数据科学导论

Introduction to Data Science

第四章 数据挖掘基础

黄振亚，陈恩红
Email: huangzhy@ustc.edu.cn, cheneh@ustc.edu.cn

课程主页：
http://staff.ustc.edu.cn/~huangzhy/Course/DS2022.html

数据挖掘基础

- 基本概念——数据挖掘是什么？
  - **数据挖掘**：从大量的数据中挖掘哪些令人感兴趣的、有用的、隐含的、先前未知的和可能有用的模式或知识，并据此更好的服务人们的生活。

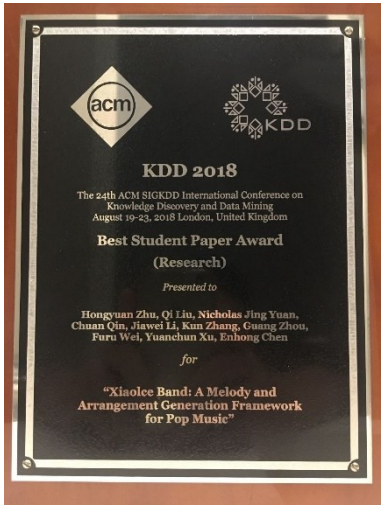
图1-2 我们的数据丰富，但信息贫乏

数据挖掘：在你的数据中搜索知识（有趣的模式）

数据挖掘基础

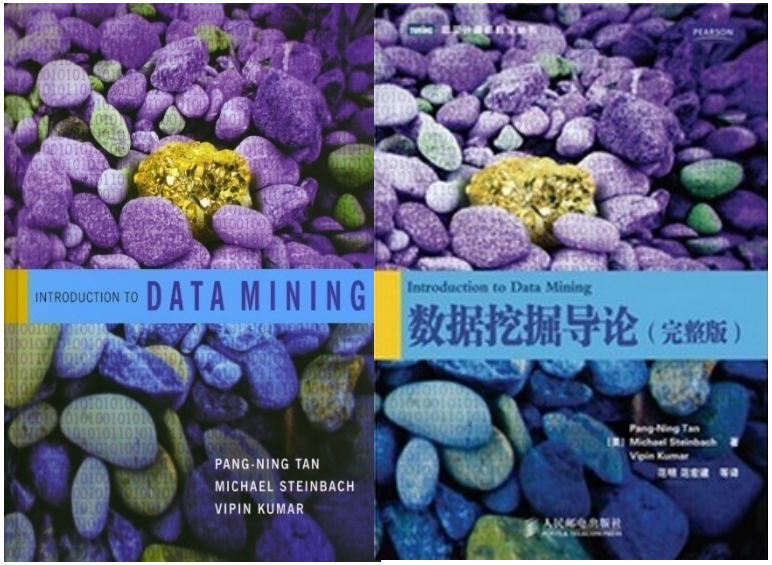
- 基本概念——数据挖掘是什么？
  - 数据挖掘的近义词
    - 从数据中挖掘知识
      - Knowledge Discovery in Data
    - 知识提炼
    - 数据/模式分析
    - 数据考古
    - 数据捕捞、信息收获、资料勘探等。

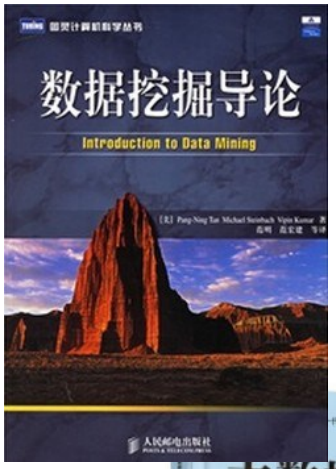
- SIGKDD: Knowledge Discovery and Data Mining

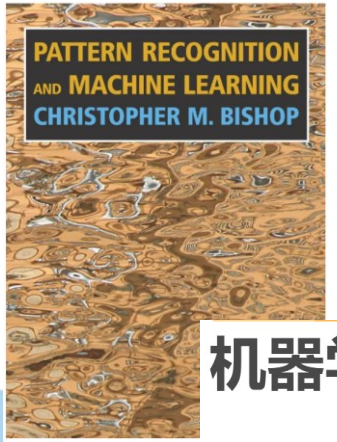


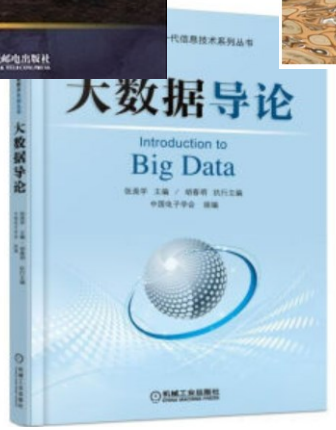
数据挖掘基础

- 参考书
  - 数据挖掘导论 （Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Addison Wesley ）







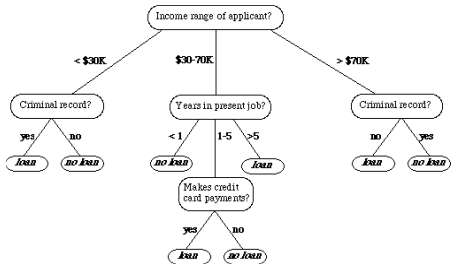


机器学习

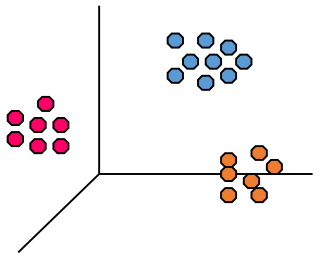
数据挖掘基础

- 数据挖掘有哪些典型任务?

分类与预测



聚类



数据

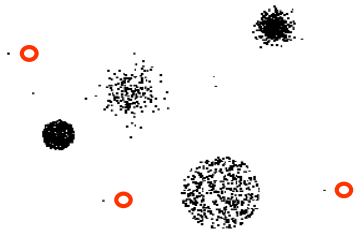
	T		H		P	
	L	H	L	H	L	H
J	-6.0	8.8	60	100	986	1044
F	-2.8	10.9	48	100	973	1025
M	-5.6	17.7	34	100	976	1037
A	-1.2	22.2	27	100	996	1036
M	-0.8	27.8	25	100	1003	1034
J	5.2	29.1	26	100	998	1030
J	9.8	30.6	23	99	997	1027
A	5.6	26.1	31	100	992	1029
S	5.2	24.8	35	100	998	1028
O	-0.4	21.3	42	100	990	1031
N	-7.6	17.3	55	100	963	1023
D	-10.4	9.2	53	100	987	1039

table 17a

2010 monthly weather variation, Cambridge (UK)

关联分析

异常检测



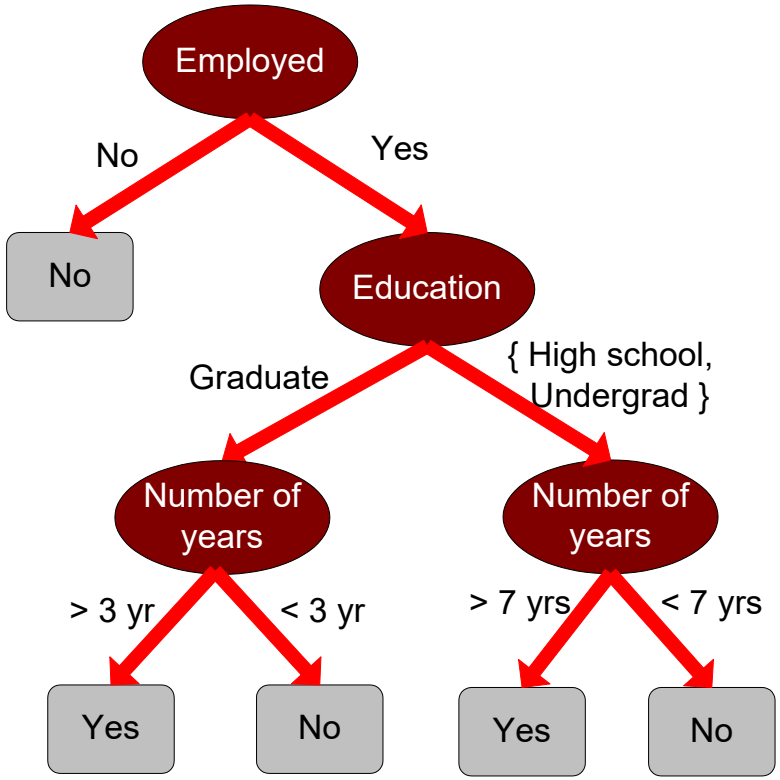
数据挖掘基础

- 数据挖掘任务——分类与预测 (Classification, Prediction)
  - 预测性建模（监督学习）
  - 寻找一个模型：特征–>类别的函数

类别

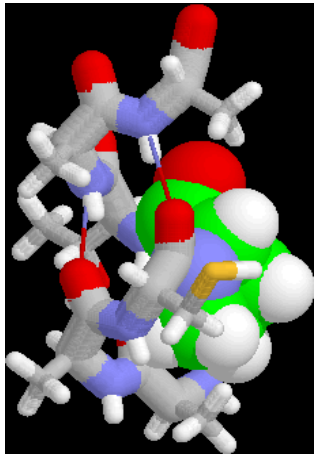
Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
…	…	…	…	…

Model for predicting credit worthiness（信誉）



数据挖掘基础

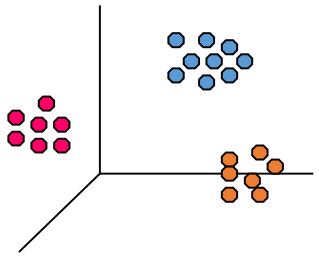
- 数据挖掘任务——分类与预测 (Classification, Prediction)
  - 邮件分类（垃圾邮件）
  - 将新闻故事分类为财经、天气、娱乐、体育等
  - 判断信用卡交易是合法的还是欺诈
  - 将蛋白质的二级结构分类为α -螺旋、β -薄片或随机螺旋
  - 预测肿瘤细胞是良性还是恶性
  - 识别网络空间的入侵者
  - 推荐系统
  - 。。。

