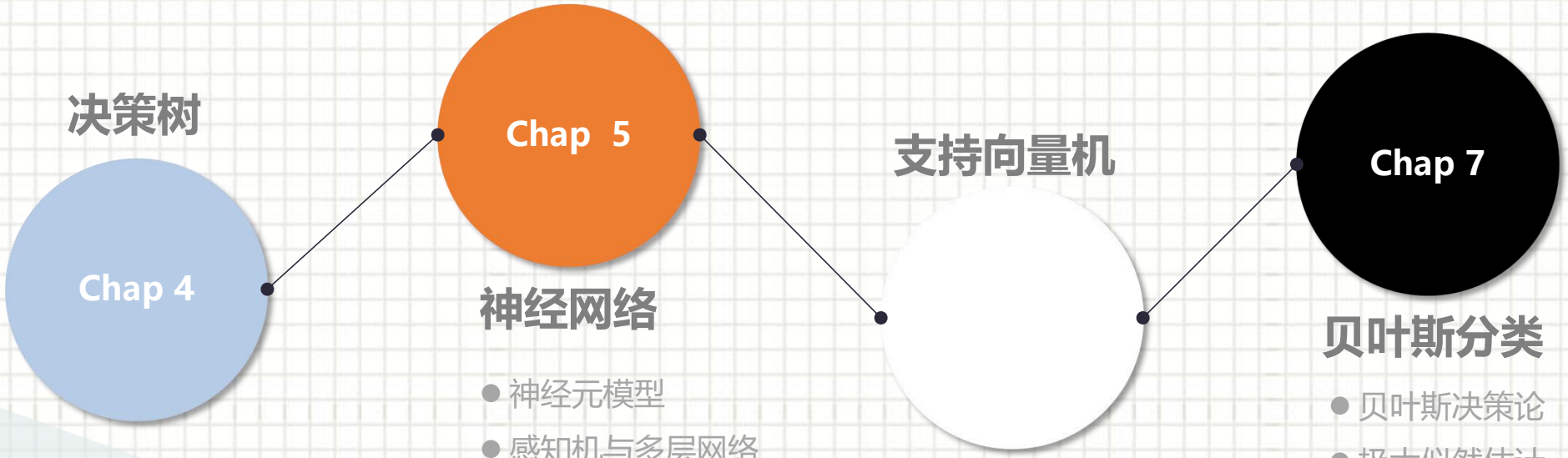


图10

机器学习 周志华 chap5、chap7知识点总结

目录 /CONTENTS

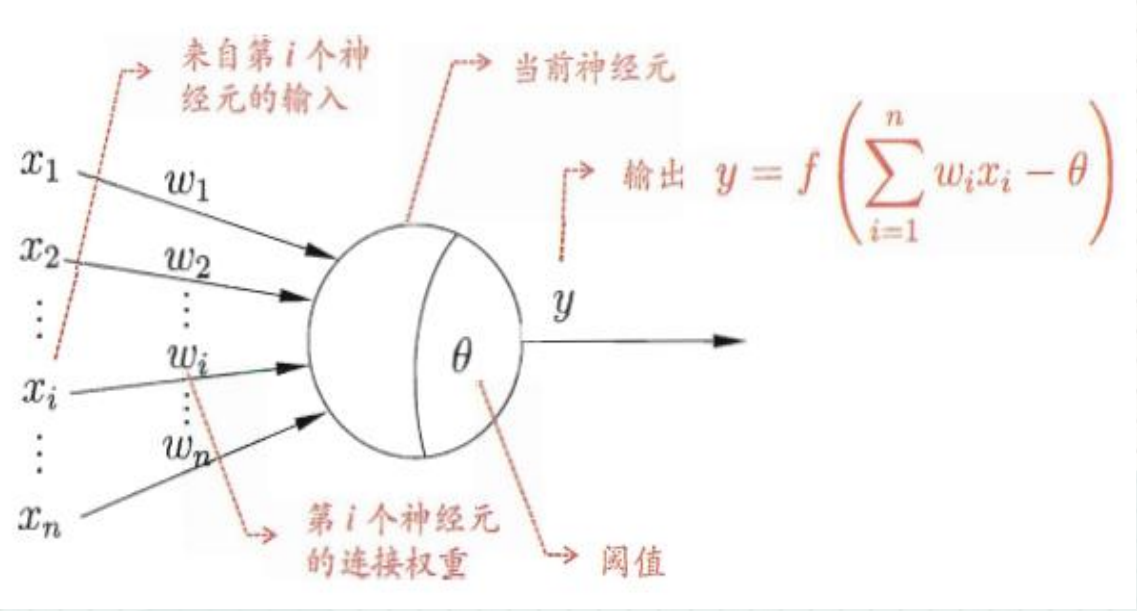


- 神经元模型
- 感知机与多层网络
- 误差逆传播算法
- 全局最小与局部最小
- 其他常见神经网络

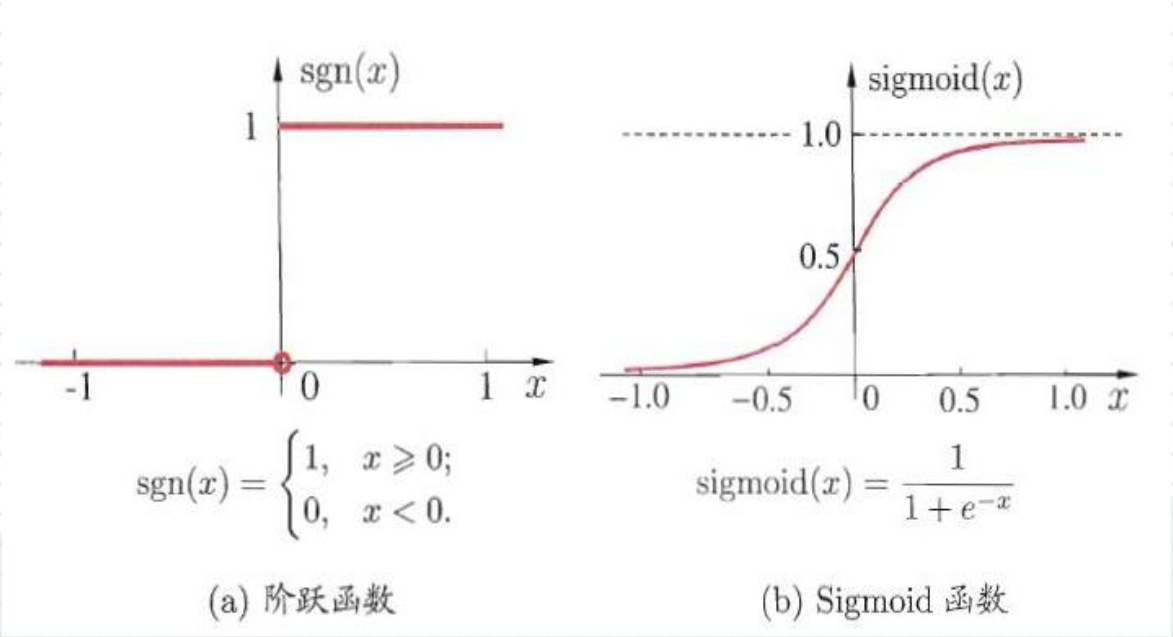
- 贝叶斯决策论
- 极大似然估计
- 朴素贝叶斯分类器
- 半朴素贝叶斯分类器
- 贝叶斯网
- EM算法

Chap 5 神经元模型

- 神经网络是由具有适应性的简单单元组成的广泛并行互连的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应.
- 神经网络中最基本的成分是神经元模型。
- M-P神经元模型中，神经元收到来自n个其他神经元传递过来的输入信号，这些输入信号通过带权重的连接进行传递，神经元接收到的总输入值将于神经元的阈值进行比较，然后通过“激活函数”处理以产生神经元的输出。



- 激活函数：理想激活函数是阶跃函数，0 表示抑制神经元，1表示激活神经元
  - 阶跃函数：不连续、不光滑；
  - Sigmoid函数：将输入挤压大（0,1）输出之间。
- 将单个神经元以一定的层次结构连接起来就得到神经网络。



