

智能制造过程大数据技术

Big Data Technology in Intelligent Manufacturing Process

第六讲：分类分析

Lecture 6: Classification analysis

丁敏 dingmin@cug.edu.cn



中国地质大学(武汉) 自动化学院

School of Automation, China University of Geosciences

- 分类分析的基本概念
- 决策树
- 贝叶斯分类
- 支持向量机
- 人工神经网络
- 分类模型的评价与选择
- 组合分类技术
- 实例



- 分类分析的基本概念
- 决策树
- 贝叶斯分类
- 支持向量机
- 人工神经网络
- 分类模型的评价与选择
- 组合分类技术
- 实例



1 分类分析的基本概念

4

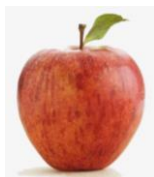
如何表现数据？

分类问题？

数据

特征

标签



红色，圆的，叶子，50g，...

苹果



绿色，圆的，没有叶子，60g，...

苹果



黄色，弯的，没有叶子，60g，...

香蕉



绿色，弯的，没有叶子，75g，...

香蕉

学习



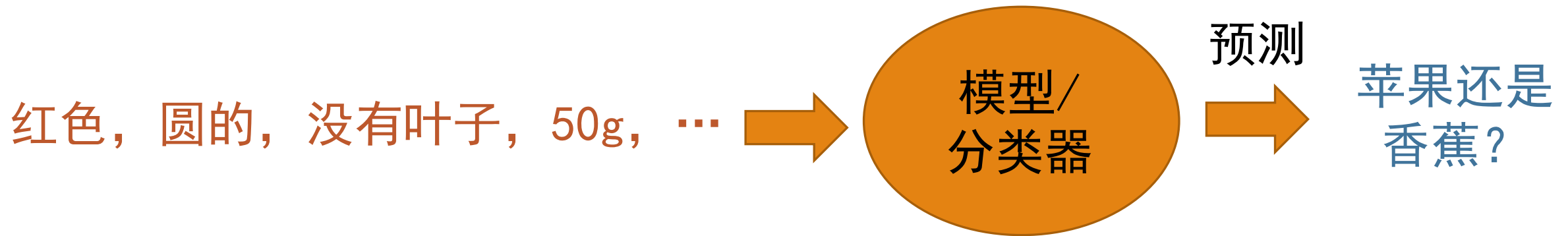
模型/
分类器

在学习/训练/归纳的时候，基于特征拟合一个可以区分苹果和香蕉的模型

1 分类分析的基本概念

5

分类问题？



模型可以根据特征对新的数据分类

1 分类分析的基本概念

6

训练集

测试集

特征

标签

红色，圆的，叶子，50g，...

苹果

绿色，圆的，没有叶子，60g，...

苹果

黄色，弯的，没有叶子，60g，...

香蕉

绿色，弯的，没有叶子，75g，...

香蕉

红色，圆的，没有叶子，50g，...

?

学习是从训练数据中概括

对训练和测试集有什么假设呢？

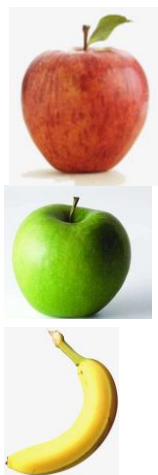
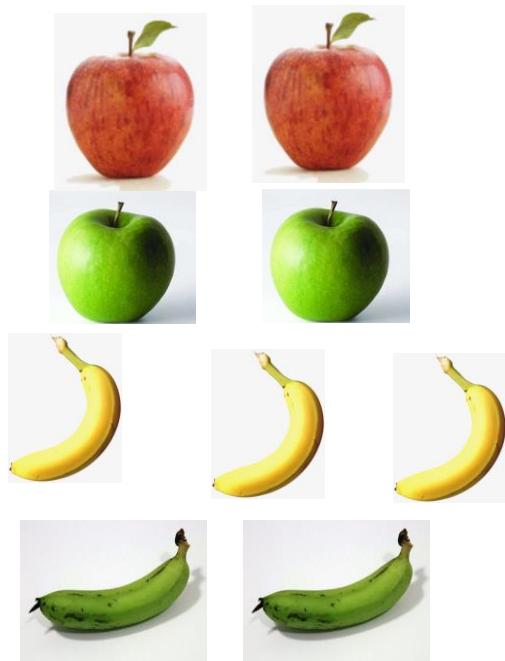
1 分类分析的基本概念

7

学习是用过去预测未来

训练集

测试集



- 从技术上讲，是用**概率模型**学习
- 数据/标签对服从某种概率分布，称为**数据生成分布**
- 训练集和测试集**都产生于这一分布**

➤ 分类的基本原理

- ❑ 定义：对样本数据进行观测与分析，选择合适的分类方法构造分类模型，使用分类模型预测未知样本数据的类标签
- ❑ 给定数据库 $D = \{t_1, t_2, \dots, t_n\}$ ，元组 $t_i \in D$ ，类的集合 $C = \{C_1, \dots, C_m\}$ ，分类问题定义为从数据库到类集合的映射 $f: D \rightarrow C$ ，即数据库中的元组 t_i 分配到某个类 C_j 中，有 $C_j = \{t_i \mid f(t_i) = C_j, 1 \leq i \leq n, \text{ 且 } t_i \in D\}$
- ❑ 分类分析应用：仪器仪表的故障类型、医疗数据对应的治疗方案、基于运营商数据的个人征信、欺诈信息预测等

➤ 分类分析概念

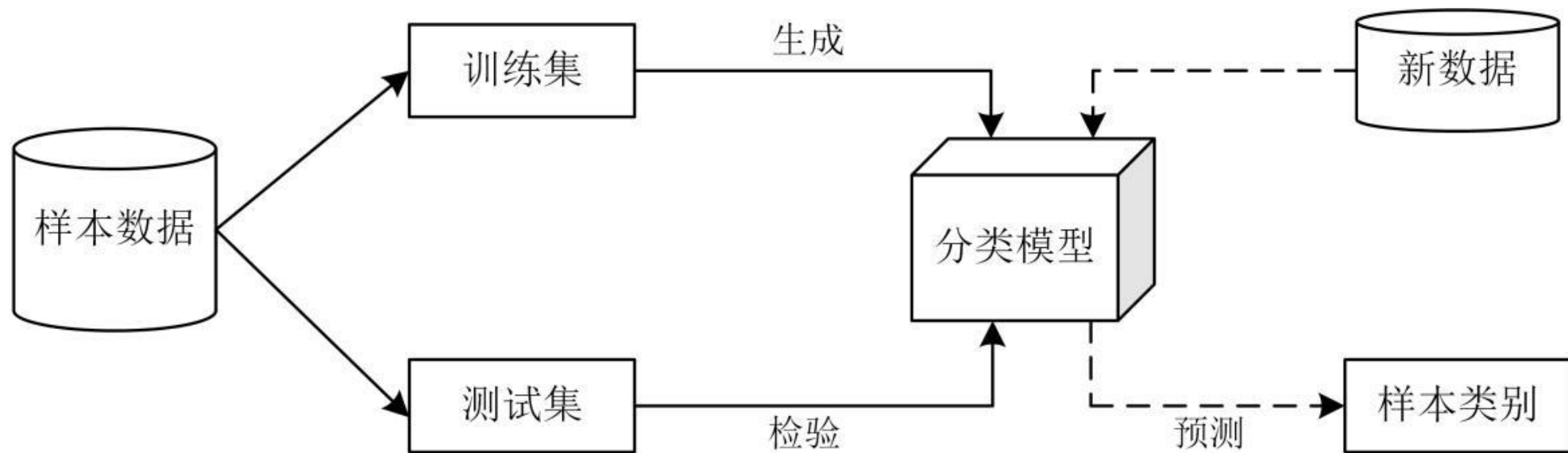
- 分类的三个阶段：模型构造、模型测试和模型应用
- 模型构造：建立描述预先定义的数据类或概念集的分类模型
 - 用来建模的样本（数据、对象、案例或记录）集合——训练集
 - 通过分析或从训练集学习来构造分类模型
 - 模型可用数学公式、决策树等表示
- 模型测试：对上述分类模型的预测准确率进行评估
 - 测试集（Test Set）参与模型准确率的评估
 - 用预测的类标签与真实的类标签比较，相同则结果正确
 - 计算分类模型在给定检验集上的准确率
- 模型应用：在模型测试之后，使用分类模型对未知样本进行分类

1 分类分析的基本概念

10

➤ 分类的基本原理

□ 分类分析的一般过程



➤ 主要分类方法

□ 决策树 (Decision Tree)

- 一种类似于流程图的树结构，它采用自顶而下的递归方式，经过一批训练集的训练生成一棵决策树

□ 朴素贝叶斯 (Naïve Bayes)

- 一种统计学分类方法，它可以预测某个给定样本属于一个特定类的概率

□ 支持向量机 (Support Vector Machine, SVM)

- 一种经典的机器学习方法

□ 人工神经网络 (Artificial Neural Networks, ANN)

- 一种类似于大脑神经突触连接的结构，训练重点是构造逻辑单元并反复调整权重系数

- 分类分析的基本概念
- **决策树**
- 贝叶斯分类
- 支持向量机
- 人工神经网络
- 分类模型的评价与选择
- 组合分类技术
- 实例

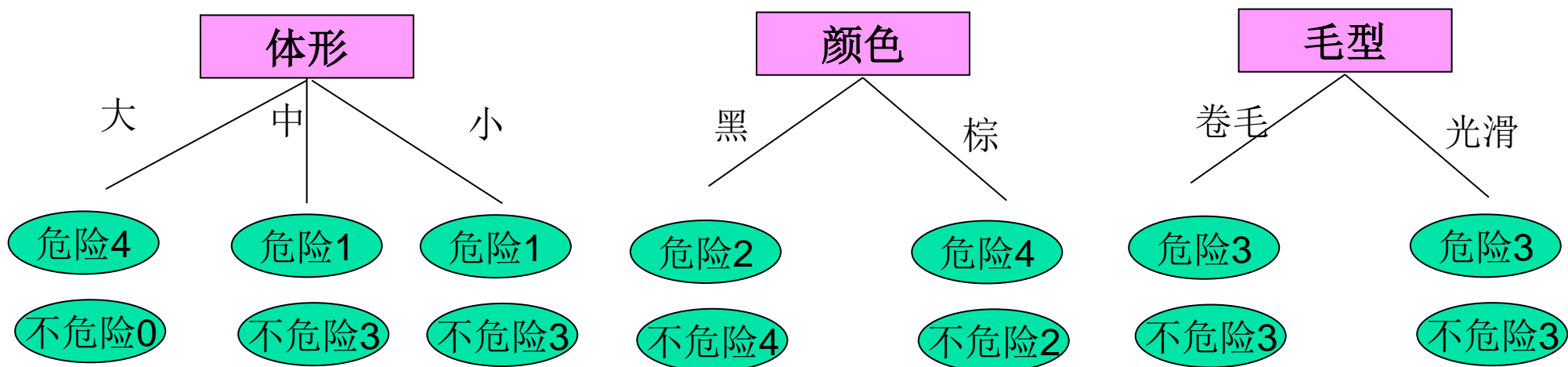


➤ 决策树基本原理

□ 决策树

- 决策树是一种描述对实例进行分类的树形结构，包含三类节点：**根节点**、**内部节点**和**叶节点**
 - **根节点**：包含所有样本，位于顶层
 - **内部节点**：按某种分类规则划分出的各类样本子集，位于中间
 - **叶节点**：分类结果，位于底层
- 采用**自顶而下**的递归方式：训练过程中，由根节点向下划分生成内部节点，每个内部节点向下继续划分
- 决策树的**根节点到某个叶节点的路径对应着一条分类规则**，整个决策树对应着一组规则。

➤ 数据的划分

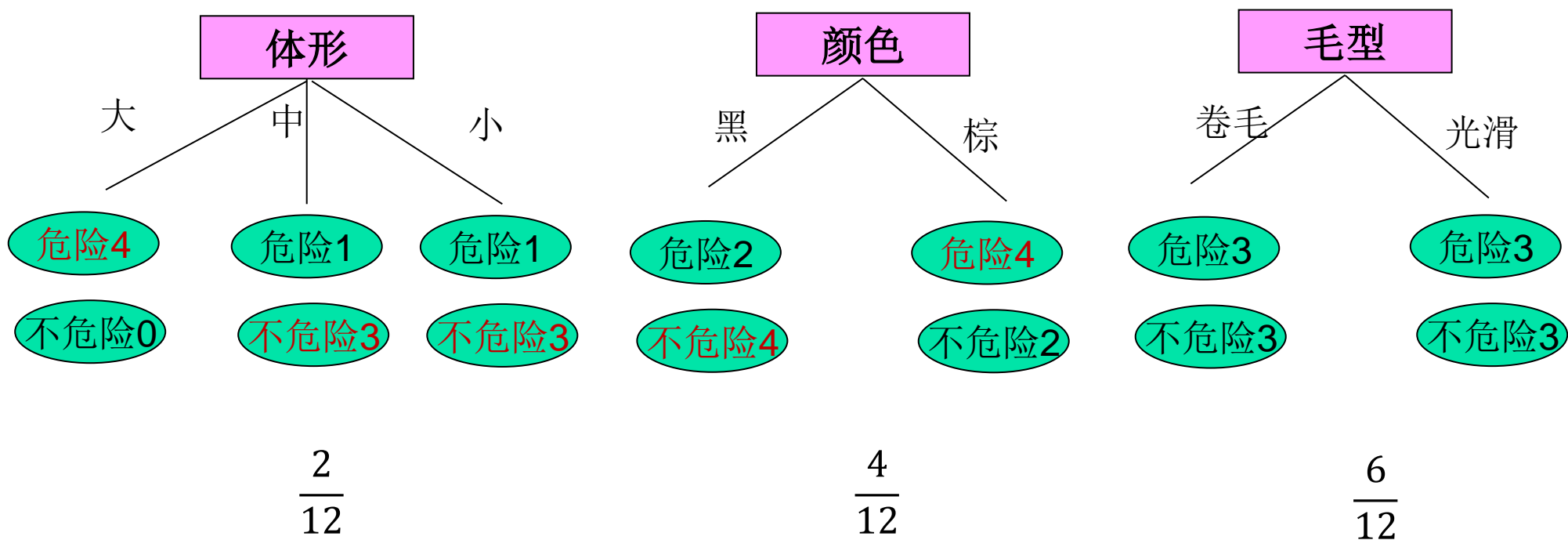


思考：计算每一个用来分割数据的特征的“得分”。

我们应该使用什么分数？

先选择哪棵树最好呢？

➤ 数据的划分

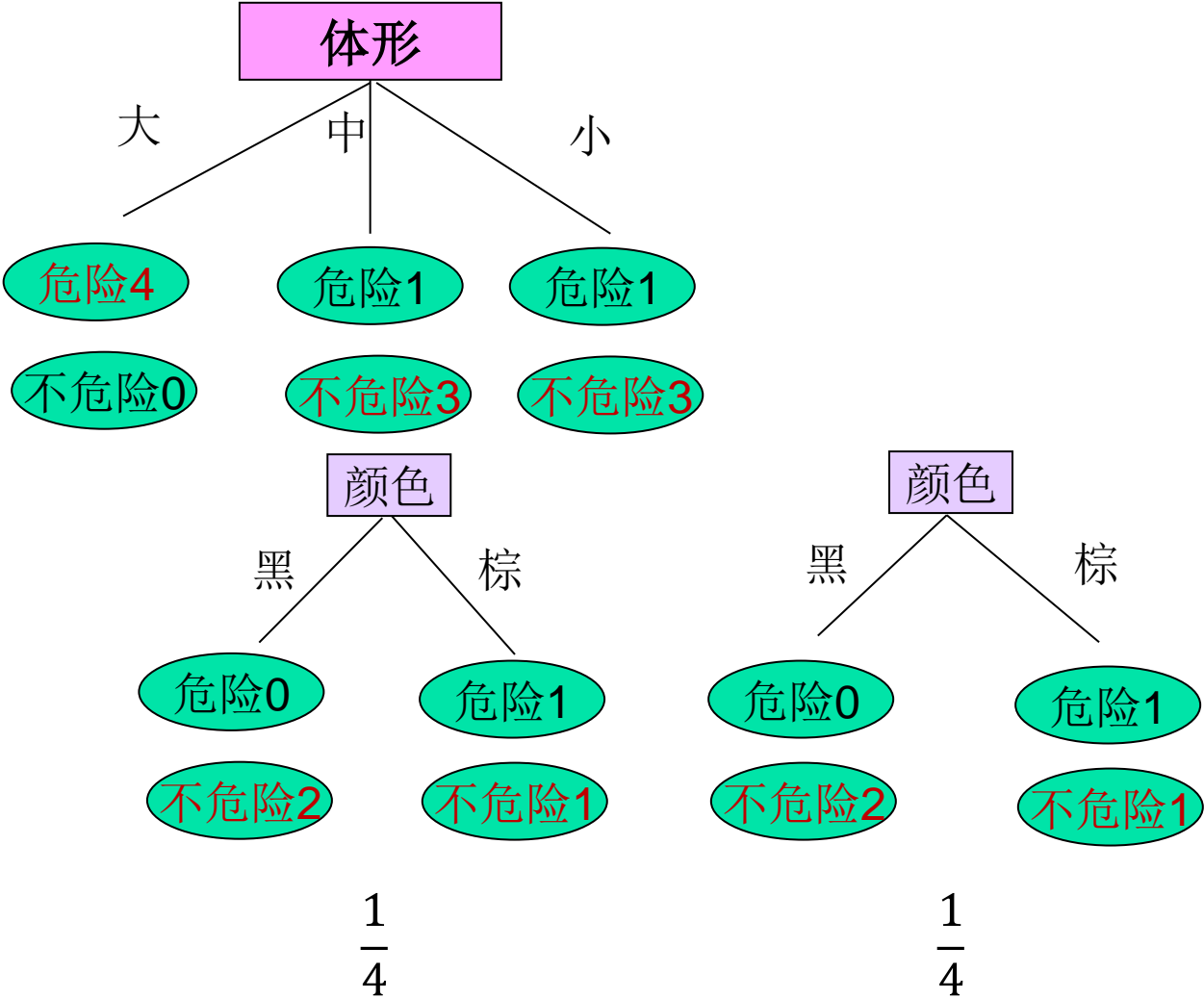


训练误差：训练集上的平均误差率

➤ 决策树过程

▣ 最终生成的决策树

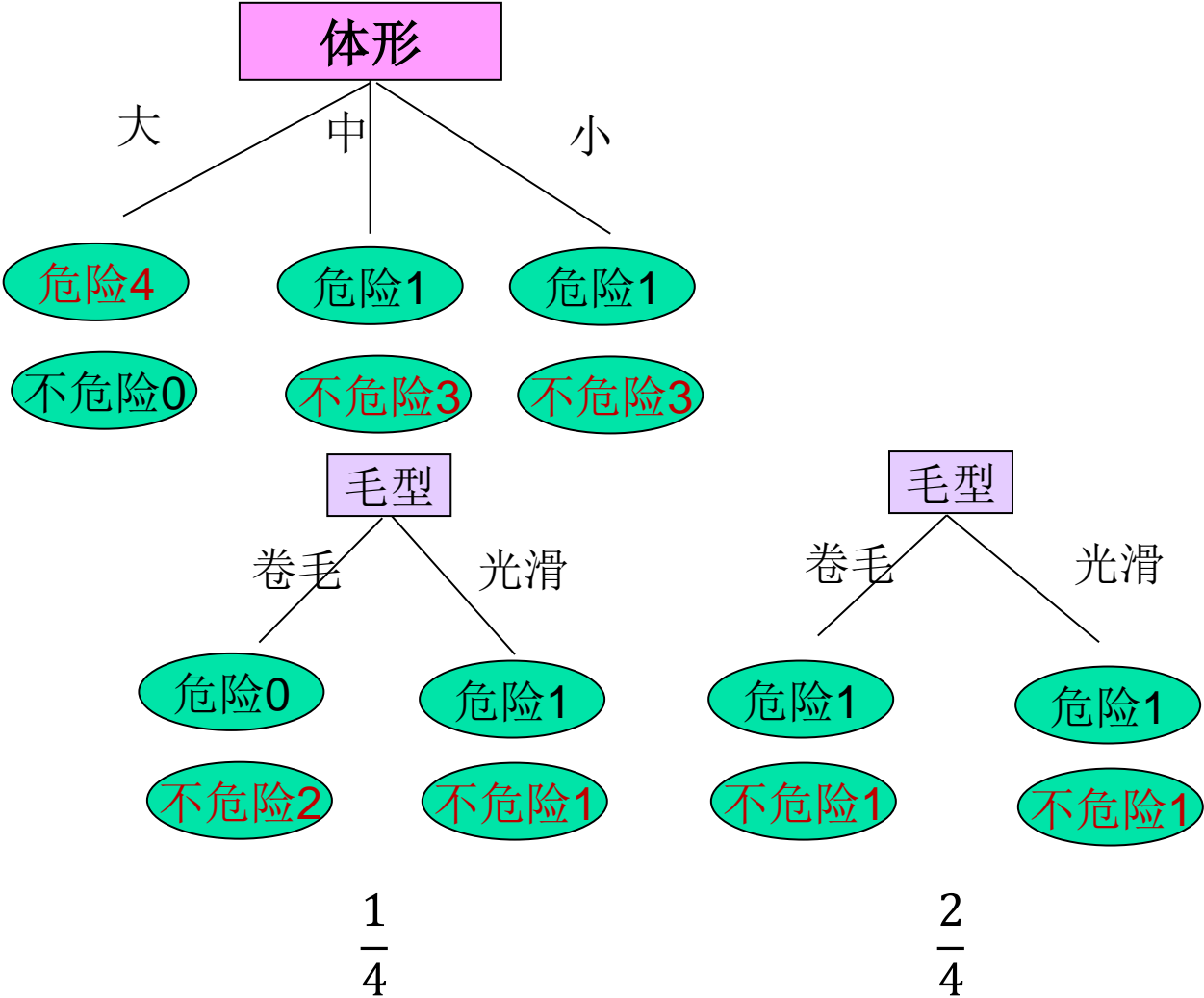
实例序号	颜色	体形	毛型	类别
1	黑	大	卷毛	危险
2	棕	大	光滑	危险
3	棕	中	卷毛	不危险
4	黑	小	卷毛	不危险
5	棕	中	光滑	危险
6	黑	大	光滑	危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
9	棕	大	卷毛	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险
12	黑	小	光滑	不危险



➤ 决策树过程

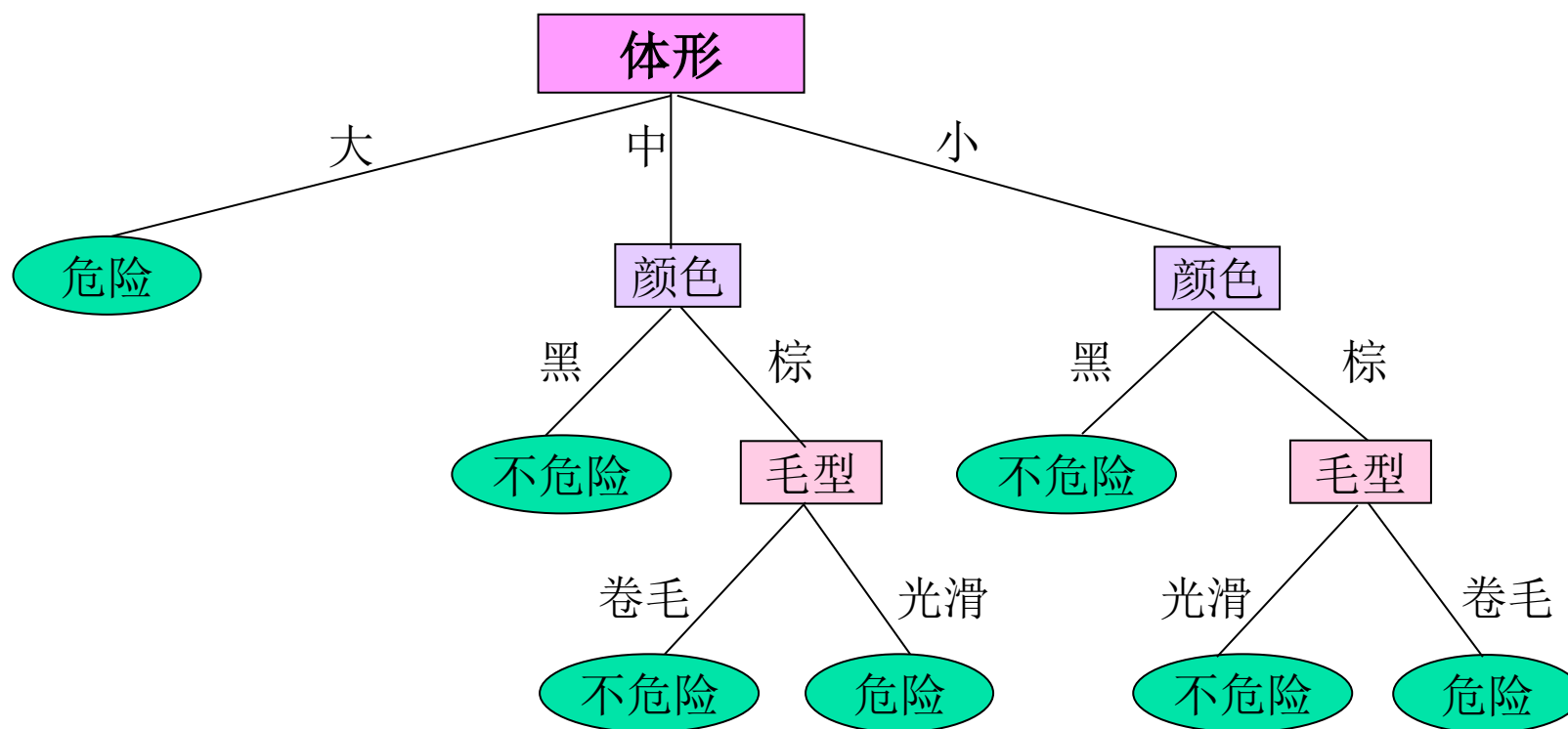
▣ 最终生成的决策树

实例序号	颜色	体形	毛型	类别
1	黑	大	卷毛	危险
2	棕	大	光滑	危险
3	棕	中	卷毛	不危险
4	黑	小	卷毛	不危险
5	棕	中	光滑	危险
6	黑	大	光滑	危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
9	棕	大	卷毛	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险
12	黑	小	光滑	不危险