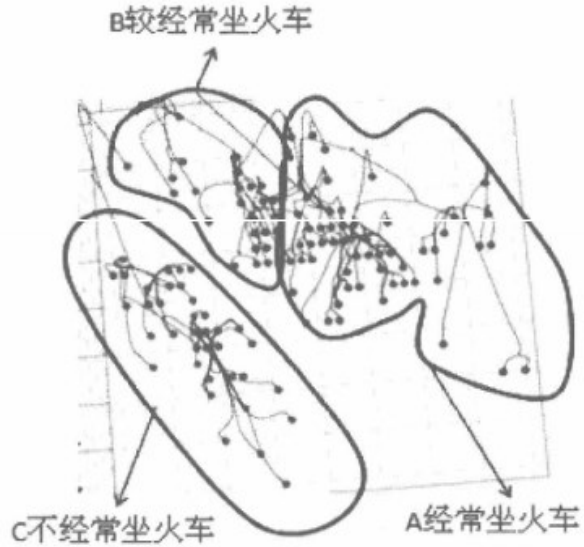


数据挖掘基础

- 数据挖掘任务——聚类(Clustering, unsupervised learning)
  - 例如：铁路票价制定
  - 问：如何制定合适的票价提高上座率?
  - 方案：将旅客进行聚类分析，根据旅客乘坐高铁的频率 提供不同的优惠政策。合适的定价是提高高铁上座率的保障

聚类





数据挖掘基础

- 数据挖掘任务——聚类(Clustering)
  - 例：搜索词条聚类(Query clustering)
  - “USTC”，“中科大”，“中国科大”，“中国科学技术大学”
  - “长城”，“颐和园”，“故宫”

数据挖掘基础

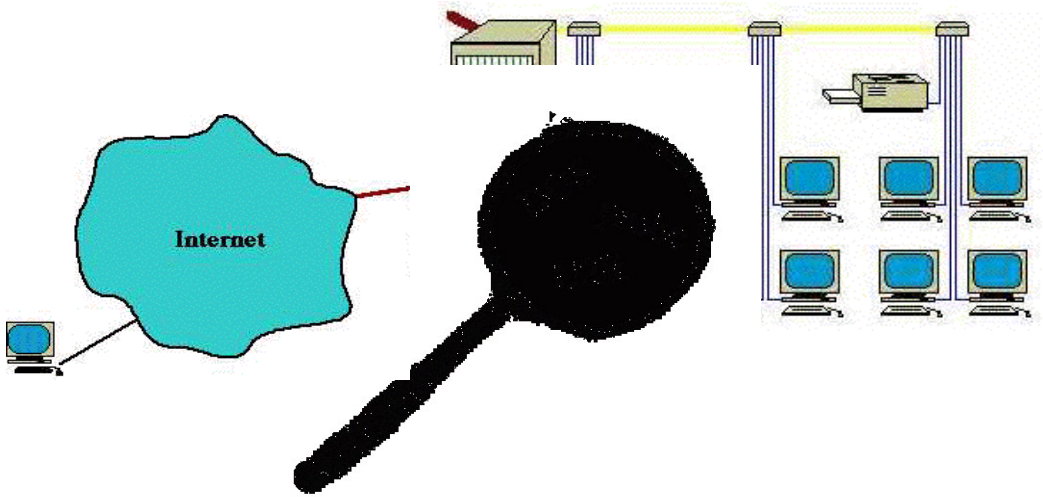
- 数据挖掘任务——关联分析(Association Analysis)
  - 例如： “啤酒与尿布”
  - 在一次圣诞节的顾客消费行为分析中，沃尔玛意外发现跟尿布一起购买最多的商品竟然是啤酒。经过深入分析后，卖场立即对两类商品的空间距离与价格都进行了调整，结果尿布与啤酒销量双双大增。

萨姆·沃尔顿
沃尔玛公司创始人

轰动一时的啤酒与尿布关联规则

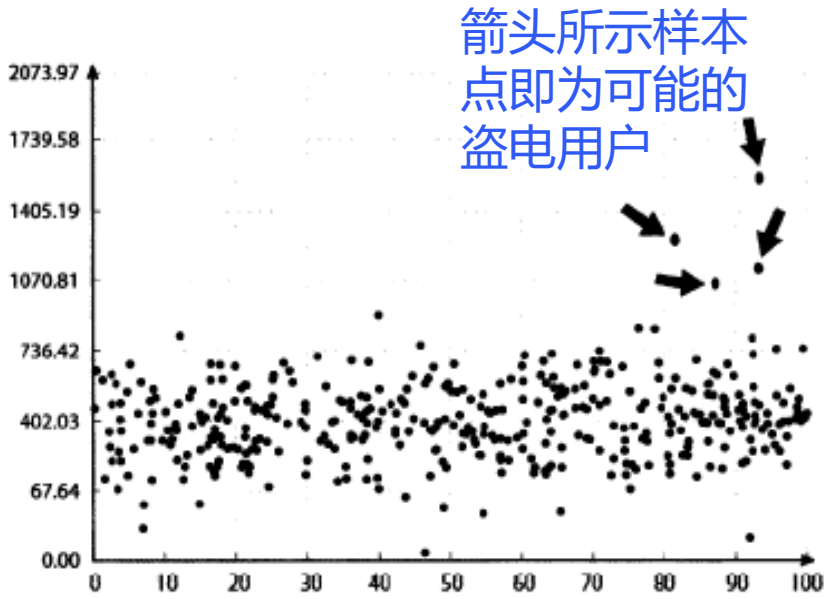
数据挖掘基础

- 数据挖掘任务——异常检测 (Anomaly Detection)
  - 检测非正常行为
  - 应用:
    - 信用卡诈骗检测
    - 网络入侵检测



数据挖掘基础

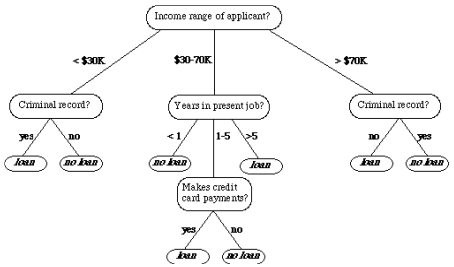
- 数据挖掘任务——异常检测 (Anomaly Detection)
  - 例：电力行业--盗电检测
    - 盗电用户行为特征与普通用户不同（电费与人员、产值、税收等形成反差）
    - 通过对用户用电数据的聚类分析，反窃电的业务人员就能对锁定的目标重点侦察，既能提高窃电客户的识别率，还能节省电力部门人力资源，为反窃电提供了另一种思路。



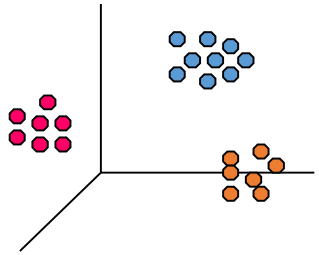
数据挖掘基础

- 数据挖掘——四个任务有哪些常用方法?

分类与预测



聚类



数据

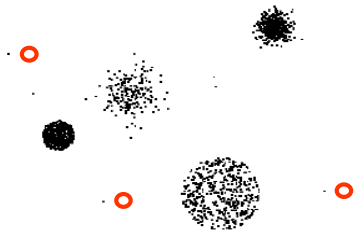
	T		H		P	
	L	H	L	H	L	H
J	-6.0	8.8	60	100	986	1044
F	-2.8	10.9	48	100	973	1025
M	-5.6	17.7	34	100	976	1037
A	-1.2	22.2	27	100	996	1036
M	-0.8	27.8	25	100	1003	1034
J	5.2	29.1	26	100	998	1030
J	9.8	30.6	23	99	997	1027
A	5.6	26.1	31	100	992	1029
S	5.2	24.8	35	100	998	1028
O	-0.4	21.3	42	100	990	1031
N	-7.6	17.3	55	100	963	1023
D	-10.4	9.2	53	100	987	1039

table 17a

2010 monthly weather variation, Cambridge (UK)

