数据科学导论

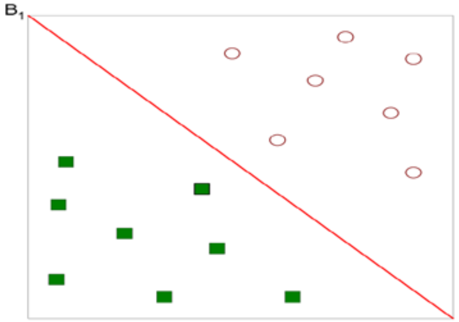
第三节

分 类

徐童

2020.3.20

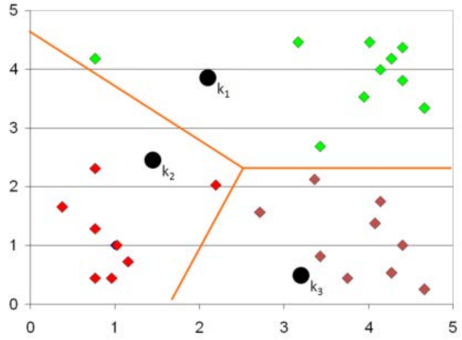
- 数据挖掘的基本方法



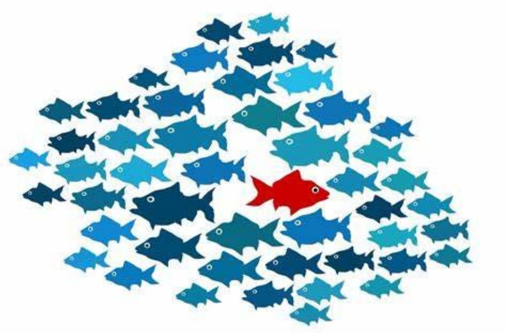
分类

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

关联规则



聚类



离群检测

问题背景

- **时常会面临的困扰**

- 这封邮件是垃圾邮件吗？

Important Message 发起会议

发件人： Maggie M... <archive@tfi-urfo.ru>

时　间： 2019年11月22日 00:03:09 (星期五)

收件人： Recipients <archive@tfi-urfo.ru>

Hi,

I have a business proposal to share with you. Contact me back for more details.

Thanks.
Maggie M. Wang

- **时常会面临的困扰**

- 如何判别垃圾邮件?


Important Message 发起会议
发件人: Maggie M... <archive@tfi-urfo.ru>
时 间: 2019年11月22日 00:03:09 (星期五)
收件人: Recipients <archive@tfi-urfo.ru>

Hi,

I have a business proposal to share with you. Contact me back for more details.

Thanks.
Maggie M. Wang

罕见的邮箱后缀

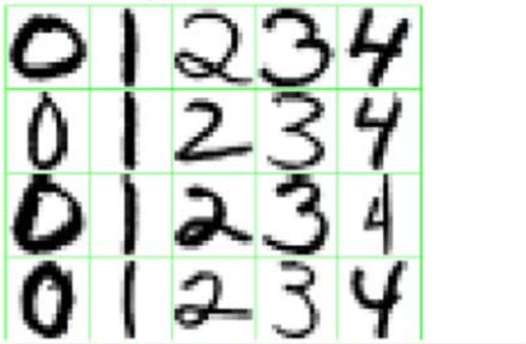
莫名其妙的收信人

不合常理的邮件内容

基于一些特征与规则，我们可以将垃圾邮件的判别视作一个分类问题

- **什么是分类问题**

- 给定一组样本 $\{(\mathbf{x}_i, y_i)\}_{i=1..N}$ ，其中x为样本的特征，y为样本对应的标签
  - 分类问题的目标在于训练一个有效的分类器（Classifier）
  - 借助分类器，当面对新样本 x 时，应将其准确映射到对应的标签 $y \in \mathcal{Y}$
  - 如何表征样本是一个问题，尤其是面向文本、图像、视频等多模态信息
- 一个基础的实例：



digits recognition;
$\mathcal{Y} = \{0, \ldots, 9\}$

- **最基本的分类方法：人工分类**

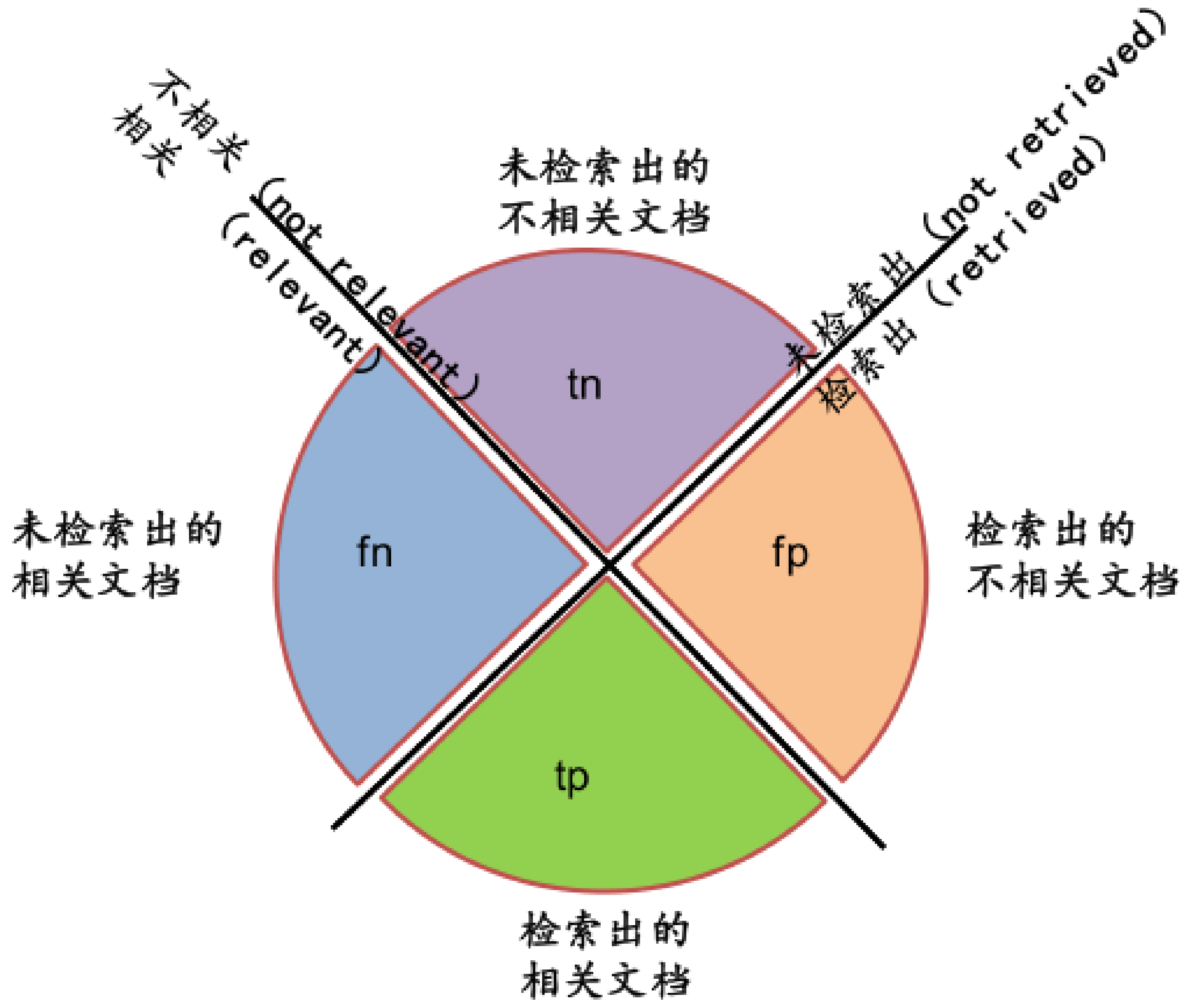
- 最为基础和直接的方法，借助领域专家的支持可以保证分类的准确性
  - 有多少人工，就有多少智能（手动）
  - 广泛应用于早期的各种搜索引擎
  - 在问题规模较小时较为有效
  - 严重缺乏可扩展性，代价高昂效率低下

1995年的Yahoo!

本节目录

- 分类评价指标
- 基于规则的分类
- 基于监督学习的分类
  - 决策树
  - 最近邻分类
  - 支持向量机
  - Logistic回归
  - 不平衡分类问题

- **文档集合的基本划分**


不相关（not relevant）
相关（relevant）
未检索出（not retrieved）
检索出（retrieved）
未检索出的
不相关文档
tn
未检索出的
相关文档
fn
检索出的
不相关文档
fp
tp
检索出的
相关文档

- **基本评价指标的矩阵化表示**

- TP、FN等字母究竟代表着什么?
  - T/F：True or False，表示二分类的正确与否
  - P/N：Positive or Negative，表示算法对样本的判断
  - 四种简写的含义：
    - TP：True Positive，样本为正例，且被判定为正，即真正
    - FN：False Negative，样本为正例，但错误地被判定为负，即假负
    - FP：False Positive，样本为负例，但错误地被判定为正，即假正
    - TP：True Negative，样本为负例，且被判定为负，即真负

	分类为正例	分类为负例
实际为正例	TP	FN
实际为负例	FP	TN

面向单查询的基本评价指标

- 准确率（Precision）
  - 指分类为正例的样本中，标签也为正例的样本比例，也称查准率
  - 计算公式为TP/(TP+FP)
- 召回率（Recall）
  - 指标签为正例的样本中，被分类为正例的样本比例，也称查全率
  - 计算公式为TP/(TP+FN)

- **一个P-R指标计算的实例**

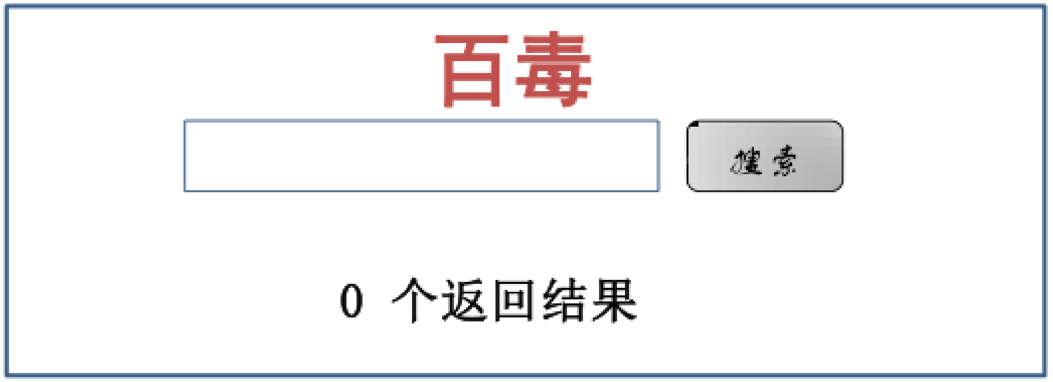
- 当查询1 的标准答案集合为 {d3,d4,d6,d9}时，可知：
  - 对于系统1，查询1：正确率2/5，召回率2/4
  - 对于系统2，查询1：正确率2/4，召回率2/4

系统&查询	1	2	3	4	5
系统1，查询1	d3√	d6√	d8	d10	d11
系统1，查询2	d1	d4	d7	d11	d13
系统2，查询1	d6√	d7	d2	d9√	/
系统2，查询2	d1	d2	d4	d13	d14

- **为什么某种方案被抛弃?**

- 既然TP与TN都是正确结果，为什么不直接计算(TP+TN)的全局比例?
  - (TP+TN)/(TP+TN+FP+FN)，即Accuracy，在模式分类中经常被使用
  - 然而，它在信息检索的相关任务中并不常见，为什么?