➤ 决策树过程

□ 最终生成的决策树



一定能保证训练误差为0吗？

➤ 决策树过程

实例序号	颜色	体形	毛型	类别
1	黑	大	卷毛	危险
2	棕	大	光滑	不危险
3	棕	中	卷毛	不危险
4	黑	小	卷毛	不危险
5	棕	中	光滑	不危险
6	黑	大	光滑	危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
9	棕	大	卷毛	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险
12	黑	小	光滑	不危险

□ 数据中错误或有噪音

□ 现有数据不足以区分或决策，
还需要一个特征

什么时候会出现问题数据？

➤ 决策树过程

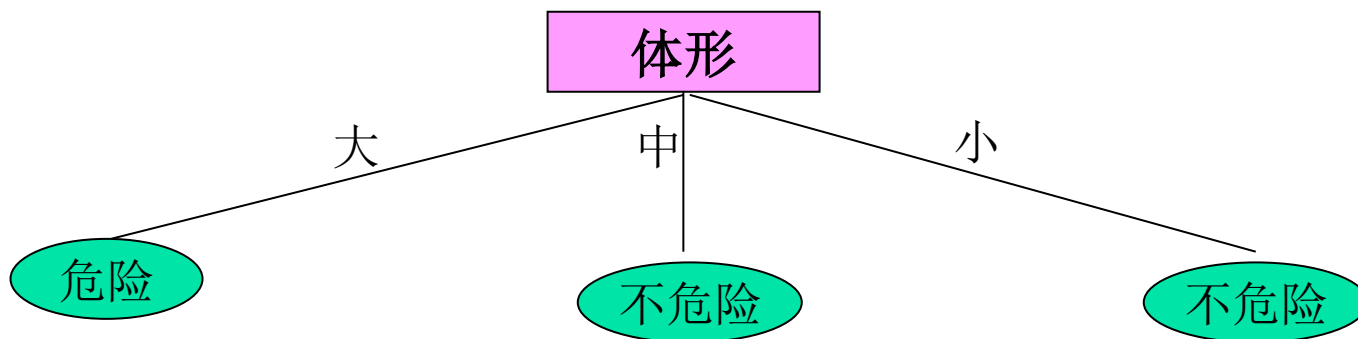
实例序号	颜色	体形	毛型	类别
1	黑	大	卷毛	危险
2	棕	大	光滑	不危险
3	棕	中	卷毛	不危险
4	黑	小	卷毛	不危险
5	棕	中	光滑	不危险
6	黑	大	光滑	危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
9	棕	大	卷毛	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险
12	黑	小	光滑	不危险

□ 如果所有数据属于同一个类，使用该标签创建叶节点或者所有数据有相同的特征值

我们一定要把数据划分到底吗？

➤ 决策树过程

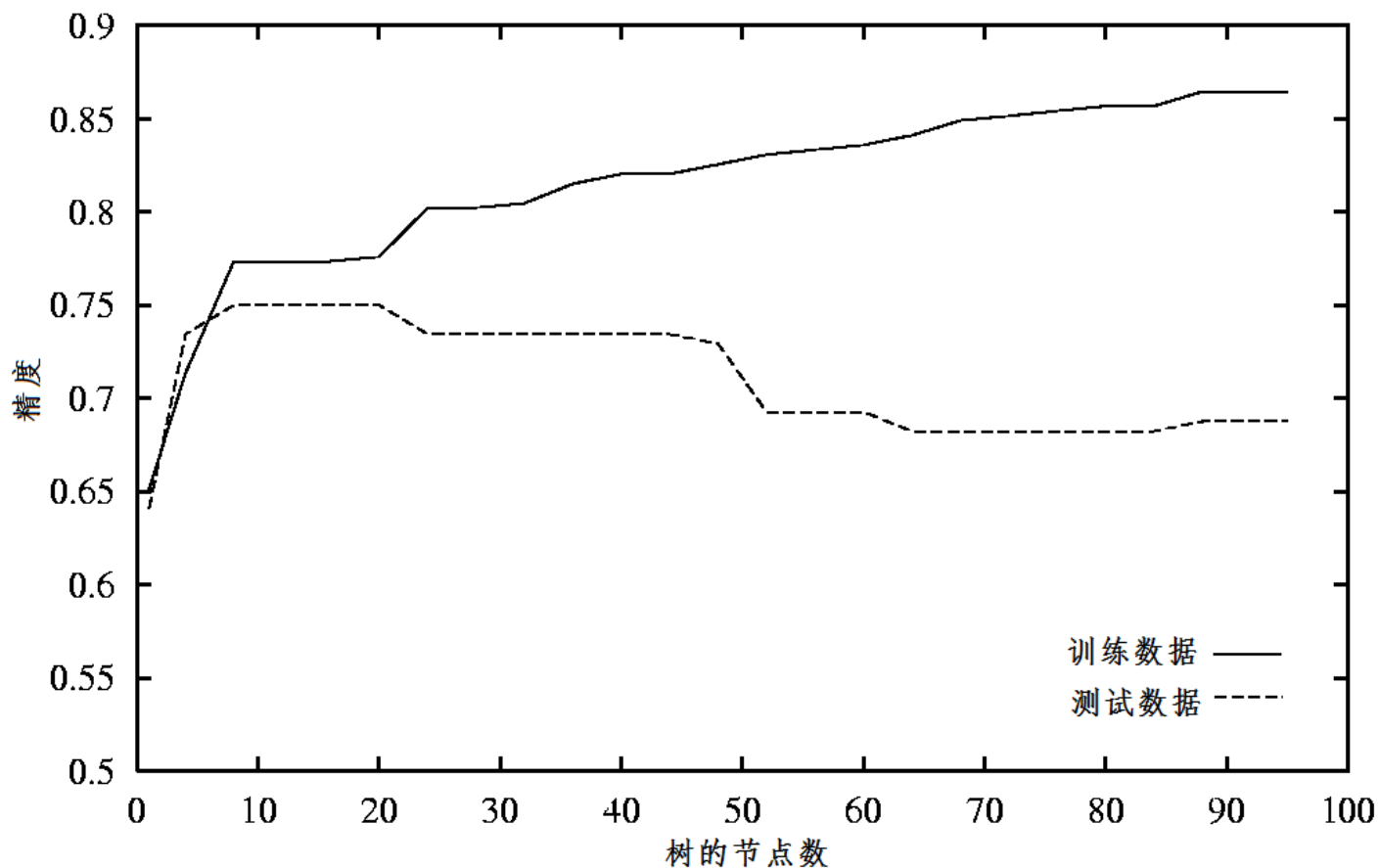
□ 最终生成的决策树



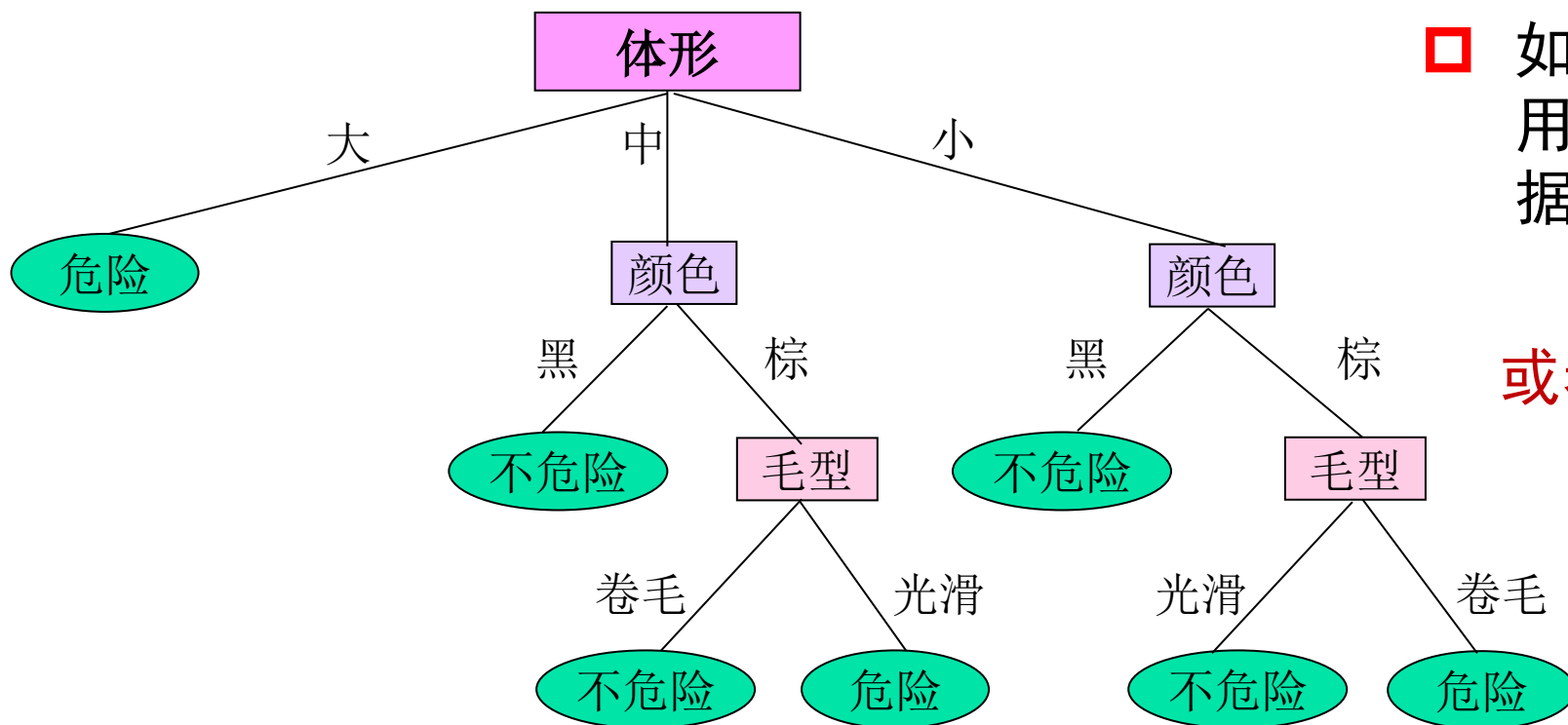
我们有时候可能不需要这么做

- **过拟合**发生在模型太过于偏向训练数据时
- 我们的目标是学习一个一般的模型，既要符合训练数据也要符合其他数据（如测试数据）

- 为将所有训练样本分类，节点不断分裂，分支越来越多，这种过程很可能过分适应了训练数据集中的“噪声”，这种现象称为“**过拟合**”



➤ 决策树过程

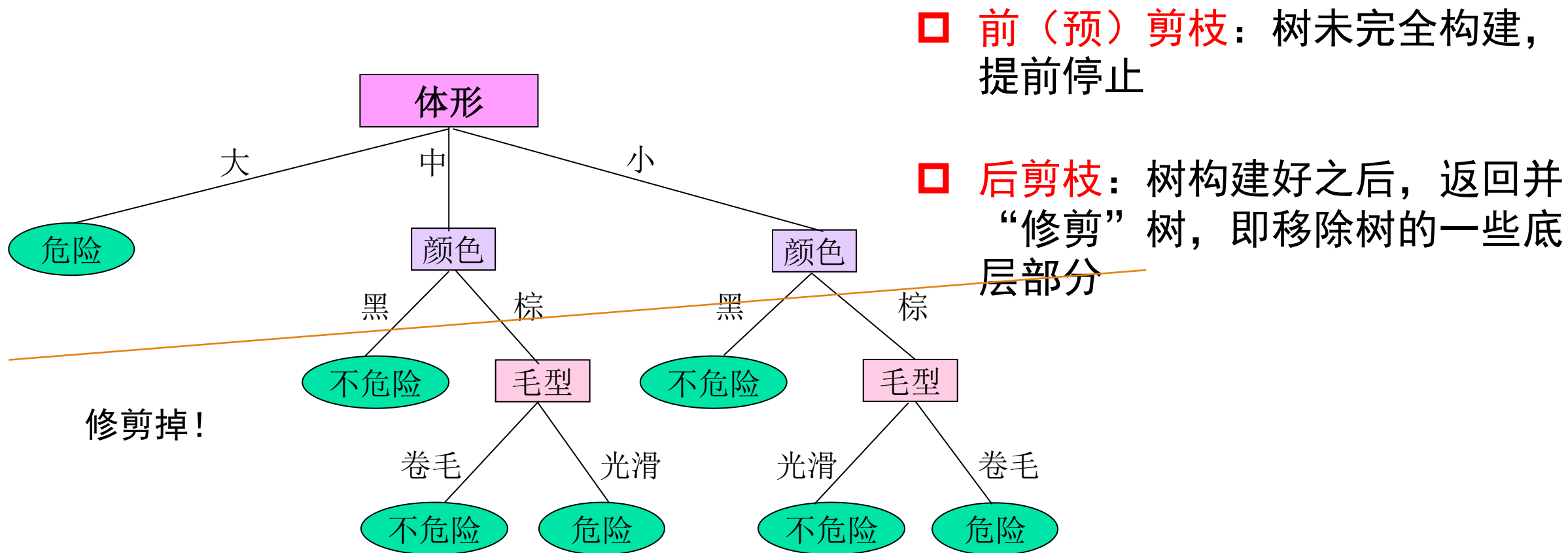


□ 如果所有数据属于同一个类，使用该标签创建叶节点或者所有数据有相同的特征值

或者我们已经达到数的一定深度？
一个思路：提前结束树的生长；
达到一个足够小的训练误差

.....

➤ 阻止过拟合



➤ 树剪枝

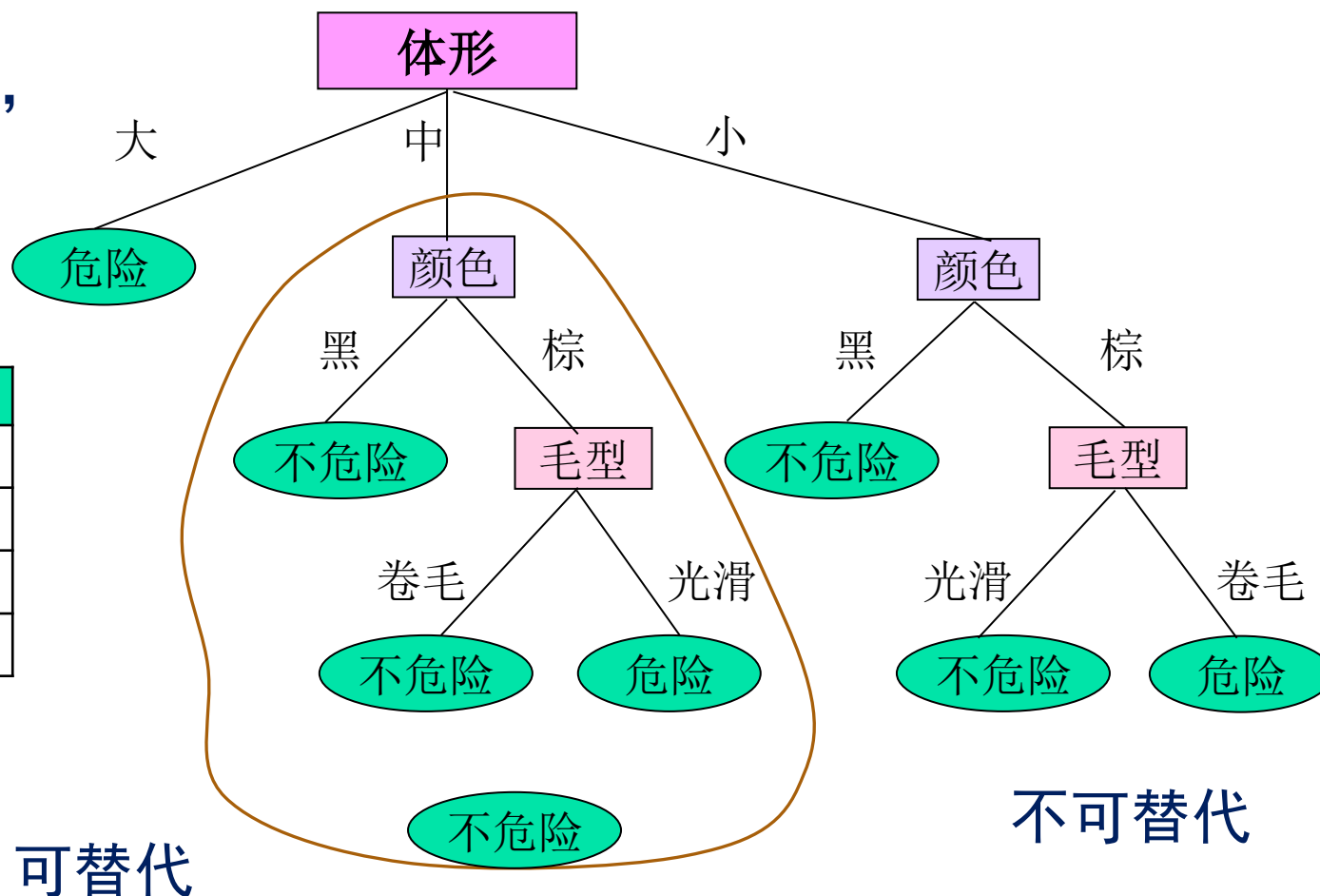
- 前（预）剪枝：提前设定一个指标或阈值，当达到预定的条件就停止划分
 - 参数控制法：利用某些参数（例如：节点的大小、树的深度等）限制树的生长
 - 分裂阈值法：设定一个分裂阈值，当分裂后的信息增益不小于该增益，才保留分支，否则停止分裂
- 不足：阈值不好设置，过大决策树过于简单；过小，有多余树枝，过于茂盛

➤ 树剪枝

□ 后剪枝：构造完整的决策树，允许过度拟合数据，然后在树的主体中删除不必要的子树

- 如果删除某个节点的子节点后，决策树的准确率 并没降低，那么就将该节点变成叶节点

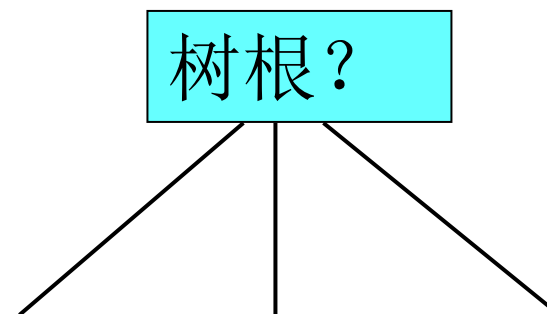
实例序号	颜色	体形	毛型	类别
13	黑	大	卷毛	危险
14	棕	中	光滑	不危险
15	棕	中	光滑	不危险
16	棕	小	卷毛	危险



➤ 决策树建立的关键

□ 建立一个好的决策树的关键是**决定树根和子树根的属性**

实例序号	颜色	体形	毛型	类别
1	黑	大	卷毛	危险
2	棕	大	光滑	危险
3	棕	中	卷毛	不危险
4	黑	小	卷毛	不危险
5	棕	中	光滑	危险
6	黑	大	光滑	危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
9	棕	大	卷毛	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险
12	黑	小	光滑	不危险



➤ 决策树基本原理

□ CLS算法

- (1) 由训练数据集生成根节点
- (2) 为当前节点选择某一分类属性作为划分依据
- (3) 根据当前节点属性不同的取值，将训练集划分为若干子集，每个取值形成一个分支。针对当前划分的若干个子集，重复步骤（2）~（3）
- (4) 达成下列条件之一时，停止划分，将该节点标记为叶节点：
 - 节点的所有样本属于同一类
 - 没有剩余属性
 - 如果某一分支没有样本，则以该节点中占大多数的样本类别创建一个叶节点
 - 决策树深度已达到设定的最大值

➤ 决策树实例

□ 试用CLS算法对电池故障实验数据构造决策树

序号	属性A	属性B	属性C	是否异常
1	> 0.06	> 0.77	> 0.18	是
2	> 0.06	0.65~0.77	> 0.18	是
3	< 0.04	< 0.65	< 0.12	否
4	0.04~0.06	> 0.77	0.12~0.18	是
5	0.04~0.06	0.65~0.77	< 0.12	否
6	0.04~0.06	< 0.65	< 0.12	否
7	< 0.04	0.65~0.77	> 0.18	是
8	> 0.06	0.65~0.77	0.12~0.18	是
9	< 0.04	< 0.65	0.12~0.18	否
10	0.04~0.06	> 0.77	> 0.18	是
11	< 0.04	0.65~0.77	< 0.12	否
12	0.04~0.06	0.65~0.77	0.12~0.18	是
13	0.04~0.06	0.65~0.77	> 0.18	是
14	> 0.06	< 0.65	0.12~0.18	否
15	< 0.04	> 0.77	0.12~0.18	是

- 属性 $A = \{a_1, a_2, a_3\}$
 $a_1(> 0.06)$ 、 $a_2(0.04 \sim 0.06)$ 、 $a_3(< 0.04)$
- 属性 $B = \{b_1, b_2, b_3\}$
 $b_1(> 0.77)$ 、 $b_2(0.65 \sim 0.77)$ 、 $b_3(< 0.65)$
- 属性 $C = \{c_1, c_2, c_3\}$
 $c_1(> 0.18)$ 、 $c_2(0.12 \sim 0.18)$ 、 $c_3(< 0.12)$

➤ 决策树实例

- 步骤1：生成根节点，位于顶端，在属性集中选择某一属性作为决根节点的划分依据。按属性B的不同取值可将训练集D划分为三个不同子集 D_{b_1} D_{b_2} D_{b_3}
- 其中子集 D_{b_1} 中只包含一种类别的样本，子集 D_{b_3} 同理，这两个子集不再向下继续划分。
 - 子集 D_{b_2} 表示如下：

序号	属性A	属性B	属性C	是否异常
1	> 0.06	0.65~0.77	> 0.18	是
2	0.04~0.06	0.65~0.77	< 0.12	否
3	< 0.04	0.65~0.77	> 0.18	是
4	> 0.06	0.65~0.77	0.12~0.18	是
5	< 0.04	0.65~0.77	< 0.12	否
6	0.04~0.06	0.65~0.77	0.12~0.18	是
7	0.04~0.06	0.65~0.77	> 0.18	是

➤ 决策树实例

- 步骤2：子集 D_{b2} 仍包含两种类别的样本，故应继续选择其它属性对子集 D_{b2} 进行划分。可选择属性A将其划分为三个不同子集 V_{a1} V_{a2} V_{a3}

序号	属性A	属性B	属性C	是否异常
1	> 0.06	0.65~0.77	> 0.18	是
2	> 0.06	0.65~0.77	0.12~0.18	是

序号	属性A	属性B	属性C	是否异常
1	0.04~0.06	0.65~0.77	< 0.12	否
2	0.04~0.06	0.65~0.77	0.12~0.18	是
3	0.04~0.06	0.65~0.77	> 0.18	是