- 感知机由两侧神经元组成：输入层接收外界输入信号后传递给输出层，输出层是M-P神经元。
- 感知机只有输出层神经元进行激活函数处理，即只拥有一层功能神经元，只能处理线性可分问题（与、或、非），不能处理非线性可分问题（异或）。
- 多层功能神经元可以解决非线性问题（异或问题），输出层与输入层之间的神经元称为隐层或隐含层，隐含层和输出层神经元都是拥有激活函数的功能神经元。

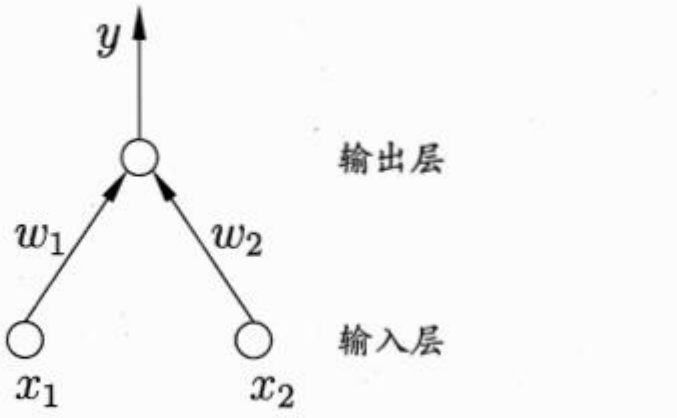
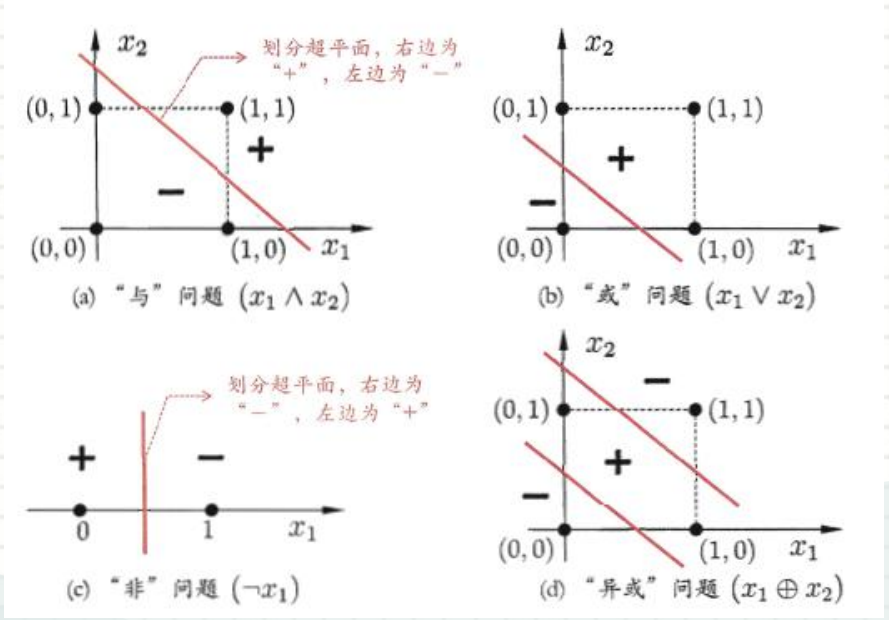


图 5.3 两个输入神经元的感知机网络结构示意图

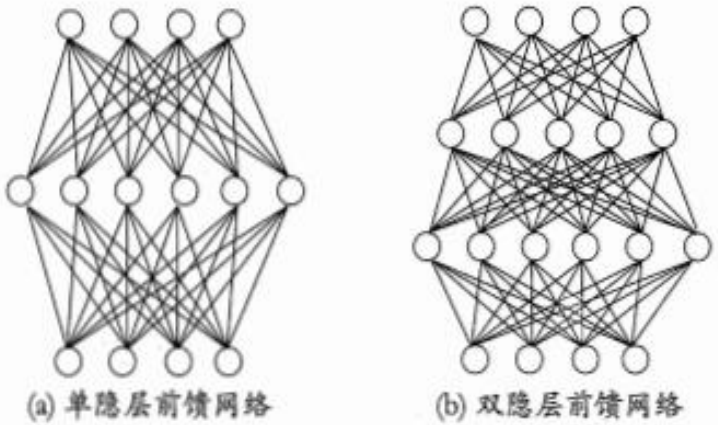


- 给定训练数据集，权重 $\omega_i (i=1,2,\ldots,n)$ 以及阈值θ 可通过学习得到。

$$y = f\left(\sum_i \omega_i x_i - \theta\right)$$

- 假定f 是阶跃函数，感知机的学习规则是：
- 对训练样例 (x,y)，若当前感知机的输出为 $\hat{y}$，则感知权重将这样调整：

$$\omega_i \leftarrow \omega_i + \Delta\omega_i$$
$$\Delta\omega_i = \eta\left(y - \hat{y}\right)x_i$$