如何以最低的代价做一个Accuracy接近 100% 的搜索引擎?



- **准确率与召回率的平衡**

- 不同应用场景中，对于准确率和召回率有着不同的侧重
  - 邮件分类：宁愿放过一些垃圾邮件，也不能错杀正常邮件
    - 牺牲（对垃圾邮件的）召回率，保证较高准确率
  - 智慧医疗：宁愿多判断一些疑似患者，不能漏掉一个病人
    - 牺牲（对病人判断的）准确率，保证较高召回率

- **从P-R的平衡到F值**

- 如前所述，准确率与召回率之间存在权衡
  - 如何综合评价一个算法在这两项指标上的性能?
- F值（F-measure），即准确率与召回率的加权调和平均数

$$F=\frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}}=\frac{(\beta^2+1)PR}{\beta^2P+R}$$

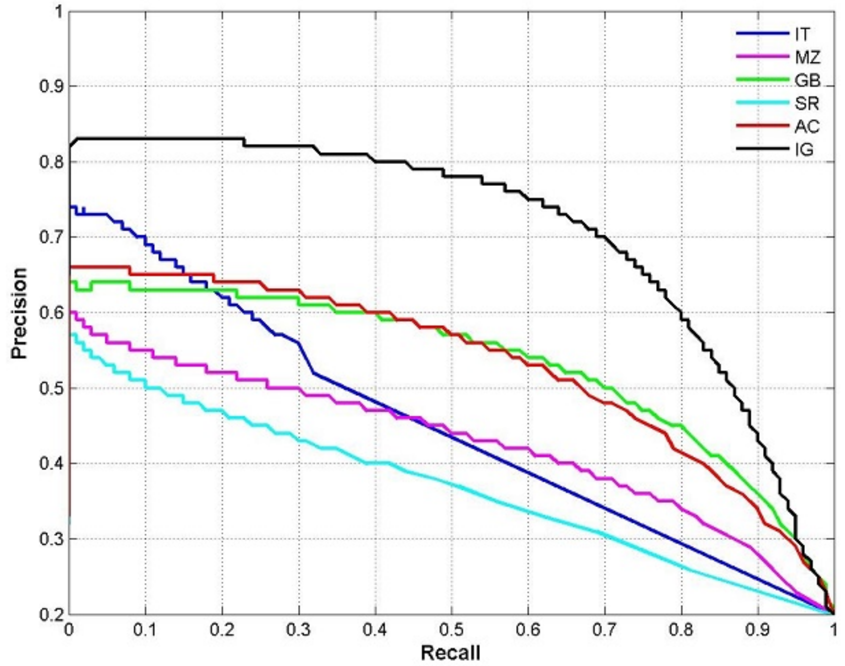
- 通常情况下，我们取α=0.5或β=1（即两者同等重要）
  - 此时，可得基本的F1值，即F=2PR/(P+R)
- 思考题：为什么采用调和平均数，而不是算术平均数?

- **P-R曲线与ROC曲线**

- 在分类问题中，准确率与召回率的平衡是通过选定不同阈值实现的
  - 例如，通过调控相关性的阈值，可以控制检索所得的文档数量
  - 较低的阈值可以使得返回更多文档，但也混入大量不相关的文档
  - 较高的阈值可以保障文档的相关性，但也会遗漏许多相关的文档
  - 如何选择合适的阈值？
    - 通过绘制不同阈值下的指标变化曲线，可以帮助我们做出选择

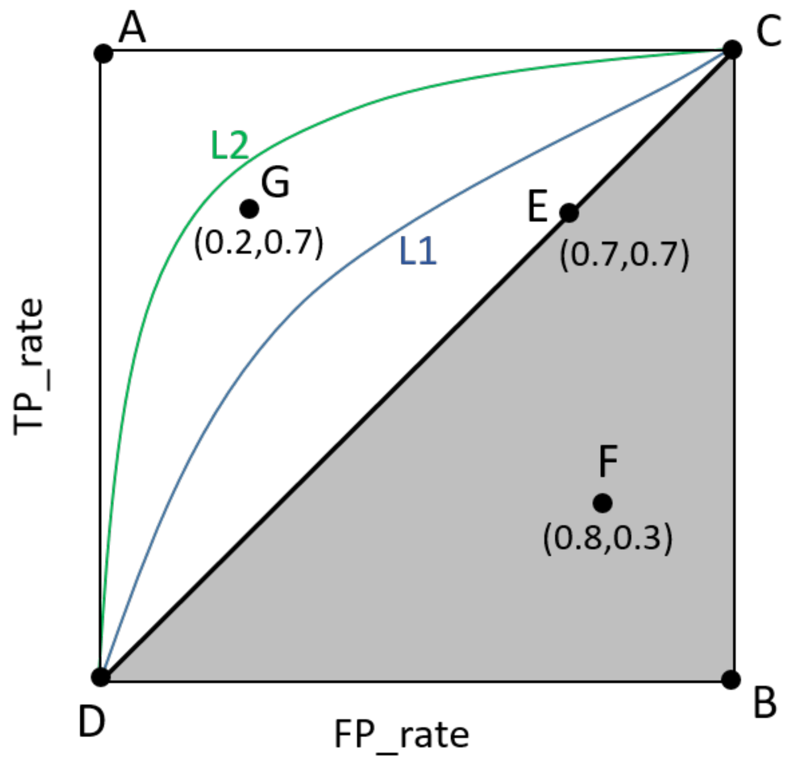
- **P-R曲线与ROC曲线**

- P-R曲线（Precision-Recall Curve）
  - 以准确率和召回率分别作为两条轴线
  - 通过选定不同的阈值得到不同的P-R点并连接成线
  - 通过P-R曲线，可以直观地看出准确率与召回率之间的平衡关系



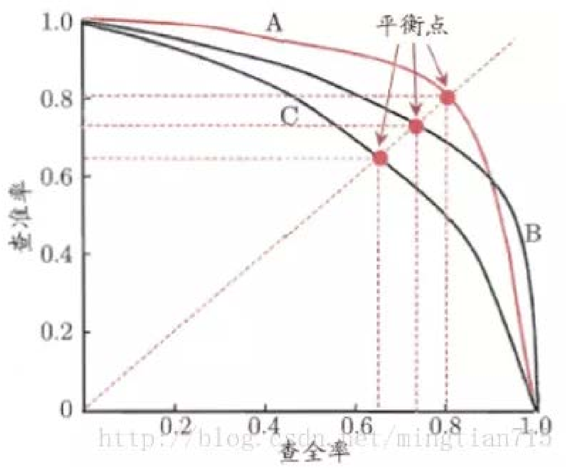
• P-R曲线与ROC曲线

- ROC曲线（Receiver Operating Characteristic Curve，接受者操作特征曲线）
  - 以真正率【TP/(TP+FN)】和假正率【FP/(FP+TN)】作为两条轴线
    - 可直观解释为“命中率”和“误报率”
  - 通过选定不同的阈值得到不同的真正率-假正率点并连接成线
    - 对角线表示区分能力为0，即**随机猜测**
    - 在对角线上端越远，效果越好
    - 低于对角线的结果无意义（无区分度）



- **P-R曲线与ROC曲线**

- 如何基于P-R曲线或ROC曲线判别算法好坏
  - 如果线A将线B完全包住，显然线A对应的算法效果更好
  - 如果两条线发生重合，则可依据以下规则判别：
    - 计算AUC，AUC更高者效果更好
      - Area under curve，即曲线下面积
      - 可通过积分近似计算
    - 另外，当使用P-R曲线时，可使用平衡点计算
      - 平衡点即Precision = Recall的点，值越高越好

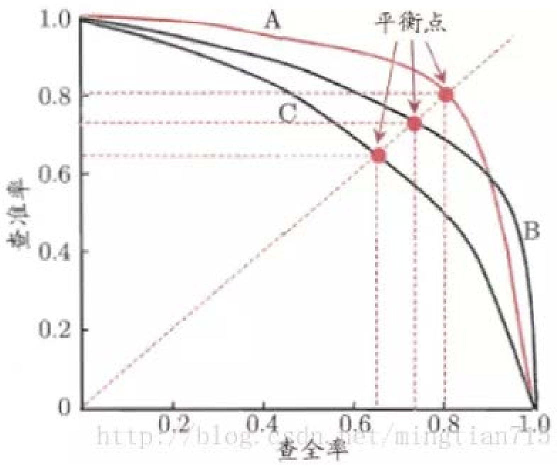


查准率-查全率曲线

单查询评价

- **P-R曲线与ROC曲线**

- P-R曲线与ROC曲线的选择
  - ROC曲线兼顾正负样例，更为全面，而P-R曲线则只考虑正例
    - 用户更关心正样本，如果面向特定数据集，P-R曲线是个好选择
  - P-R曲线受分布影响大，多份数据且正负比例不一时ROC曲线更合适
    - ROC各指标不受正负比例变化影响，但Precision受样本数影响
  - 正负样本比例失调时，P-R曲线更合适
    - 此时，ROC曲线各指标波动很小，难以体现出性能的差异性



查准率-查全率曲线

本节目录

- 分类评价指标
- **基于规则的分类**
- 基于监督学习的分类
  - 决策树
  - 最近邻分类
  - 支持向量机
  - Logistic回归
  - 不平衡分类问题