关联分析

异常检测



分类与预测

- 数据挖掘任务 — 分类与预测

该图片有关联

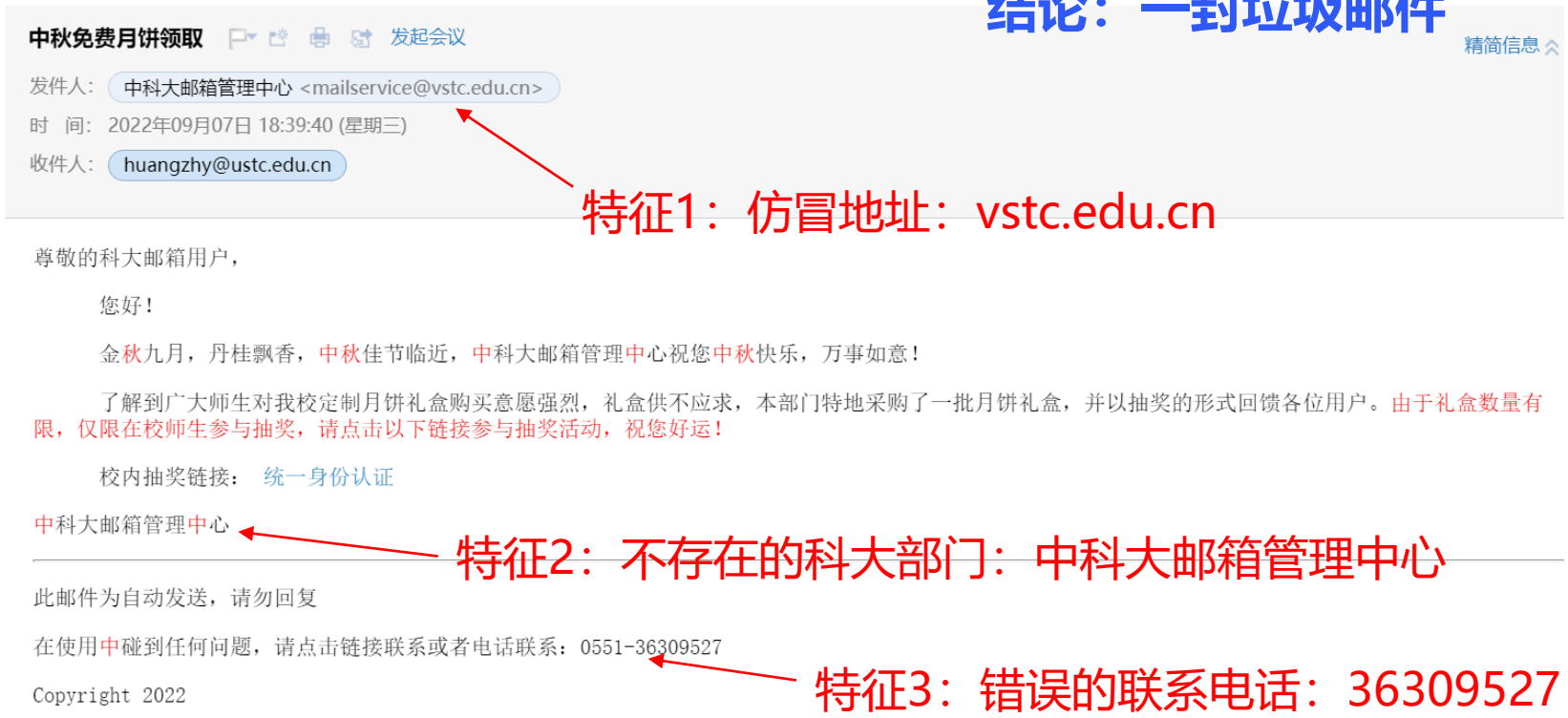
该图片是科大

如果数据有标签，即已知图片是科大，则可以预测新图片的类

分类与预测

- 案例一：垃圾邮件分类 — 中科大安全演练
  - 判断：下面这封邮件是垃圾邮件吗？

结论：一封垃圾邮件



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

分类与预测

- 案例二：电影评分预测
  - 预测：用户对电影《功夫》的评分是多少？

    已知：他对4部电影的评分分别为：5.0，4.8，4.9，4.5

特征1：喜欢周星驰

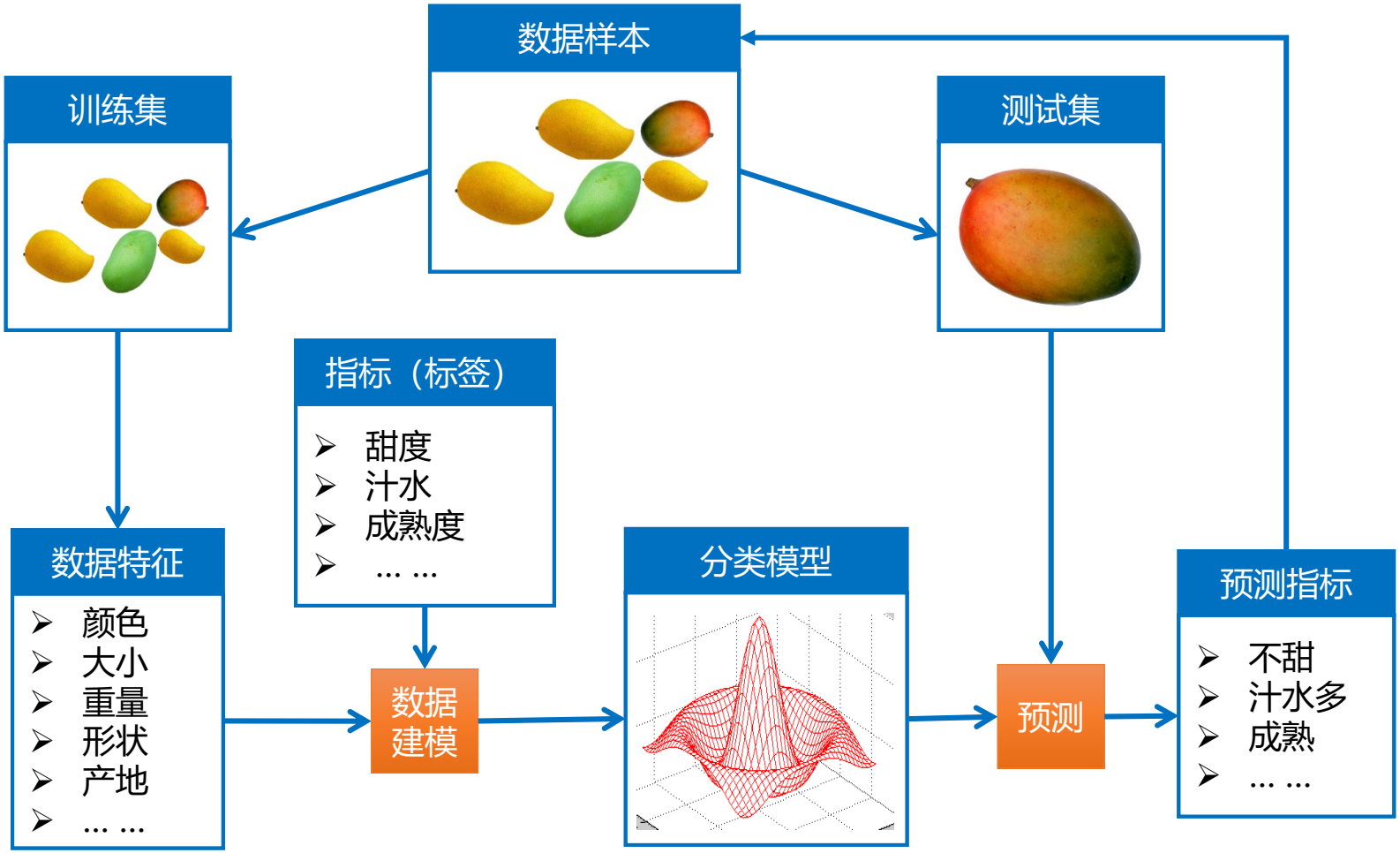
特征2：喜欢喜剧

结论：预测评分5分

基于一些特征与规则，我们可以将电影评分(连续值)估计视作一个预测问题

分类与预测

- 生活中的案例：直观理解买芒果



分类与预测

- 分类与预测 — 基本定义
  - 已知：一组数据（训练集） (X, Y)
  - 如右图，每一条记录表示为 (x, y)
    - x：数据特征/属性（如收入）
    - y：类别标记（是否有借款）
  - 任务:
    - 学习一个模型，利用每一条记录的特征x去预测它对应的类别y

即：输入未标记的数据（含特征x），预测数据的类别y

分类 / 数值预测 取决于 类别标签是离散型 / 数值型

3个特征：

- 是否有住房
- 婚姻状态
- 年收入

类别：

是否拖欠贷款

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

分类与预测

- 分类与预测 — 回顾前例

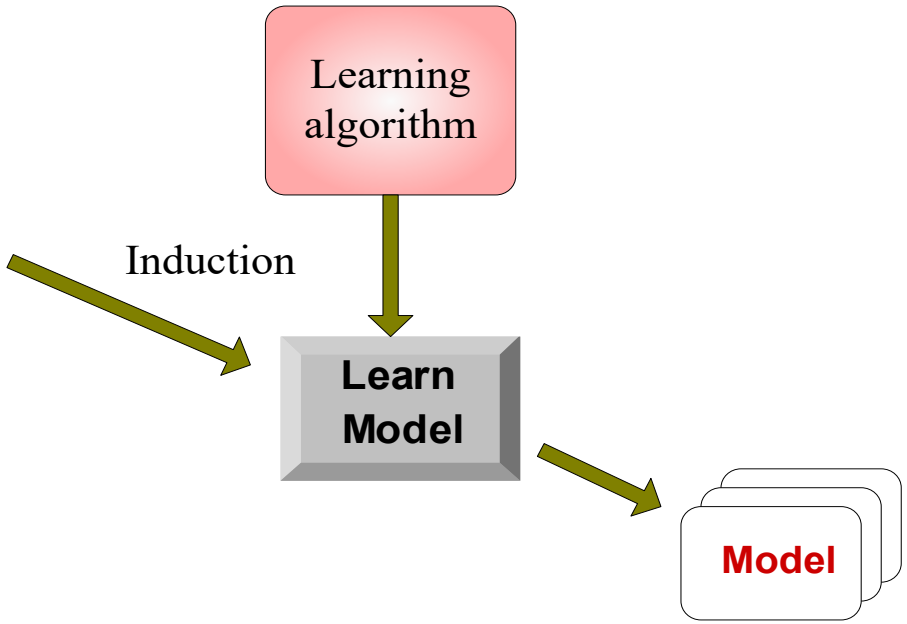
任务	特征x	类别y
垃圾邮件分类	收件人、邮箱名、邮件内容等	是否时垃圾邮件 **离散型**
电影评分预测	用户在其他电影的评分 电影的演员，类型等	实值评分[0,5] **数值型**
芒果好坏预测	芒果的颜色、大小、重量、形状、产地等	芒果的甜度、水分、成熟与否 **离散型 或 数值型**

分类与预测

- 如何建立分类与预测模型？
  - 一般流程：有监督学习
  - 通常包括两个阶段：模型训练、模型预测
    - 模型训练：目标是利用训练数据，学习一个分类或预测模型

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

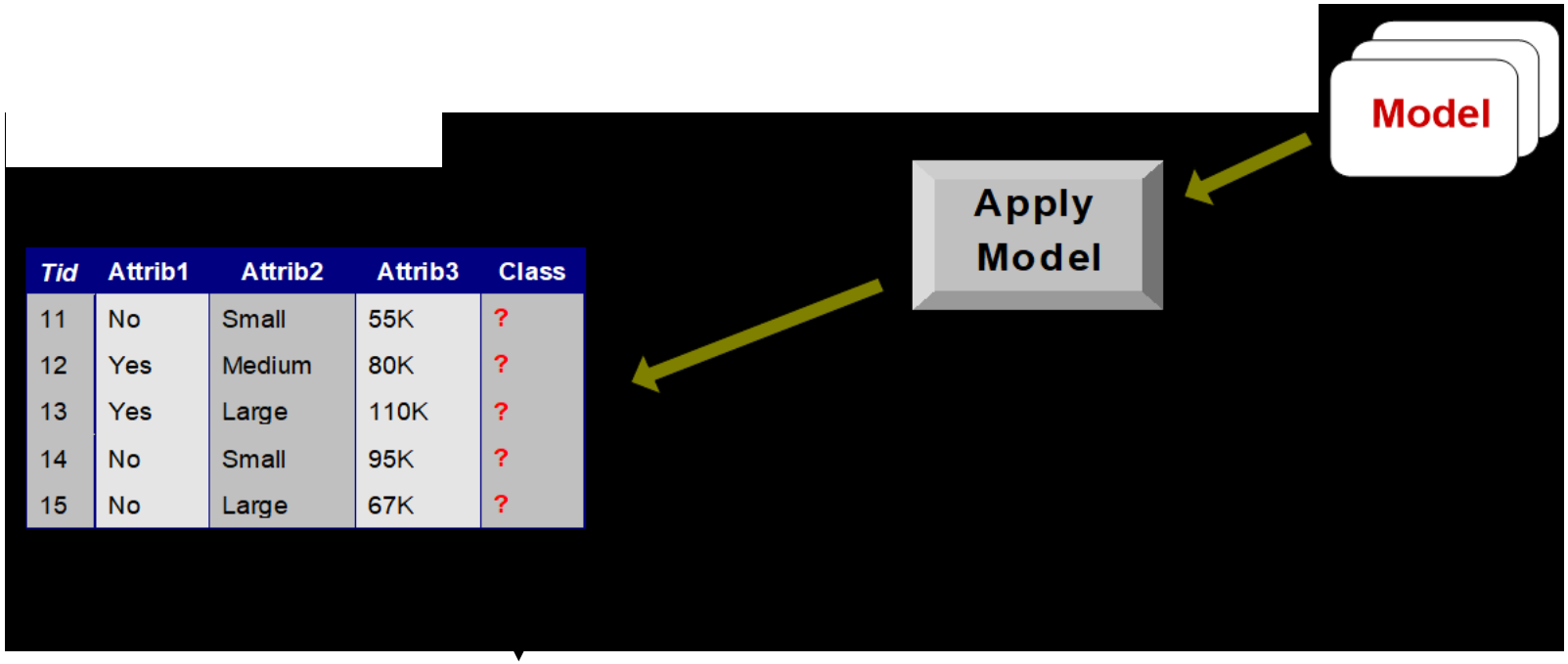
Training Set



训练集有类别标签

分类与预测

- 如何建立分类与预测模型？
  - 一般流程：有监督学习
  - 通常包括两个阶段：模型训练、模型预测
    - 模型预测：目标是利用学习的模型，预测测试数据的标签



