
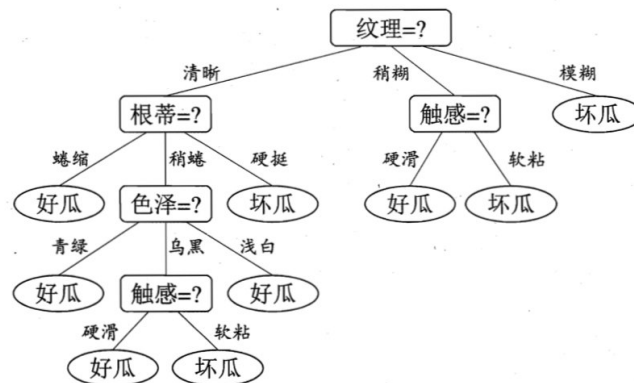


# Classification

 by kingno

## Classification I

### 1、决策树



#### 2.1 生成过程

决策树的生成是一个递归的过程：

1. 从属性集  $A$  中选择 最优划分属性  $a$
2. 为  $a$  中的每一个值生成一个分支

递归返回的情形：

1. 当前节点包含的样本全属于同一类别，无需划分
2. 当前属性集为空，或是所有样本在所有属性上取值相同，无法划分
3. 当前结点包含的样本集合为空，不能划分

## 2.1 划分选择

决策树学习的关键是：如何选择最优划分属性  $a$ 。有信息增益、增益率、基尼指数三种算法：

### 1. 信息增益 (Information Gain)

**信息熵**：度量样本集合纯度的指标。假定当前样本集合  $D$  中第  $k$  类样本所占比例为  $p_k$ ，则  $D$  的信息熵为  $\text{Ent}(D) = -\sum p_k \log_2 p_k$ ， $\text{Ent}(D)$  的**值越小**，则  $D$  的**纯度越高**

假定属性  $a$  有  $V$  个可能取值  $\{a^1, a^2, \dots, a^v\}$ ，用该属性对  $D$  划分，得到  $v$  个子节点  $\{D^1, D^2, \dots, D^v\}$

信息增益： $\text{Gain}(D, a) = \text{Ent}(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} \text{Ent}(D^v)$ ，信息增益越大，意味着用  $a$  划分所获得的“纯度提升”越大。即，**信息增益准则对可取值数目较多的属性有所偏好**

### 2. 增益率 (Gain Ratio)

为减少信息增益准则的偏好带来的不利影响，使用增益率来选择最优划分准则。其定义为：

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

$\text{IV}$  称为属性  $a$  的固有值， $a$  的可能取值越多， $\text{IV}(a)$  越大

**增益率对可能取值数目较少的属性有所偏好**

### 3. 基尼指数 (Gini Index)

**数据集  $D$  的基尼指数**：

$$\text{Gini}(D) = 1 - \sum p_k^2$$

它反映了从数据集  $D$  中随机抽取两个样本，其类别标记不一致的概率。因此， $\text{Gini}(D)$  越小，数据集  $D$  的纯度越高

**属性  $a$  的基尼指数**：

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

于是，我们在候选属性集  $A$  中，选择那个使得划分后基尼指数最小的属性作为最优划分属性

## 2.2 剪枝处理

应对“过拟合”

- 预剪枝：对每个节点在划分前进行估计，若当前节点的划分不能带来决策树泛化性能提升，则停止划分并将当前节点标记为叶节点
- 后剪枝：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节点进行考察，若将该节点对应的子树替换为叶节点能带来决策树泛化能力的提升，则将该子树替换为叶节点

## 2.3 连续值处理

对连续属性进行离散化。最简单的策略是二分法

要选取最佳分割点 A：对于属性  $a$ ，它在 D 上出现了  $n$  个不同的取值，则将这些值从小到大进行排序，取所有相邻两个点的平均值  $(a_i + a_{i+1})/2$ ，一共  $n - 1$  个，找那个使得指标最大化的点

## 2.4 缺失值处理

1. 给缺失值赋出现最多的值
2. 根据各种取值的概率进行选择

# Classification II

## 1、贝叶斯分类

概念太难懂，直接看例题

### 1.1 朴素贝叶斯

age	income	student	credit_rating	buy
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