序号	属性A	属性B	属性C	是否异常
1	< 0.04	0.65~0.77	> 0.18	是
2	< 0.04	0.65~0.77	< 0.12	否

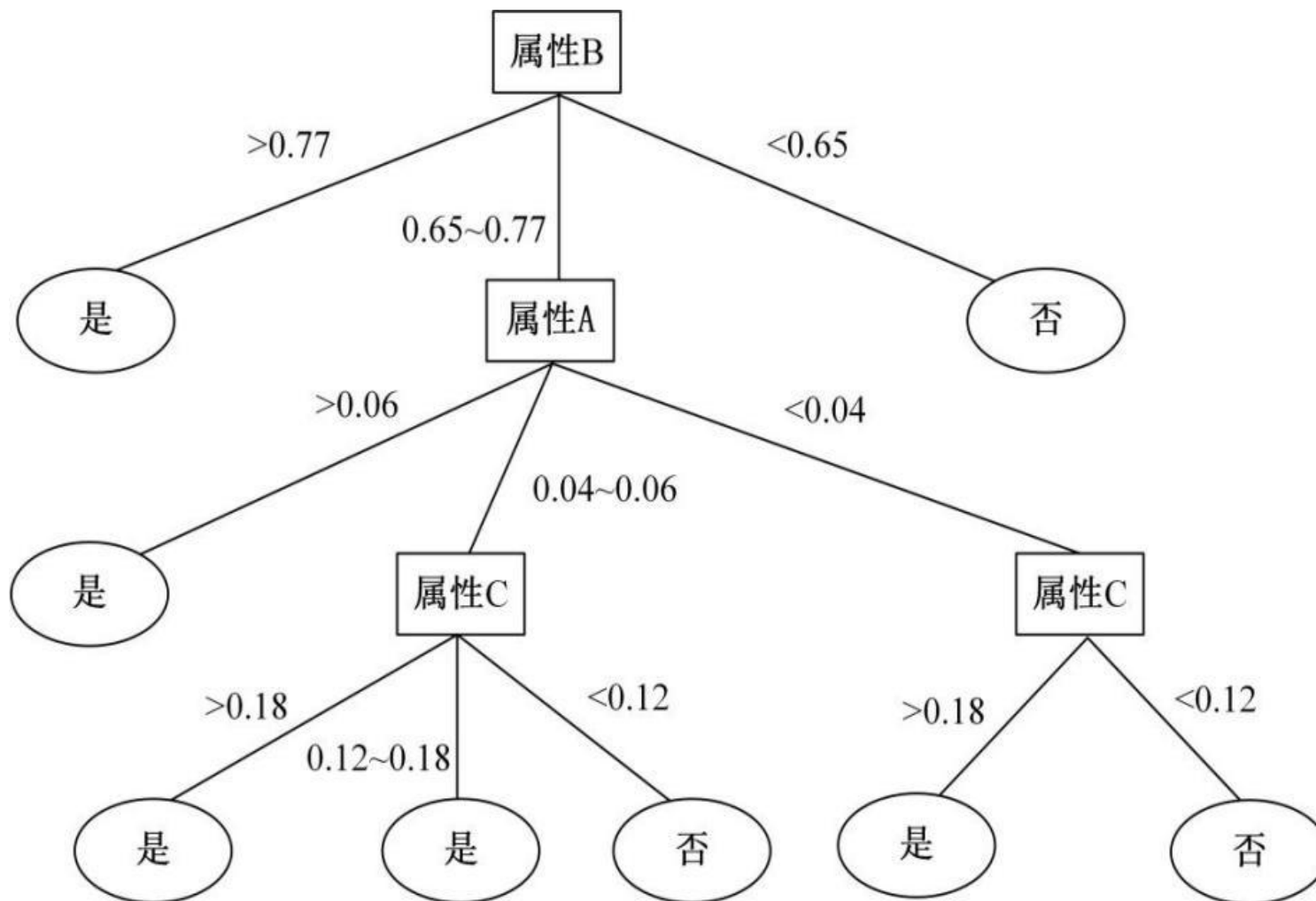
➤ 决策树实例

- 步骤3：子集 V_{a1} 中只含一种类别，无须继续划分进行划分。子集 V_{a2} V_{a3} 需要按属性C继续划分
- 以 V_{a3} 为例，它可继续划分为两个不同子集

序号	属性A	属性B	属性C	是否异常
1	< 0.04	0.65~0.77	> 0.18	是

序号	属性A	属性B	属性C	是否异常
1	< 0.04	0.65~0.77	< 0.12	否

➤ 决策树实例



➤ 决策树实例

决策树可以用如下的IF-THEN分类规则描述：

IF $B > 0.77$ THEN 电池异常

IF AND $A > 0.06$ THEN 电池异常

IF AND AND $C > 0.18$ THEN 电池异常

IF AND AND THEN 电池异常

IF AND AND $C < 0.12$ THEN 电池正常

IF AND $A < 0.04$ AND $C > 0.18$ THEN 电池异常

IF AND $A < 0.04$ AND $C < 0.12$ THEN 电池正常

IF $B < 0.65$ THEN 电池正常

➤ 属性选择度量

- 假设当前样本集合 D 包含 n 个类别的样本 C_1, C_2, \dots, C_n ，其中第 k 类样本 C_k ，在所有样本中出现的频率为 p_k 。属性 A 将 D 划分成 m 份， D_i 表示 D 的第 i 个子集， $|D|$ 和 $|D_i|$ 分别表示 D 和 D_i 中的样本数量

- 信息增益：划分前后样本数据集的熵的差值

样本 D 的信息熵 $Ent(D) = Ent(p_1, p_2, \dots, p_n) = -\sum_{k=1}^n p_k \log_2 p_k$

条件熵 $Ent(D, A) = -\sum_{i=1}^m \frac{|D_i|}{|D|} Ent(D_i)$

信息增益 $Gain(D, A) = Ent(D) - Ent(D, A)$

选择获得最大信息增益的属性作为分支属性

➤ 属性选择度量

- 增益率

$$Gain_ratio(D, A) = \frac{Gain(D, A)}{SplitInfo(D, A)}$$

$$SplitInfo(D, A) = -\sum_{i=1}^m \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$$

$SplitInfo(D, A)$ 反映属性A的纯度，A的取值越少，A的纯度就越高， $Ent(D, A)$ 的值也就越小，因此最后得到的信息增益率也就越高

➤ 属性选择度量

■ 基尼指数

$$Gini(D) = \sum_{k=1}^{|n|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|n|} p_k^2$$

- 反映了从数据集中随机抽取的样本其类别标志不一致的概率。
- $Gini(D)$ 越小，则数据集 D 的纯度越高

定义在属性 A 下数据集的基尼指数

$$Gini_A(D) = \frac{x_1}{N} Gini(A_1) + \frac{x_2}{N} Gini(A_2) + \cdots + \frac{x_m}{N} Gini(A_m)$$

$$Gini(A_i) = 1 - \sum_{j=1}^k \left(\frac{x_{ij}}{x_i} \right)^2$$

基尼指数越小表明该属性越适合作为分支的属性

➤ ID3算法

- 采用信息增益作为度量标准，在选择根节点和各个内部节点属性时，选择当前样本集中具有**最大信息增益值的属性**作为划分标准

优点：

建树方法简单，学习能力较强，但偏向于选择取值较多的属性作为分支属性

缺点：

只能构造出离散数据集的决策树，而对连续性属性不能直接进行处理，且对噪声数据敏感，抗噪性能较差

➤ 决策树实例

□ 使用ID3算法构造决策树

序号	属性A	属性B	属性C	是否异常
1	> 0.06	> 0.77	> 0.18	是
2	> 0.06	0.65~0.77	> 0.18	是
3	< 0.04	< 0.65	< 0.12	否
4	0.04~0.06	> 0.77	0.12~0.18	是
5	0.04~0.06	0.65~0.77	< 0.12	否
6	0.04~0.06	< 0.65	< 0.12	否
7	< 0.04	0.65~0.77	> 0.18	是
8	> 0.06	0.65~0.77	0.12~0.18	是
9	< 0.04	< 0.65	0.12~0.18	否
10	0.04~0.06	> 0.77	> 0.18	是
11	< 0.04	0.65~0.77	< 0.12	否
12	0.04~0.06	0.65~0.77	0.12~0.18	是
13	0.04~0.06	0.65~0.77	> 0.18	是
14	> 0.06	< 0.65	0.12~0.18	否
15	< 0.04	> 0.77	0.12~0.18	是

- 属性 $A = \{a_1, a_2, a_3\}$
 $a_1(> 0.06)$ 、 $a_2(0.04 \sim 0.06)$ 、 $a_3(< 0.04)$
- 属性 $B = \{b_1, b_2, b_3\}$
 $b_1(> 0.77)$ 、 $b_2(0.65 \sim 0.77)$ 、 $b_3(< 0.65)$
- 属性 $C = \{c_1, c_2, c_3\}$
 $c_1(> 0.18)$ 、 $c_2(0.12 \sim 0.18)$ 、 $c_3(< 0.12)$

➤ 决策树实例

□ 计算样本总的信息熵

$$Ent(D) = -\frac{9}{15} \log_2 \frac{9}{15} - \frac{6}{15} \log_2 \frac{6}{15} = 0.970$$

□ 样本集在属性A划分的条件下，各子集的熵为

$$\begin{cases} Ent(D, a_1) = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 0.811 \\ Ent(D, a_2) = -\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0.918 \\ Ent(D, a_3) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971 \end{cases}$$

$$Ent(D, A) = \frac{4}{15} \times 0.811 + \frac{6}{15} \times 0.918 + \frac{5}{15} \times 0.971 = 0.907$$

$$Ent(D, B) = 0.403$$

$$Ent(D, C) = 0.367$$

$$Gain(D, A) = Ent(D) - Ent(D, A) = 0.063$$

$$Gain(D, B) = 0.567$$

$$Gain(D, C) = 0.603$$

属性C具有最高信息增益，故将其作为划分属性

➤ 决策树实例

□ 样本训练集分为三个子集： D_{c_1} D_{c_2} D_{c_3} ， D_{c_1} D_{c_3} 均只含一类样本，不再继续划分

□ 子集 D_{c_2} 继续划分，计算总的信息熵

$$Ent(D, c_2) = -\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0.918$$

□ 计算样本子集在用属性A划分的条件下，子集的熵为

序号	属性A	属性B	属性C	是否异常
4	0.04~0.06	> 0.77	0.12~0.18	是
8	> 0.06	0.65~0.77	0.12~0.18	是
9	< 0.04	< 0.65	0.12~0.18	否
12	0.04~0.06	0.65~0.77	0.12~0.18	是
14	> 0.06	< 0.65	0.12~0.18	否
15	< 0.04	> 0.77	0.12~0.18	是

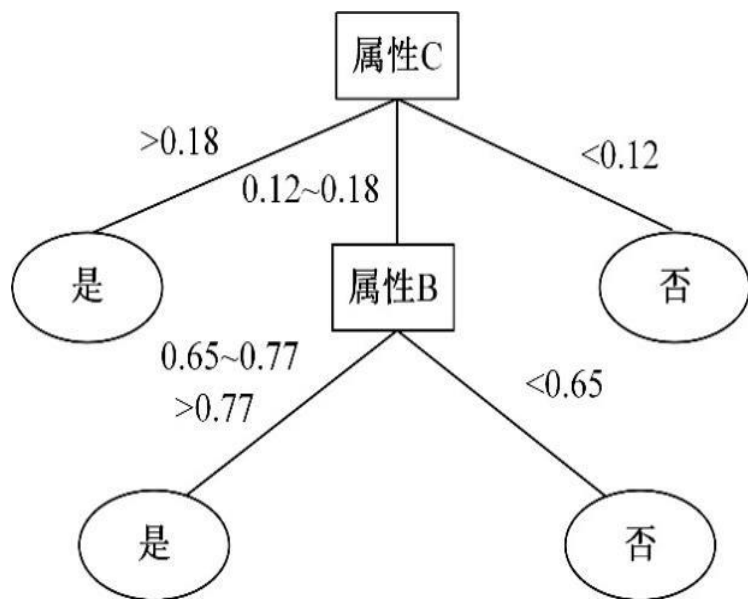
$$\left\{ \begin{aligned} Ent(D_{c_2}, a_1) &= -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \\ Ent(D_{c_2}, a_2) &= -\frac{2}{2} \log_2 \frac{2}{2} = 0 \\ Ent(D_{c_2}, a_3) &= -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \end{aligned} \right.$$

➡

$$\begin{aligned} Ent(D_{c_2}, B) &= 0 \\ Gain(D_{c_2}, A) &= 0.251 \\ Gain(D_{c_2}, B) &= 0.918 \end{aligned}$$

➤ 决策树实例

□ 建立完整的决策树



IF $C > 0.18$ THEN 电池异常

IF $0.12 \leq C \leq 0.18$ AND $C \geq 0.65$ THEN 电池异常

IF $0.12 \leq C \leq 0.18$ AND $B < 0.65$ THEN 电池正常

IF $C < 0.12$ THEN 电池正常

➤ C4.5算法

□ 基于ID3算法改进而来的决策树算法，它采用信息增益率作为判定划分属性好坏的标准

优点：

- (1) 可以有效减少特征属性值的多少对算法的影响
- (2) 可以采用二分法处理连续属性，还能通过忽略、补全等方法处理缺失值

缺点：

要对数据集进行多次的顺序扫描和排序，可能会导致算法的低效

➤ C4.5算法例题

□ 使用C4.5算法构造决策树

序号	属性A	属性B	属性C	是否异常
1	> 0.06	> 0.77	> 0.18	是
2	> 0.06	0.65~0.77	> 0.18	是
3	< 0.04	< 0.65	< 0.12	否
4	0.04~0.06	> 0.77	0.12~0.18	是
5	0.04~0.06	0.65~0.77	< 0.12	否
6	0.04~0.06	< 0.65	< 0.12	否
7	< 0.04	0.65~0.77	> 0.18	是
8	> 0.06	0.65~0.77	0.12~0.18	是
9	< 0.04	< 0.65	0.12~0.18	否
10	0.04~0.06	> 0.77	> 0.18	是
11	< 0.04	0.65~0.77	< 0.12	否
12	0.04~0.06	0.65~0.77	0.12~0.18	是
13	0.04~0.06	0.65~0.77	> 0.18	是
14	> 0.06	< 0.65	0.12~0.18	否
15	< 0.04	> 0.77	0.12~0.18	是

- 属性 $A = \{a_1, a_2, a_3\}$
 $a_1(> 0.06)$ 、 $a_2(0.04 \sim 0.06)$ 、 $a_3(< 0.04)$
- 属性 $B = \{b_1, b_2, b_3\}$
 $b_1(> 0.77)$ 、 $b_2(0.65 \sim 0.77)$ 、 $b_3(< 0.65)$
- 属性 $C = \{c_1, c_2, c_3\}$
 $c_1(> 0.18)$ 、 $c_2(0.12 \sim 0.18)$ 、 $c_3(< 0.12)$

➤ 决策树实例

- 在使用ID3算法构造决策树时，已知

$$Gain(D, A) = 0.063 \quad Gain(D, B) = 0.567 \quad Gain(D, C) = 0.603$$

- 计算此时各属性在D上的分裂信息，以A属性为例

$$SplitInfo(D, A) = -\frac{4}{15} \log_2 \frac{4}{15} - \frac{5}{15} \log_2 \frac{5}{15} - \frac{6}{15} \log_2 \frac{6}{15} = 1.566$$



$$SplitInfo(D, B) = 1.530 \quad SplitInfo(D, C) = 1.585$$

➤ 决策树实例

□ 若用属性A对D进行划分，所得的信息增益率为

$$Gain_ratio(D, A) = \frac{Gain(D, A)}{SplitInfo(D, A)} = 0.040$$

属性C具有最高信息增益率，故将作为根节点的划分属性



$$Gain_ratio(D, B) = 0.370 \quad Gain_ratio(D, C) = 0.385$$

□ 根据C属性建立三个子集，样本子集 D_{c_1} D_{c_3} 已分类完成。依次计算其它属性对子集划分后的信息增益率。

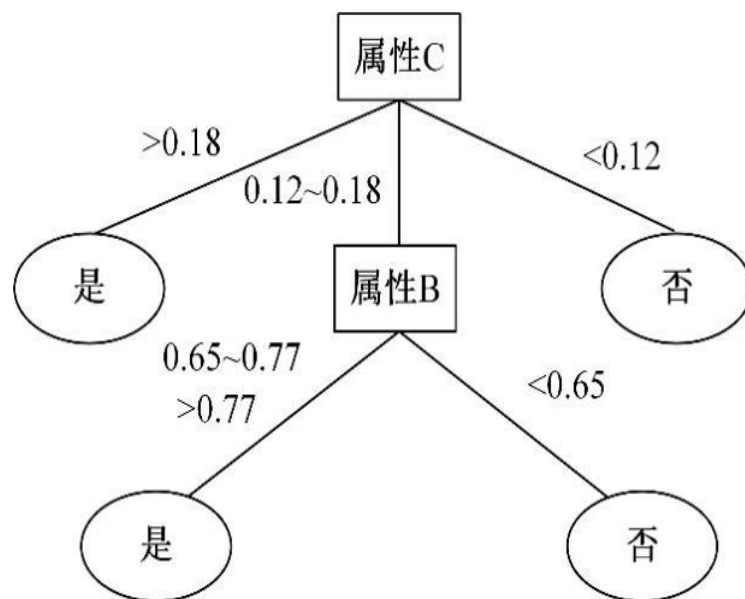
$$SplitInfo(D_{c_2}, A) = -\frac{2}{6}\log_2 \frac{2}{6} - \frac{2}{6}\log_2 \frac{2}{6} - \frac{2}{6}\log_2 \frac{2}{6} = 1.585 \quad \longrightarrow \quad SplitInfo(D_{c_2}, B) = 1.585$$

➤ 决策树实例

□ 若用属性A对D进行划分，所得的信息增益率为

$$Gain_ratio(D_{c_2}, A) = \frac{Gain(D_{c_2}, A)}{SplitInfo(D_{c_2}, A)} = 0.158 \quad \longrightarrow \quad Gain_ratio(D_{c_2}, B) = 0.580$$

□ 建立完整的决策树



IF C > 0.18 THEN 电池异常

IF 0.12 ≤ C ≤ 0.18 AND C ≥ 0.65 THEN 电池异常

IF 0.12 ≤ C ≤ 0.18 AND B < 0.65 THEN 电池正常

IF C < 0.12 THEN 电池正常

➤ CART算法

- ❑ CART算法是一种以基尼指数作为属性选择度量的方法。
- ❑ CART算法假设决策树是一棵二叉树，因此在每个分支节点出将当前的样本数据集分割成两个互不相交的子集，以基尼指数最小的属性为最佳的分支变量

优点：

对异常点和干扰数据的抵抗性强，在面对诸如存在缺失值、变量数多等问题时
CART显得非常稳健

缺点：

存在偏向多值属性的问题，且计算量较大

➤ CART算法例题

□ 使用CART算法构造决策树

序号	属性A	属性B	属性C	是否异常
1	> 0.06	> 0.77	> 0.18	是
2	> 0.06	0.65~0.77	> 0.18	是
3	< 0.04	< 0.65	< 0.12	否
4	0.04~0.06	> 0.77	0.12~0.18	是
5	0.04~0.06	0.65~0.77	< 0.12	否
6	0.04~0.06	< 0.65	< 0.12	否
7	< 0.04	0.65~0.77	> 0.18	是
8	> 0.06	0.65~0.77	0.12~0.18	是
9	< 0.04	< 0.65	0.12~0.18	否
10	0.04~0.06	> 0.77	> 0.18	是
11	< 0.04	0.65~0.77	< 0.12	否
12	0.04~0.06	0.65~0.77	0.12~0.18	是
13	0.04~0.06	0.65~0.77	> 0.18	是
14	> 0.06	< 0.65	0.12~0.18	否
15	< 0.04	> 0.77	0.12~0.18	是

- 属性 $A = \{a_1, a_2, a_3\}$
 $a_1(> 0.06)$ 、 $a_2(0.04 \sim 0.06)$ 、 $a_3(< 0.04)$
- 属性 $B = \{b_1, b_2, b_3\}$
 $b_1(> 0.77)$ 、 $b_2(0.65 \sim 0.77)$ 、 $b_3(< 0.65)$
- 属性 $C = \{c_1, c_2, c_3\}$
 $c_1(> 0.18)$ 、 $c_2(0.12 \sim 0.18)$ 、 $c_3(< 0.12)$

➤ 决策树实例

- 计算样本数据D中属性的基尼指数，找出**基尼指数减小幅度最大**的属性作为根节点属性。根节点 V_0 的基尼指数为

$$Gini(V_0) = 1 - \left(\frac{9}{15}\right)^2 - \left(\frac{6}{15}\right)^2 = 0.480$$

- 计算用属性A对D划分后各样本子集的基尼指数减小值

按 $\{a_1\} / \{a_2, a_3\}$ 划分D时，基尼指数为

$$Gini(D_{\{a_1\}/\{a_2, a_3\}}, V_0) = \frac{4}{15} \times [1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2] + \frac{11}{15} \times [1 - \left(\frac{6}{11}\right)^2 - \left(\frac{5}{11}\right)^2] = 0.464$$

基尼指数减小值

$$\Delta Gini(D_{\{a_1\}/\{a_2, a_3\}}, V_0) = 0.48 - 0.464 = 0.016$$



$$\Delta Gini(D_{\{a_2\}/\{a_1, a_3\}}, V_0) = 0.006$$

$$\Delta Gini(D_{\{a_3\}/\{a_1, a_2\}}, V_0) = 0.040$$

➤ 决策树实例

- 计算用属性B对D划分后各样本子集的基尼指数减小值

$$\Delta Gini(D_{\{b_1\}/\{b_2, b_3\}}, V_0) = 0.116$$

$$\Delta Gini(D_{\{b_2\}/\{b_1, b_3\}}, V_0) = 0.023$$

$$\Delta Gini(D_{\{b_3\}/\{b_1, b_2\}}, V_0) = 0.262$$

- 计算用属性C对D划分后各样本子集的基尼指数减小值

$$\Delta Gini(D_{\{c_1\}/\{c_2, c_3\}}, V_0) = 0.160$$

$$\Delta Gini(D_{\{c_2\}/\{c_1, c_3\}}, V_0) = 0.006$$

$$\Delta Gini(D_{\{c_3\}/\{c_1, c_2\}}, V_0) = 0.262$$

选择**B**对**D**按 $\{b_3\}/\{b_1, b_2\}$
进行划分，向下生成子集

D_{b_3} $D_{\{b_1, b_2\}}$

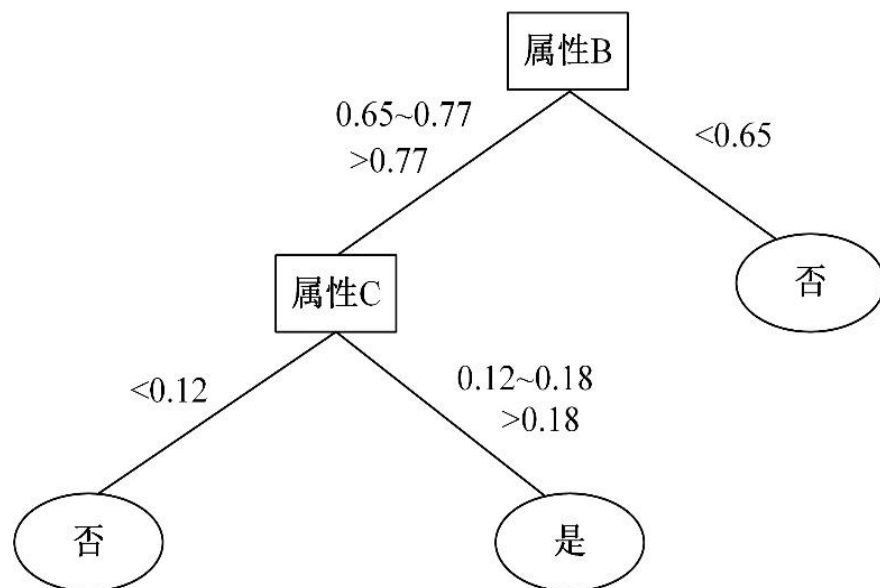
➤ 决策树实例

□ 子集 D_{b_3} 只含一类样本，故无法继续划分。 $D_{\{b_1, b_2\}}$ 的节点的基尼指数为

$$Gini(V_{11}) = 1 - \left(\frac{2}{11}\right)^2 - \left(\frac{9}{11}\right)^2 = 0.298 \quad \longrightarrow \quad Gini(D_{\{c_3\}/\{c_2, c_1\}}, V_{11}) = 0$$