测试集无类别标签，需要预测

分类与预测

- 有监督学习：分类与预测
- 常用方法
  - 规则方法
  - 决策树
  - 贝叶斯方法
  - 最近邻方法
  - 支持向量机（SVM）
  - 神经网络
  - 集成方法
- 分类的评价指标

分类与预测

- 回顾垃圾邮件分类的例子
  - 判断：下面这封邮件是垃圾邮件吗？

结论：一封垃圾邮件

分类：规则方法

- **规则方法**
  - 基于规则的分类器（Rule-based Classifier）就是使用一组 if-then 的模式来进行分类
  - 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
  - 最基础的获得规则的方法：人工制定规则进行分类

↓

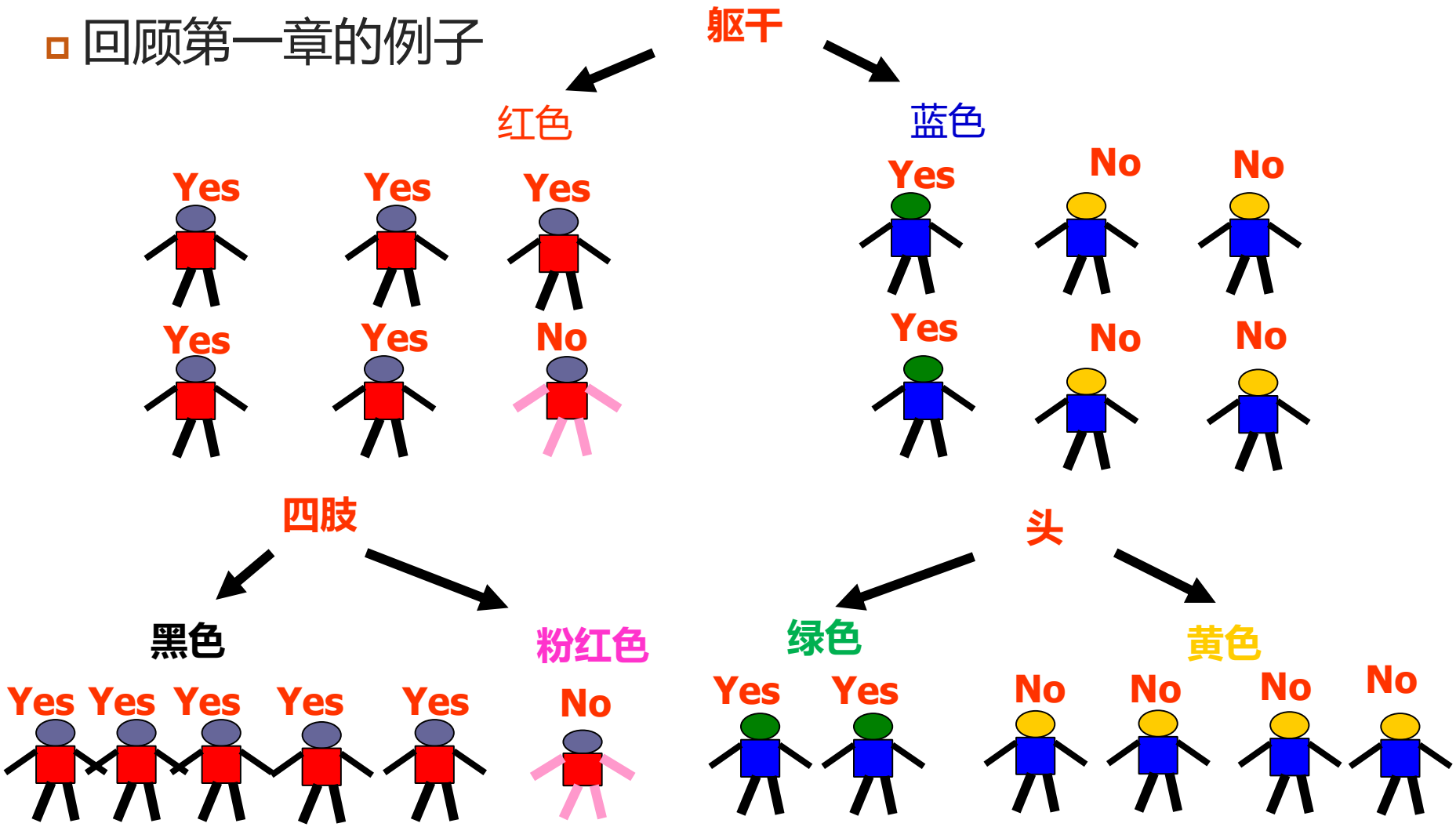
不足：人工定义规则、效率低，难以处理复杂问题

↓

自动生成规则？决策树

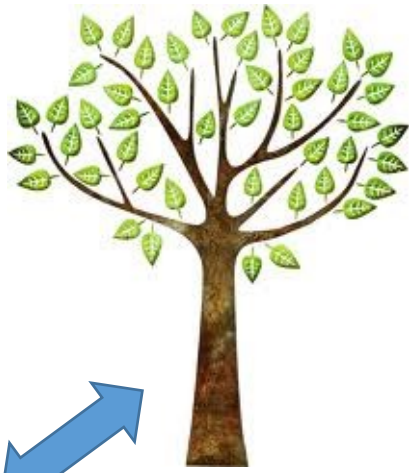
分类：决策树

- 回顾第一章的例子



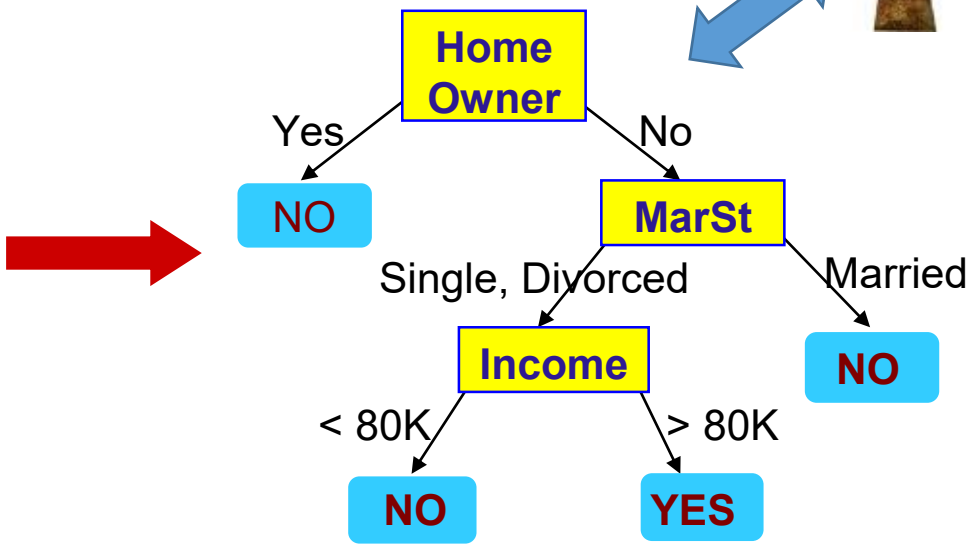
分类：决策树

- 什么是决策树
  - 对数据进行处理，利用归纳算法生成可读的规则
  - 模型以树状形式呈现出来



ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

训练数据

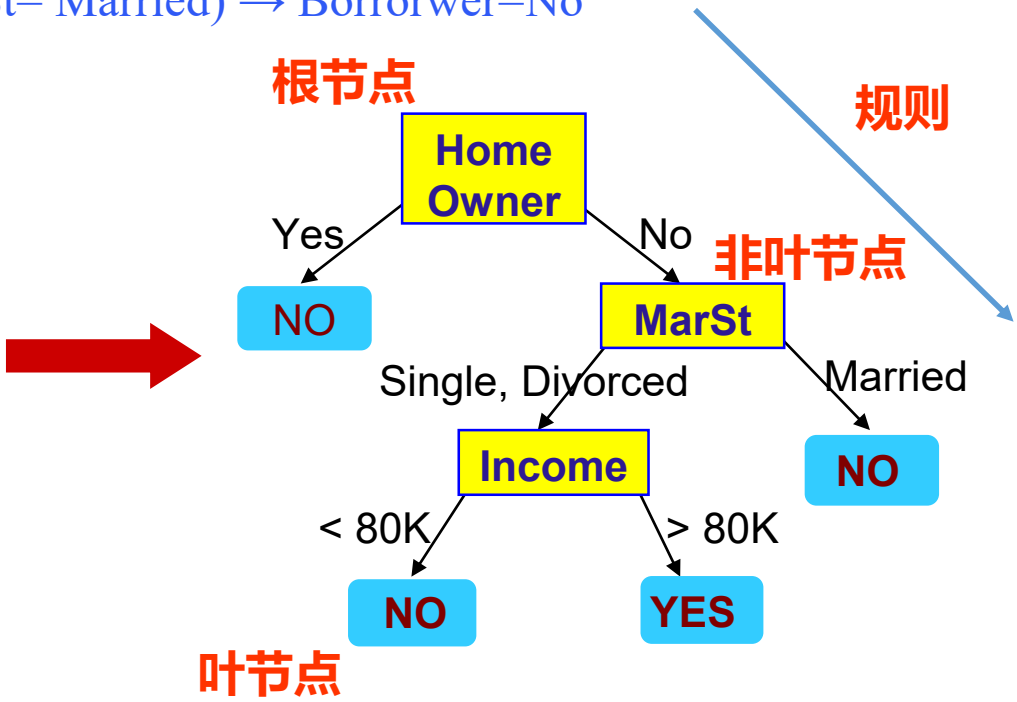


模型：决策树

分类：决策树

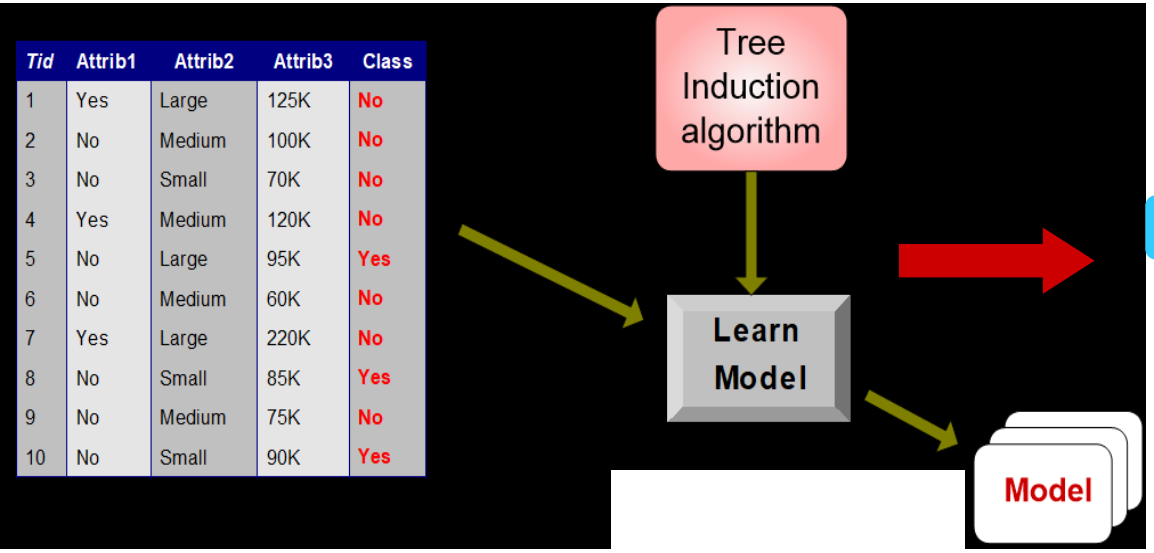
- 什么是决策树 —— 基本概念
  - 非叶节点：一个属性上的测试，每个分枝代表该测试的输出
  - 叶节点：存放一个类标记
  - 规则：从根节点到叶节点的一条属性取值路径
    - (HomOwn= No) ∧ (MarSt= Married) → Borrorwer=No

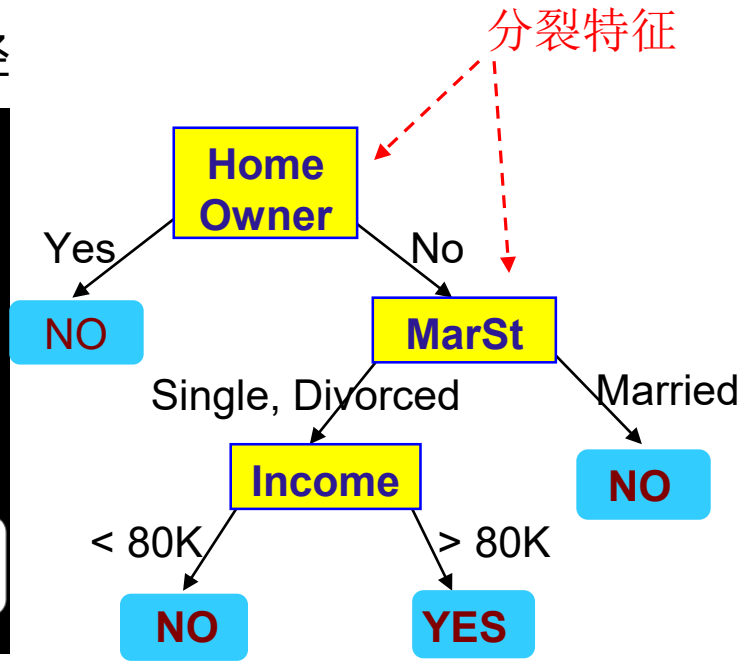
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



分类：决策树

- 建立决策树分类模型的流程
  - 模型训练：从已有数据中生成一棵决策树
    - 分裂数据的特征，寻找决策类别的路径





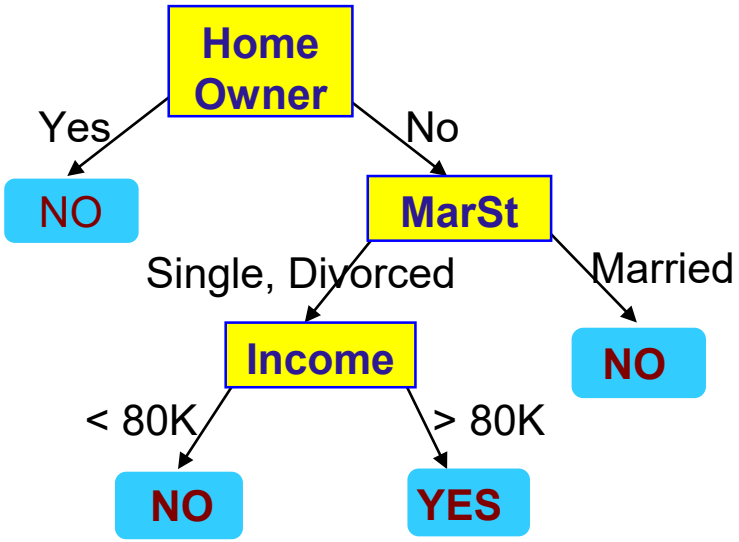
分裂特征:Home Owner, MarSt, Income

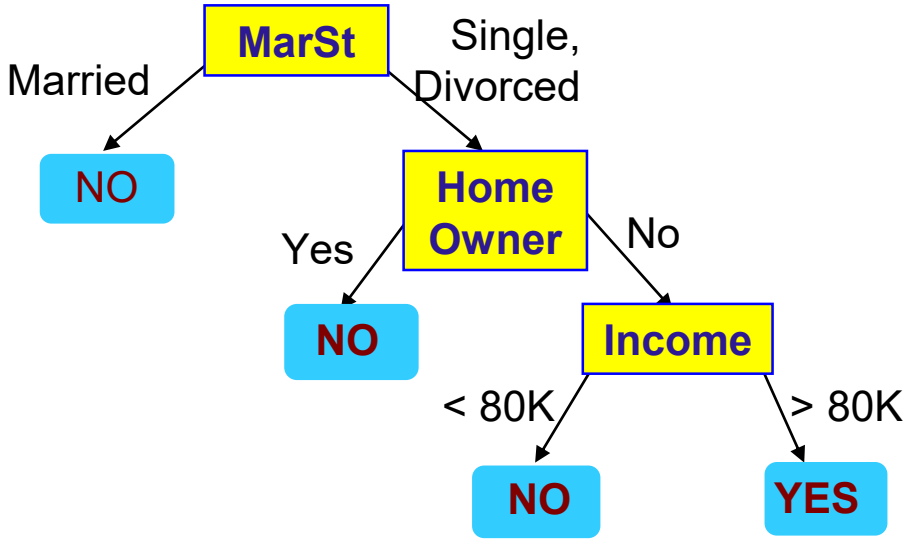
生成模型：决策树

分类：决策树

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- 是否有其他决策树?