假设分类的类别有

- $C_1$  : 买电脑 yes
- $C_2$  : 不买电脑 no

对于一个人  $X = (\text{age} \leq 30, \text{Income}=\text{medium}, \text{Student}=\text{yes}, \text{Credit\_rating}=\text{Fair})$ ，我们要预测他是否会买电脑，也就是  $P(C_1|X), P(C_2|X)$  哪一个值更大？

1. 计算  $P(C_i)$ :  $P(C_1) = 9/14 = 0.643$ ,  $P(C_2) = 5/14 = 0.357$
2. 朴素贝叶斯假设**每个属性都是独立的**。所以，我们分别计算不同属性下的条件概率：

$$P(\text{age} = "<=30" \mid \text{buys\_computer} = \text{"yes"}) = 2/9 = 0.222$$

$$P(\text{age} = "<= 30" \mid \text{buys\_computer} = \text{"no"}) = 3/5 = 0.6$$

$$P(\text{income} = \text{"medium"} \mid \text{buys\_computer} = \text{"yes"}) = 4/9 = 0.444$$

$$P(\text{income} = \text{"medium"} \mid \text{buys\_computer} = \text{"no"}) = 2/5 = 0.4$$

$$P(\text{student} = \text{"yes"} \mid \text{buys\_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{student} = \text{"yes"} \mid \text{buys\_computer} = \text{"no"}) = 1/5 = 0.2$$

$$P(\text{credit\_rating} = \text{"fair"} \mid \text{buys\_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{credit\_rating} = \text{"fair"} \mid \text{buys\_computer} = \text{"no"}) = 2/5 = 0.4$$

3. 计算总的概率：

- $P(X \mid \text{buys\_computer}=\text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$
- $P(X \mid \text{buys\_computer}=\text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$

所以，这个人可能会买电脑

## 1.2 零概率问题

假设数据集有1000个元组，低收入0人，中等收入990人，高收入10人

使用拉普拉斯修正 (Laplacian correction) : 给每个类别 + 1, 所以:

- 低收入:  $1/1003$
- 中收入:  $991/1003$
- 高收入:  $11/1003$

## 1.3 避免计算下溢

全都是小数，计算机可能会出现下溢

解决方法: 取  $\log$

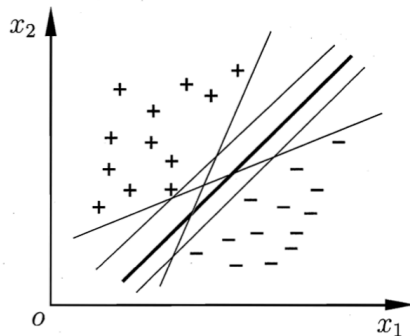
## 2、反向传播

看 PPT 的篇幅，应该只需要了解基本的知识，没什么重点

## 3、支持向量机

了解基本知识即可，不需要懂公式

给定训练样本集  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m), y_i \in \{-1, +1\}\}$ , 分类学习最基本的想法就是基于训练集  $D$  在样本空间中找到一个划分超平面，将不同类别的样本分开。但是，能将训练样本分开的划分超平面可能有很多，比如下图。我们应该找哪一个呢？



直观上看，应该去找位于两类训练样本“正中间”的划分超平面，即图中加粗的那条线。因为该划分超平面对训练样本局部扰动“容忍”性最好。例如，由于训练集的局限性或噪声的因素，训练集外的样本可能比上图中的训练样本更接近两个类的分隔界，这将使许多划分超平面出现错误，而加粗的超平面受影响最小。换言之，这个划分超平面对未见示例的泛化能力最强。