$\Delta Gini(D_{\{c_3\}/\{c_2, c_1\}}, V_{11})$ 最大，故由属性C分

□ 生成决策树



IF B ≥ 0.65 AND C ≥ 0.12 THEN 电池异常

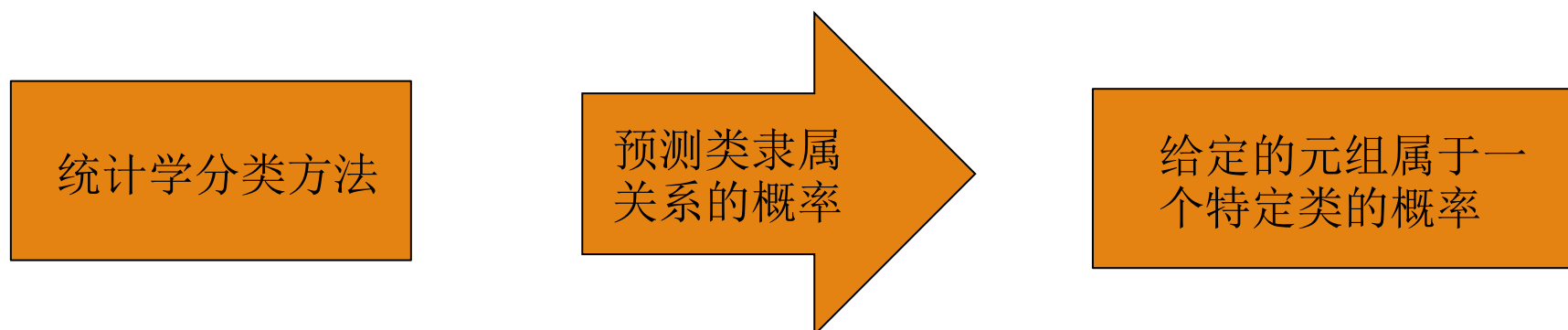
IF B ≥ 0.65 AND C < 0.12 THEN 电池正常

IF B < 0.65 THEN 电池正常

- 分类分析的基本概念
- 决策树
- 贝叶斯分类
- 支持向量机
- 人工神经网络
- 分类模型的评价与选择
- 组合分类技术
- 实例



➤ 什么是贝叶斯分类



➤ 贝叶斯的基本原理

□ 先验概率

- ✓ 根据历史数据或主观判断所确定的各事件发生的概率

□ 后验概率

- ✓ 基于试验或调查所得到的各事件发生的概率，亦可看作是在考虑相关背景和前提下得到的一个条件概率

□ 条件概率

- ✓ 某一事件在另一事件已经发生的条件下发生的概率

➤ 贝叶斯的基本原理

- $P(A|B)$ 是事件B发生情况下，A发生的概率，叫做事件B发生下事件A的**条件概率**；
- $P(B|A)$ 是根据A参数值判断某类别B的概率，称为**后验概率**，指事件A发生之后，对事件B重新估计和调整后再算出的概率值；
- $P(B)$ 是直接判断某样本属于B的概率，表示事件A发生之前，通过现有信息对B的理解和判断，称为**先验概率**

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}$$

$$P(AB) = P(B|A)P(A)$$



当A、B相互独立

$$P(AB) = P(A)P(B)$$

➤ 贝叶斯的基本原理

□ 全概率公式

若样本空间 D 被划分为 n 个子集 B_1, B_2, \dots, B_n ，它们两两互斥、互不相容，且每个子集发生的概率 $P(B_i) > 0, i = 1, 2, \dots, n$ 。则在样本空间上事件 A 发生的概率为

$$P(A) = \sum_{i=1}^n P(B_i)P(A|B_i)$$

□ 贝叶斯定理

当 $P(A) > 0$ 时，由上式可得到贝叶斯定理

$$P(B_i|A) = \frac{P(B_iA)}{P(A)} = \frac{P(B_i)P(A|B_i)}{\sum_{i=1}^n P(B_i)P(A|B_i)}$$

➤ 贝叶斯分类实例

- 例：某电子设备厂所用的元件由三家元件厂提供，根据以往记录，这三个厂家的次品率分别为0.02，0.01和0.03，提供元件的份额分别为0.15，0.8和0.05，设这三家的产品在仓库是均匀混合的，且无区别的标志。
 - 问题1：在仓库中，随机抽取一个元件，求它是次品的概率；
 - 问题2：在仓库中，随机抽取一个元件，若已知它是次品，则该次品来自三家供货商的概率分别是多少？

➤ 贝叶斯分类实例

- 【解】设A表示“取到的元件是次品”， B_i 表示“取到的元件是由第i个厂家生产的”，则

$$P(B_1)=0.15, P(B_2)=0.8, P(B_3)=0.05$$

- 对于问题1，由全概率公式可得：

$$P(A) = P(B_1)*P(A|B_1) + P(B_2)*P(A|B_2) + P(B_3)*P(A|B_3)$$

$$= 0.15*0.02+0.8*0.01+0.05*0.03$$

$$= 0.0125$$

➤ 贝叶斯分类实例

- 【解】设A表示“取到的元件是次品”， B_i 表示“取到的元件是由第*i*个厂家生产的”，则

$$P(B_1)=0.15, P(B_2)=0.8, P(B_3)=0.05$$

- 对于问题2，由贝叶斯公式可得：

$$\begin{aligned} P(B_1|A) &= P(B_1)*P(A|B_1)/P(A) \\ &= 0.15*0.02/0.0125 = 0.24 \end{aligned}$$

$$\begin{aligned} P(B_2|A) &= P(B_2)*P(A|B_2)/P(A) \\ &= 0.8*0.01/0.0125 = \mathbf{0.64} \end{aligned}$$

$$\begin{aligned} P(B_3|A) &= P(B_3)*P(A|B_3)/P(A) \\ &= 0.05*0.03/0.0125 = 0.12 \end{aligned}$$

➤ 朴素贝叶斯

- 举例：你在街上看到一个黑人小伙，你第一反应十有八九猜他来自于非洲，其实别人有可能来源于美洲或亚洲，但是你会下意识的选择后验概率最大的类别



➤ 朴素贝叶斯分类

□ 基本思想：若事件A表示样本数据的属性集合，事件B表示数据的类标签集合。则对于某个待分类样本，可以通过**贝叶斯定理**计算出该样本隶属于某个类别的**后验概率**，将**后验概率最大的类别**视作该样本所属类别。

□ 具体描述

- 给定一个训练样本集D，假设D中样本包含 m 个类别，记作 C_1, C_2, \dots, C_m
- 训练样本含 n 个属性，记作 A_1, A_2, \dots, A_n
- 现给定某个未知类别的待分类样本 X ， X 可以表示为一组 n 维向量

$X = \{x_1, x_2, \dots, x_n\}$ ，其中 x_i 表示该样本在属性 A_i 的测试值

□ 任务：预测样本 X 所属的类标签

➤ 朴素贝叶斯分类

□ 后验概率：由贝叶斯定理可得

$$P(C_i | \mathbf{X}) = \frac{P(C_i)P(\mathbf{X} | C_i)}{P(\mathbf{X})}$$

■ 计算先验概率 $P(C_i)$

$$P(C_i) = |C_{i,D}| / |D|$$

其中， $|C_{i,D}|$ 是样本D中属于 C_i 类的样本数， $|D|$ 是D的总样本数量

■ 计算条件概率 $P(\mathbf{X} | C_i)$

待分类样本 $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ ，其中 x_k 表示该样本在属性 A_k 上的测试值
作如下假设：各数据属性之间相互独立，不存在任何关联性和依赖关系。因此

$$P(\mathbf{X} | C_i) = P(x_1 | C_i)P(x_2 | C_i)P(x_3 | C_i) \cdots P(x_n | C_i)$$

➤ 朴素贝叶斯分类

- 如果 A_k 是离散属性，则可作如下估计：

$$P(x_k | C_i) = \frac{|D_{C_i, x_k}|}{|C_i, D|}$$


其中， $|D_{C_i, x_k}|$ 是 C_i 类数据样本中属性 A_k 的值为 x_k 的样本数

- 如果是连续值属性，通常假定连续值属性服从均值为 μ 、标准差为 σ 的高斯分布

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$P(x_k | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

□ 计算概率乘积

$P(C_i)P(\mathbf{X} | C_i) > P(C_j)P(\mathbf{X} | C_j), 1 \leq j \leq n, j \neq i$ 成立  朴素贝叶斯分类预测 \mathbf{X} 属于类 C_i

➤ 朴素贝叶斯分类算法流程

输入：待预测样本 X ，属性集 A ，类别集合 $C = \{C_1, C_2, \dots, C_n\}$

输出：样本 X 预测类别

1. for C 中的每一个类别 C_i
2. 计算该类样本在所有样本中出现的概率 $P(C_i)$
3. 计算 $P(X|C_i)$
4. 计算 $P(X|C_i)P(C_i)$
5. end for
6. 找出最大值 $P(X|C_m)P(C_m)$
7. 判定 $X \in C_m$

➤ 朴素贝叶斯分类实例

□ 使用朴素贝叶斯分类来预测样本 $X = \{A = a_3, B = b_2, C = c_2\}$ 的预测类标签

序号	属性A	属性B	属性C	是否异常
1	> 0.06	> 0.77	> 0.18	是
2	> 0.06	0.65~0.77	> 0.18	是
3	< 0.04	< 0.65	< 0.12	否
4	0.04~0.06	> 0.77	0.12~0.18	是
5	0.04~0.06	0.65~0.77	< 0.12	否
6	0.04~0.06	< 0.65	< 0.12	否
7	< 0.04	0.65~0.77	> 0.18	是
8	> 0.06	0.65~0.77	0.12~0.18	是
9	< 0.04	< 0.65	0.12~0.18	否
10	0.04~0.06	> 0.77	> 0.18	是
11	< 0.04	0.65~0.77	< 0.12	否
12	0.04~0.06	0.65~0.77	0.12~0.18	是
13	0.04~0.06	0.65~0.77	> 0.18	是
14	> 0.06	< 0.65	0.12~0.18	否
15	< 0.04	> 0.77	0.12~0.18	是

- 属性 $A = \{a_1, a_2, a_3\}$
 $a_1(> 0.06)$ 、 $a_2(0.04 \sim 0.06)$ 、 $a_3(< 0.04)$
- 属性 $B = \{b_1, b_2, b_3\}$
 $b_1(> 0.77)$ 、 $b_2(0.65 \sim 0.77)$ 、 $b_3(< 0.65)$
- 属性 $C = \{c_1, c_2, c_3\}$
 $c_1(> 0.18)$ 、 $c_2(0.12 \sim 0.18)$ 、 $c_3(< 0.12)$

电池异常: Y_1

电池正常: Y_2

➤ 朴素贝叶斯分类实例

□ 计算各类别先验概率

$$P(Y_1) = 9/15 = 0.6 \quad P(Y_2) = 6/15 = 0.4$$

□ 计算类条件概率

$$P(A = a_3 | Y_1) = 2/9 = 0.222 \quad P(A = a_3 | Y_2) = 3/6 = 0.500$$

$$P(B = b_2 | Y_1) = 5/9 = 0.556 \quad P(B = b_2 | Y_2) = 2/6 = 0.333$$

$$P(C = c_2 | Y_1) = 4/9 = 0.444 \quad P(C = c_1 | Y_2) = 2/6 = 0.333$$



$$P(X | Y_1) = P(A = a_3 | Y_1)P(B = b_2 | Y_1)P(C = c_2 | Y_1) = 0.055$$

$$P(X | Y_2) = P(A = a_3 | Y_2)P(B = b_2 | Y_2)P(C = c_2 | Y_2) = 0.055$$

➤ 朴素贝叶斯分类实例

□ 计算概率 $P(\mathbf{X} | Y_i)P(Y_i)$

$$P(\mathbf{X} | Y_1)P(Y_1) = 0.055 \times 0.6 = 0.033$$

$$P(\mathbf{X} | Y_2)P(Y_2) = 0.055 \times 0.4 = 0.022$$

□ 找出上述结果最大的类

$$P(\mathbf{X} | Y_1)P(Y_1) > P(\mathbf{X} | Y_2)P(Y_2)$$



样本 \mathbf{X} 电池状态异常

➤ 优点：

- ❑ 朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率
- ❑ 简单、容易实现、速度快，需要估计的参数少
- ❑ 对缺失数据不敏感

➤ 缺点

- ❑ 模型**假设属性之间相互独立**，忽略了变量间可能存在的依赖关系，这种假设一定程度上降低了朴素贝叶斯的分类准确率
- ❑ 在属性之间相关性较大时，分类效果不好；而在属性相关性较小时，朴素贝叶斯性能最为良好

- 分类分析的基本概念
- 决策树
- 贝叶斯分类
- **支持向量机**
- 人工神经网络
- 分类模型的评价与选择
- 组合分类技术
- 实例





Vladimir Vapnik

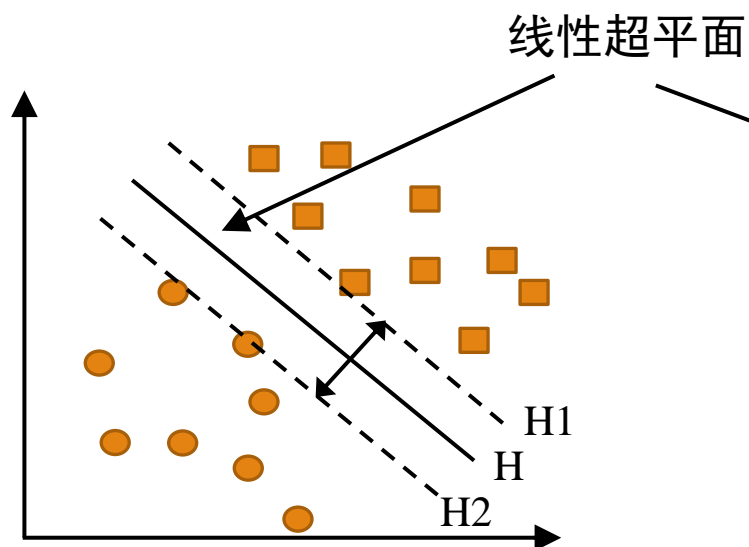
- ❑ 俄罗斯统计学家、数学家
- ❑ 上个世纪70年代已经创造
- ❑ 发表于欧美主流学术期刊

➤ 支持向量机基本原理

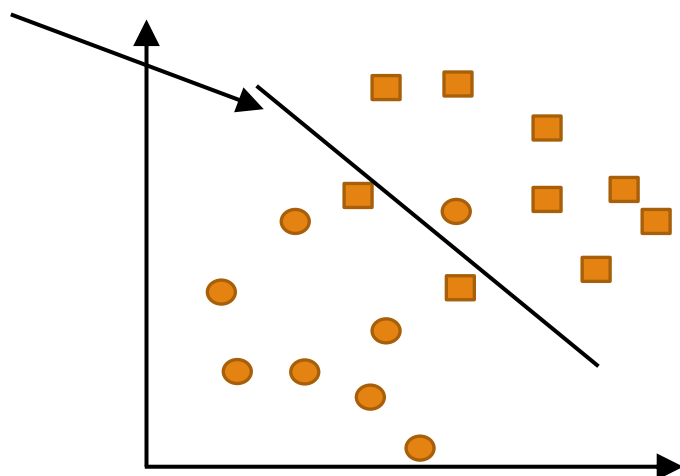
- 建立在统计学习理论基础上的的一种**预知性机器学习**方法
 - 使类与类之间的**间隔最大化**
 - 在解决**小样本、非线性及高维模式识别**中表现出特有优势
 - 能够推广到**函数拟合**等其他机器学习问题中
- 基本思想：针对两类分类问题，寻找一个**超平面**作为两类训练样本点的分割，以保证**最小的分类错误率**。

➤ 支持向量机基本原理

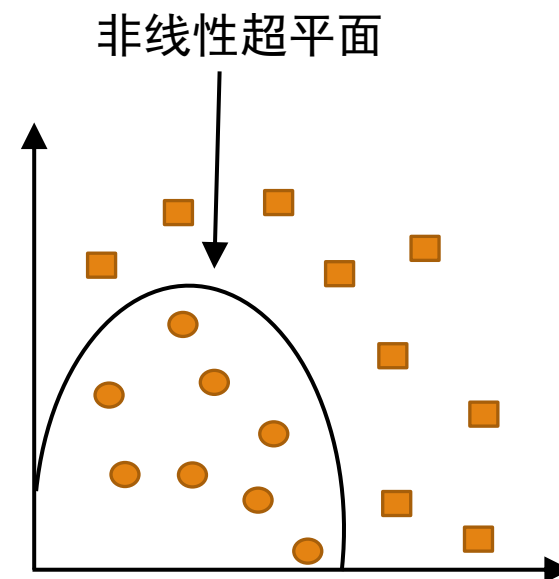
□ 不失一般性，将分类问题限制于**二分类**，即数据样本具有两种类别



线性可分情况



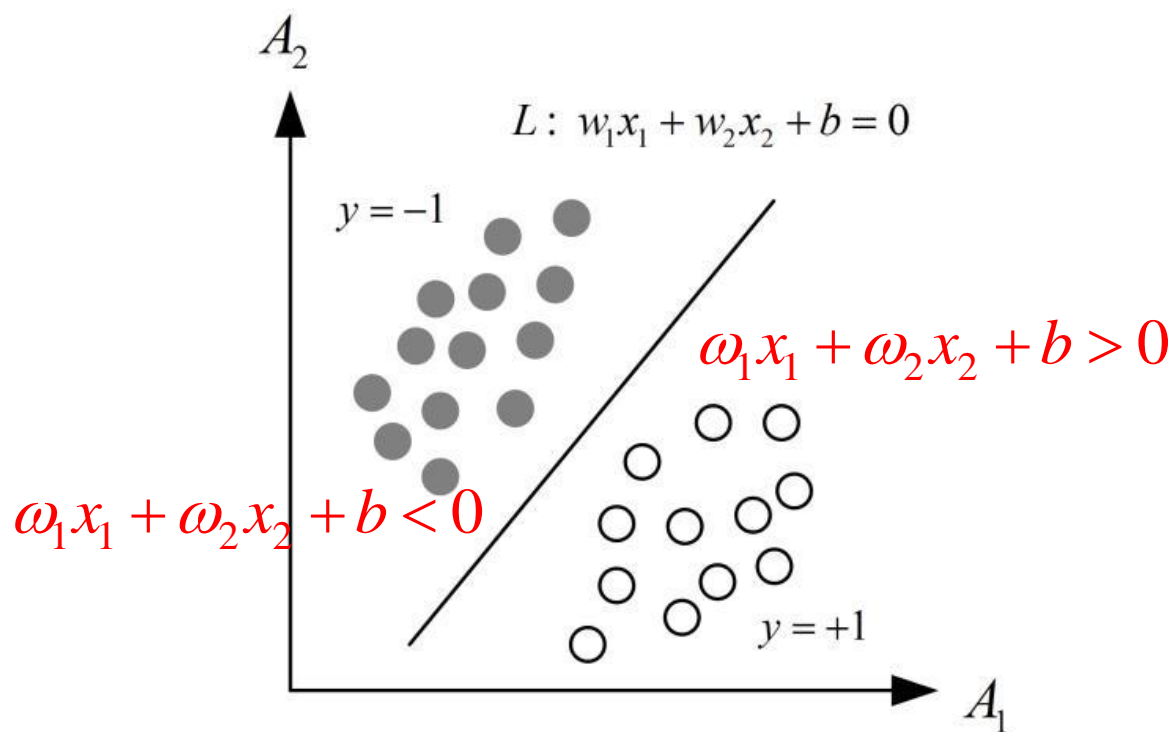
线性不可分情况



线性不可分情况

➤ 线性可分的数学定义

□ 设线性可分情况的训练样本 $D = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \dots, (\mathbf{X}_n, y_n)\}, y_i \in \{-1, 1\}$



直线方程: $w_1x_1 + w_2x_2 + b = 0$

权重: $w_1 \ w_2$

偏置: b

线性可分的定义: $y_i(w_1x_1 + w_2x_2 + b) > 0, \forall i$



$$y_i(\mathbf{w}^T \cdot \mathbf{X}_i + b) > 0, \forall i$$

$$\mathbf{w} = (w_1, w_2, \dots, w_n)^T \quad \mathbf{X}_i = (x_1, x_2, \dots, x_n)^T$$

➤ 线性可分的数学定义

□ 用数学严格定义训练样本以及他们的标签

□ 假设：

我们有 N 个训练样本和他们的标签

$$\{(X_1, y_1), (X_2, y_2), \dots, (X_N, y_N)\}$$

其中

$$X_i = [x_{i1}, x_{i2}]^T$$

$$y_i = \{+1, -1\}$$

X_i 属于 c_1

X_i 属于 c_2

➤ 线性可分的数学定义

□ 线性可分的严格定义：一个训练样本集 $\{(X_1, y_1), (X_2, y_2), \dots, (X_N, y_N)\}$

在 $i=1 \sim N$ 线性可分，是指存在 (ω_1, ω_2, b)

使得对 $i=1 \sim N$ ，有：

(1) 若 $y_i = +1$ ，则 $\omega_1 x_{i1} + \omega_2 x_{i2} + b > 0$

(2) 若 $y_i = -1$ ，则 $\omega_1 x_{i1} + \omega_2 x_{i2} + b < 0$

➤ 线性可分的数学定义

□ 用向量形式来定义线性可分线性

假设：

$$X_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}^T \quad \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}^T$$

(1) 若 $y_i = +1$ 则 $\omega^T X_i + b > 0$

(2) 若 $y_i = -1$ 则 $\omega^T X_i + b < 0$

➤ 线性可分的数学定义

□ 线性可分定义的最简化形式

$$(1) \text{ 若 } y_i = +1 \text{ 则 } \omega^T x_i + b > 0$$

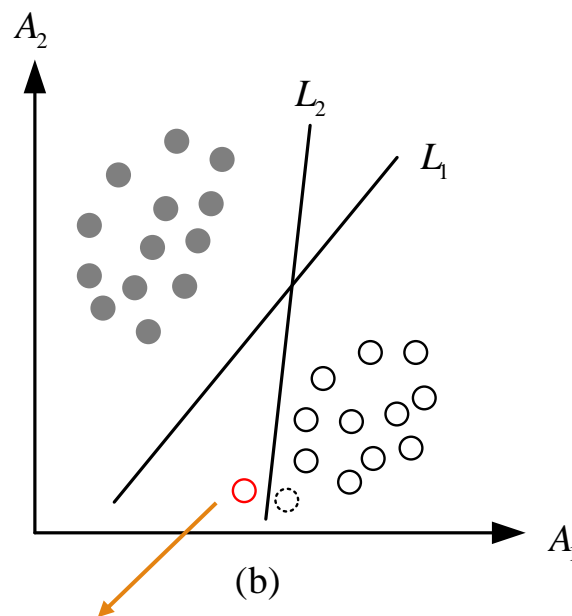
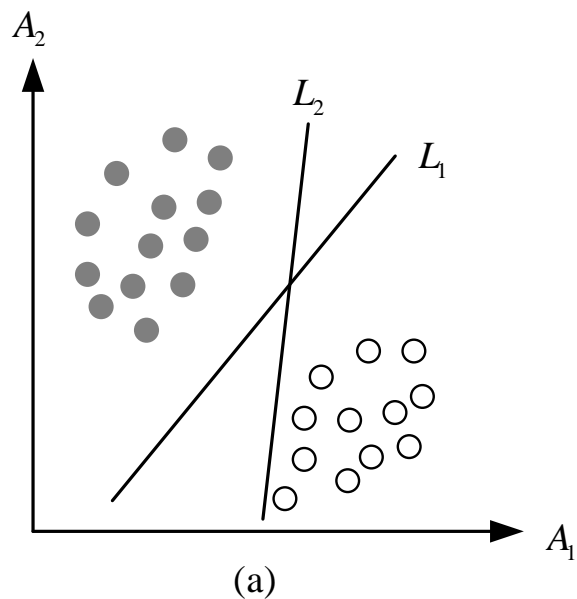
$$(2) \text{ 若 } y_i = -1 \text{ 则 } \omega^T x_i + b < 0$$

如果 $y_i = +1$ 或 $y_i = -1$

一个训练样本集 $\{(x_i, y_i)\}$ ，在 $i=1 \sim N$ 线性可分，是指存在 (ω, b) ，使得对 $i=1 \sim N$ ，有：
$$y_i (\omega^T x_i + b) > 0$$