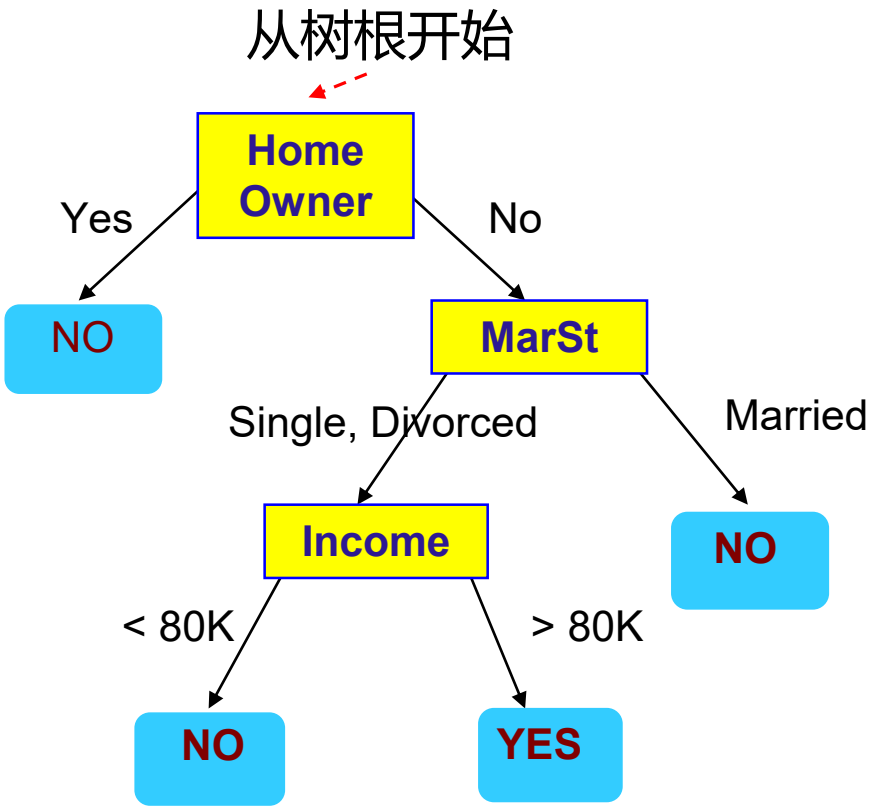
特征顺序: Home Owner, MarSt, Income

特征顺序: MarSt, Home Owner, Income

相同的数据，根据不同的特征顺序，可以建立多种决策树

分类：决策树

- 决策树分类模型的测试过程
  - 模型测试：根据规则将样本分类到某个叶子节点



测试数据（预测标签）

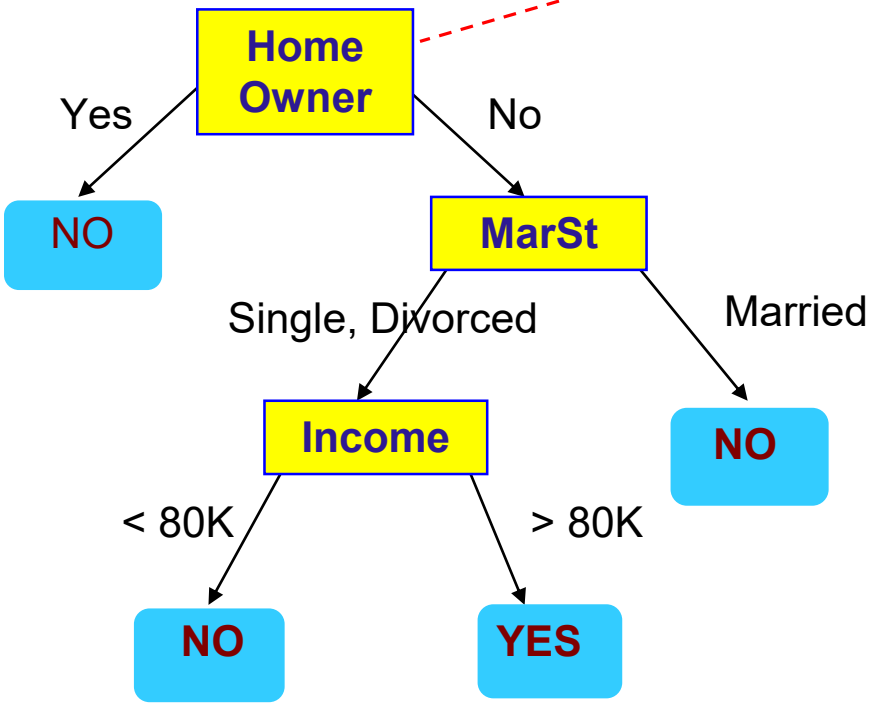
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