在样本空间中，划分超平面可以通过如下线性方程来描述：

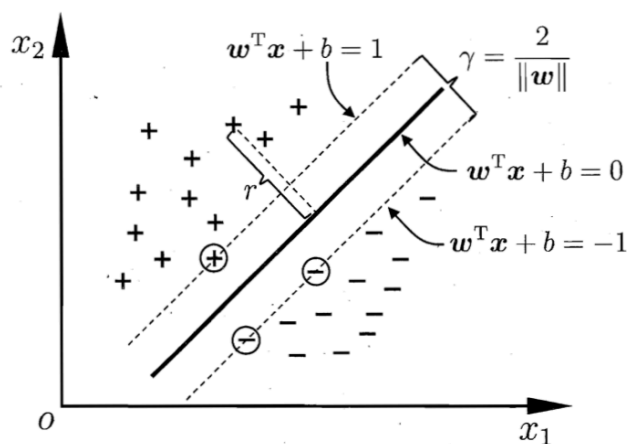
$$\mathbf{w}^T \mathbf{x} + b = 0$$

其中， $\mathbf{w} = (w_1; w_2; \dots; w_d)$  为法向量，决定了超平面的 **方向**； $b$  为位移项，决定了超平面 **与原点之间的距离**。显然，超平面可以被法向量  $\mathbf{w}$  和位移  $b$  确定，下面我们记为  $(w, b)$ 。样本空间中任意点  $x$  到超平面  $(w, b)$  的距离可写为

$$r = \frac{|w^T x + b|}{\|\mathbf{w}\|}$$

假设超平面能将训练样本正确分类，即对于  $(x_i, y_i) \in D$ ，若  $y_i = +1$ ，则有  $w^T x_i + b > 0$ ；若  $y_i = -1$ ，则有  $w^T x_i + b < 0$ 。令

$$\begin{cases} w^T x_i + b \geq +1, & y_i = +1 \\ w^T x_i + b \leq -1, & y_i = -1 \end{cases}$$



如上图所示，距离超平面最近的这几个训练样本点是的不等式的等号成立，它们被称为“**支持向量**”，两个异类支持向量到超平面的距离之和为

$$\gamma = \frac{2}{\|w\|}$$

它被称为“**间隔**”。想要找到具有“**最大间隔**”的划分超平面，也就是要找到能满足上述不等式方程组中的约束的参数 $w, b$ ，使得 $\gamma$  最大，即

$$\begin{aligned} \max_{w, b} \quad & \frac{2}{\|w\|} \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 \end{aligned}$$

为了最大化间隔，只需要最大化 $\|w\|^{-1}$ ，这等价于最小化 $\|w\|^2$ ，于是，上式课重写为

$$\begin{aligned} \max_{w, b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 \end{aligned}$$

这就是支持向量机的基本型。这是一个凸二次规划（convex quadratic programming）问题

上述的假设，都基于训练样本是线性可分的。对于线性不可分的问题，可将样本从原始空间映射到一个更**高维**的特征空间，是的样本在这个特征空间内线性可分。

## Classification III

### 1、KNN

#### 1.1 急切学习和懒惰学习

- 急切学习：在利用算法进行判断之前，先 利用训练集数据 通过训练得到一个 目标函数 ，在需要进行判断时利用已经训练好的函数进行决策
- 惰性学习：在最开始的时候不会根据已有的样本创建目标函数，只是简单的把训练用的样本储存好，后期需要对新进入的样本进行判断的时候才开始分析新进入样本与已存在的训练样本之间的关系，并据此确定新实例（新进入样本）的目标函数值

- 惰性学习使用了更多的假设；急切学习需要用 一个假设 来涵盖整个样例空间
- KNN 是典型的懒惰学习算法

## 1.2 KNN 算法

对于空间中的点  $x_q$ ，KNN 算法会返回离  $x_q$  最近的  $k$  个点（欧几里得距离）

- **维数灾难**：在高维情形下出现的数据样本稀疏、距离计算困难等问题

解决方法：降维

## 1.3 案例式推理（CBR）

利用数据库中已有的问题的解法来解决新的问题。数据库中存储的是具体的符号描述，而不是坐标点。  
利用相似度算法来查找相似问题

分类数据中标签数量不平衡（正样本很少，负样本过多）的解决方法：

- Oversampling：从正样本中再次采样
- Under-sampling：随机删除负样本
- Threshold-moving：改变决策的阈值  $t$