- **基于规则的分类器**

- 未来的程序猿（媛）们都知道的基本语法：if…then…
- 基于规则的分类器（Rule-based Classifier）就是使用一组if-then来进行分类
  - 基本形式：Condition → y（标签）
  - 其中，Condition是一组属性的组合，也被称作规则的前提
  - 例如：
    - (胎生 = 否) ∧ (飞行动物 = 是) → 鸟类
    - (胎生 = 是) ∧ (体温 = 恒温) → 哺乳类

- **规则分类器的基本原理**

- 基于规则的分类器所产生的规则集，应具有以下两个重要性质：
- **互斥原理**（Mutually Exclusive Rule）
  - 如果规则集中不存在两条规则被同一条记录触发，则规则是互斥的
  - 该性质确保每条记录**至多**被一条规则所覆盖
- **穷举原理**（Exhaustive Rule）
  - 对于属性值的任一组合，规则集中都存在某条规则加以覆盖
  - 该性质确保每条记录**至少**被一条规则所覆盖

• 规则分类器的基本原理

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) → Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) → Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

- Turtle同时触发R4与R5，因此出现了矛盾。
  - 解决方法：重新制定规则、采用有序规则，或采用投票机制
- Dogfish Shark无法对应任何一条规则，因此无法分类
  - 解决方法：制定一条“兜底”规则，解决所有不被其他规则包含的样本

- **规则分类器的有序性**

- 在互斥原理未被遵守的情况下，一条记录可能被多条规则覆盖
  - 此时，这些规则的结果之间可能存在冲突
- 一种解决方法是：对规则集按照优先级降序排列
  - 对于一条记录，按照所触发的排序最高的规则进行执行

规则分类器的有序性

- 常见的两种规则排序方法
  - 基于规则的排序方案：按照规则的质量（如准确性）进行排序
  - 基于类的排序方案：同类的规则排在一起，相对顺序被忽略
    - 此时排序的对象不是规则，而是类

Rule-based Ordering

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Class-based Ordering

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

(Refund=No, Marital Status={Married}) ==> No

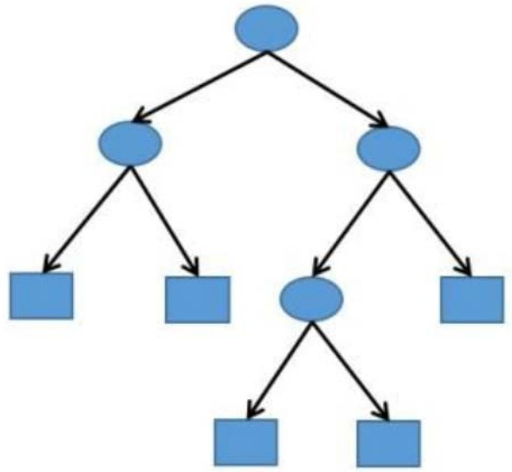
(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

- **如何制定规则分类器**

- 最基础的方法：人工制定规则进行分类
  - 相较于纯手工打造，效率更高，运用更广泛
    - 例如，各大门户网站持续不断维护更新的敏感词列表……
  - 如果人工规则得到定期更新，效果往往较好
  - 然而，维护规则成本较高，需要持续更新

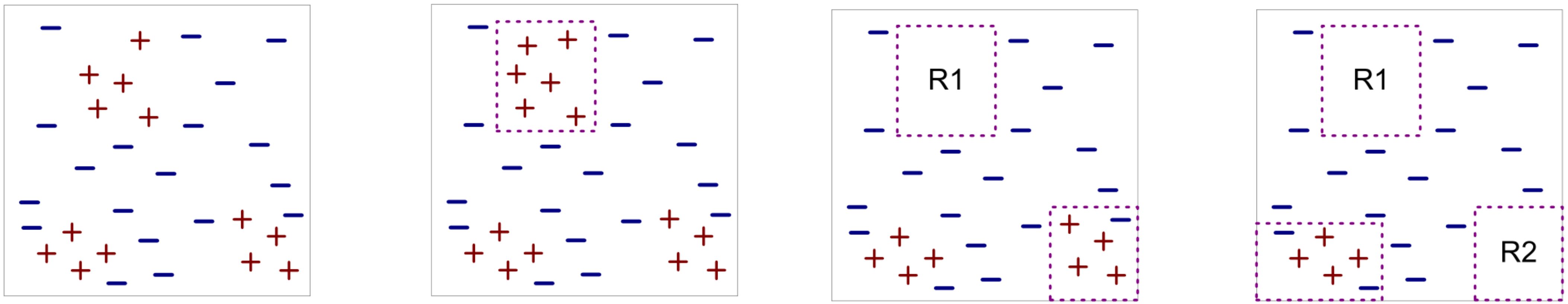
- **如何制定规则分类器**

- 如何利用算法自动生成规则?
  - 直接方法：从数据中自动学习规则
    - 例如，顺序覆盖（Sequential Covering）、RIPPER、CN2等算法
  - 间接算法：借助其他分类模型学习规则
    - 典型例子：从决策树中提炼规则



- **直接方法：顺序覆盖**

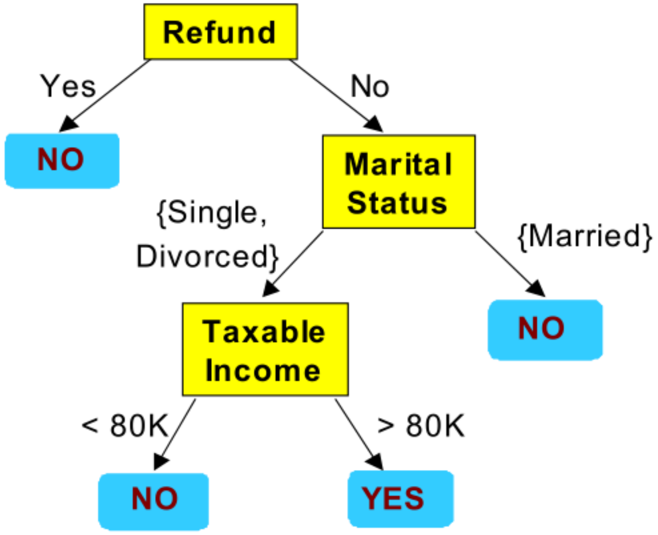
- 一种基础的规则生成算法，采用某种评估以贪心方式进行学习



(i) Original Data　(ii) Step 1　(iii) Step 2　(iv) Step 3

• 间接方法：基于决策树生成规则

- 事实上，决策树从根节点到叶节点的每一条路径都对应一条分类规则
  - 路径中的条件构成规则的前提，而叶节点的类标号对应规则的分类结果
  - 注意：这种情况下生成的规则是满足**<u>互斥</u>**和**<u>穷举</u>**两条原理的



Classification Rules

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

- **基于规则分类器的优点**

- 基于规则的分类器具有较好的可解释性和直观性
  - 这一点与决策树类似，区别主要在于呈现方式
- 基于规则的分类器生成较为简便，分类也较为迅速
  - 当然，如果面临违反互斥原则的情况，可能涉及多条规则投票，相对较为复杂

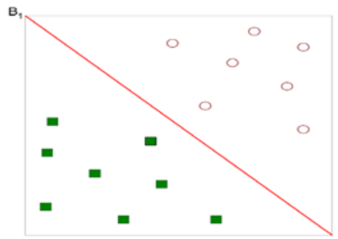
本节目录

- 分类评价指标
- 基于规则的分类
- **基于监督学习的分类**
    - 决策树
    - 最近邻分类
    - 支持向量机
    - Logistic回归
    - 不平衡分类问题

监督学习

- **更为普遍的分类方法：基于有监督学习**

- 通过训练样本，学习从样本到类别标签的映射方式，实现标签分类
  - 常见的有监督分类方法包括
    - 决策树：类似于基于规则的分类方法，简单易行
    - K-最近邻分类法：简单，有效，但取决于数据分布
    - 支持向量机：效果较好，但核函数的选取是个问题
    - 朴素贝叶斯、逻辑斯蒂回归、人工神经网络……
  - 通常情况下，实用中往往采用多种技术的混合模型进行分类

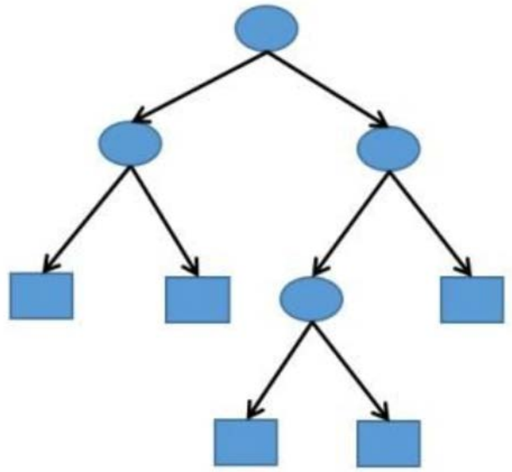


本节目录

- 分类评价指标
- 基于规则的分类
- **基于监督学习的分类**
  - **决策树**
  - 最近邻分类
  - 支持向量机
  - Logistic回归
  - 不平衡分类问题