➤ 线性可分的数学定义

□ 扰动问题



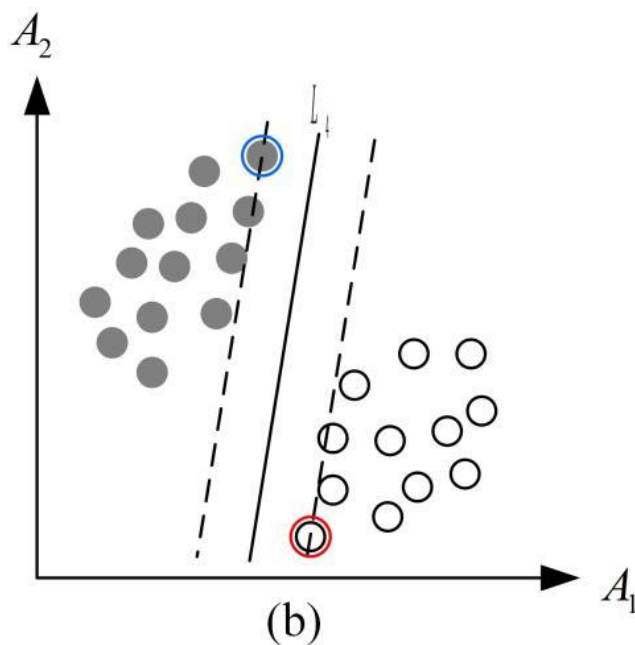
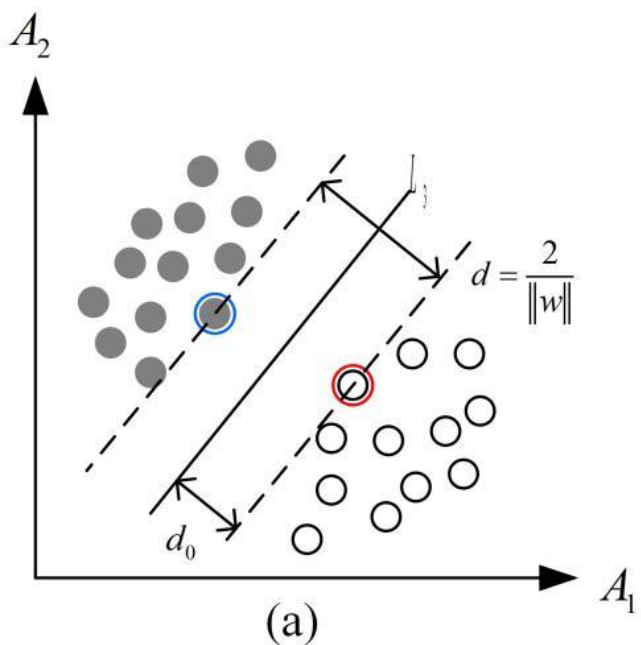
超平面 L_1 比 L_2 容错率更高，鲁棒性更强

如何寻找最佳超平面？

噪声影响， L_2 错误分类

➤ 最佳超平面应该满足的条件

- (1) 超平面准确无误地分开了两类样本；
- (2) 超平面与两类支持向量的距离相等；
- (3) 超平面使得间隔最大化。



如何用严格的数学



寻找最优分类超平面的过程

➤ 支持向量机问题描述

□ 支持向量 \mathbf{x}_0 到分类超平面的距离

$$d_0 = \frac{|\mathbf{w}^T \mathbf{x}_0 + b|}{\|\mathbf{w}\|} = \frac{y_0 (\mathbf{w}^T \mathbf{x}_0 + b)}{\|\mathbf{w}\|}$$

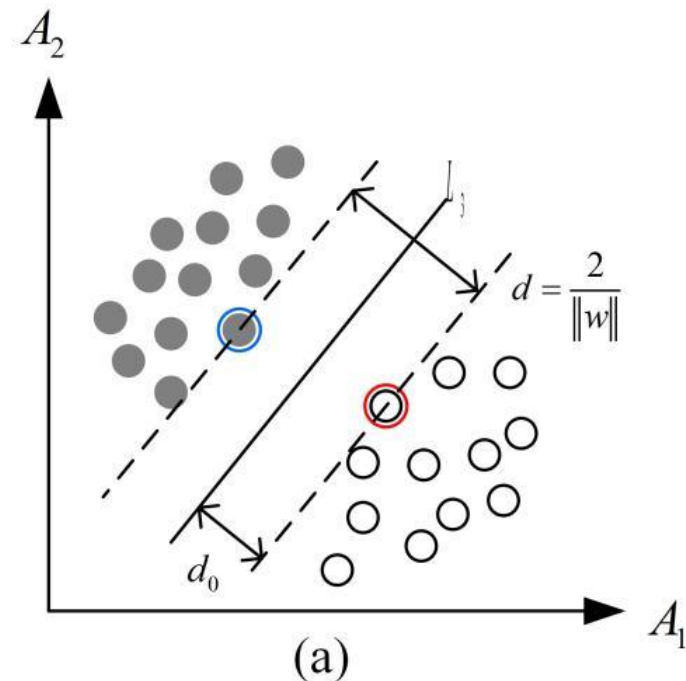
$$\|\mathbf{w}\| = \sqrt{\mathbf{w}_1^2 + \mathbf{w}_2^2 + \cdots + \mathbf{w}_n^2}$$

□ 最佳超平面应使间隔 $d = 2d_0$ 最大化

$$\begin{aligned} & \max_{\mathbf{w}, b} d \\ \text{s.t. } & y_i (\mathbf{w}^T \mathbf{X}_i + b) \geq 0, \quad \forall i \end{aligned}$$

求解 \mathbf{w} 、 b

最佳超平面



➤ 两个事实

□ 事实1:

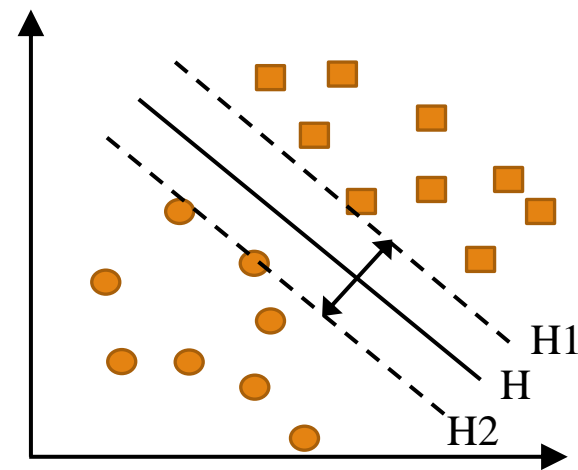
$(\omega^T x + b) = 0$ 与 $(a\omega^T)x + (ab) = 0$ 是同一个超平面 $a \neq 0$

□ 事实2: 一个点 x_0 到超平面 $(\omega^T x + b) = 0$ 的距离

$$d = \frac{|\omega^T x_0 + b|}{\|\omega\|}$$

一个点 (x_0, y_0) 到超平面 $\omega_1 x + \omega_2 y + b = 0$

$$d = \frac{|\omega_1 x_0 + \omega_2 y_0 + b|}{\sqrt{\omega_1^2 + \omega_2^2}}$$



➤ 支持向量机优化问题推导中最难理解的部分

□ 用 a 去缩放 ωb

$$(\omega, b) \rightarrow (a\omega, ab):$$

最终使在支持向量 x_0 上有 $|\omega^T x_0 + b| = 1$ ，而在非支持向量上 $|\omega^T x_0 + b| > 1$

由于： (ω, b) 和 $(a\omega, ab)$ 表示的超平面是同一个平面，可以用 a 进行缩放

□ 事实2：支持向量 x_0 到超平面 的距离将会变为：

$$d = \frac{|\omega^T x_0 + b|}{\|\omega\|} = \frac{1}{\|\omega\|}$$

最大化支持向量到超平面的距离，等价于最小化 $\|\omega\|$

➤ 支持向量机问题描述

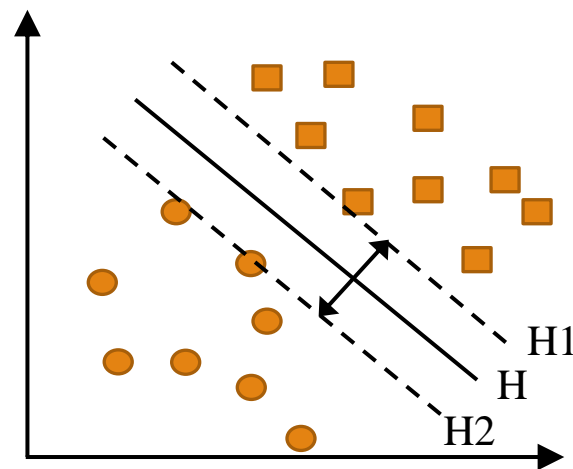
□ 支持向量机需要寻找的是最大化间隔的超平面

已知：训练样本集 $\{(X_i, y_i)\}$, $i=1\sim N$;

待求： (ω, b)

支持向量机要找一个超平面，使它的间隔最大：

离两边所有支持向量的距离相等。



➤ 支持向量机求解

□ 线性可分情况下，支持向量机寻找最佳超平面的优化问题可以表示为：

$$\text{最小化: } \frac{1}{2} \|\omega\|^2$$

$$\text{限制条件: } y_i(\omega^T x_i - b) \geq 1$$

， $i = 1 \sim N$ 是已知的 $\{(X_i, y_i)\}$ 是待求的 (ω, b)

□ 这是一个凸优化问题中的二次规划问题：

(1) 目标函数是二次项； (2) 限制条件是一次项

这种凸优化问题要么无解，要么只有唯一的最小值解（只有唯一的一个全局极值）。

➤ 支持向量机求解

□ 使用拉格朗日乘子法得到“对偶问题”

- 满足约束条件的优化问题可由拉格朗日算符 $L(\mathbf{w}, b)$ 转化

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i \{(\mathbf{w}^T \cdot \mathbf{X}_i + b)y_i - 1\} \quad \text{拉格朗日乘数: } \alpha_i \geq 0$$

- 拉格朗日函数参数的最小化:

$$\begin{aligned} \frac{\partial L}{\partial b} = 0 &\Rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \quad \text{代入原目} \\ \frac{\partial L}{\partial \mathbf{w}} = 0 &\Rightarrow \mathbf{w} = \sum_{i=1}^N \mathbf{X}_i \alpha_i y_i \quad \text{标函数} \end{aligned}$$

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

对偶问题求解: $\boldsymbol{\alpha} = (\alpha_1, \alpha_1, \dots, \alpha_n)$
SMO 算法求解

满足KKT条件

$$\begin{cases} \alpha_i \geq 0, \\ y_i f(\mathbf{x}_i) \geq 1, \\ \alpha_i (y_i f(\mathbf{x}_i) - 1) = 0. \end{cases}$$

必有 $\alpha_i = 0$ 或 $y_i f(\mathbf{x}_i) = 1$

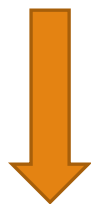
$$\text{最终模型} \quad f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

解具有稀疏性，模型仅与支持向量有关，得名：Support Vector Machine

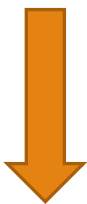
➤ 支持向量机优化问题-线性不可分情况 (1)

- 线性不可分情况下，找不到对应的 ω 和 b ，需要放松限制条件（软间隔SVM）

基本思路： 对每个训练样本及标签 (x_i, y_i)



松弛变量： ξ_i



限制条件改写为： $y_i (\omega^T x_i + b) \geq 1 - \xi_i$

➤ 支持向量机优化问题-线性不可分情况 (1)

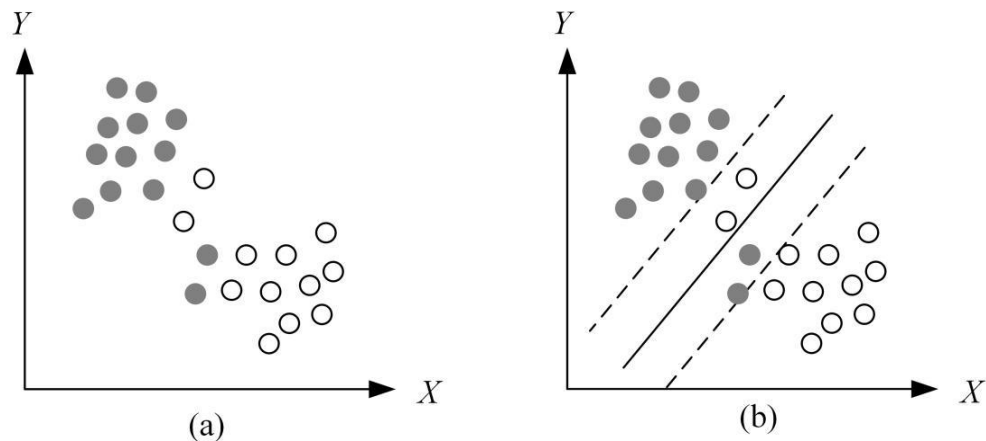
□ 支持向量机优化版本——软间隔支持向量机

$$\text{最小化: } \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^N \xi_i$$

限制条件: (1) 松弛变量: , $i = 1 \sim N$ $\xi_i \geq 0$

$$(2) \quad , i = 1 \sim N \quad y_i (\omega^T x_i + b) \geq 1 - \xi_i$$

- 以前的目标函数只需要最小化: $\frac{1}{2} \|\omega\|^2$
- 现在的目标函数增加了一项: 所有 ξ_i 的和
- 比例因子C: 平衡两项 (事先设定的参数叫做算法的超参数)



支持向量机是超参数很少的算法模型

➤ 线性不可分情况-线性不可分情况（1）

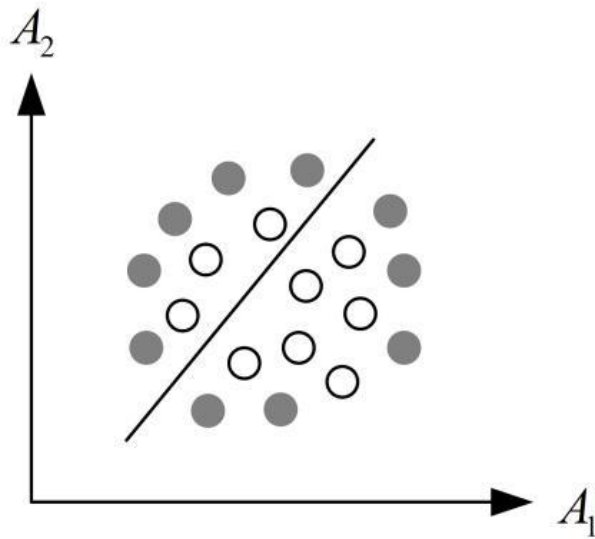
□ 线性不可分时不存在 w 和 b 满足上述优化问题的限制条件

□ 解决：

一般在SVM中定义核函数，实现训练集向更高维空间（特征）的映射，将其转化为线性可分的情况处理。

□ 常见的核函数

线性核函数、多项式核函数、高斯核函数等

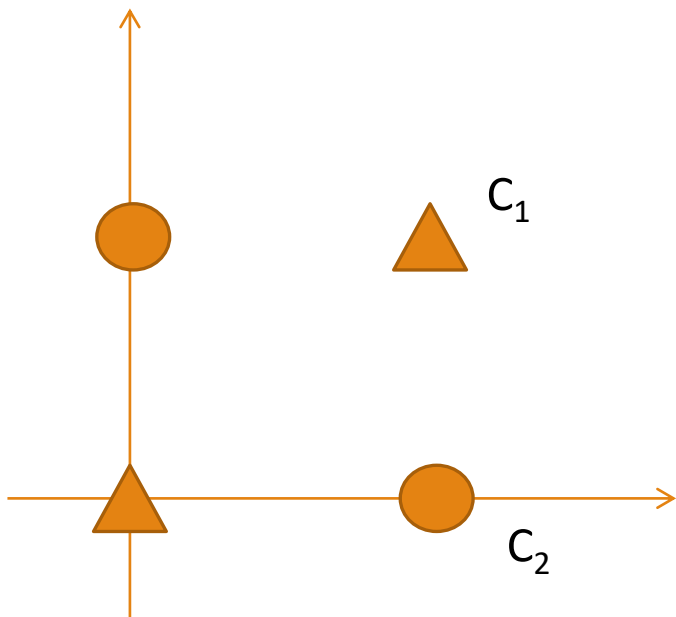


➤ 支持向量机优化问题-线性不可分情况 (2)

□ 将特征空间由低维映射到高维

□ 用线性超平面对数据进行分类

考察如图的异或问题



即:

$$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

C_1

$$x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$x_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

C_2

$$x_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

线性不可分的。

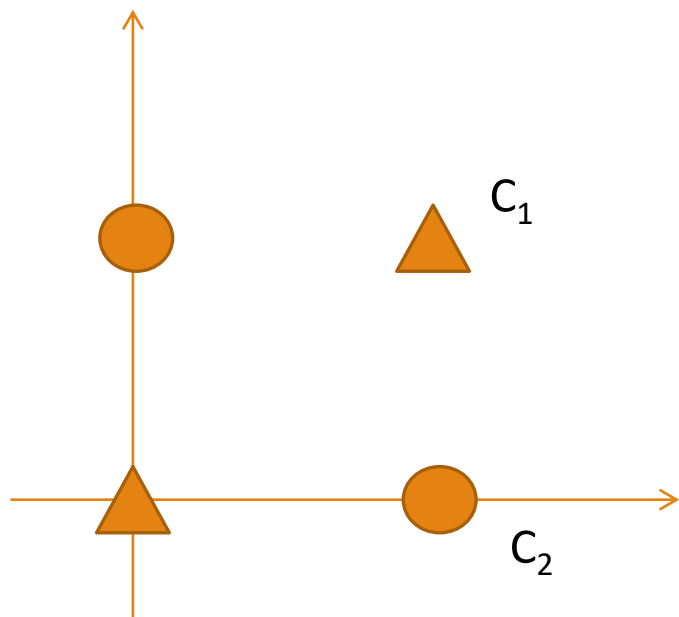
➤ 支持向量机优化问题-线性不可分情况 (2)

□ 构造一个二维到五维的映射 $\varphi(x)$

$$\varphi(x) : x = \begin{bmatrix} a \\ b \end{bmatrix} \longrightarrow \varphi(x) = \begin{bmatrix} a^2 \\ b^2 \\ a \\ b \\ ab \end{bmatrix}$$

$$\varphi(x_1), \varphi(x_2), \varphi(x_3), \varphi(x_4)$$

线性可分



$$\varphi(x_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\varphi(x_2) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\varphi(x_3) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\varphi(x_4) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

➤ 支持向量机优化问题-线性不可分情况 (2)

□ 设:

$$\omega = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 6 \end{bmatrix} \quad b = 1$$

$$\omega^T \varphi(\triangle x_1) + b = 1 \geq 0$$

$$\omega^T \varphi(\triangle x_2) + b = 3 \geq 0$$

$$\omega^T \varphi(\circ x_3) + b = -1 < 0$$

$$\omega^T \varphi(\circ x_4) + b = -1 < 0$$

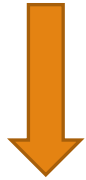
➡ 线性可分了!

➤ 支持向量机优化问题-线性不可分情况 (2)

- 在一个 M 维空间上随机取 N 个训练样本，随机的对每个训练样本赋予标签+1或-1
- 这些训练样本线性可分的概率为 $P(M)$

当 M 趋于无穷大时， $P(M)=1$

将训练样本由低维映射到高维

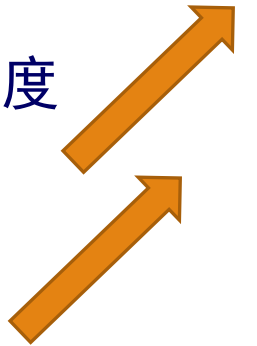


增大线性可分的概率

特征空间的维度 M

待估计参数 (ω, b) 的维度

整个算法模型的自由度



➤ 支持向量机优化问题-线性不可分情况 (2)

□ 假设 $\varphi(x)$ 已经构造好

$$\text{最小化: } \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^N \xi_i$$

$\varphi(x_i)$ 替换 x_i

限制条件: (1) $\delta_i \geq 0$, $i=1 \sim N$

$$(2) y_i [\omega^T \varphi(x_i) + b] \geq 1 - \xi_i , i=1 \sim N$$

ω_i 维度与 x_i 维度相同

ω_i 维度与 $\varphi(x_i)$ 维度相同

高维情况下优化问题的解法和低维情况完全类似：凸优化问题的求解

➤ 核函数

□ 核函数 K 和映射 $\varphi(x)$ 是一一对应的关系，核函数的形式不能随意的取



满足Mercer定理

两个 $\varphi(x)$ 内积的形式

□ Mercer定理

$K(X_1, X_2)$ 能写成 $\varphi(X_1)^T \varphi(X_2)$ 的充要条件

(1) $K(X_1, X_2) = K(X_2, X_1)$ (交换性)

(2) $\forall C_i (i=1 \sim N), \forall N$ 有 $\sum_{i=1}^N \sum_{j=1}^N C_i C_j K(X_i, X_j) \geq 0$ (半正定性)

➤ 常用核函数

□ (1) 多项式形式的核函数

$$K(x, y) = \{(x \cdot y) + 1\}^d$$

□ (2) 径向基函数形式的核函数

$$K(x, y) = e^{-\|x-y\|^2/2\sigma^2}$$

□ (3) Sigmoid函数形式的核函数

$$K(x, y) = \tanh(kx \cdot y - \delta)$$

□ (4) 点积形式的核函数

$$K(x, y) = x \cdot y$$

变成在已知 K ，而不知 $\varphi(x)$



求解支持向量机的优化问题

➤ SVM算法流程

□ 训练流程

输入 $\{(X_i, y_i)\}, i = 1 \sim N$

解优化问题：

最大化： $\theta(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(X_i, X_j)$

限制条件： ① $0 \leq \alpha_i \leq C$ ② $\sum_{i=1}^N \alpha_i y_i = 0$

算b, 找一个 $0 < \alpha_i < C$, $b = \frac{1 - y_i \sum_{j=1}^N \alpha_j K(X_i, X_j)}{y_i}$

➤ SVM算法流程

□ 测试流程

测试样本X

若 $W^T \varphi(X) + b \geq 0$ 则 $y = +1$

若 $W^T \varphi(X) + b \leq 0$ 则 $y = -1$

SVM中的泛化误差代表什么

- ☐ A 分类超平面与支持向量的距离
- ☐ B SVM模型的复杂程度
- ☒ C SVM对新的测试数据的分类精度
- ☐ D SVM中的误差阈值

提交

SVM算法的性能取决于

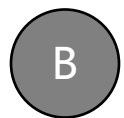
- ☒ A 核函数的选择
- ☒ B 核函数的参数
- ☒ C 软间隔
- ☐ D 算法复杂程度

提交

判断题：常见的核函数有sigmoid核函数、径向基核函数等



正确



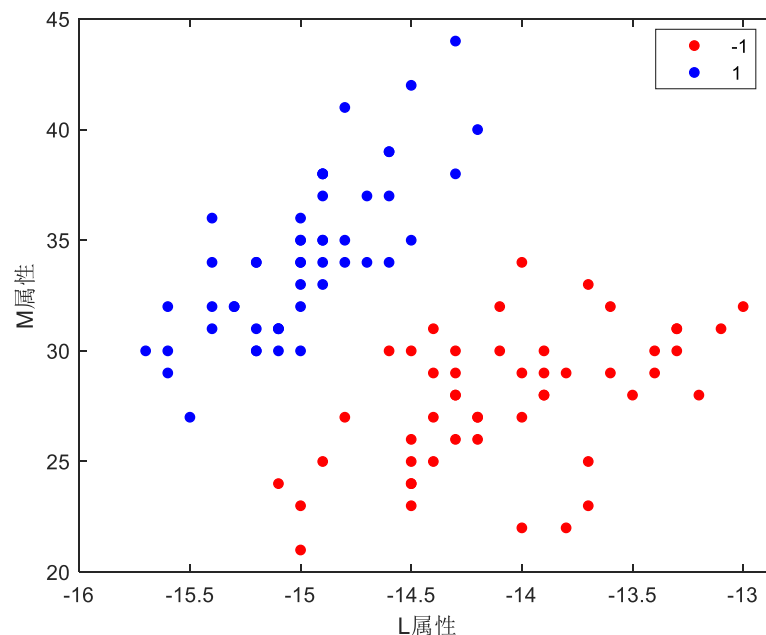
错误

提交

➤ 支持向量机实例

- 在生产现场，轴承的检测主要是通过时域信号完成的。在以“异响”为主要考虑指标的轴承质量检测中，可选择相关性较大的两个指标，它们分别是：加速度信号均方根值的分贝值L，超过某幅值的点数M。质量检测将产品分为合格P与不合格N两个类别，选择合格与不合格样本各50个作为SVM的训练样本。 $y_i \in \{-1, 1\}$ ，质量合格时记为1，否则记为-1。试求取SVM的分类超平面