(a) 单隐层前馈网络　(b) 双隐层前馈网络

- 其中$\eta \in (0,1)$ 称为学习率。
- 若感知机对训练样例 (x,y)预测正确，即 $\hat{y} = y$ ，则感知机不发生变化，否则将根据错误的程度进行权重调整。

- 多层前馈神经网络：
  - 多层：包含隐层；
  - 前馈：神经元之间不存在同层连接也不存在跨层连接。

多层前馈网络有强大的表示能力，设置隐层神经元数，通常使用“试错法”。

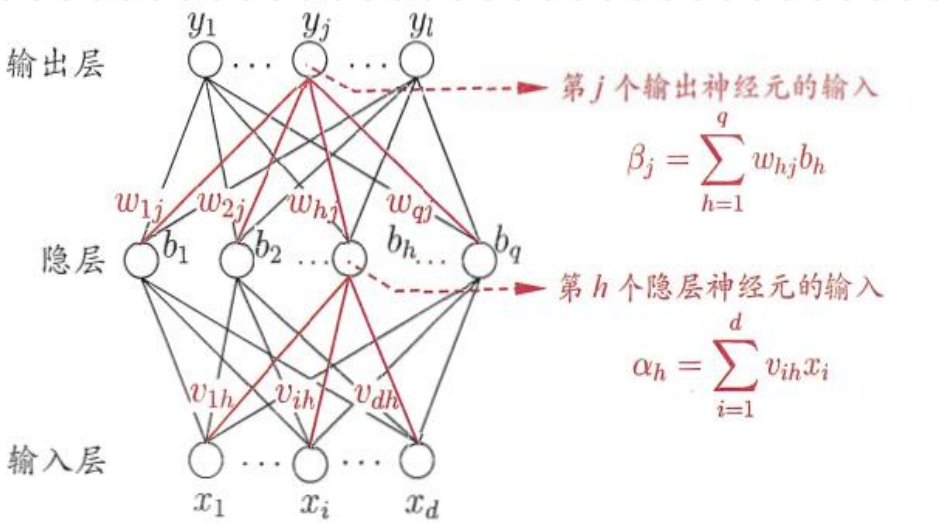
- BP神经网络：是迄今为止最成功的神经网络学习算法，不仅可用于多层前馈神经网络，还可用于其他类型的神经网络。
- 给定训练集 $D=\{(x_1,y_1),(x_2,y_2),\ldots(x_m,y_m)\},x_i\in\mathbb{R}^d,y_i\in R^l$

  输入：d维特征向量；

  输出：l个输出值；

  隐层：假定使用q个隐层神经元；

  假定功能单元均使用Sigmoid函数。
- BP算法基于梯度下降策略，以目标的负梯度方向对参数进行调整。
- BP算法工作流程：
  - 对每个训练样例，先将输入示例提供给输入层神经元；
  - 然后逐层将信号前传，直到输出层的结果；
  - 然后计算输出层的误差，再将误差逆向传播至隐层神经元，
  - 最后根据隐层神经元的误差来对连接权和阈值进行调整。



- 标准BP算法：每次仅针对一个训练样例更新连接权值
  - 缺点：参数更新频繁，不同样例更新效果可能抵消，需要多次迭代
- 累计BP算法：直接针对累计误差最小化
  - 优点：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。

在很多任务中，累计误差下降到一定程度后，进一步下降会非常缓慢，这是标准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。

- 缓解BP网络过拟合的两种策略：
  - 早停：将数据分成训练集和验证集，训练集用来计算梯度、更新连接权和阈值，验证集用来估计误差，若训练集误差降低但验证集误差升高，则停止训练，同时返回具有最小验证集误差的连接权和阈值。
  - 正则化：在误差目标函数中增加一个用于描述网络复杂度的部分，训练过程偏好比较小的连接权和阈值，使网络输出更加光滑，缓解过拟合