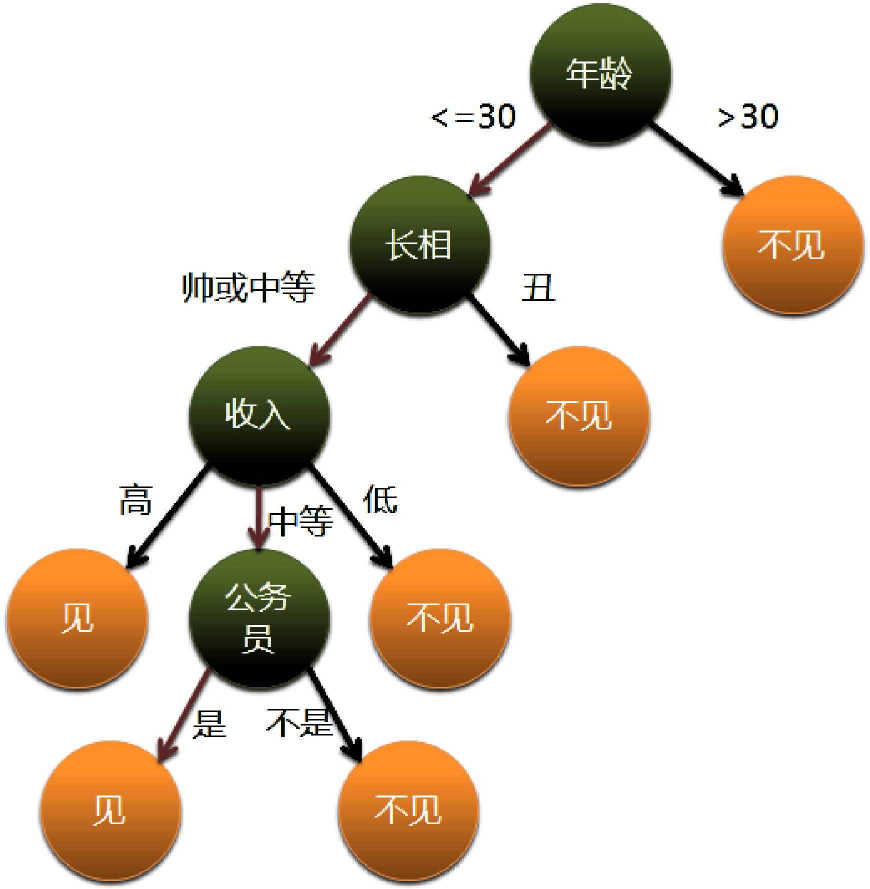
- **什么是决策树**

- 在前面，我们介绍了基于规则进行分类的技术
  - 在满足特定要求（两大原理）的条件下，将规则以树状形式呈现出来
- 决策树（Decision Tree）的定义
  - 一种典型的分类方法，首先对数据进行处理，利用归纳算法生成可读的规则和决策树，然后使用决策对新数据进行分析。
  - 本质上，决策树仍是通过一系列规则对数据进行分类的过程。



- **喜闻乐见的决策树实例**

- 假如你的家人给你安排了一次相亲，你去不去呢?



- **决策树如何生成?**

- 基本的决策树学习过程，可以归纳为以下三个步骤：
  - 特征选择：选取对于训练数据有着较强区分能力的特征
    - 例如，相亲时“有房”是个具有竞争力的要素
  - 生成决策树：基于选定的特征，逐步生成完整的决策树
    - 将所有相亲条件依次整合，最终形成上一页择偶标准
  - 决策树剪枝：简化部分枝干，避免过拟合因素影响
    - 例如，把“学历”等条件剪去，最后保留“真爱”这一项

• 决策树的特征选择问题

- 选取对训练数据具有区分能力的特征，从而提高决策树学习的效率
  - 如果某特征分类的结果与随机结果没有很大的差别，则称这个特征是没有分类能力的
  - 经验上，扔掉这样的特征对决策树学习的精度影响不大

表 5.1　贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

• 基本概念：信息熵

- 了解特征选择的准则之前，先了解基本的信息熵（Entropy）的概念
  - 熵是表示随机变量不确定性的度量，不确定性越高，熵越高
  - 设D是一个取自有限个值的离散随机变量，其概率分布如下：

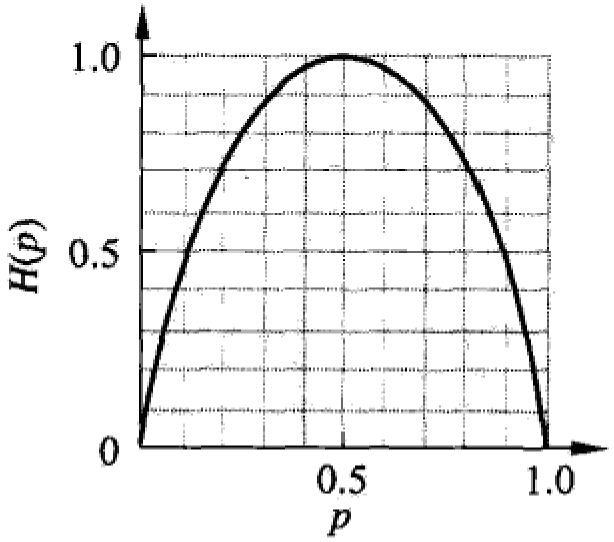
$$P(\mathrm{D} = d_k) = p_k, \qquad k = 1, 2, \ldots, |\mathrm{y}|$$

  - 那么，随机变量X的熵定义为：

$$Ent(D) = -\sum_{K=1}^{|y|} p_k \log_2 p_k$$

• 基本概念：信息熵

- 假设随机变量的取值为非0即1时，可以直观看出熵的变化趋势
  - 此时，熵的计算公式为： Ent(X)= $-p\log_2 p-(1-p)\log_2(1-p)$
  - 熵随概率p变化的曲线如图所示。
    - 可以看出，p=0或1时，Ent(X)=0，即没有不确定性。
    - 当p=0.5时，熵取值最大，同时随机变量不确定性最大。



• 特征选择准则（1）信息增益

- 特征A对训练数据集D的信息增益$Gain(D,a)$，定义为集合D的经验熵$Ent(D)$与特征A在给定条件下对D的经验条件熵$Ent(D|a)$之差，即：

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

- 其中，公式的后半部分，也就是经验条件熵$Ent(D|a)$的部分，即

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

• 特征选择准则（1）信息增益

- 一个计算实例，来自周志华老师的西瓜书（《机器学习》）

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

• 特征选择准则（1）信息增益

- 一个计算实例，来自周志华老师的西瓜书（《机器学习》）
  - 首先，整体的熵可以根据“是否为好瓜”进行计算如下：

$$\mathrm{Ent}(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = -\left(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}\right) = 0.998 .$$

  - 其次，考虑特定特征，以“色泽”为例

D^1 (色泽=青绿), D^2 (色泽= 乌黑), D^3 (色泽=浅白)

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

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

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

- **特征选择准则（1）信息增益**

- 一个计算实例，来自周志华老师的西瓜书（《机器学习》）
  - 由此，可以计算特征“色泽”的经验条件熵如下：

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

  - 类似的，计算其他特征的信息增益，可以最终选择最合适的特征

- **特征选择准则（2）信息增益率**

- 信息增益虽然能够较好地体现某个特征在降低信息不确定性方面的贡献
  - 信息增益越大，说明信息纯度提升越快，最后结果的不确定性越低
- 但是，信息增益也具有一定的局限性，尤其体现在更偏好可取值较多的属性
  - 取值较多，不确定性相对更低，因此得到的熵偏低
- 如何改进？引入信息增益率：
  - 本质上是引入一个惩罚项
  - 取值越少，惩罚项越小

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

其中

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

• 特征选择准则（2）信息增益率

- 信息增益率有矫枉过正的危险
  - 采用信息增益率的情况下，往往倾向于选择取值较少的特征
    - 当取值较少时，IV(a)较小，因此惩罚项相对较小
  - 目前通常采用折中的方法
    - 先从候选特征中，找到信息增益高于平均水平的集合
    - 再从这一集合中，找到信息增益率最大的特征

• 特征选择准则（3）基尼指数

- 此基尼非彼基尼，我们的目标是衡量信息的纯度
  - 基尼指数的目的，在于表示样本集合中一个随机样本被分错的概率
  - 基尼指数越低，表明被分错的概率越低，相应的信息纯度也就越高
  - 假设有K个类，样本点属于第k类的概率为p_k，则基尼指数定义为

$$\mathrm{Gini}(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}{p_k}^2$$

- **决策树的生成过程**

- 决策树的最终目标，在于使每个节点所对应的样本类别均为“纯”的
- 以十大经典算法之一的C4.5为例，当某个节点对应的样本集合 “不纯”时
  - 计算当前节点的类别信息熵
  - 计算当前节点各个属性的信息熵，并进而计算得到该属性对应的信息增益率
  - 基于最大信息增益率的属性，对节点对应的样本集合进行分类
  - 重复上述过程，直至节点对应的样本集合为“纯”的集合（即样本类别统一）
- 其他决策树生成算法过程类似，区别在于准则不同
  - ID3（原型算法）采用信息增益，而CART采用基尼指数