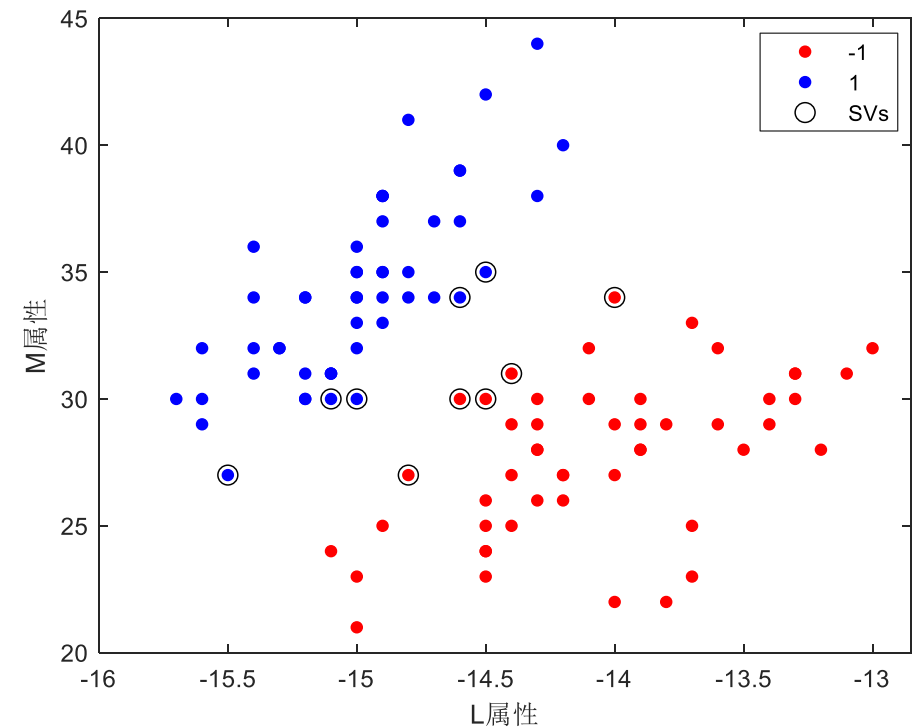
(1) 样本散点图



➤ 支持向量机实例

(2) 优化问题求解：通过SMO算法编程求得最优解，确定个支持向量机SVs属性值与其对应的拉格朗日乘数 a_i 如下：

序号	L	M	a	序号	L	M	a
1	-15.1	30	1	6	-14.8	27	0.5431
2	-14.6	34	1	7	-14.4	31	1
3	-15	30	1	8	-14.6	30	1
4	-14.5	35	0.5431	9	-14	34	1
5	-15.5	27	1	10	-14.4	30	1



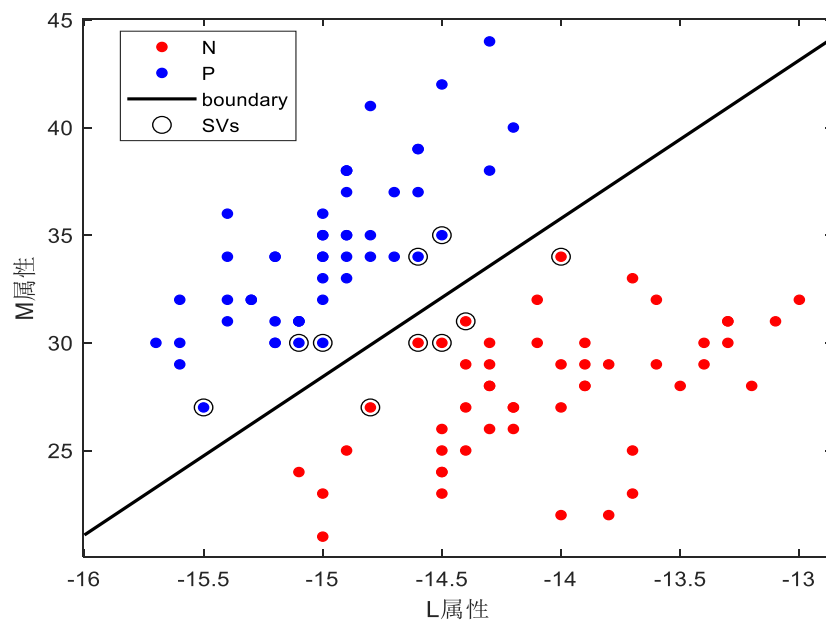
➤ 支持向量机实例

(3) 计算分类超平面

由 a_i 可确定权重向量 w 、 b

$$w = (w_1, w_2)^T = (-2.537, 0.345)^T$$

w 、 b 对应分类超平面如下图, 其数学表达为 $-0.2537x_1 + 0.345x_2 - 47.867 = 0$



- 分类分析的基本概念
- 决策树
- 贝叶斯分类
- 支持向量机
- **人工神经网络**
- 分类模型的评价与选择
- 组合分类技术
- 实例



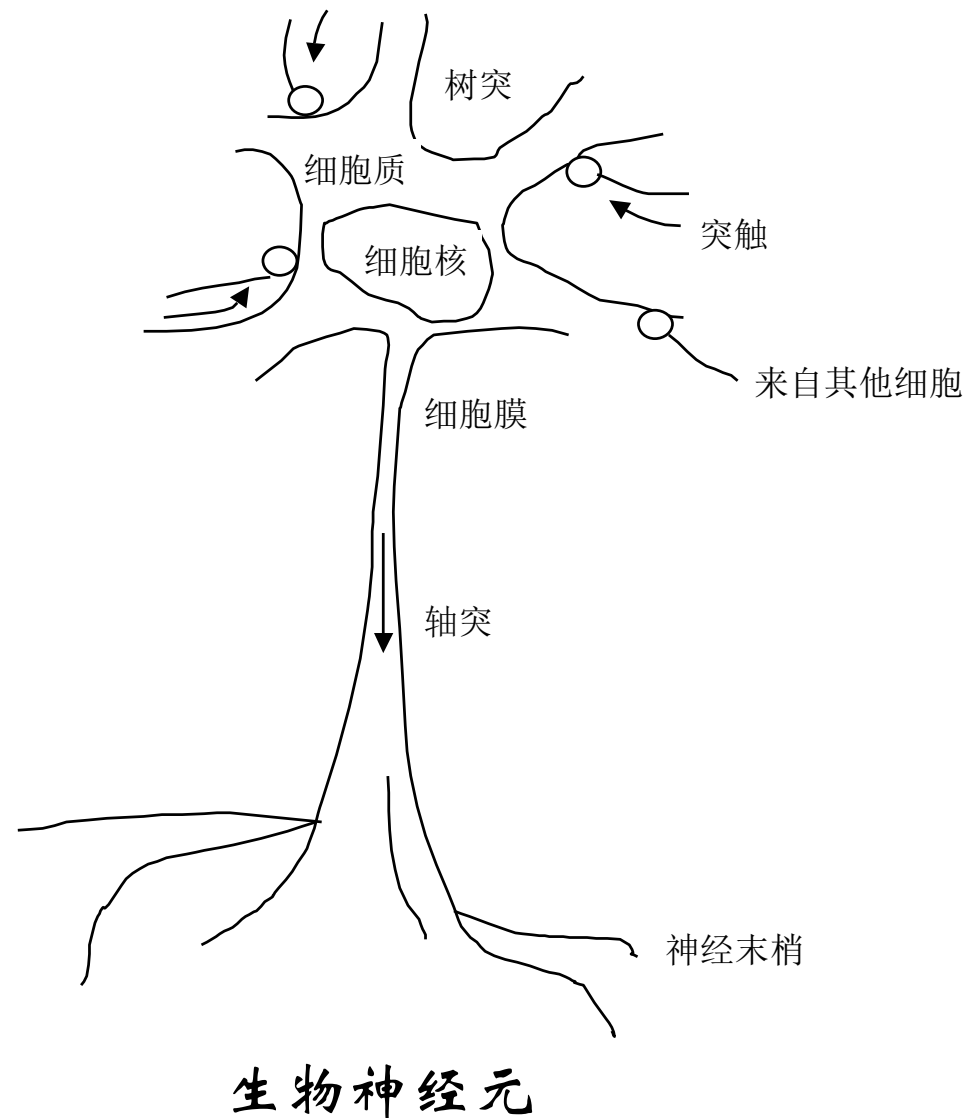
➤ 人工神经网络基本概念

□ 从生物神经元到人工神经网络

- 我们已知的最智能的东西就是我们的大脑
- 那么能否模仿人类的大脑，造出可以思考的机器呢
- 人脑智慧的物资基础，是人体的神经网络
- 生物神经系统的基本单元是神经细胞，也称神经元

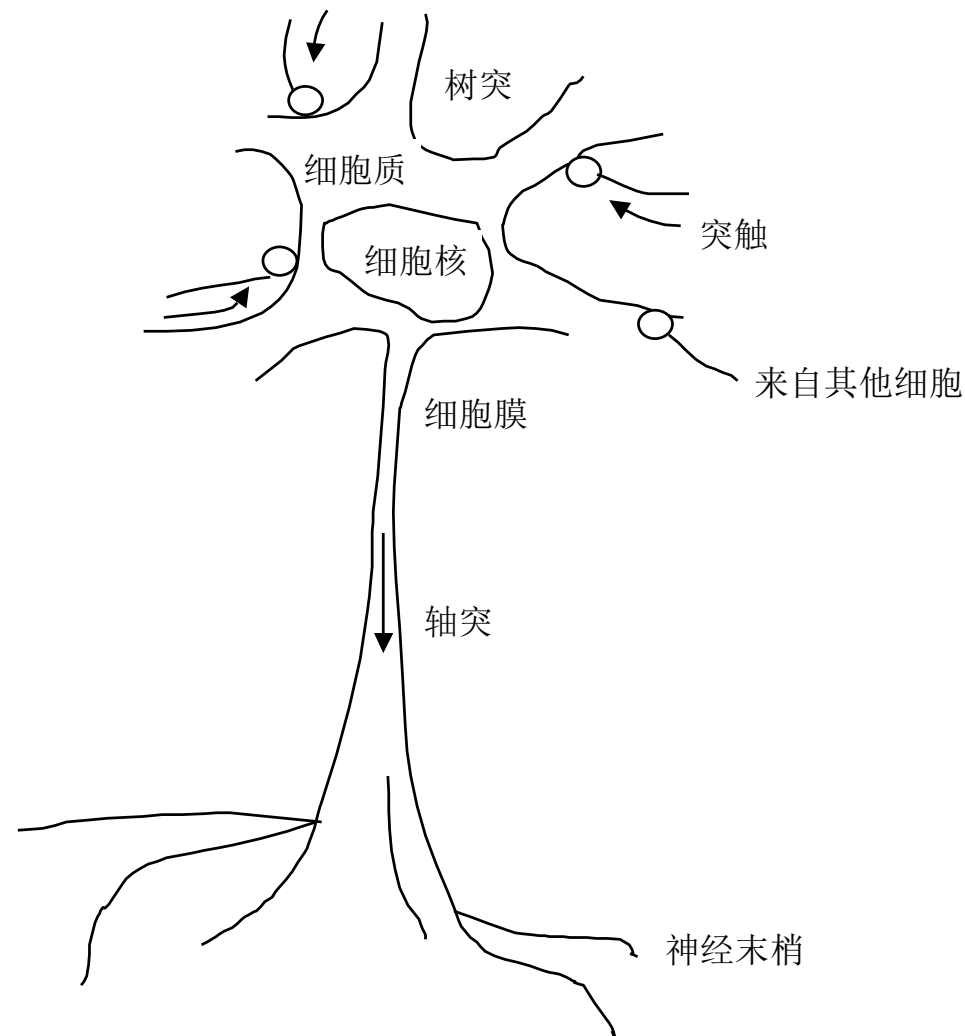
➤ 人工神经网络基本概念

- ❑ 神经元由细胞体、树突和轴突组成
- ❑ 细胞体是神经元的主体,它由细胞核、细胞质和细胞膜三部分构成
- ❑ 细胞突起是延伸出来的细长部分,每个突起又会伸出更多细分的触手,这些触手与其他神经元的触手相连接,形成神经网络
- ❑ 这些细胞突起又分为树突和轴突,树突是神经元的输入,轴突是神经元的输出,每个神经元只有一个轴突,轴突的末端是神经末梢



➤ 人工神经网络基本概念

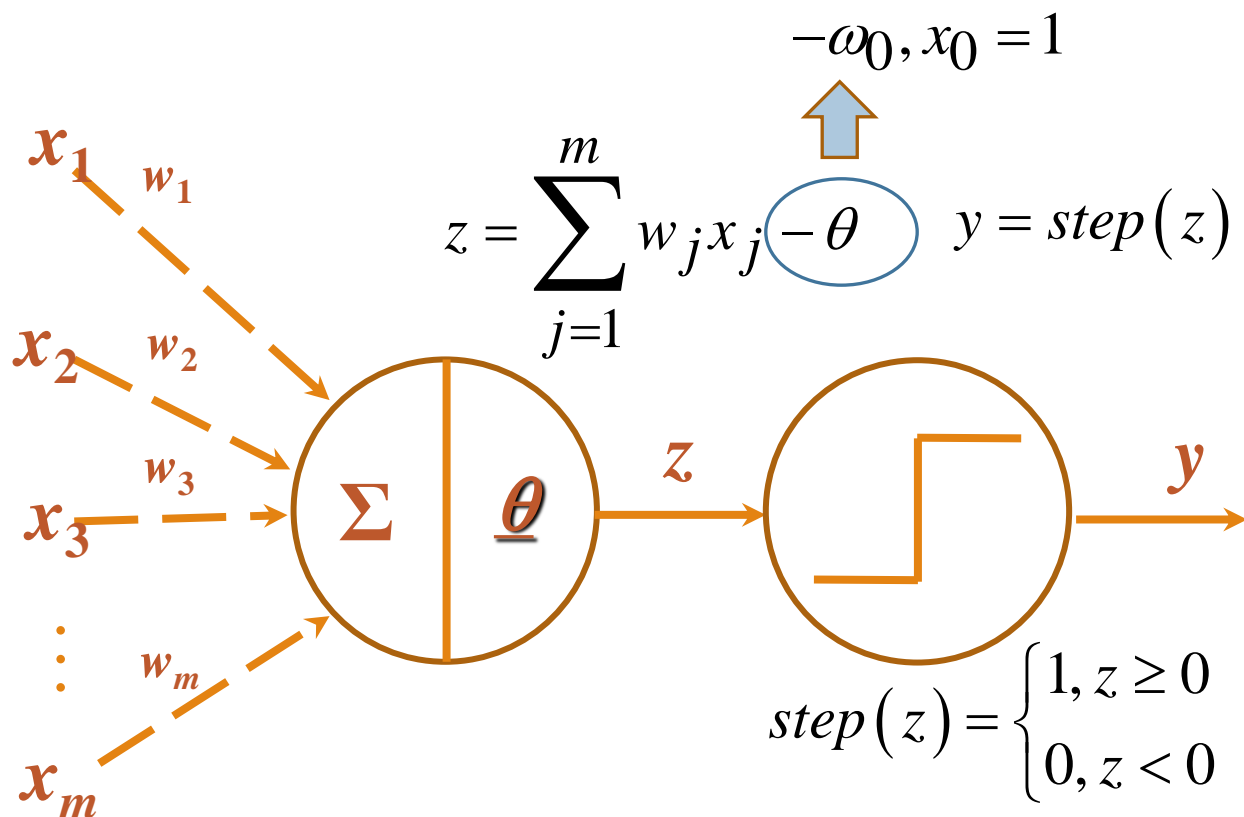
- ❑ 树突和轴突之间的连接点成为突触
- ❑ 通过突触，神经元就可以接收来自其他神经元的刺激，并且发送信号给其他神经元
- ❑ 生物神经元有两种状态，兴奋和抑制
- ❑ 神经元处于抑制状态时，轴突并不向外输出信号
- ❑ 当树突中输入的刺激累计达到某个程度，达到某个阈值时，神经元就会由抑制状态转为兴奋状态，同时通过轴突向其他神经元发送信号
- ❑ 这种能够传导的兴奋称为神经兴奋



生物神经元

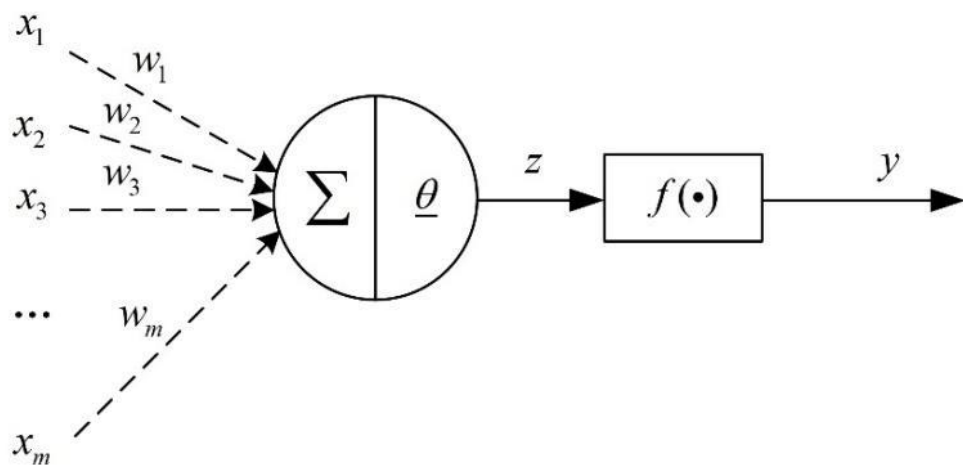
➤ 人工神经网络基本概念

- ❑ MP神经元的输入信号模拟神经元的树突
- ❑ 输入信号的来源不同，对神经元的影响也不同，因此给他们分配不同的权重
- ❑ 计算单元模拟神经元的细胞核，对接收到的输入信号加权求和之后与产生神经兴奋的阈值相比较
- ❑ 阶跃函数又称激活函数，模拟神经兴奋
- ❑ 输出y模拟生物神经元中的轴突，将神经元的输出信号传递给其他神经元



1943年，MP 神经元模型

➤ 人工神经网络拓扑



单个（MP）神经元的数理模型

□ 一个基本的神经元包含：

- ✓ 输入信号
- ✓ 求和单元
- ✓ 激活函数
- ✓ 输出信号

□ 一个神经元：

$$z = \sum_{i=1}^m w_i x_i - \theta$$

$$y = f(z)$$

输入信号： x_i

输入权值： w_i

加法器输入： z

激活函数： $f(\bullet)$

神经元输出： y

➤ 人工神经网络拓扑

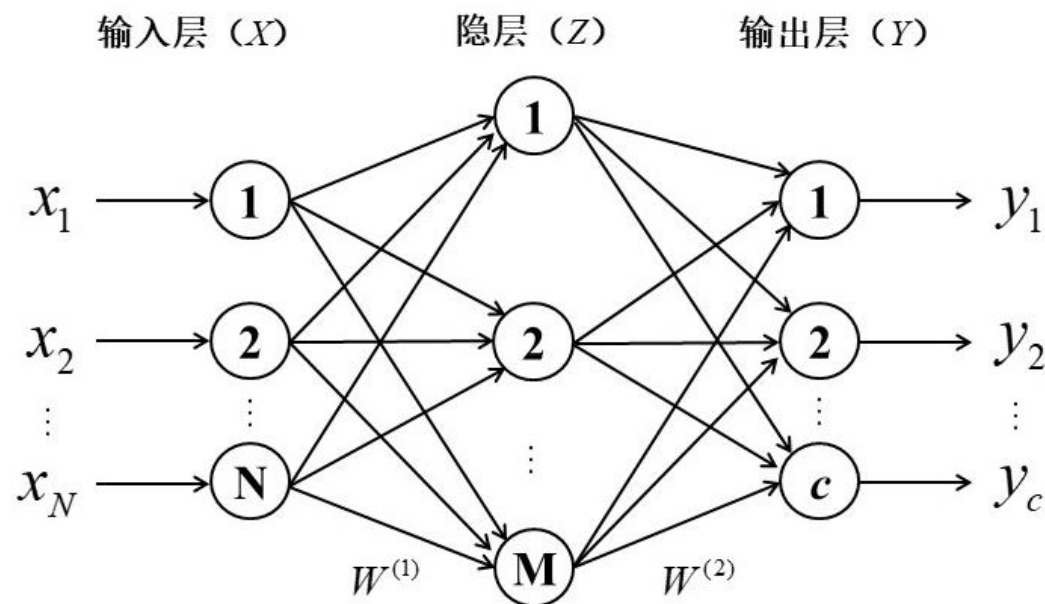
□ 多层网络结构

- ✓ 输入层：接收外部环境的输入信号
- ✓ 隐藏层：神经网络的内部处理单元
- ✓ 输出层：输出神经网络信号

□ 训练结束的评估——损失函数

✓ 绝对值损失函数
$$Loss = \sum_{i=1}^n |y_i - \hat{y}_i| = \sum_{i=1}^n |y_i - (wx_i + b)|$$

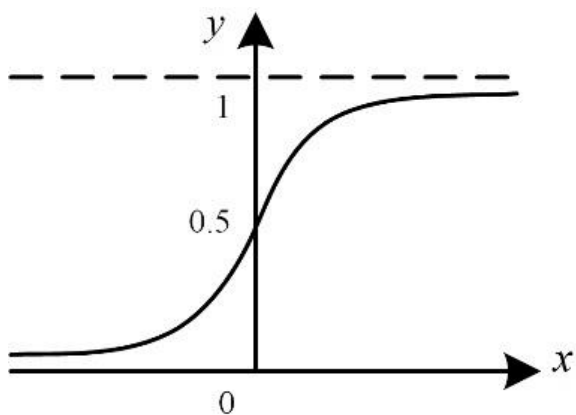
✓ 平方损失函数
$$Loss = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{2} \sum_{i=1}^n (y_i - (wx_i + b))^2$$



损失函数越小，说明所得结果越接近期望值

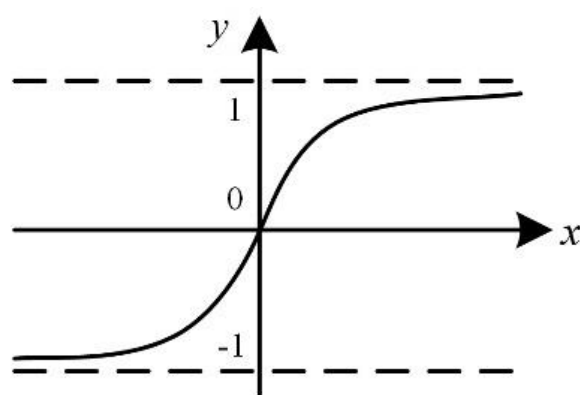
➤ 激活函数

□ 常用激活函数



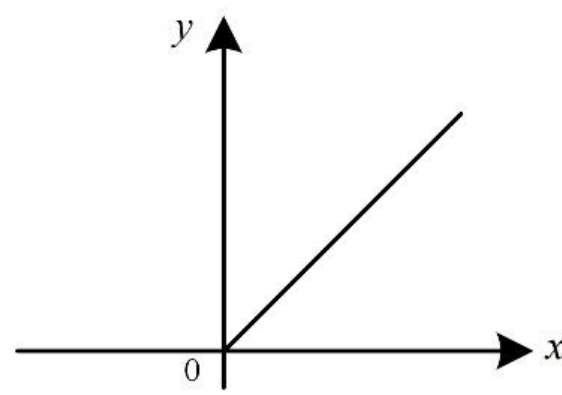
(a) Sigmoid函数

$$f(x) = \frac{1}{1 + e^{-x}}$$



(b) Tanh函数

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



(c) ReLU函数

$$f(x) = \max\{0, x\}$$

➤ 梯度下降法

- 梯度下降法求解线性回归模型的过程----函数求极值的过程

- 解析解：根据严格的推导和计算得到，是方程的精确解

能够在任意精度下满足方程

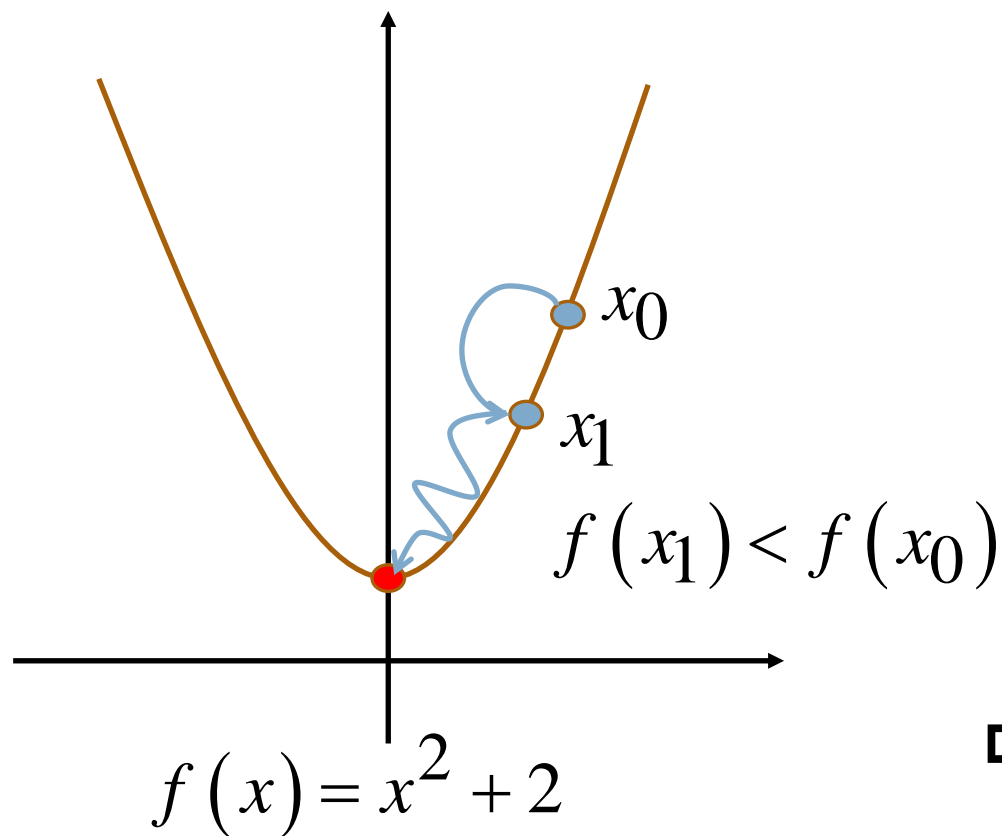
- 数值解：通过某种近似计算得到的解

能够在给定的精度条件下满足方程

➤ 梯度下降法

□ 梯度下降法：一种常用的求数值解的方法

□ 一元凸函数求极值 $x_0 = 3$ $f(3) = 11$ 取步长0.2, x 可能分别向2个方向移动



$$x_1 = 3.2 \quad f(3.2) = 12.24$$

$$x_1 = 2.8 \quad f(2.8) = 9.84$$

$$x_2 = 3 \quad f(3) = 11$$

$$x_2 = 2.6 \quad f(2.6) = 8.76$$

...