分类：决策树

- 决策树分类模型的测试过程

测试数据（预测标签）

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

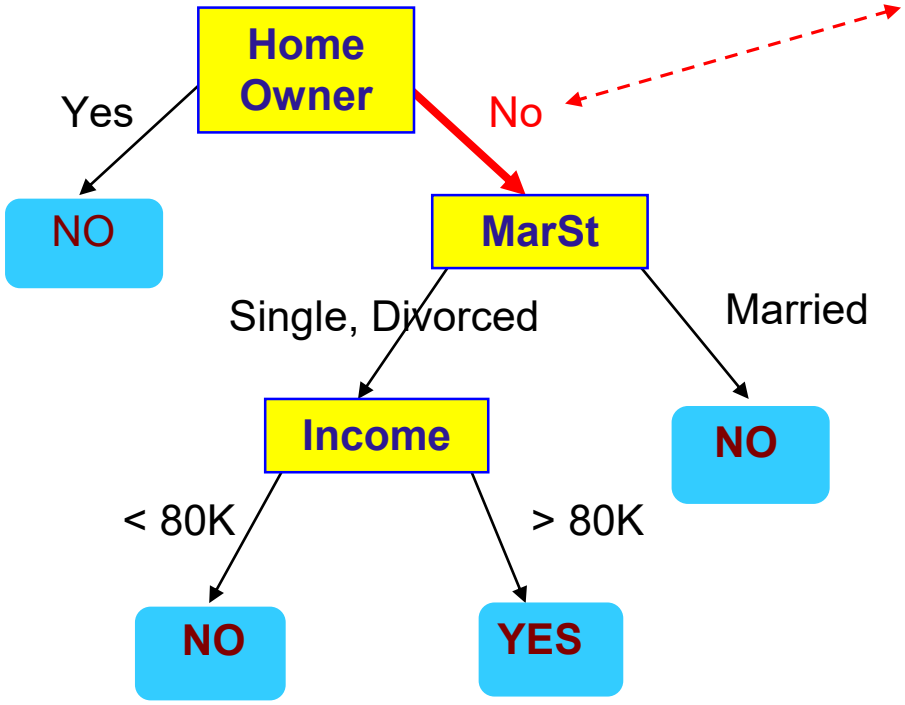


分类：决策树

- 决策树分类模型的测试过程

测试数据（预测标签）

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

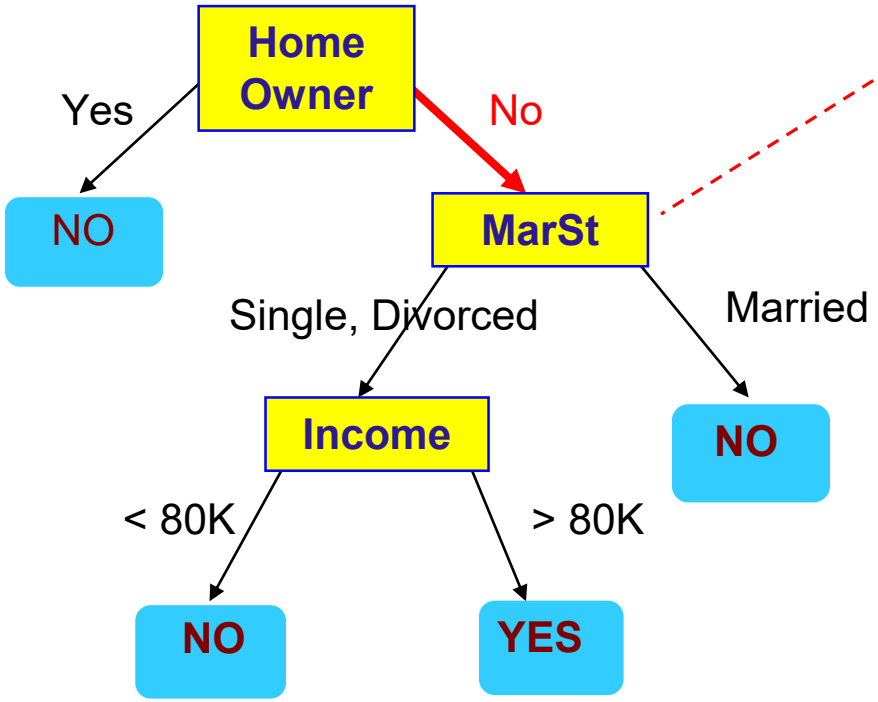


分类：决策树

- 决策树分类模型的测试过程

测试数据（预测标签）

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

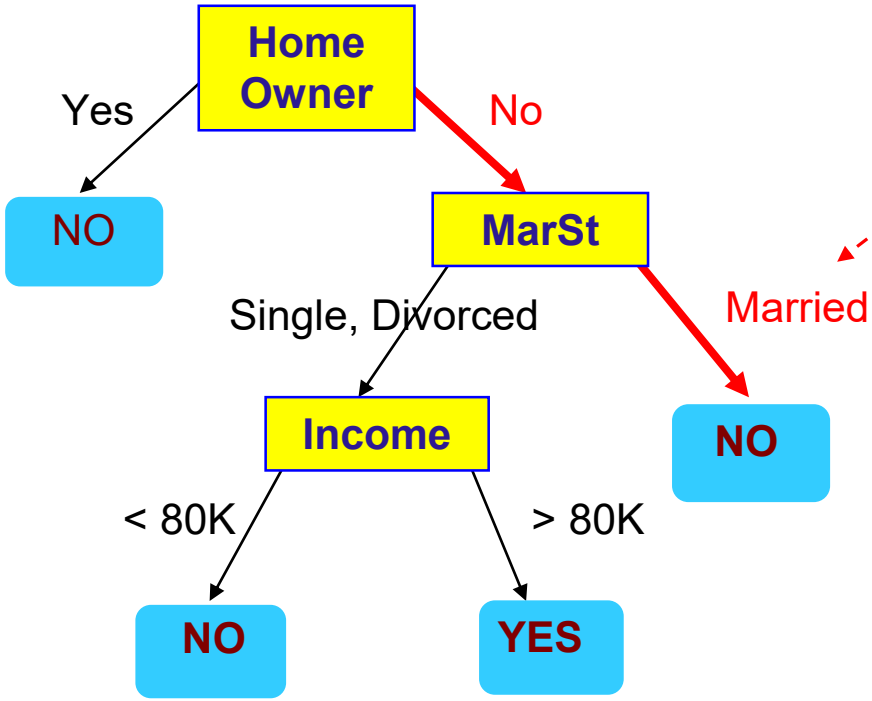


分类：决策树

- 决策树分类模型的测试过程

测试数据（预测标签）

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

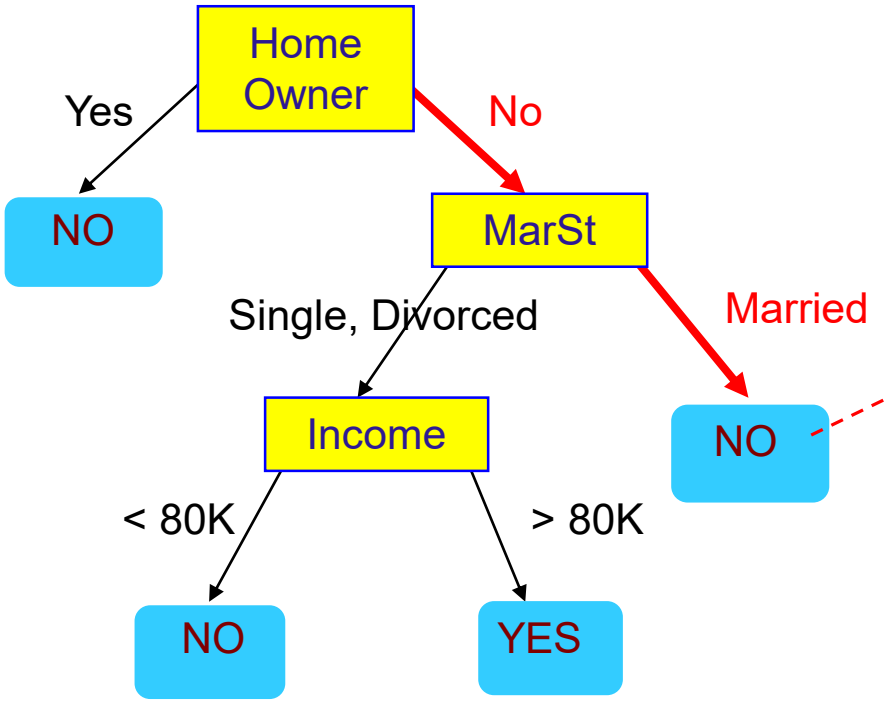


分类：决策树

- 决策树分类模型的测试过程

测试数据（预测标签）

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



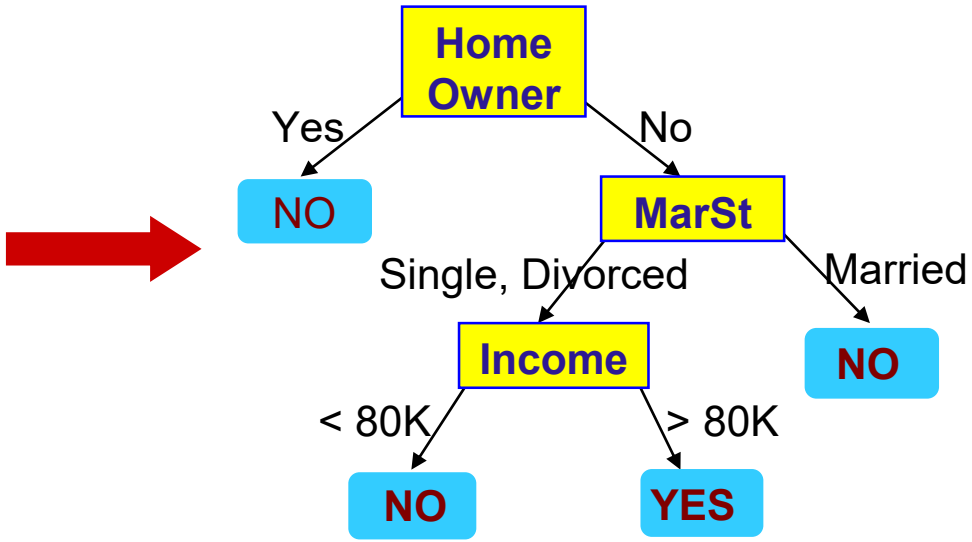
类别Defaulted Borrower 为 “No”

决策过程：未使用所有的特征/属性

分类：决策树

- 如何建立决策树?
  - 基本的决策树学习过程，可以归纳为以下三个步骤：
  - **1. 特征选择：** 选取对于训练数据有着较强区分能力的特征