• 决策树的生成实例

- 迭代式地对节点进行分类，直到某个节点上的样本类别统一
    - 此时，该节点为决策树的叶子节点

晴

温度	湿度	风速	活动
寒冷	正常	弱	进行
适中	正常	强	进行
炎热	高	弱	取消
炎热	高	强	取消
适中	高	弱	取消

正常

温度	风速	活动
寒冷	弱	进行
适中	强	进行

高

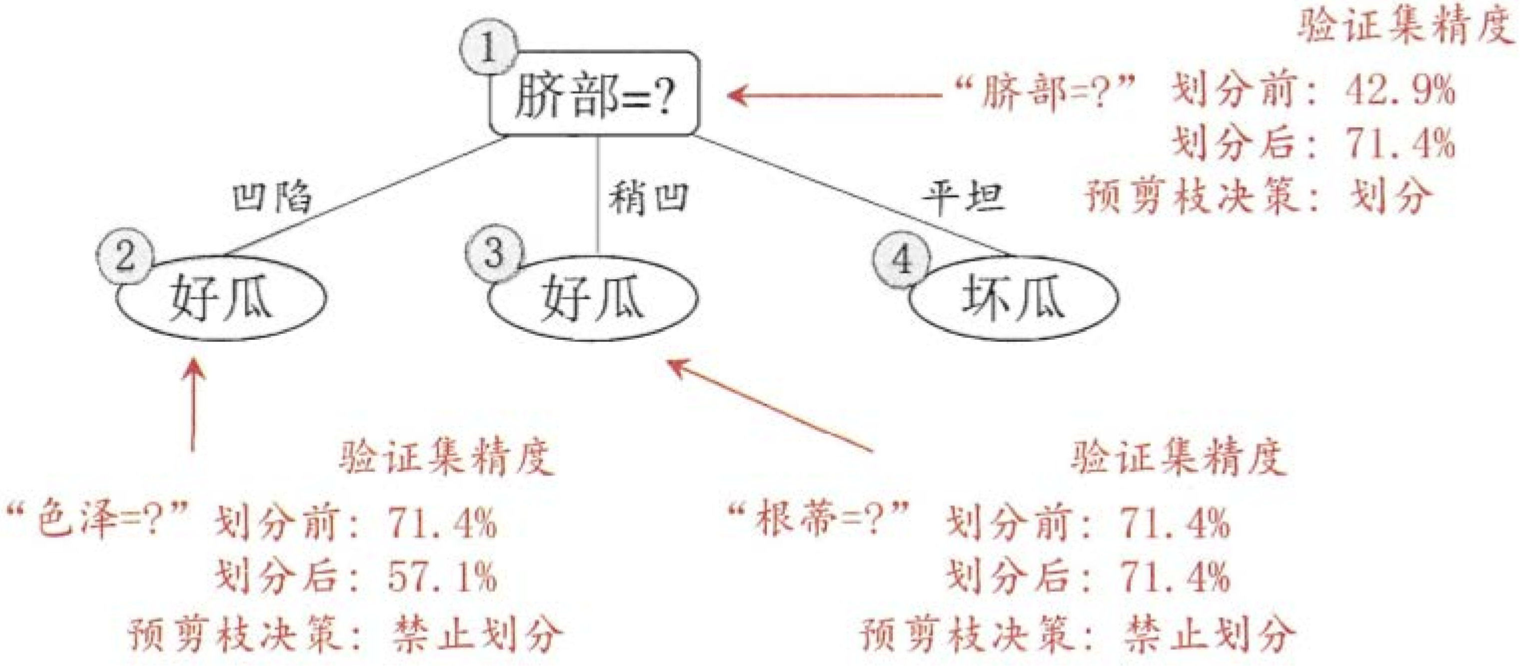
温度	风速	活动
炎热	弱	取消
炎热	强	取消
适中	弱	取消

决策树的剪枝

- 在生成决策树之后，我们还将根据实际情况，对决策树进行剪枝
  - 剪枝的原因在于训练过程的“过拟合”问题
    - 如果训练集与测试集效果都不好，说明出现“欠拟合”现象
    - 如果训练集效果好，而测试集效果不好，说明出现“过拟合”现象
  - 过拟合出现的原因，在于训练过程中过度迁就训练数据特性，而导致构造出过于复杂、过于细枝末节的决策树，泛化能力较差
    - 可能由于训练集本身的特有样本导致了干扰，无法代表测试数据的分布
  - 解决这一问题的办法在于对已生成的决策树进行简化，即“剪枝”

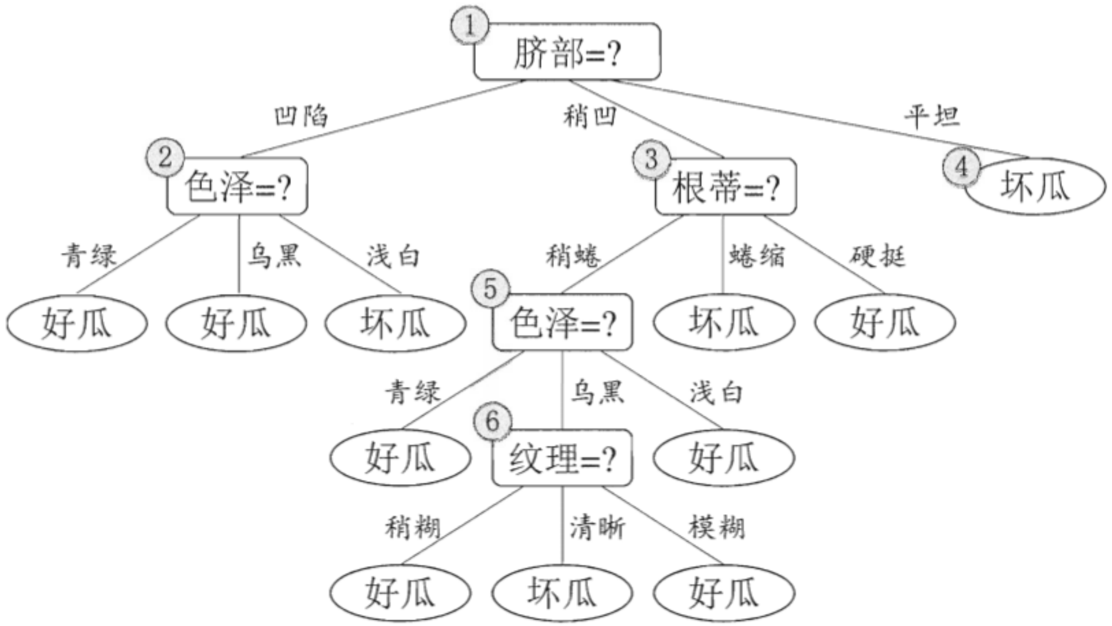
- **决策树的剪枝策略（1）预剪枝**

- 在生成决策树的过程中即进行剪枝，称作“预剪枝”
  - 每个节点划分前，衡量当前节点的划分能否提高决策树的泛化能力



• 决策树的剪枝策略（2）后剪枝

- 在生成决策树之后再进行剪枝，称作“后剪枝”
  - 自底向上考察每个非叶子节点，考虑将该节点替换成叶子节点后能否提高泛化性能

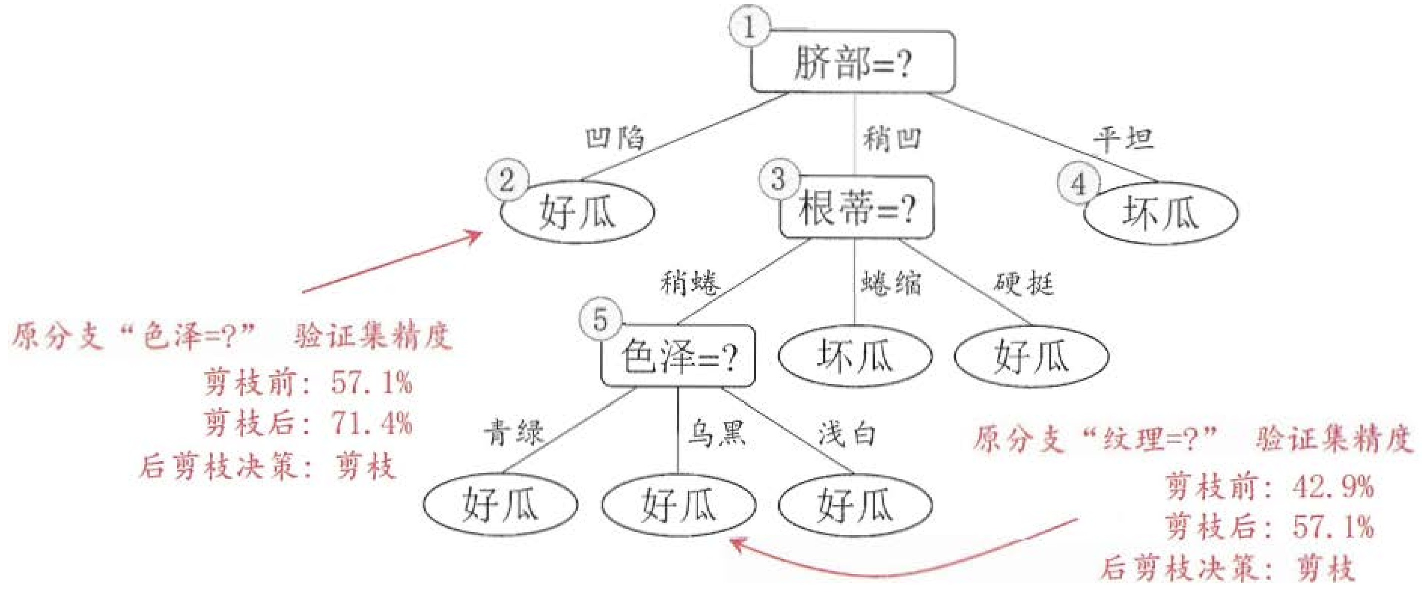


剪枝前

图 4.5 基于表 4.2 生成的未剪枝决策树

• 决策树的剪枝策略（2）后剪枝

- 在生成决策树之后再进行剪枝，称作“后剪枝”
  - 自底向上考察每个非叶子节点，考虑将该节点替换成叶子节点后能否提高泛化性能



剪枝后

图 4.7 基于表 4.2 生成的后剪枝决策树

- **衍生算法：随机森林概述**

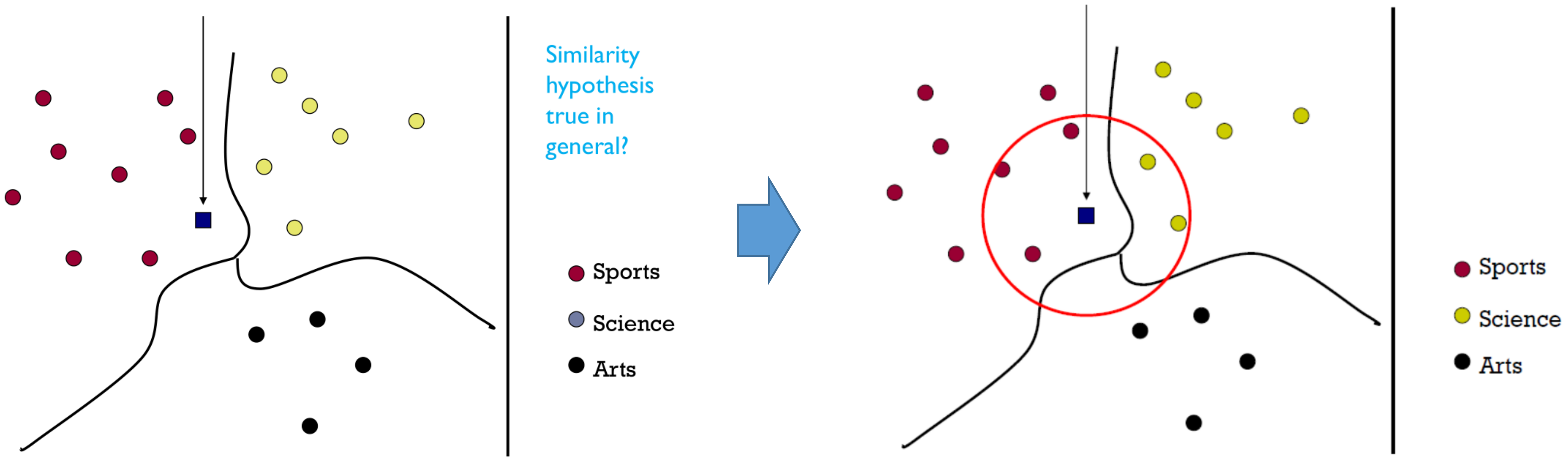
- 决策树简单可行，但单一决策树往往效果有限
- 然而，决策树的拓展方法“**随机森林**”，却往往取得比深度学习更好的效果
  - 两个基本概念：“随机”与“森林”
    - 随机：每棵决策树的生成依赖于有放回随机采样与随机特征
      - 每棵树的样本和特征不尽相同，避免过拟合问题
    - 森林：最终的分类结果取决于多棵决策树集成学习的结果
      - 只有多数决策树出错时才会分类错误，结果相对稳定