Chap 5 全局最小与局部最小

- 神经网络的训练过程可以看作一个参数寻优过程，即在参数空间中，寻找一组最优参数使得误差最小。
  - 局部极小点：参数空间内梯度为零的点，只要其误差函数值小于邻点的误差函数值，就是局部极小点
  - 全局最小点：参数空间中所有点的误差函数值均不小于该点的误差函数值

  可能存在多个局部极小值，但是全局最小值只有一个。

- 如何避免陷入“局部最小”：
  - 以多组不同参数值初始化多个神经网络，按标准方法训练后，取其中误差最小的解作为最终参数（相当于从多个不同的初始点开始搜索）。
  - “模拟退火”技术，在每一步都以一定概率接收比当前解更差的结果（次优解）。
  - 随机梯度下降，在计算梯度时加入随机因素。
  - 遗传算法

Chap 5 其他常见神经网络

常见神经网络	分类	主要思想	优点	缺点
RBF（径向基函数）网络	单隐层前馈神经网络	使用径向基函数作为隐层神经元激活函数，输出层是隐层神经元输出的线性组合	具有足够多隐层神经元的RBF网络能以任意精度逼近任意连续函数。	
ART（自适应谐振理论）网络	竞争学习型的无监督网络	网络由比较层、识别层、识别阈值和重置模块构成。	ART比较好的缓解了竞争性学习中的“可塑性-稳定性窘境”,可进行增量学习和在线学习。	识别阈值对ART网络的性能有重要影响。
SOM（自组织映射）网络	竞争学习型的无监督神经网络	将高维数据映射到低维空间，每个神经元拥有一个权向量，目标是为每个输出层神经元找到合适的权向量以保持拓扑结构。	将高维空间中相似的样本点映射到网络输出层中的邻近神经元。	
级联相关网络	结构自适应网络	开始训练时只有输入层和输出层，训练过程中加入隐层神经元。通过最大化新神经元的输出与网络误差之间的相关性来训练相关的参数。	无需设置网络层数、隐层神经元数目，训练速度较快	数据较小时容易陷入过拟合
Elman网络	递归神经网络	隐层神经元的输出被反馈回来，与下一刻输入层神经元的输出一起作为隐层神经元下一时刻的输入。		
标准的Boltzmann机	基于能力的模型	显层表示数据的输入与输出，隐层是数据的内在表达，神经元都是布尔型，只能取“0”抑制态或“1”激活态		训练网络复杂度太高
受限Boltzmann机		仅保留显层与隐层之间的连接，常用对比散度（CD）算法进行训练	降低网络复杂度	

- 典型的深度学习模型就是很深层的神经网络。
- 多隐层神经网络：
  - 缺点：难以直接用经典算法（如标准BP算法）进行训练，因为误差在多隐层内逆传播时，会“发散”不能收敛到稳定状态。
  - 多隐层网络训练方法：无监督逐层训练
    - 思想：预训练+微调

      预训练：每次训练一层隐结点时将上一层隐结点的输出作为输入，本层隐结点的输出作为下一层隐结点的输入；

      微调：预训练全部完成后，在对整个网络进行“微调”。

      应用：深度信念网络（DBN）

      优点：将大量参数分组，先对每组找到局部最优解，联合进行全局寻优，利用模型大量参数的同时有效的节省了训练开销。
    - 思想：权共享，让一组神经元使用相同的连接权。

      应用：卷积神经网络（CNN）

- 贝叶斯决策论是概率框架下实施决策的基本方法。基于后验概率可获得将样本x误分类所产生的期望损失，即在样本x上的“条件风险”。
- 我们的任务是寻找一个判定准则，以最小化总体风险。

- 贝叶斯判定准则：最小化总体风险，只需在每个样本上选择使条件风险最小的类别标记。此时的分类器称为“贝叶斯最优分类器” 与之对应的总体风险称为“贝叶斯风险” 。
  - 贝叶斯风险反映了分类器所能达到的最好性能，即通过机器学习所能产生的模型精度的理论上限。
  - 欲使用贝叶斯判定准则来最小化决策风险，首先要获得后验概率，往往难以直接获得。机器学习要实现的是基于有限训练样本集尽可能准确地估计出后验概率。
  - 主要有两种策略：“判别式模型” 、“生成式模型” 。