  - **2. 生成决策树：** 基于选定的特征，逐步生成完整的决策树
  - **3. 决策树剪枝：** 简化部分枝干，避免过拟合因素影响

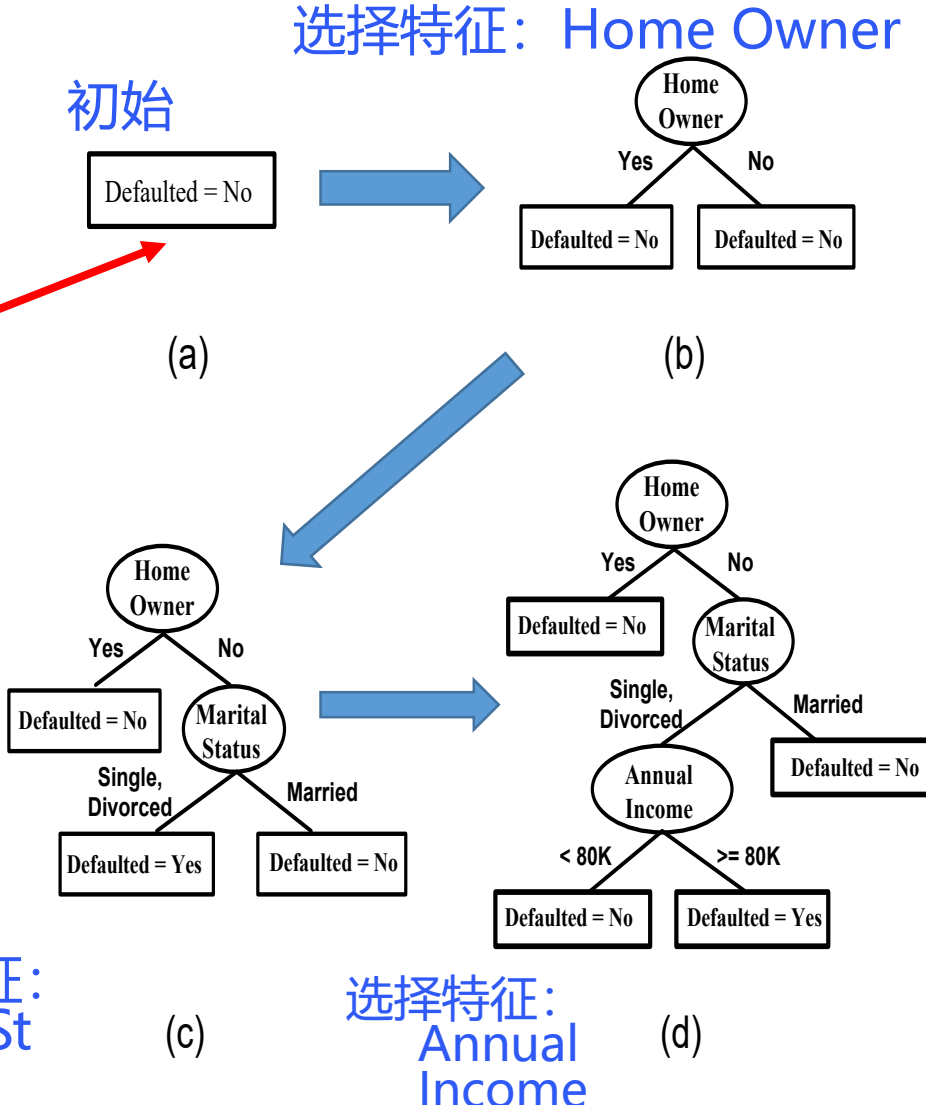
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



分类：决策树

- 1. 特征选择
  - 决策树的基本思想：特征分裂
  - 初始节点包含所有的数据样本，我们希望这些样本能划分到同个类里，如defaulted=No
  - 但往往不成立，因此通过选择特征和取值，将样本集合不断划分

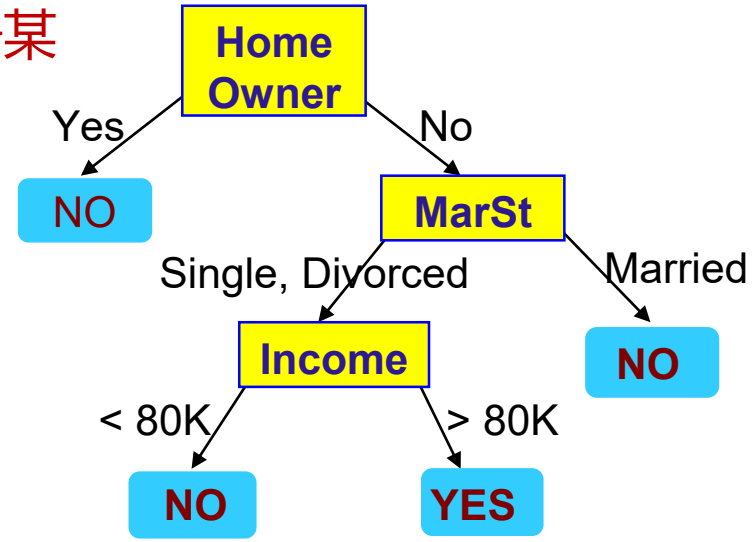
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



分类：决策树

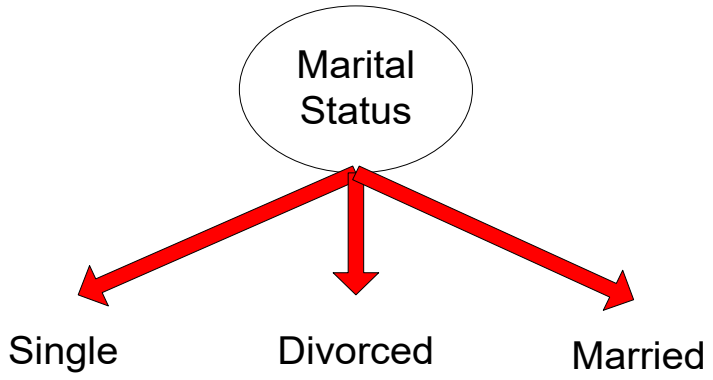
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

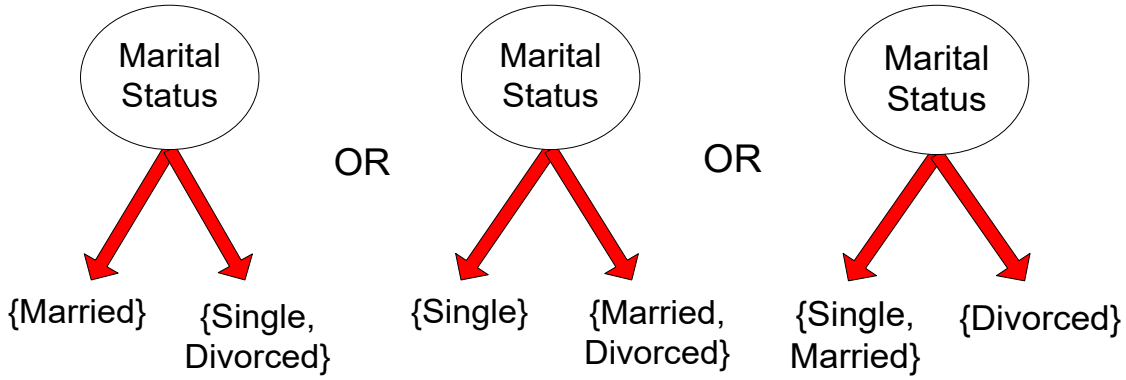
- **1. 特征选择：** 分裂思想的形式化
- Hunt贪心算法
  - D_t：为树中结点t的所有训练样本
  - 若D_t 中的样本属于同一类别y_t, 则t 作为叶子节点，标签为y_t
  - 若D_t 中的样本不属于同一类别, 则根据某特征将D_t 分为更小的样本集
  - 重复上述过程



分类：决策树

- **1. 特征选择：** 分裂过程的形式
  - 多路分裂(Multi-way split)
    - 不同取值均作为一个子集
  - 二分裂(Binary split)
    - 只划分两个子集
    - 需找到最优划分方法