本节目录

- 分类评价指标
- 基于规则的分类
- **基于监督学习的分类**
  - 决策树
  - **最近邻分类**
  - 支持向量机
  - Logistic回归
  - 不平衡分类问题

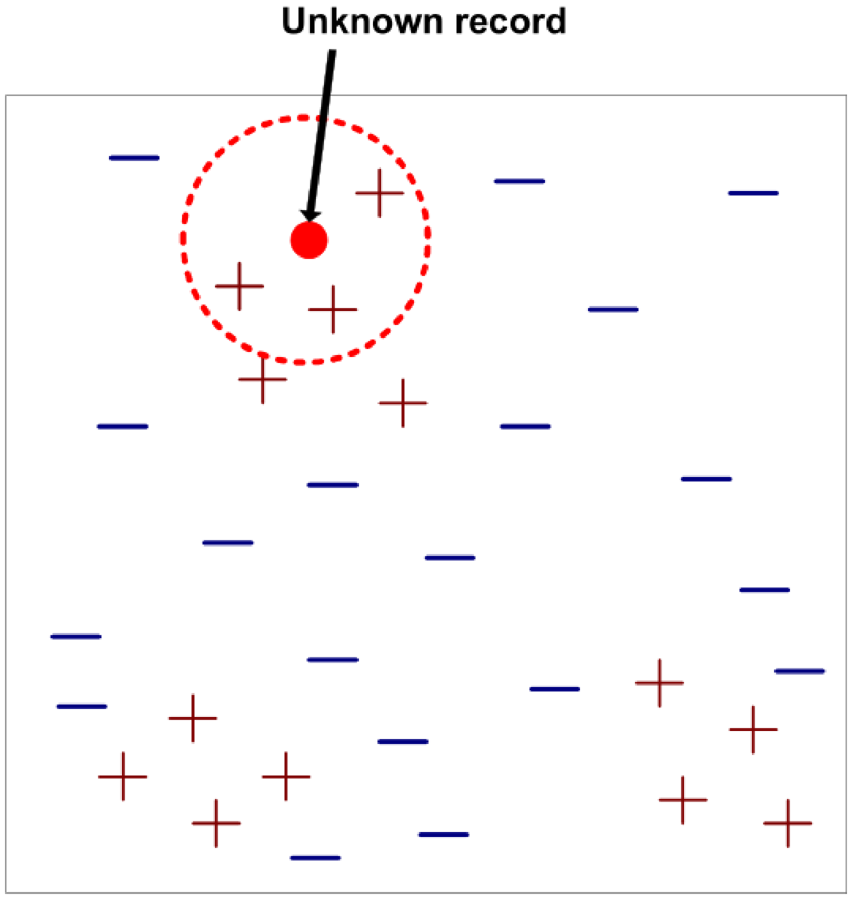
- **相似文档与最近邻假设**

- 经典的向量空间模型遵循一个思路：表征空间上相近的文档是相似的
- 同样的，可以提出合理假设：表征空间上相近的文档应该属于同一个类别



- **最近邻分类的基本要素**

- K-最近邻分类（K Nearest Neighbor）
- 模型输入：
  - 训练样本集合与距离度量
  - K值，用于限定最近邻的数量
- 算法流程：
  - 给定未知样本，首先计算与其他样本的距离，找到K-最近邻
  - 基于K-最近邻的类别确定分类结果



- **最近邻分类的关键问题（1）距离度量**

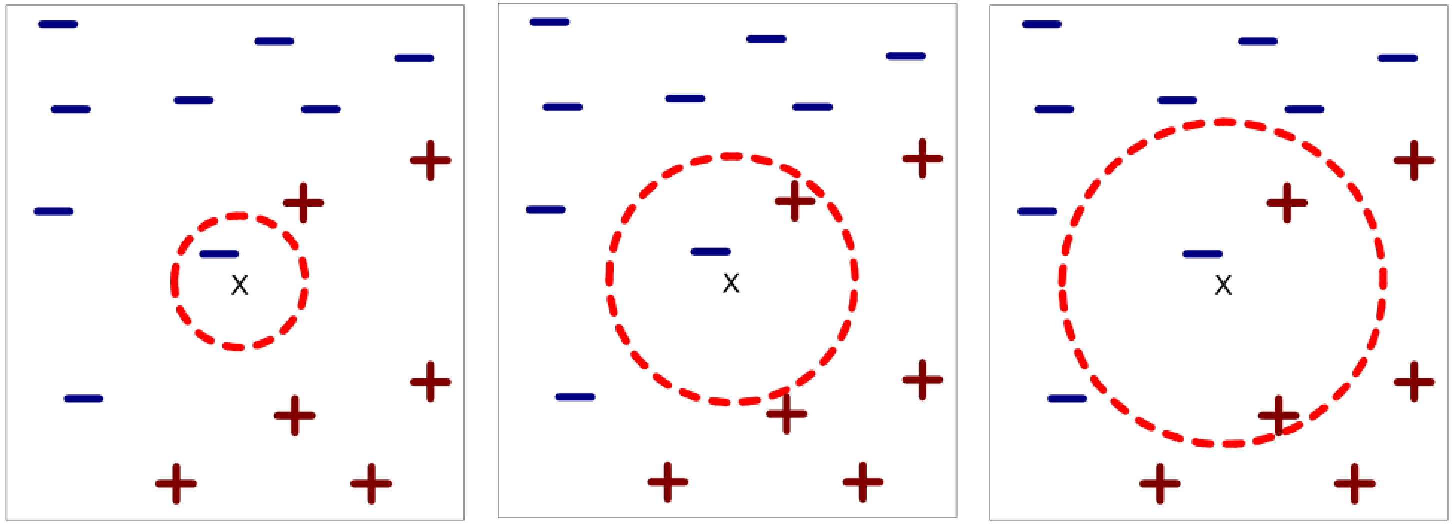
- 最近邻分类的效果严重依赖于距离（或相似性）的度量
  - 对于高维空间而言，最基本的度量方式为欧式距离

$$D(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_i \sigma_i^2 (\mathbf{x}_i - \mathbf{x}_i')^2}$$

  - 如果采用0/1向量，则可使用汉明距离（Hamming）代替
    - 相当于统计有多少维的数字不相同
  - 除此之外，对于文本而言（如采用TF-IDF），可使用余弦相似度
  - 其他可采用的度量如马氏距离、无穷范数（向量最大值）等

- **最近邻分类的关键问题（2）K的取值**

- 最近邻分类的效果同样严重依赖于K的取值（即邻居的数量）
  - K太小，则容易受噪声干扰，K太大，又可能导致错误涵盖其他类别样本



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

- **最近邻分类的特点**

- 最近邻分类是一种典型的基于实例的学习
  - 使用具体实例进行预测，而不需要对数据进行抽象（如提取特征）
- 最近邻分类是一种消极学习，不需要模型，但分类过程开销很大
  - 相比之下，积极学习方法训练模型较为费时费力，但基于模型分类很快
- 最近邻分类基于局部信息进行判别，受噪声影响很大
- 最近邻分类需要慎重选择度量并预处理数据，否则可能被误导
  - 例如，借助身高体重进行分类，身高波动范围不大，而体重差距巨大

本节目录

- 分类评价指标
- 基于规则的分类
- **基于监督学习的分类**
    - 决策树
    - 最近邻分类
    - **支持向量机**
    - Logistic回归
    - 不平衡分类问题