Chap 7 极大似然估计

- 极大似然估计是根据数据采样估计概率分布参数的经典方法。是试图在参数所有可能的取值中，找到一个能使数据出现在数据集中的“可能性”最大的值。

- 需要注意的是，这种参数化的方法虽然能使类条件概率估计变得相对简单，但估计结果的准确性严重依赖于所假设的概率分布形式是否符合潜在的真实数据分布。

Chap 7 朴素贝叶斯分类器

- 解决的问题：基于贝叶斯公式来估计后验概率的主要困难在于类条件概率是所有属性的联合概率，难以从有限的训练样本直接估计而得。

  “朴素贝叶斯分类器” 采用了“属性条件独立性假设”，对已知类别，假设所有属性相对独立。

- 朴素贝叶斯分类器的训练过程就是基于训练集D来估计类先验概率，并为每个属性估计条件概率。
- 拉普拉斯修正： 拉普拉斯修正避免了因训练样本不充分而导致概率估计值为零的问题，而且在训练集变大时，修正过程所引入的先验的影响也会逐渐变得可忽略，使得估值渐趋向于实际概率值。

- 实际应用：
  - 对预测速度要求较高：将涉及的所有概率估值事先计算好存储起来，预测时“查表”即可判别；
  - 任务数据更替频繁：采用“懒惰学习” 方法，先不进行任何训练，待接收到预测请求时再根据当前数据集进行概率估值；
  - 数据不断增加：在现有估值基础上，仅对新增样本的属性值所涉及的概率估值进行计数修正即可实现“增量学习”。

Chap 7 半朴素贝叶斯分类器

- 解决的问题：为降低贝叶斯公式中估计后验概率的困难，朴素贝叶斯分类器采用了属性条件独立性假设，但现实任务中此假设也很难成立，于是人们尝试对属性条件独立性假设进行一定程度的放松，由此产生了一类为“半朴素贝叶斯分类器” 的学习方法。

- 方法策略：采用“独依赖估计” 策略：假设每个属性在类别之外最多仅依赖于一个其他属性。

- 描述：贝叶斯网借助有向无环图来刻画属性之间的依赖关系，并使用条件概率来描述属性的联合概率分布。
- 结构：贝叶斯网结构有效地表达了属性间的条件独立性。给定父节点集，贝叶斯网假设每个属性与它的非后裔属性独立。
- 贝叶斯网中三个变量之间的典型依赖关系
  - 同父结构
  - V型结构
  - 顺序结构
- 学习：贝叶斯网学习的首要任务就是根据训练数据集来找出结构最“恰当”的贝叶斯网。
  - 评分搜索： 先定义一个评分函数，以此来评估贝叶斯网与训练数据的契合程度，然后基于这个评分函数来寻找结构最优的贝叶斯网。符合“最小描述长度” 准则。
  - 缺点：从所有可能的网络结构空间搜索最优贝叶斯网结构是一个NP难问题，难以快速求解。
  - 解决办法：求近似解
    - 贪心法；
    - 通过网络结构施加约束来削减搜索空间。
- 推断：通过抑制变量观测值来推测待查询变量的过程称为“推断”， 已知变量观测值称为“证据”。
  - 吉布斯采样：一种随机采样方法。

Chap 7 EM算法

- 隐变量：属性中存在“未观测”的变量，称为“隐变量”。
- EM算法常用于估计参数隐变量，是一种迭代式的方法：若模型参数已知，则可根据训练数据推断出最优隐变量的值；反之，若隐变量的值已知，则可方便地对模型参数做极大似然估计。

- 算法步骤：
  - 期望（E）步：利用当前估计的参数值来计算对数似然的期望值；
  - 第二步是最大化（M）步，寻找能使E步产生的似然期望最大化的参数值。然后得到的参数值重新被用于E步，直至收敛到局部最优解。

- 隐变量估计问题的其他解决办法：梯度下降等优化算法
  - 缺点：求和项数将随着隐变量的数目以指数级上升，计算复杂。