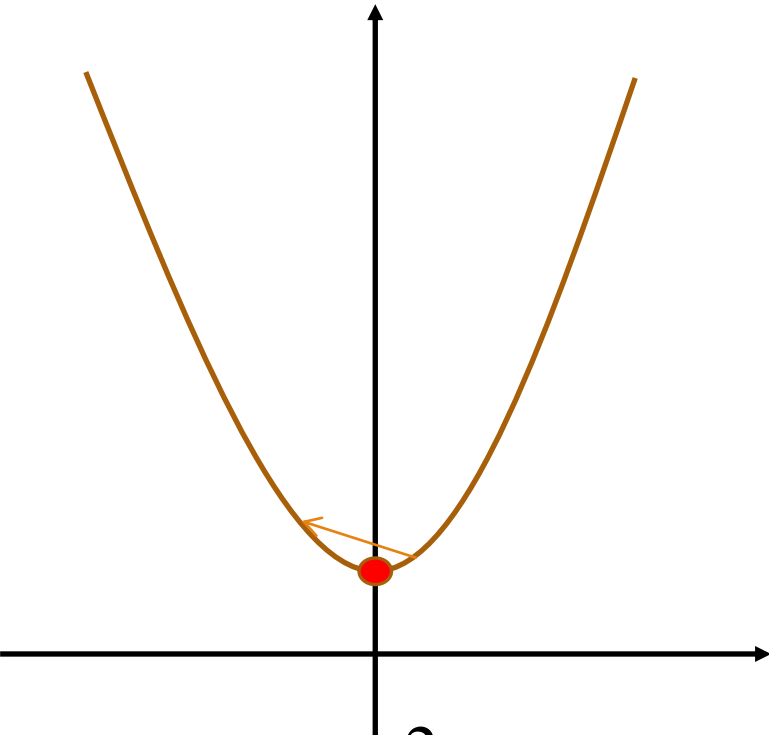
...

$$x_{15} = 0 \quad f(0) = 2$$

□ 步长取0.5, 经过6次迭代就可以达到极小值点

➤ 梯度下降法

步长取0.7



$f(x) = x^2 + 2$

发生了震荡

迭代次数	x_i	候选值	y	移动方向
0	3	2.3	7.29	✓
		3.7	15.69	
1	2.3	1.6	4.56	✓
		3	11	
2	1.6	0.9	2.81	✓
		2.3	7.29	
3	0.9	0.2	2.04	✓
		1.6	4.56	
4	0.2	-0.5	2.25	✓
		0.9	2.81	
5	-0.5	-1.2	3.44	
		0.2	2.04	✓
6	0.2	-0.5	2.25	✓
		0.9	2.81	
7	-0.5	-1.2	3.44	
		0.2	2.04	✓
8	0.2	-0.5	2.25	✓
		0.9	2.81	

➤ 人工神经网络原理

□ 步长对收敛过程的影响

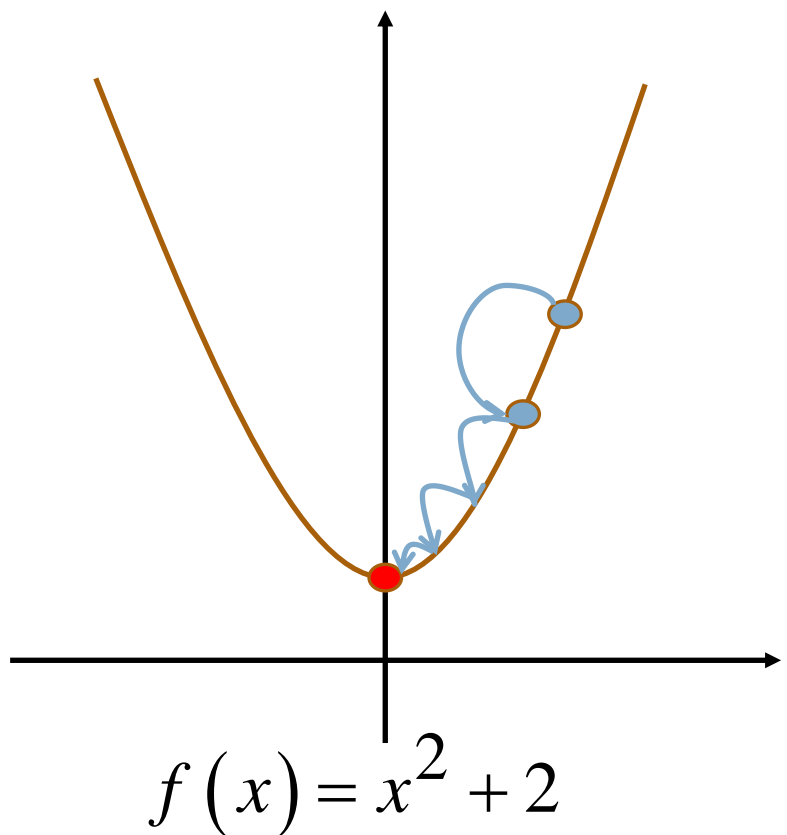
- 步长太小，迭代次数多，收敛慢
- 步长太大，引起震荡，可能无法收敛
- 自动调节步长

距离极值点比较远的时候，选用大一些的步长，避免收敛太慢

距离极值点比较近的时候，选用小一些的步长，避免发生震荡

想法不错，如何实现呢？斜率！

➤ 人工神经网络原理



□ 斜率大，步长大

□ 斜率小，步长小

□ 步长 = $\eta \frac{df(x)}{dx}$

$$x^{(k+1)} = x^{(k)} - \eta \frac{df(x)}{dx}$$

□ 二元凸函数求极值

$$z = f(x, y)$$

$$x^{(k+1)} = x^{(k)} - \eta \frac{\partial f(x, y)}{\partial x}$$

$$y^{(k+1)} = y^{(k)} - \eta \frac{\partial f(x, y)}{\partial y}$$

梯度

$$\frac{df(x)}{dx}$$

η : 学习率

✓ 自动调节步长

✓ 自动确定下一次更新的方向

✓ 保证收敛性

➤ 人工神经网络原理

- 导数：函数的变化率
- 偏导数： $\frac{\partial f(x, y)}{\partial x}$ 函数在x方向的变化率
- $\frac{\partial f(x, y)}{\partial y}$ 函数在y方向的变化率
- 方向导数：函数沿着某一个方向的变化率
- 梯度： $\text{grad} f(x, y) = \frac{\partial f}{\partial x} \vec{i} + \frac{\partial f}{\partial y} \vec{j}$
- 模为方向导数的最大值
- 方向为取得最大方向导数的方向

$$\nabla \begin{pmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{pmatrix}$$

函数在某个点的梯度就是指在这个点沿着这个方向的变化率最大！

$$z = f(x, y)$$

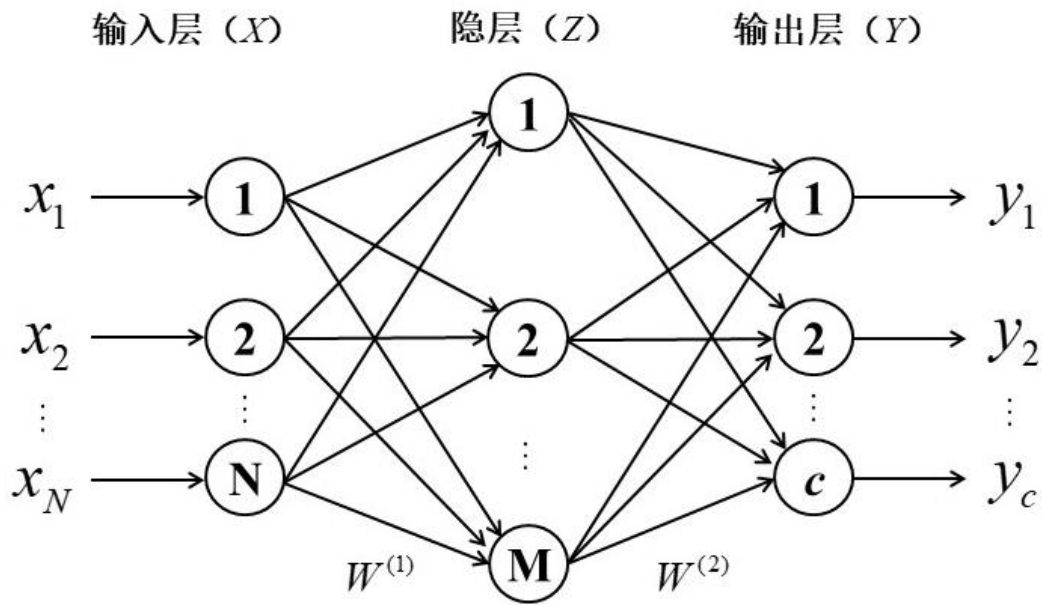
$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \eta \frac{\partial f(x, y)}{\partial x} \\ y^{(k+1)} &= y^{(k)} - \eta \frac{\partial f(x, y)}{\partial y} \end{aligned} \quad \text{梯度}$$

只要能将损失函数描述成凸函数，就可以采用梯度下降法，以最快的速度更新权值向量 w ，找到使损失函数达到最小值点的位置！

➤ 反向传播过程

□ 误差反向传播算法（BP）

n 个神经元 m 个神经元 c 个神经元



输入: $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$

输出: $\mathbf{y} = (y_1, y_2, \dots, y_m)^T$

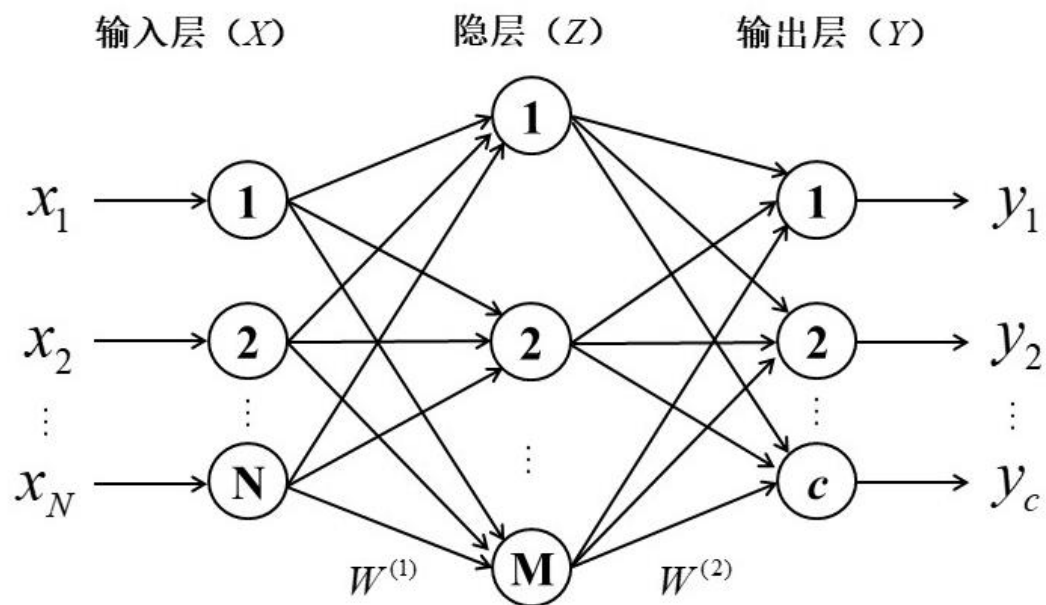
期望输出

权值: w_{ij} w_{jk}

激活函数: f 、 g

➤ 误差的反向传播算法

- 利用链式法则，反向传播损失函数的梯度信息，计算出损失函数对网络中所有模型参数的梯度



➤ 梯度下降算法

□ 误差反向传播算法（BP）

- 一个事实：若实值函数 $f(x)$ 在点 x_k 处可微且有定义，那么函数 $f(x)$ 在点 x_k 沿着负梯度（梯度的反方向）下降最快，沿着梯度下降方向求解最小值。

- 算法步骤

(1) 任取一点 w_i ，将 $f(w)$ 对 w 求导，计算梯度 $\frac{\partial f(w)}{\partial w} \Big|_{w=w_i}$

(2) 令 $k = 0$ 。若 $\frac{\partial f(w)}{\partial w} \Big|_{w=w_i} = 0$ ，退出运算；否则调整 w_i 的值： $w_i(t+1) = w_i(t) - \eta \frac{\partial f(w)}{\partial w} \Big|_{w=w_i}$

学习率：控制算法每一轮迭代的更新步长

➤ BP迭代过程

□ 步骤1：设置模型参数初始值

- 确定网络的初始权值与阈值，初始参数的设置直接影响着神经网络的学习性能。一般设置在 $[-1, 1]$ 之间。

□ 步骤2：计算正向传播过程中各节点的输出，包括隐层各节点和输出层各节点

- 隐藏层第 j 个神经元的输出： $z_j = f(\sum_{i=0}^n w_{ij}x_i - \theta_j)$
- 输出层第 k 个神经元的输出： $y_k = g(\sum_{j=0}^h w_{jk}z_j - \gamma_k)$



θ_j γ_k 可看作固定输入为-1.0的“哑结点”所对应的连接权重

$$z_j = f(\sum_{i=0}^n w_{ij}x_i) \quad y_k = g(\sum_{j=0}^h w_{jk}z_j)$$

➤ BP迭代过程

□ 步骤3：计算输出误差

- 网络输出与目标输出的均方误差为 $Loss = \frac{1}{2} \sum_{k=1}^m (y_k - \hat{y}_k)^2$


误差会逐渐减小

□ 步骤4：误差反向传播，调整权值

- 按梯度下降算法调整权值，使误差减小。每次权值的调整为 $\Delta w_{pq} = -\eta \frac{\partial Loss}{\partial w_{pq}}$
- BP神经网络的调整顺序

(1) 先调整隐藏层到输出层的权值

$$\frac{\partial Loss}{\partial w_{jk}} = \frac{\partial Loss}{\partial y_k} \frac{\partial y_k}{\partial z_k} \frac{\partial z_k}{\partial w_{jk}}$$

 v_k : 输出层第 k 个神经元的输入

➤ BP迭代过程

■ BP神经网络的调整顺序

✓ 隐藏层到输出层的权值调整迭代公式为 $w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}(t)$

(2) 再调整输入层到隐藏层的权值，同理

$$\frac{\partial Loss}{\partial w_{ij}} = \frac{\partial Loss}{\partial y_j} \frac{\partial y_j}{\partial z_j} \frac{\partial z_j}{\partial x_j} \frac{\partial x_j}{\partial w_{ij}}$$

u_k : 隐藏层第j个神经元的输入

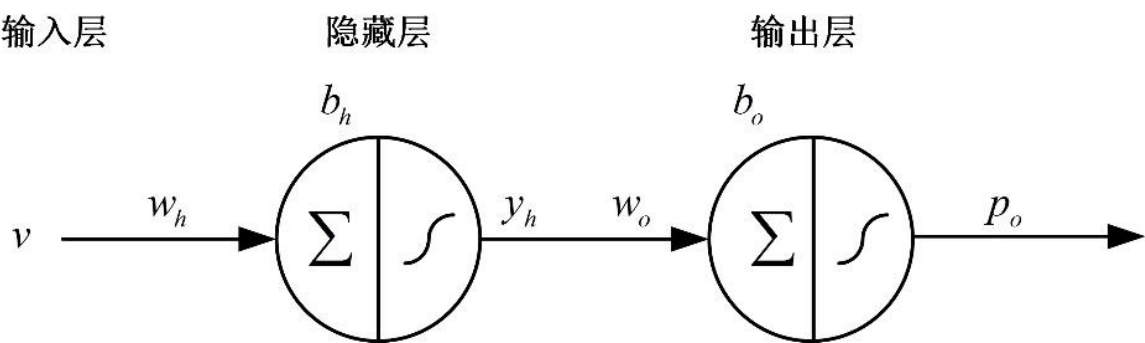


✓ 从输入层到隐藏层的权值调整迭代公式为 $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$

□ 步骤5: 判断是否满足终止条件，满足则停止，否则重复步骤2-4

➤ 人工神经网络实例

❑ 风电功率预测技术是指对未来一段时间内风电场所能输出的功率大小进行预测，以便安排调度计划。可利用BP神经网络算法对风电功率进行预测，假设有三层BP神经网络结构如下图所示，使用Sigmoid函数作为网络的激活函数。神经网络的初始赋值如下表，计算该网络第一次迭代的过程。



输入：风速 v

输出：功率 p_v

输入层到隐藏层的权值和阈值： $w_h \quad \theta_h$

输入层到隐藏层的权值和阈值： $w_o \quad \theta_o$

v	p	w_h	w_o	θ_h	θ_o	η
0.7	1	0.3	0.2	-0.1	-0.2	0.5

➤ 人工神经网络实例

□ 步骤1：信号前向传播，计算各节点输出

(2) 输入层到隐藏层

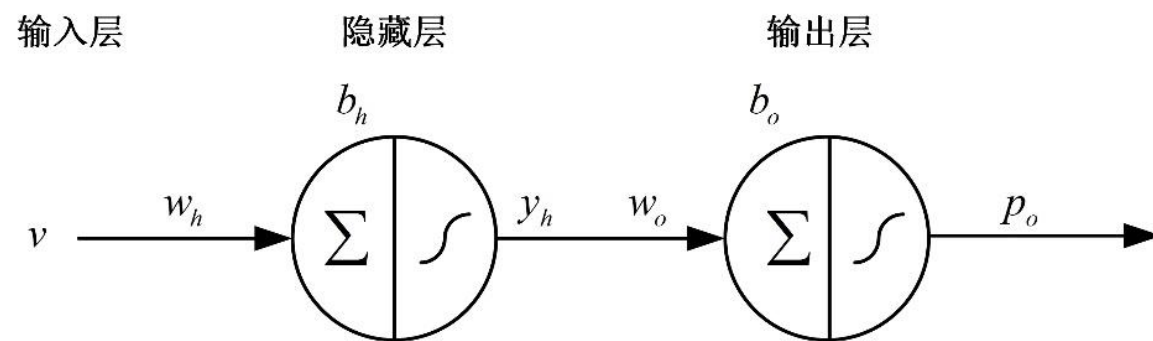
■ 隐藏层神经元的输入： $z_h = w_h v - \theta_h = 0.3 \times 0.7 + 0.1 = 0.310$

■ 隐藏层神经元输出： $y_h = f(z_h) = \frac{1}{1 + e^{-z_h}} = \frac{1}{1 + e^{-0.31}} = 0.577$

(2) 隐藏层到输出层

■ 输出层神经元的输入： $z_o = w_o y_h - \theta_o = 0.2 \times 0.577 + 0.2 = 0.315$

■ 隐藏层神经元输出： $p_o = f(z_o) = \frac{1}{1 + e^{-z_o}} = \frac{1}{1 + e^{-0.315}} = 0.578$



➤ 人工神经网络实例

□ 步骤2：计算输出误差

$$Loss = \frac{1}{2}(p - p_o)^2 = \frac{1}{2}(1 - 0.578)^2 = 0.089$$

□ 步骤3：误差反向传播过程，更新参数

(1) 计算输出层节点的误差率

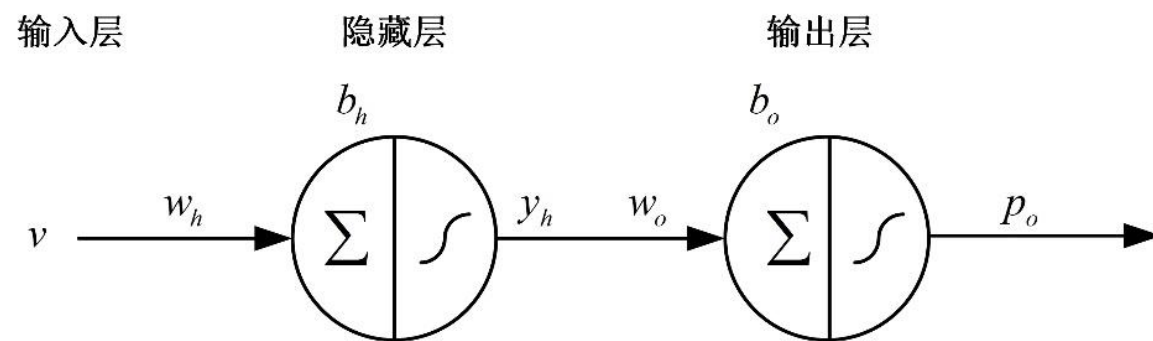
$$\frac{\partial Loss}{\partial p_o} = -(p - p_o) = -(1 - 0.578) = -0.422$$

$$\frac{\partial p_o}{\partial z_o} = p_o(1 - p_o) = 0.578 \times (1 - 0.578) = 0.244$$

$$\frac{\partial z_o}{\partial w_o} = y_h = 0.577$$



$$\frac{\partial Loss}{\partial w_o} = \frac{\partial Loss}{\partial p_o} \frac{\partial p_o}{\partial z_o} \frac{\partial z_o}{\partial w_o} = -0.422 \times 0.244 \times 0.577 = -0.059$$



➤ 人工神经网络实例

□ 步骤3：误差反向传播过程，更新参数

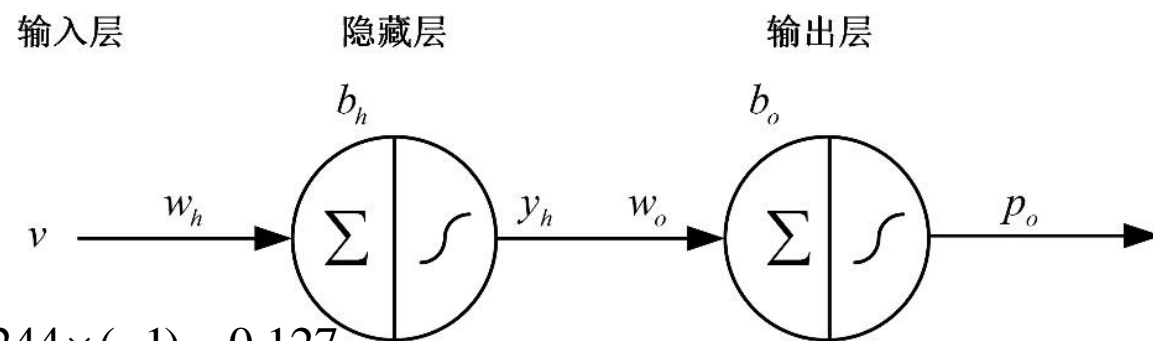
$$\frac{\partial z_o}{\partial \theta_o} = -1 \quad \longrightarrow \quad \frac{\partial Loss}{\partial \theta_o} = \frac{\partial Loss}{\partial p_o} \frac{\partial p_o}{\partial z_o} \frac{\partial z_o}{\partial \theta_o} = -0.422 \times 0.244 \times (-1) = 0.127$$

(2) 计算隐藏层节点的误差率

$$\frac{\partial z_o}{\partial y_h} = w_o = 0.2$$

$$\frac{\partial y_h}{\partial z_h} = y_h(1 - y_h) = 0.577 \times (1 - 0.577) = 0.244 \quad \longrightarrow \quad \frac{\partial Loss}{\partial w_h} = \frac{\partial Loss}{\partial p_o} \frac{\partial p_o}{\partial z_o} \frac{\partial z_o}{\partial y_h} \frac{\partial y_h}{\partial z_h} \frac{\partial z_h}{\partial w_h} = -0.00317$$

$$\frac{\partial z_h}{\partial w_h} = v = 0.7$$



➤ 人工神经网络实例

□ 步骤3：误差反向传播过程，更新参数

(2) 计算隐藏层节点的误差率

$$\frac{\partial z_h}{\partial \theta_h} = -1 \quad \longrightarrow \quad \frac{\partial Loss}{\partial \theta_h} = \frac{\partial Loss}{\partial p_o} \frac{\partial p_o}{\partial z_o} \frac{\partial z_o}{\partial y_h} \frac{\partial y_h}{\partial z_h} \frac{\partial z_h}{\partial \theta_h} = -0.422 \times 0.244 \times 0.2 \times 0.244 \times (-1) = 0.00502$$