SVM

- **写在前面：一本参考书**

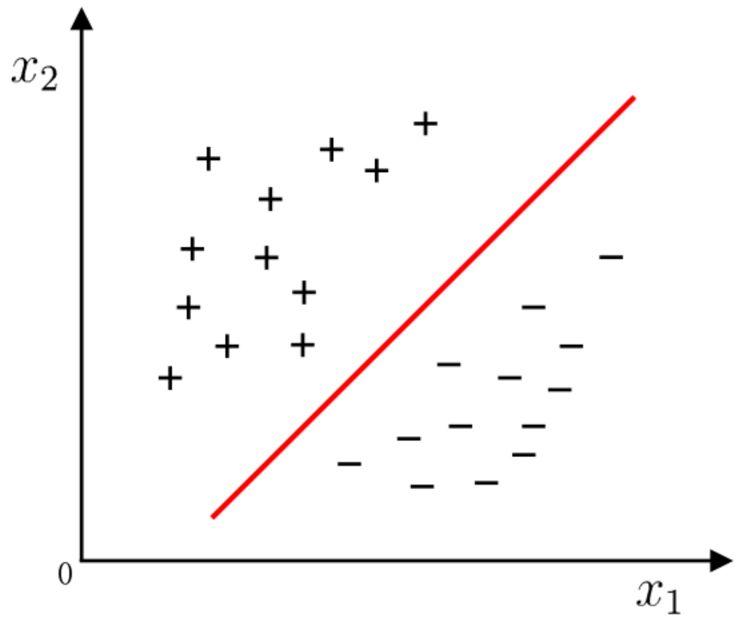
参考书目：支持向量机：理论、算法与拓展，

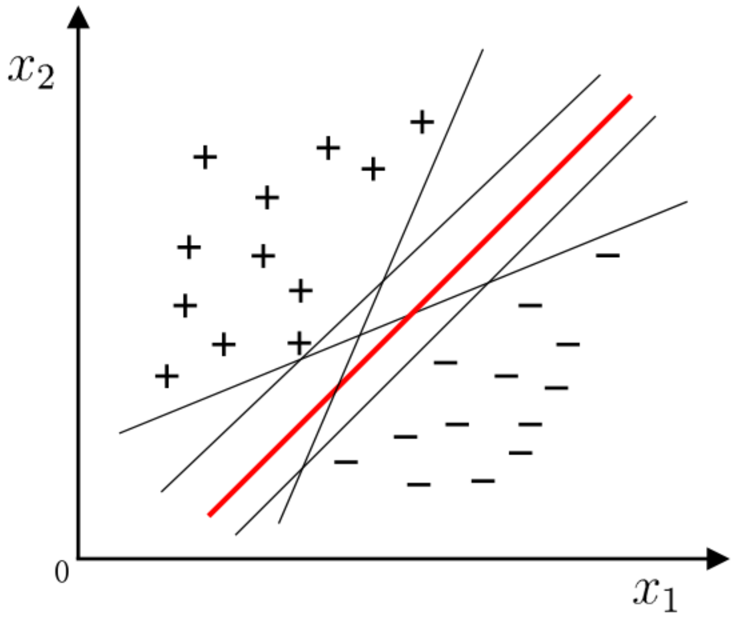
邓乃扬 / 田英杰，科学出版社，2009

SVM

- **高维空间中的节点分割问题**

- 如果仅考虑二分类问题，那么分类问题可以转化为寻找一个超平面，实现对于高维空间中的节点进行有效分割

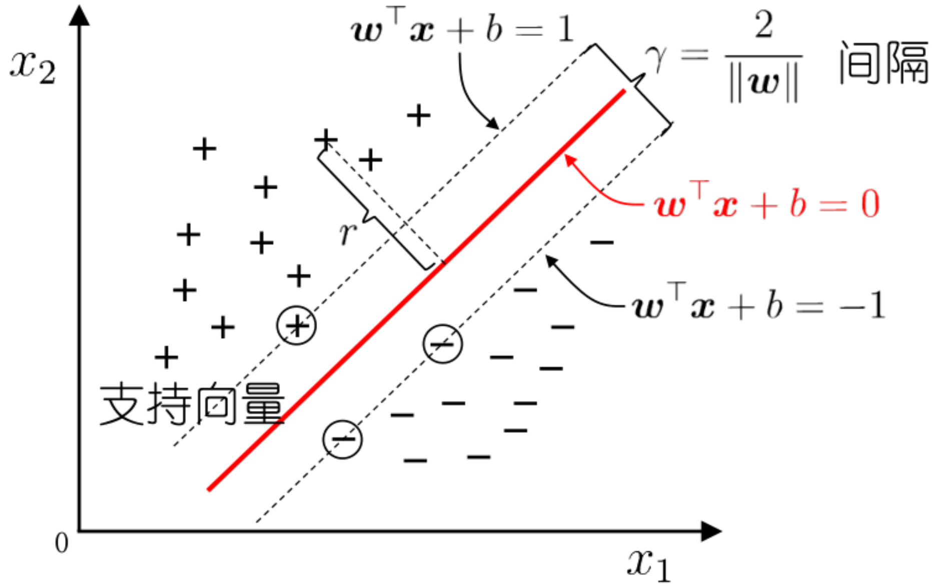




将训练样本分开的超平面可能有很多, 哪一个好呢?

- **高维空间中的节点分割问题**

- 显然，应该选择“正中”的最大间隔超平面，容忍性好，泛化能力强
  - 符合这样条件的超平面，在线性可分的条件下“存在且唯一” （为什么？）



其中，w,b 为参数，w向量方向垂直于超平面。离超平面最近的节点被称作“支持向量”

有超平面方程如右： $\boldsymbol{w}^\top\boldsymbol{x}+b=0$

- **支持向量机及其基本形式**

- 支持向量机（Support Vector Machine）的目的在于找到这样一个最大间隔超平面，使得超平面对应的间隔 γ 最大化

$$\arg\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|}$$
$$\text{s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \; i = 1, 2, \ldots, m.$$

$$\Downarrow$$

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \; i = 1, 2, \ldots, m.$$

- **支持向量机的求解方式**

- 问题：如何求解支持向量机的基本形式?
- 我们注意到，该形式符合凸二次规划问题特征，可以借助拉格朗日对偶性，通过求解对偶问题加以求解

$$\begin{aligned} \min \quad & f_0(x),\ x \in R^n, \\ \text{s.t.} \quad & f_i(x) \leqslant 0\ ,\ i = 1, \cdots, m\ , \\ & h_i(x) = a_i^{\mathrm{T}} x - b_i = 0\ ,\ i = 1, \cdots, p, \end{aligned}$$

满足以上形式的最优化问题为凸规划问题，其中f_0与f_i为连续可微凸函数，h_i为一般线性函数

- **支持向量机的求解方式**

- 凸规划问题对应的拉格朗日函数如下：

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x),$$

  - 其中，λ 与 v 为拉格朗日算子
  - 理论证明可知，该函数的下确界为原凸规划问题最优解的一个下界
    - 证明其实不复杂，只要知道 $L(x, \lambda, \nu) \leqslant f_0(x)$. 就不难理解
  - 因此，求解该问题可以获得原凸规划问题的最优解

- **支持向量机的求解方式**

- 接下来，考虑如何将支持向量机的基本形式套入凸规划问题：

$$
\begin{aligned}
\min \quad & \boxed{f_0(x)}, x \in R^n, \\
\text{s.t} \quad & \boxed{f_i(x)} \leqslant 0, \ i = 1, \cdots, m, \\
& \cancel{h_i(x) = a_i^\top x - b_i = 0, \ i = 1, \cdots, p,}
\end{aligned}
$$

$$
\begin{aligned}
\underset{\boldsymbol{w}, b}{\arg\min} \quad & \boxed{\frac{1}{2}\|\boldsymbol{w}\|^2} \\
\text{s.t.} \quad & \boxed{y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1}, \quad i = 1, 2, \ldots, m.
\end{aligned}
$$

SVM

- **支持向量机的求解方式**

- 由此，我们采用拉格朗日算子法求解支持向量机
  - 第一步：引入拉格朗日乘子 $\alpha_i \geq 0$ 得到拉格朗日函数

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1\right)$$

  - 第二步：令 $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ 对 $\boldsymbol{w}$ 和 b 的**偏导**为零可得

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i, \quad \sum_{i=1}^{m} \alpha_i y_i = 0.$$

  - 第三步：回代

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j - \sum_{i=1}^{m} \alpha_i$$

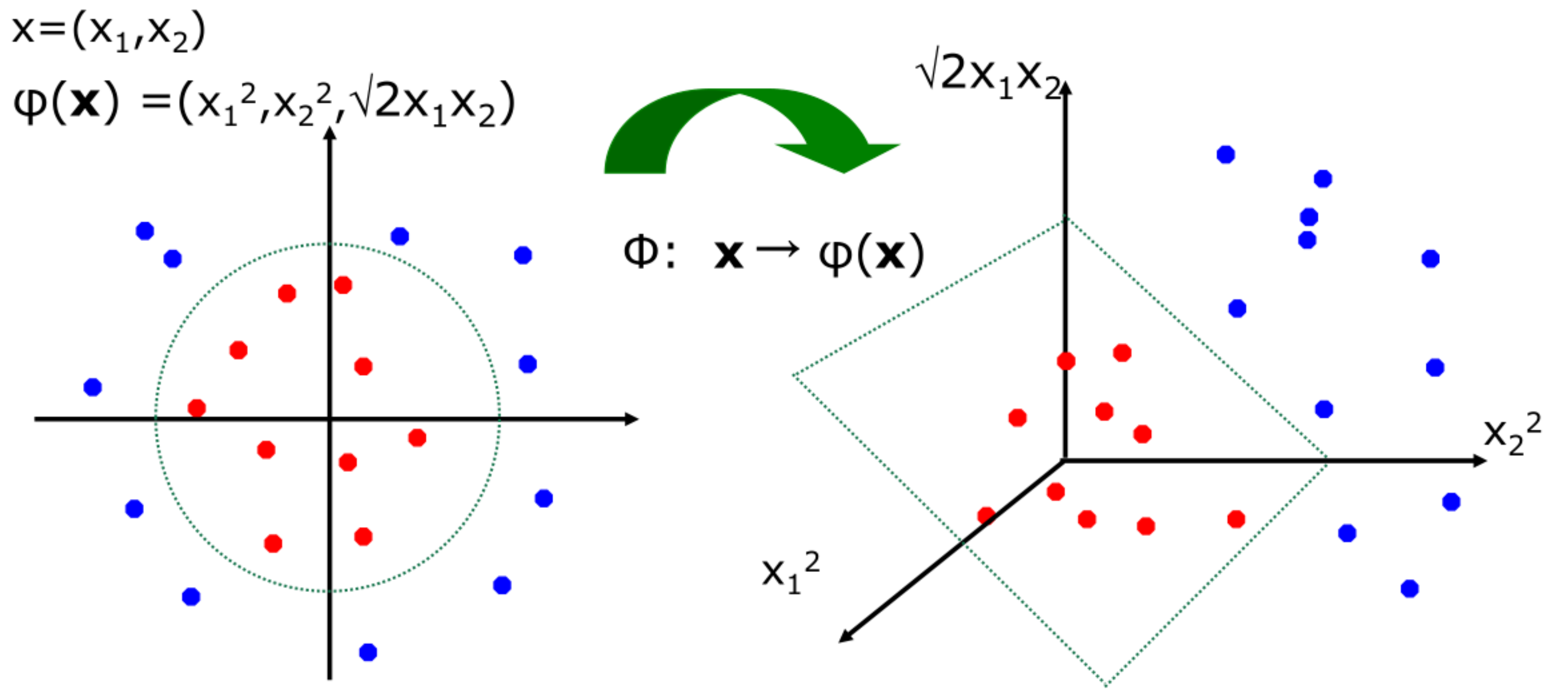
$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i y_i = 0, \ \alpha_i \geq 0, \ i = 1, 2, \ldots, m.$$

- **支持向量机的求解方式**

- 具体而言，求解方式可采用序列最小优化算法（SMO）进行求解
  - 基本思路：不断执行如下两个步骤直至收敛
    - 第一步：选定以对需要更新的向量α_i 和 α_j
    - 第二步：固定α_i 和 α_j以外的参数，求解对偶问题以更新α_i 和 α_j
      - 挑选α_i 和 α_j时，注意挑选差别较大的一对以尽快提升目标函数
  - 在得到所有的α_i之后，即可借助 $\alpha_i(y_i f(\boldsymbol{x}_i)-1)=0$ 获得其他参数w和b的解，其中 $f(\boldsymbol{x})=\boldsymbol{w}^\top\boldsymbol{x}+b$

- **线性不可分问题与核函数**

- 如果数据集线性不可分，不存在这样一个超平面，怎么办？
  - 将样本映射到一个更高维的特征空间，使得在这个特征空间线性可分



- **为什么需要核函数**

- 基于核函数的方法计算两个样本之间的内积
  - 从前面的回代式可以发现，优化过程需要计算内积 $\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boxed{\boldsymbol{x}_i^{\top}\boldsymbol{x}_j} - \sum_{i=1}^{m}\alpha_i$
  - 核函数的概念：
    - 某些样本在低维空间时线性不可分，通过非线性映射将其映射到高维空间的时候则线性可分，但非线性映射的形式、参数等难以确定。
    - 核函数的目的，在于将高维空间下的内积运算转化为低维空间下的核函数计算，从而避免高维空间可能遇到的“维度灾难”问题。