中间删除若干章节，感觉不是很重要。考试的时候再看也来得及

## 2、模型评估

### 2.1 混淆矩阵

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

查准率  $P$  (precision) :  $P = \frac{TP}{TP+FP}$

查全率  $R$  (recall) :  $R = \frac{TP}{TP+FN}$



$$F = (2 \times P \times R) / (P + R)$$

$$F_\beta = ((1 + \beta^2) \times P \times R) / (\beta^2 \times P + R)$$

## 2.2 误差计算

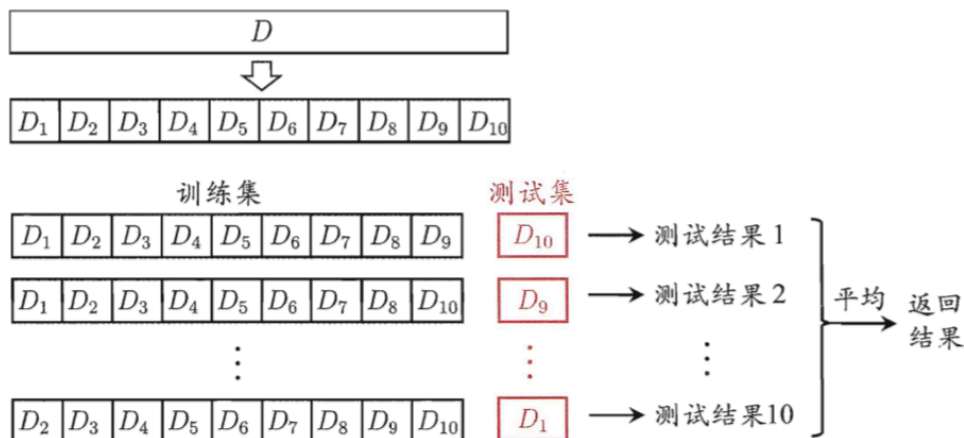
- $|y_i - y'_i|$
- $(y_i - y'_i)^2$

- Mean absolute error:  $\frac{\sum_{i=1}^d |y_i - y'_i|}{d}$  Mean squared error:  $\frac{\sum_{i=1}^d (y_i - y'_i)^2}{d}$
- Relative absolute error:  $\frac{\sum_{i=1}^d |y_i - y'_i|}{\sum_{i=1}^d |y_i - \bar{y}|}$  Relative squared error:  $\frac{\sum_{i=1}^d (y_i - y'_i)^2}{\sum_{i=1}^d (y_i - \bar{y})^2}$

## 2.2 留出法 (hold-out)

将数据集划分为两个互斥的集合，其中一个集合作为训练集，另一个集合作为测试集

## 2.3 交叉验证



## 2.4 自助法 (bootstrapping)

给定包含  $m$  个样本的数据集  $D$ ，我们对它进行采样产生数据集  $D'$ ：每次随机从  $D$  中挑选一个样本，将其拷贝放入  $D'$ ，然后再将该样本放回初始数据集  $D$  中。重复该步骤  $m$  次，我们得到了包含  $m$  个样本的数据集  $D'$ 。显然， $D$  中有一部分数据会在  $D'$  中重复出现，而另一部分不出现。样本在  $m$  次采样中始终不被采到的概率是  $(1 - \frac{1}{m})^m$ ，求极限

$$\lim_{m \rightarrow \infty} (1 - \frac{1}{m})^m = \frac{1}{e} = 0.368$$

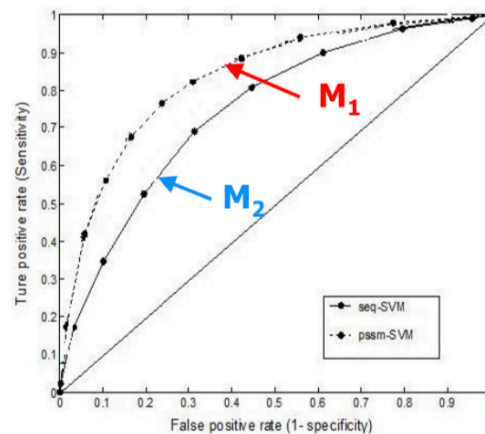
即，数据集D中有36.8%的样本未出现在 $D'$ 中，于是我们可以将 $D'$ 用作训练集， $D \setminus D'$ 用作测试集

自助法适用于数据集较小的情况

## 2.5 置信区间

对于两个模型  $M_1, M_2$ ，计算它们的误差。假设它们的误差只是基于概率，我们该选择哪一个？使用置信区间估计。（和概率论有关，估计不会考）

## 2.6 ROC 曲线



该展示了 TP 和 FP 的权衡。曲线下的面积是对模型准确率的评估。曲线越接近中间的那条对角线，模型的准确率越低

## 3、模型融合

- Bagging：取不同分类器预测的平均
- Boosting：对每个分类器加权
- Ensemble：将一系列不同的分类器混合