(3) 更新各节点权值与阈值：利用梯度下降法更新

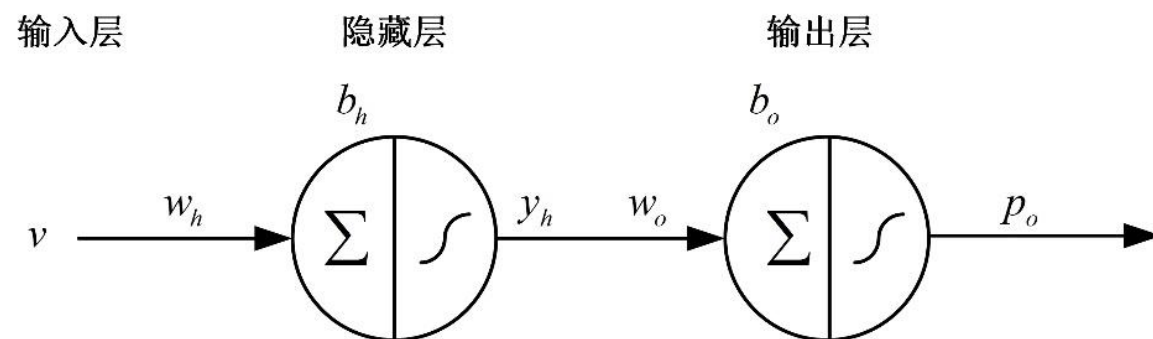
$$w_o^{(1)} = w_o^{(0)} - \eta \frac{\partial Loss}{\partial w_o} = 0.230$$

$$w_h^{(1)} = w_h^{(0)} - \eta \frac{\partial Loss}{\partial w_h} = 0.30158$$

$$\theta_o^{(1)} = \theta_o^{(0)} - \eta \frac{\partial Loss}{\partial \theta_o} = -0.264$$

$$\theta_h^{(1)} = \theta_h^{(0)} - \eta \frac{\partial Loss}{\partial \theta_h} = -0.10251$$

随着训练次数的增加，误差会越来越小



➤ BP神经网络

优点：

- (1) 具有高度的自组织和学习能力、良好的鲁棒性和容错性
- (2) 可以实现大规模数据的并行处理

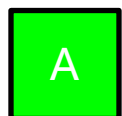
缺点：

收敛速度慢，并且容易陷入局部最优解

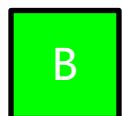
解决方法：

- (1) 在局部极小值方面，优化初值选取和改变网络结构
- (2) 在收敛速度方面，采用自适应学习率和引入陡度因子等方法加快训练速度

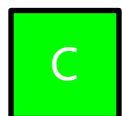
下列哪些是神经网络中的超参数



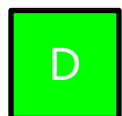
学习率



隐层神经元个数



迭代次数



神经网络层数

提交

- 分类分析的基本概念
- 决策树
- 贝叶斯分类
- 支持向量机
- 人工神经网络
- 分类模型的评价与选择
- 组合分类技术
- 实例



➤ 分类器评价指标: 混淆矩阵

□ 混淆矩阵

混淆矩阵		实际类别	
		正例	反例
预测类别	正例	TP	FP
	反例	FN	TN

混淆矩阵展示了分类模型检验的结果，但不够直观

- 1) 真正例 (TP): 表示预测结果为正, 实际为正的样本数。
- 2) 假正例 (FP): 表示预测结果为正, 实际为负的样本数。
- 3) 真反例 (TN): 表示预测结果为负, 实际为负的样本数。
- 4) 假反例 (FN): 表示预测结果为负, 实际为正的样本数。

➤ 分类器评价指标：样本类别分布相对平衡

□ 准确率（Accuracy）：分类器预测正确的样本数占总样本数的比重

$$Accuracy = \frac{TP + TN}{P + N}$$

□ 错误率（Error Rate）：分类器预测错误的样本数占总样本数的比重

$$Error\ rate = \frac{FP + FN}{P + N}$$

➤ 分类器评价指标: 数据集实际类别不平衡

- 测试集包含97个正常工件和3个故障工件

混淆矩阵		实际类别	
		正常	故障
预测类别	正常	97	2
	故障	0	1

$$Accuracy = 98\%$$

准确率如此高，分类效果就真的好吗？

□ 对故障识别能力不佳

➤ 分类器评价指标：样本类别不平衡

□ 灵敏性（Sensitive）：

- 灵敏性用于评估分类器正确地识别正样本的情况

$$Sensitivity = \frac{TP}{P}$$

□ 特效性（Specificity）

- 特效性用于评估分类器正确地识别负样本的情况

$$Specificity = \frac{TN}{N}$$

➤ 分类器评价指标: 准确性, 识别率

混淆矩阵		实际类别	
		正常	故障
预测类别	正常	97	2
	故障	0	1

$$Sensitivity = 100\%$$

$$Specificity = 33\%$$

如何衡量正样本的能力?

➤ 分类器评价指标：样本类别不平衡

□ 精度（Precision）：分类器预测类别为正的样本中，实际类别为正的样本的比重

$$Precision = \frac{TP}{TP + FP}$$

□ 召回率（Recall）：实际类别为正的样本中，被分类器预测为正的样本的比重

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{P}$$

□ F 度量——精度和召回率的调和均值

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

精度和召回率是相互矛盾的

➤ 示例

□ 使用相同的混淆矩阵，计算刚刚引入的度量

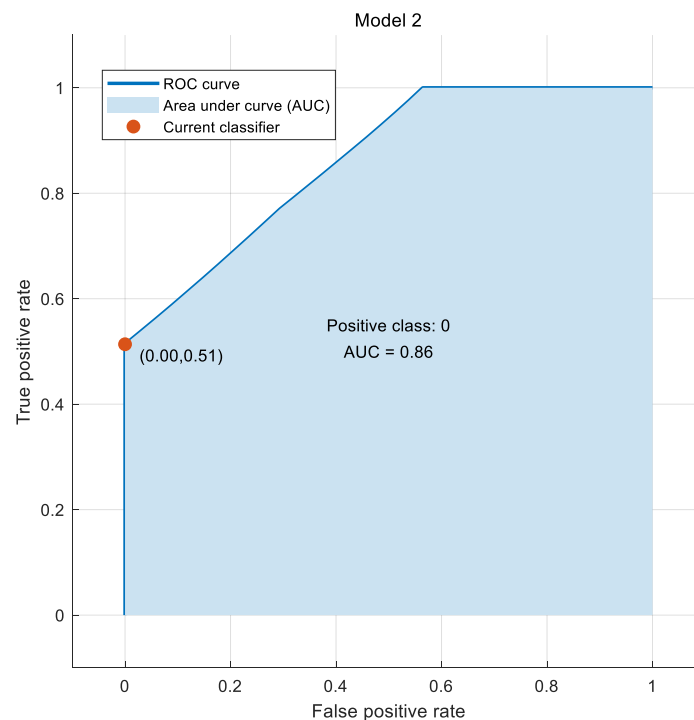
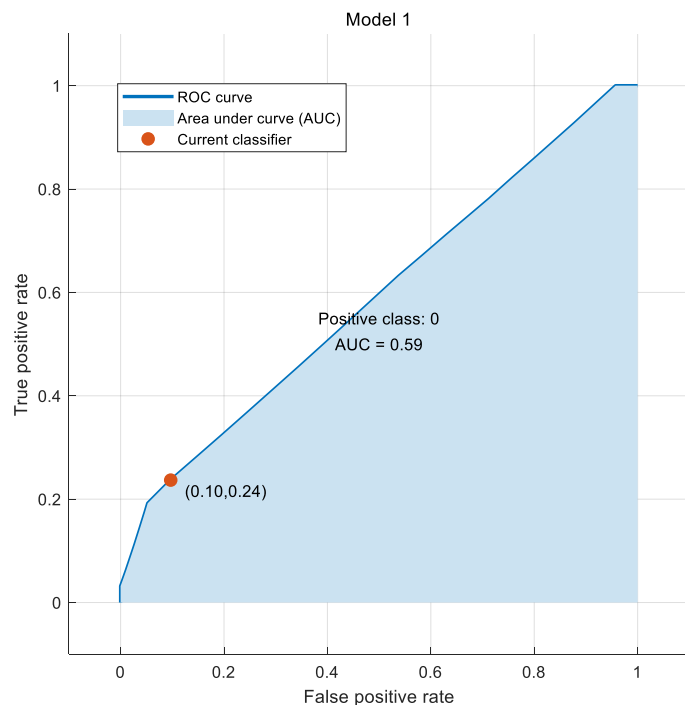
Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	90	210	300	30.00 (<i>sensitivity</i>)
cancer = no	140	9560	9700	98.56 (<i>specificity</i>)
Total	230	9770	10000	96.50 (<i>accuracy</i>)

- $\text{Sensitivity} = \text{TP}/\text{P} = 90/300 = 30\%$
- $\text{Specificity} = \text{TN}/\text{N} = 9560/9700 = 98.56\%$
- $\text{Accuracy} = (\text{TP} + \text{TN})/\text{All} = (90+9560)/10000 = 96.50\%$
- $\text{Error rate} = (\text{FP} + \text{FN})/\text{All} = (140 + 210)/10000 = 3.50\%$
- $\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) = 90/(90 + 140) = 90/230 = 39.13\%$
- $\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) = 90/(90 + 210) = 90/300 = 30.00\%$
- $F = 2 \text{ P} \times \text{R} / (\text{P} + \text{R}) = 2 \times 39.13\% \times 30.00\% / (39.13\% + 30\%) = 33.96\%$

➤ 分类器模型评估

□ ROC曲线

- 衡量“二分类问题”机器学习算法性能（泛化能力），用于分类模型的可视化比较
- ROC曲线下面积(AUC: area under curve)是衡量模型准确性的指标
- 越接近对角线(即面积越接近0.5)，模型的精度越低



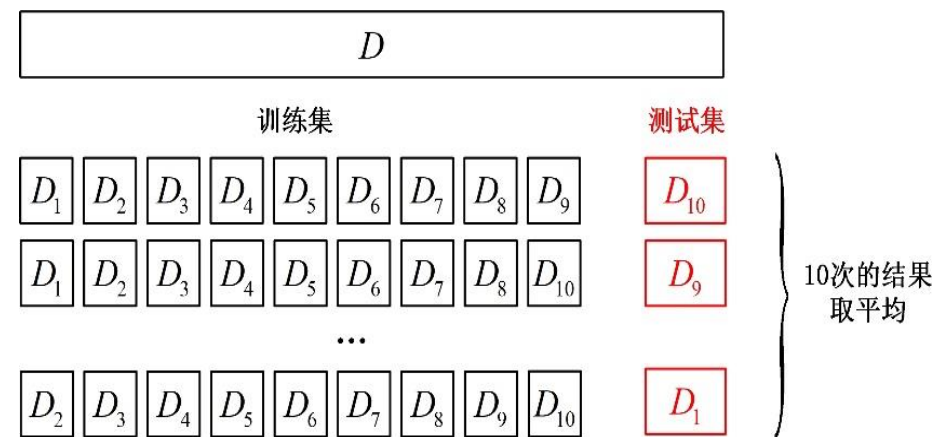
➤ 分类器评价方法

□ 保持法(Hold out)

- 将给定的数据随机划分为两个独立的数据集
- 模型构建的训练集(如2/3);用于准确度估计的测试集(例如, 1/3)
- 多次划分所得的性能指标取平均值

□ 交叉验证(k-fold)

- 将数据随机分成k个互斥的子集, 每个子集的大小近似相等
- 第i次迭代时, 使用 D_i 作为测试集, 其他作为训练集
- 相当于同时训练k个模型取均值, 也相当于扩充了数据集



➤ 分类器的评估

□ 自助法

- 适用于小数据集
- 对给定的训练元组进行置换均匀抽样
- 每当一个元组被选中时，它再次被选中并重新添加到训练集的可能性是相等的

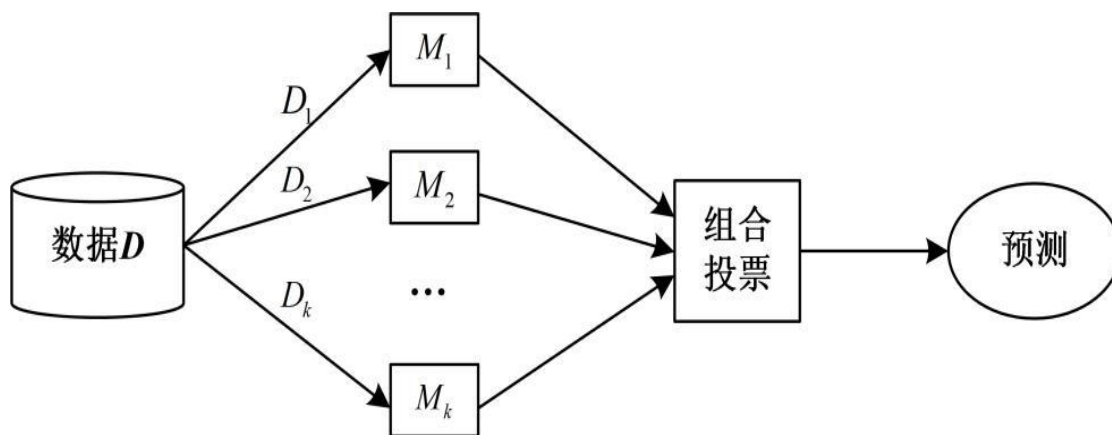
- 分类分析的基本概念
- 决策树
- 贝叶斯分类
- 支持向量机
- 人工神经网络
- 分类模型的评价与选择
- **组合分类技术**
- 实例



➤ 组合分类技术

□ 组合分类器：由多个分类器组合而成的复合模型

- 优势：能提高分类的准确性，常可以获得比单一学习器显著优越的泛化性能



组合分类的基本原理

■ 常用方法：

✓ Bagging算法

✓ Boosting算法

➤ 组合分类技术

□ Bagging算法

- 基于自助法，用训练集的子集和并行学习的模型来训练每个模型。通常对分类任务使用简单投票法，即每个基分类器使用相同的权重投票，最终将票数最高的类赋予样本数据。如：随机森林。

□ Boosting算法

- 先从初始训练集训练出一个基分类器 M_i ，更新样本权重，使分类器 M_i 更加关注被 M_i 上一个分类器错误分类的训练样本。反复更新，直到基分类器的数目达到指定值，最终将这些分类器进行加权集合。如：Adaboost

- 分类分析的基本概念
- 决策树
- 贝叶斯分类
- 支持向量机
- 人工神经网络
- 分类模型的评价与选择
- 组合分类技术
- 实例



➤ 基于分类方法的变压器故障诊断

□ 变压器内部故障及诊断

- 电力变压器是整个电力设备中最关键的组成部分，其运行可靠性直接影响电力系统的安全运行。
- 油中溶解气体的成分和含量在一定程度上反映出变压器的内部故障，可作为识别变压器故障类型的特征量。

□ 故障类型及特征属性

- 电力变压器的内部故障一般有过热故障与放电故障
- 用于分析的主要气体种类为：

氢气 H_2 、甲烷 CH_4 、乙烷 C_2H_6 、乙烯 C_2H_4 、乙炔 C_2H_2

➤ 基于分类方法的变压器故障诊断

▣ 原始数据集

■ 现对一批电力变压器样本的油中气体含量进行检测，得到70组实验数据：

序号	氢气	甲烷	乙烷	乙烯	乙炔	故障类别
1	44.3	17.3	3.6	23.3	10.4	放电故障
2	673.6	423.5	77.5	988.9	344.4	放电故障
3	550.0	53	34.0	20.0	0	放电故障
4	13.6	5.3	8.2	29.0	2.1	正常
5	4670.0	3500	2120.0	5040	2560	放电故障
6	26.6	22.7	22.5	109	0	过热故障
.....
69	189.0	157.0	17.0	62.0	7.4	放电故障
70	27.0	90.0	42.0	63.0	0.2	过热故障

➤ 基于分类方法的变压器故障诊断

□ 训练集划分

- 挑选前50组样本作为训练集，后20组样本作为测试集

□ 数据预处理

- 查看数据集类别分布
 各类别分布相对均衡
- 缺失值处理
- 数据标准化

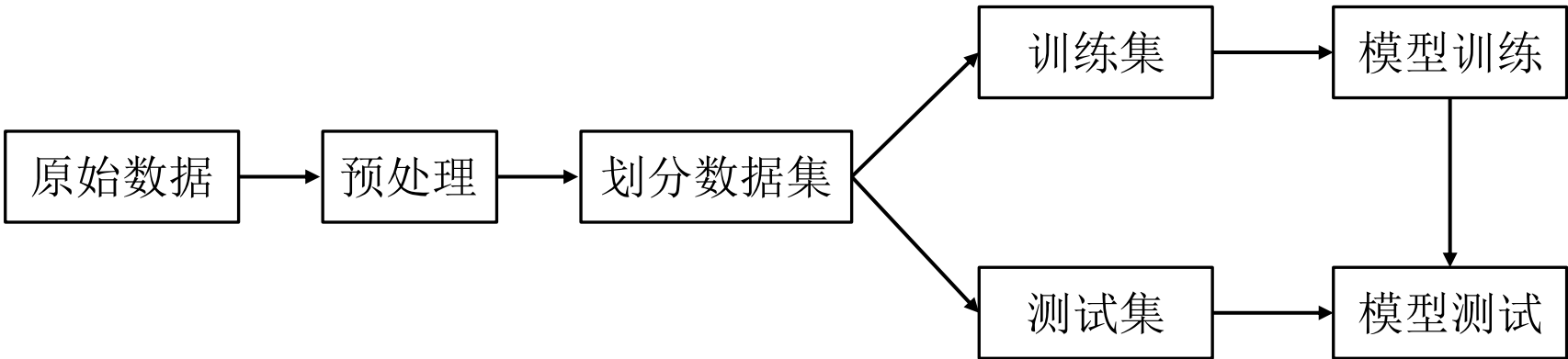
类型	样本数量		
	训练集	测试集	数据集
放电故障	17	6	23
过热故障	18	7	25
正常	15	7	22
合计	50	20	70

为了降低各特征维度不一致（取值大小范围不一致）给分类模型造成的影响，通过标准差归一化方法对各个特征进行标准化处理。

➤ 基于分类方法的变压器故障诊断

□ 分类方法与框架

- 本例分别采用决策树模型与BP神经网络进行模型训练与分类

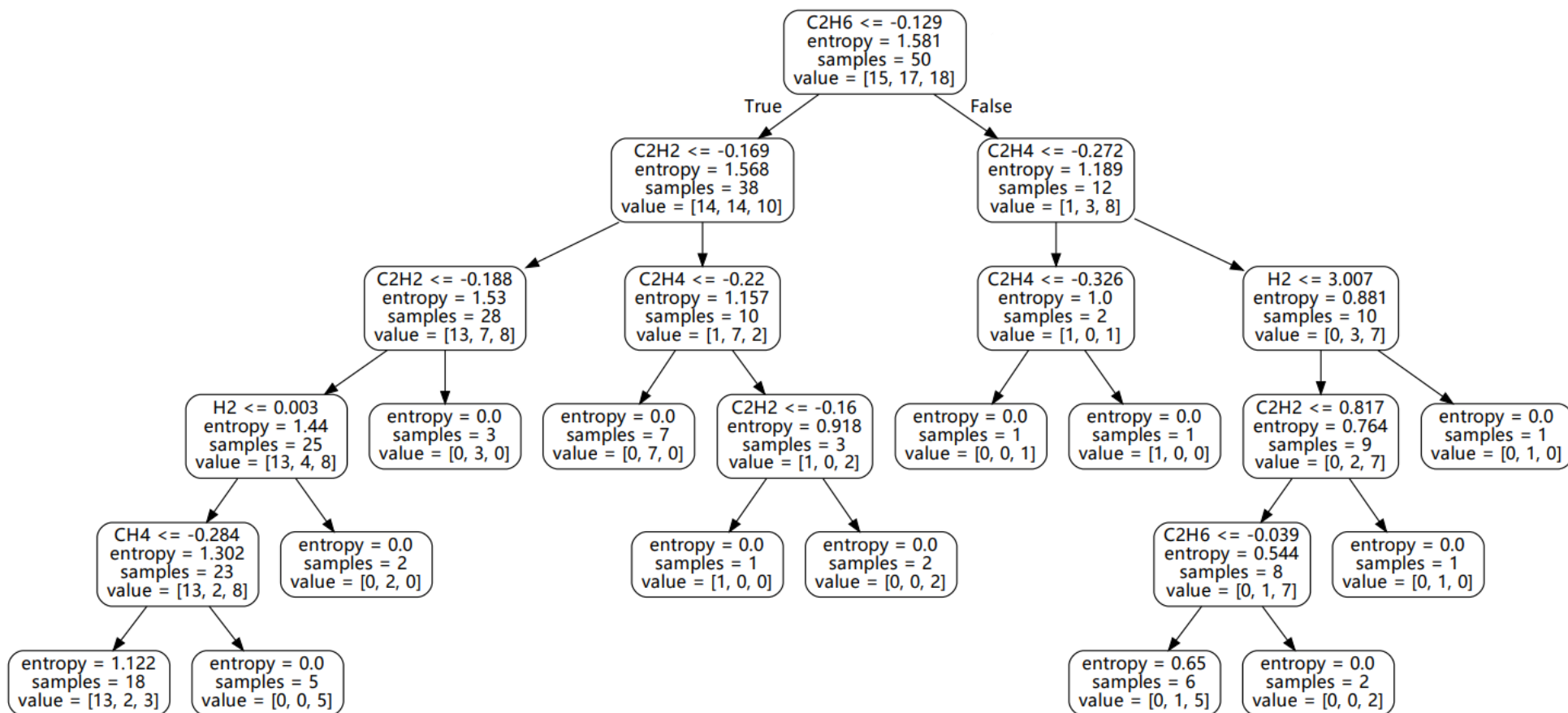


- 采用网格搜索算法对以下模型参数进行寻优

决策树				BP神经网络	
不纯度计算指标	决策树最佳切分点	最大深度	分裂最小样本数	隐层层数及神经元个数	迭代次数

➤ 基于分类方法的变压器故障诊断

□ 决策树分类模型



➤ 基于分类方法的变压器故障诊断

□ 分类结果对比

■ 混淆矩阵

决策树		实际类别		
		正常	放电故障	过热故障
预测类别	正常	5	0	2
	放电故障	0	6	0
	过热故障	1	1	5

BP神经网络		实际类别		
		正常	放电故障	过热故障
预测类别	正常	7	0	0
	放电故障	0	6	0
	过热故障	1	2	4

BP神经网络误分类的样本数略少于决策树。

➤ 基于分类方法的变压器故障诊断

□ 分类结果对比

■ 准确率、精度、召回率以及F度量

指标	决策树	BP神经网络
准确率	0.8	0.85
精度	0.7988	0.8813
召回率	0.8	0.85
F1度量	0.7962	0.8384

就本例而言，以网格搜索方法遍历搜索最优的参数建立分类模型，BP神经网络模型的分分类效果优于决策树。

□ 基本概念

- 分类分析的概念、基本思路和一般过程

□ 决策树

- 决策树理论、属性选择度量及剪枝操作

□ 贝叶斯分类

- 策树理论、属性选择度量及剪枝操作
- ID3算法, C4.5算法, CART算法

□ 支持向量机

- 支持向量机的基本理论、优化、求解

□ 人工神经网络

- 激活函数、损失函数
- BP神经网络

□ 分类模型评价与选择

- 评价指标
- 划分测试集的方法

□ 组合分类技术

- 组合分类方法的一般流程

- Jiawei Han and Michel Kamber. Data Mining: Concepts and Techniques[M]. *Morgan Kaufmann Publishers*, 2001, 70-95.
- Min Ding, Hao Zhou, Hua Xie, Min Wu*, Kangzhi Liu, Yosuke Nakanishi and Ryuichi Yokoyama, A time series model based on hybrid-kernel least-squares support vector machine for short-term wind power forecasting, *ISA Transactions*, 108(2021): 58-68, 2021.
- Xiaoming Wu, Dianhong Wang, Weihua Cao and Min Ding*, A Genetic-Algorithm Support Vector Machine and D-S Evidence Theory Based Fault Diagnostic Model for Transmission Line, *IEEE Transactions on power systems*, 34(6): 4186-4194, 2019.
- T. Cover and P. Hart, Nearest neighbor pattern classification, *IEEE Transactions on Information Theory*, 13(1):21-27 , 1967.
- Shaoning Pang, S. Ozawa and N. Kasabov, Incremental linear discriminant analysis for classification of data streams, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5): 905-914, 2005.