• 如何选择核函数

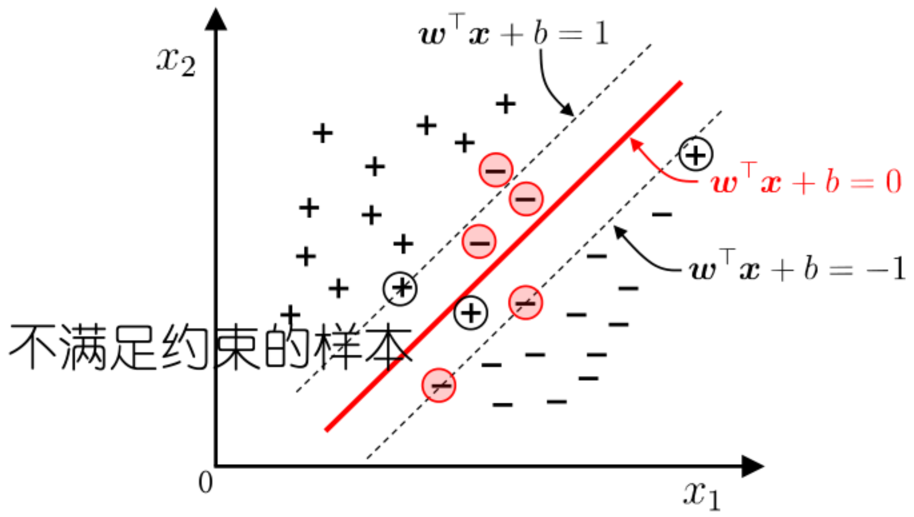
- 常见的核函数如下表所示：

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j)^d$	$d \geq 1$为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\lVert\boldsymbol{x}_i - \boldsymbol{x}_j\rVert^2}{2\delta^2}\right)$	$\delta > 0$为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\lVert\boldsymbol{x}_i - \boldsymbol{x}_j\rVert}{\delta}\right)$	$\delta > 0$
Sigmoid核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \boldsymbol{x}_i^\top \boldsymbol{x}_j + \theta)$	tanh为双曲正切函数，$\beta > 0$，$\theta < 0$

- 其中，线性核函数与高斯核函数（径向基）是最为常用的
- 常见挑选方法一般为以下两种：
  - 穷举法：一个个试过来，选择效果最好的一种
  - 混合法：将多个不同的核函数混合起来使用

- **软间隔问题**

- 现实中，有时可能既没有办法找到一个线性可分的超平面，也无法确定合适的核函数使得训练样本线性可分。另外，也有过拟合的可能
  - 一种解决方法是：引入“软间隔”，允许少数样本不满足超平面约束



本节目录

- 分类评价指标
- 基于规则的分类
- **基于监督学习的分类**
  - 决策树
  - 最近邻分类
  - 支持向量机
  - **Logistic回归**
  - 不平衡分类问题

- **基于Logistic回归的分类**

- 在上节课中，介绍了最基本的线性回归形式：

$$y = w_1 x_1 + \cdots + w_d x_d = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}$$

- 然而，在分类问题中，我们需要的y是离散而非连续的
  - 一种处理方法：对 y 进行分段，对应不同标签
    - 标签有序性假设，如何确定分段阈值?
  - 另一种处理方法：对线性回归的形式进行变更

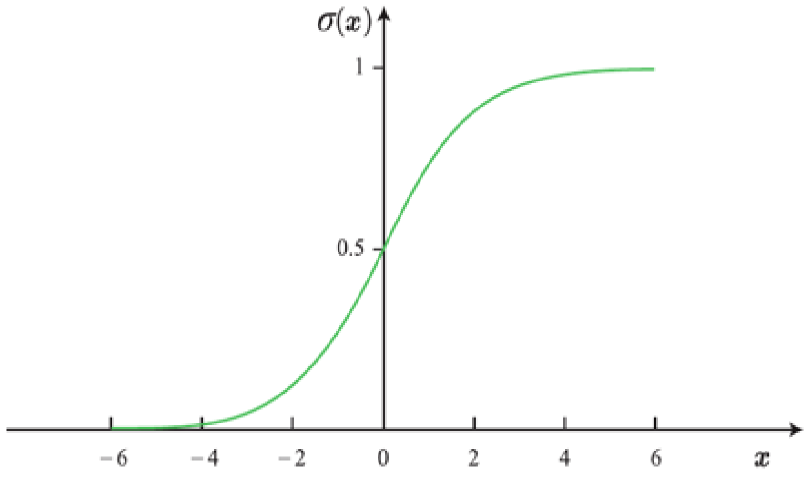
Logistic

- **基本方法：基于Logistic回归的分类**

- Logistic回归提供了解决方案：将连续输入映射到[0, 1]区间
  - 类似的形式在涉及信号处理的问题中广泛应用
  - 通过这种形式，函数值本身具有了概率含义

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

- 一个重要的性质
  - x取值较大时，函数值趋近于1
  - x取值较小时，函数值趋近于0



- **基本方法：基于Logistic回归的分类**

- 利用这个属性，我们能做什么？
  - 将回归函数代替前式中的 x，则有

$$p(y=1|\mathbf{x})=\sigma(\mathbf{w}^\top\mathbf{x})=\frac{1}{1+exp(-\mathbf{w}^\top\mathbf{x})}=\frac{exp(\mathbf{w}^\top\mathbf{x})}{1+exp(\mathbf{w}^\top\mathbf{x})}$$

  - 从概率含义出发，该函数就定义了当给定样本 x 和参数 w 时，标签为 y=1的概率
  - 相应的，对于分类问题而言：
    - 一种思路是概率之间的比较，即P(y=1|x; w) > P(y=0|x; w) 意味着标签为1
    - 另一种则是根据事先划定的阈值，即 P(y=1|x; w) > *thres* 则标签为1
      - 阈值通常为0.5，但也可根据实际情况选定

- **基本方法：基于Logistic回归的分类**

- 那么，如何估计所需的参数?
  - 在给定参数 w 有，每个样本属于各自对应的**真实类别**的概率为

$$
\begin{aligned}
p(y_i|\mathbf{x}_i;\mathbf{w}) &= \begin{cases} \sigma(\mathbf{w}^\top\mathbf{x}_i) & \text{if} \quad y_i = 1, \\ 1-\sigma(\mathbf{w}^\top\mathbf{x}_i) & \text{if} \quad y_i = 0 \end{cases} \\
&= \sigma(\mathbf{w}^\top\mathbf{x}_i)^{y_i}(1-\sigma(\mathbf{w}^\top\mathbf{x}_i))^{1-y_i}
\end{aligned}
$$

  - 那么，相应的，可以采用极大似然估计法来估计所需的参数 w，即极大化以下函数

$$
\begin{aligned}
\ell(\mathbf{w}) &= \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i;\mathbf{w}) \\
&= \sum_{i=1}^{N} y_i \log \sigma(\mathbf{w}^\top\mathbf{x}_i) + (1-y_i)\log(1-\sigma(\mathbf{w}^\top\mathbf{x}_i))
\end{aligned}
$$

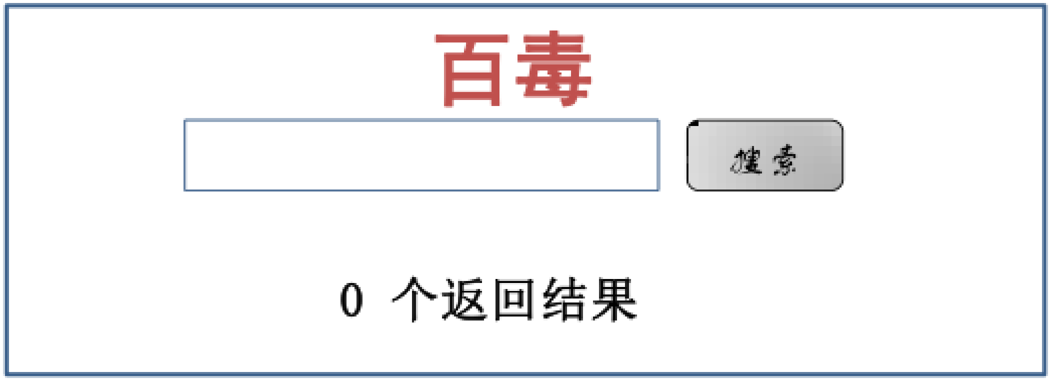
本节目录

- 分类评价指标
- 基于规则的分类
- **基于监督学习的分类**
  - 决策树
  - 最近邻分类
  - 支持向量机
  - Logistic回归
  - **不平衡分类问题**

- **不平衡分类问题**

- 具有不平衡分布的数据集在日常应用中极为常见
  - 例如，考试中及格的学生一般是绝大多数（神捕的课除外）
  - 如果依赖于效果指标优化分类结果，在不平衡分类条件下可能被误导

如何以最低的代价做一个Accuracy接近 100% 的搜索引擎？



• 解决方案（1）代价敏感学习

- 引入代价矩阵，衡量将一个类错分到另一个类的代价
  - 例如，宁可多判断一些疑似患者，不能漏掉一个病人的情况下，可将“病人”错分至“健康”的代价设为100，而“健康”错判的代价为1
  - 最终优化目标由原先的准确/召回变更为加权后的代价

		True Class j			
		1	2	…	k
Predicted Class i	1	$C(1,1)$	$C(1,2)$	…	$C(1,k)$
	2	$C(2,1)$	…	…	⋮
	⋮	⋮	…	…	⋮
	k	$C(k,1)$	…	…	$C(k,k)$

• 解决方案（2）抽样方法

- 另一种方法是基于采样的方法，通过改变样本分布来缓解不平衡问题
- 常见的策略有以下三种：
  - 过采样：提升少数类的样本比例，例如从少数类中进行重复随机采样
  - 欠采样：降低多数类的样本比例，例如用多数类的采样样本而非全样本
  - 混合采样：同时采用前两种策略
    - 也可以利用已有的少数类样本，通过K-最近邻生成新样本
    - 采样中要注意噪声问题，噪声也可能被复制多次

本章小结

分类

- 分类的评判指标
- 基于规则的分类方法
- 基于监督学习的分类方法
  - 决策树
  - 最近邻分类
  - 支持向量机
  - Logistic回归
  - 不平衡分类问题

tongxu@ustc.edu.cn