分类：决策树

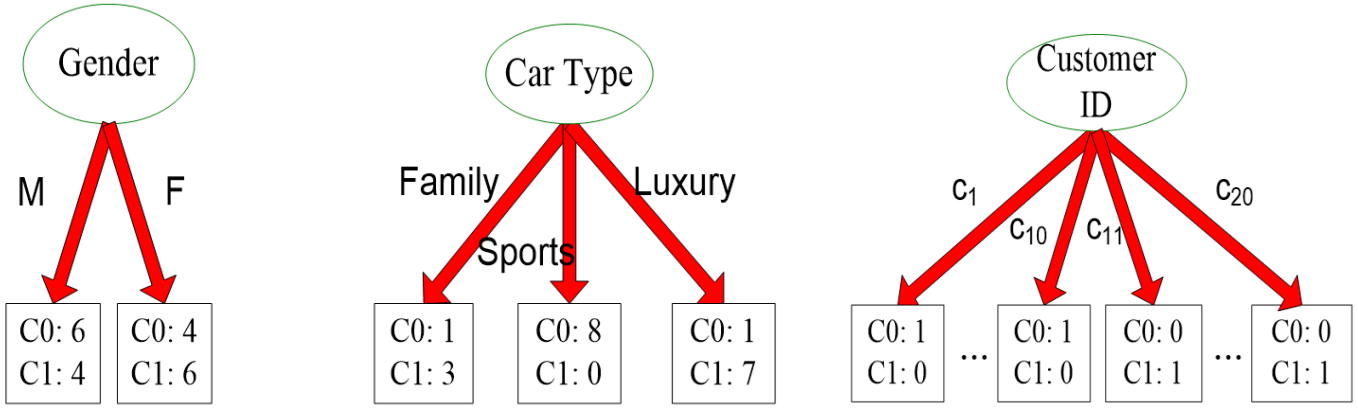
- **1. 特征选择：**分裂过程的两个问题
  - 训练样本如何分裂?
    - 选择分裂特征
    - 评价测试条件
  - 分裂过程何时停止?
    - 理想终止
      - 如果所有记录属于同一类
      - 所有数据有相同的属性值
    - 提前终止

决策树特征选择

- 问题1：如何选择分裂特征？
  - 例子：右图的数据
    - 10个记录类别为0
    - 10个记录类别为1

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1

对不同特征属性进行分裂



哪一种分裂方式最优？

决策树特征选择

- 问题1：如何选择分裂特征？
  - 目标：选取对于训练数据有着**较强区分能力**的特征
    - 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，扔掉这样的特征对学习的精度影响不大
  - 常用特征选择准则
    - 信息增益 → 回顾第二章：数据离散化
    - 信息增益率
    - 基尼指数

决策树特征选择

- 信息增益：信息熵（回顾第二章）
  - 信息熵：计算数据的不确定性

$$Entropy(t) = -\sum_{j} p(j|t) \log p\,(j|t)$$

  - 此时：表示某个节点t （即某个特征）的信息不确定性
    - $p(j \mid t)$ 是节点特征t的属于类别j的样本的比例

- 特点：对于该节点特征t
  - 当样本均匀地分布在各个类别时，熵达到最大值 $\log(n_c)$, 此时包含的信息最少
  - 当样本只属于一个类别时，熵达到最小值 0, 此时包含的信息最多

决策树特征选择

- 计算 某个节点 特征的信息熵（回顾第二章）

$$Entropy(t) = -\sum_{j} p(j|t) \log_2 p(j|t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0　　P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6　　P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6　　P(C2) = 4/6

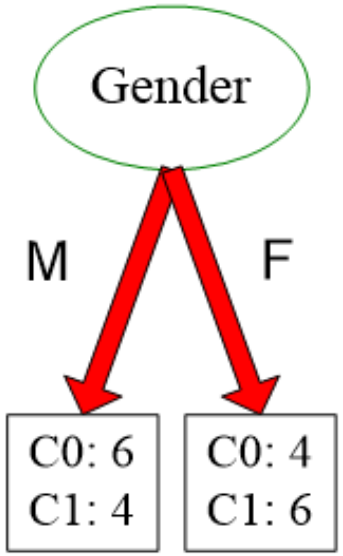
Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

决策树特征选择：信息增益

- 特征选择准则一：信息增益
  - 信息增益: 按某个特征划分之后，数据不确定性降低的程度

$$GAIN(m) = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

  - 第一项表示数据未划分时的信息熵
  - 第二项表示按特征m划分后，数据的信息熵
    - 按特征m划分后，父节点分裂成k个子节点
    - n表示父节点的样本个数
    - n_i 表示子节点i的样本个数



- 选择准则：选择最大的$GAIN$ 对应的特征m

信息增益在ID3和C4.5决策树算法中被应用

决策树特征选择：信息增益

- 特征选择准则—: 信息增益
  - 选择A或B两个特征构造节点，哪种方式好?

特征A	Yes	Yes	No	…	No	No
特征B	No	Yes	Yes	…	Yes	No
类别	C0	C0	C1		C1	C1

$Entropy(p)$

以特征A划分

特征A	Yes	Yes
特征B	No	Yes
类别	C0	C0

特征A	No	…	No	No
特征B	Yes	…	Yes	No
类别	C1		C1	C1

$Entropy(A_{Yes})$ $Entropy(A_{No})$

$$M = \frac{|A_{Yes}|}{n} Entropy(A_{Yes}) + \frac{|A_{No}|}{n} Entropy(A_{No})$$

$$Gain(A) = Entropy(p) - M$$

决策树特征选择：信息增益

- 特征选择准则—: 信息增益
  - 选择A或B两个特征构造节点，哪种方式好?

特征A	Yes	Yes	No	…	No	No
特征B	No	Yes	Yes	…	Yes	No
类别	C0	C0	C1		C1	C1

$Entropy(p)$

以特征B划分

特征A	Yes	No	…	No
特征B	Yes	Yes	…	Yes
类别	C0	C1		C1

特征A	Yes	No
特征B	No	No
类别	C0	C1

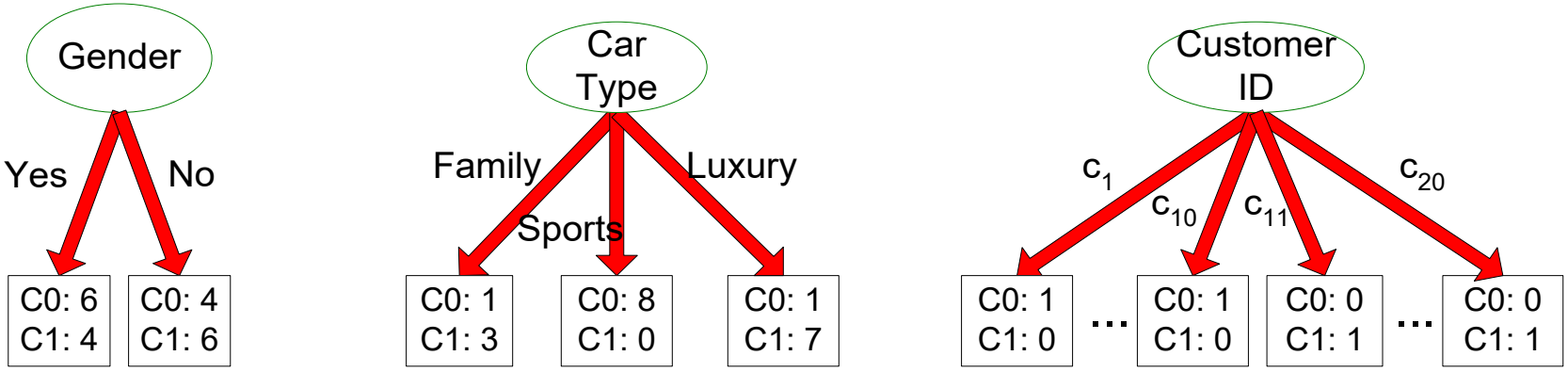
$Entropy(B_{Yes})$ $Entropy(B_{No})$

$$M = \frac{|B_{Yes}|}{n} Entropy(B_{Yes}) + \frac{|B_{No}|}{n} Entropy(B_{No})$$

$$Gain(B) = Entropy(p) - M \quad < \quad Gain(A)$$

决策树特征选择：信息增益

- 特征选择准则一：信息增益
- **结论：**信息增益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
- **不足：**信息增益的局限性，尤其体现在更偏好可取值较多的特征
  - 取值较多，不确定性相对更低，因此得到的熵偏低



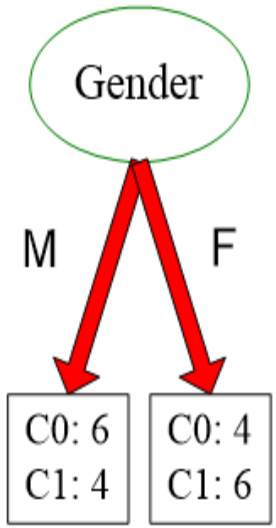
特征Customer ID有最大的信息增益，因为每个子节点的熵均为0

决策树特征选择：信息增益率

- 特征选择准则二：信息增益率
  - 信息增益率(Gain ratio): 综合考虑划分结果信息增益和划分数量的信息

$$GAIN_{ratio(m)} = \frac{GAIN(m)}{IV}, \qquad IV = -\left(\sum_{i=1}^{k} \frac{n_i}{n} log_2 \frac{n_i}{n}\right)$$

  - 相比于信息增益，增加了一个惩罚项IV, 考虑产生划分的数量带来的划分信息
  - 即，若某个特征产生的划分数量很大，则划分信息很大，降低增益率



- 选择准则：选择最大的 信息增益率 对应的特征m

信息增益率在C4.5决策树算法中被应用

决策树特征选择：信息增益率

- 特征选择准则二：信息增益率
  - 结论：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
    - 当特征的取值较少时，IV较小，因此惩罚项相对较小
  - 实际应用中，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

决策树特征选择：基尼指数

- 特征选择准则三：基尼指数
  - 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
  - 计算特征节点t的基尼指数：

$$GINI(t) = 1 - \sum_{j} [p(j|t)]^2$$

    - $p(j \mid t)$ 是特征节点t上属于类别j的样本的比例

- 特点：对于该节点特征t
  - 当样本均匀地分布在各个类别时，基尼指数达到最大值 $1 - \frac{1}{n_c}$, 此时包含的信息最少
  - 当样本只属于一个类别时，基尼指数达到最小值 0, 此时包含的信息最多

决策树特征选择：基尼指数

- 计算 某个节点 特征的 基尼指数

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0　　P(C2) = 6/6 = 1

Gini = 1 – P(C1)^2 – P(C2)^2 = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6　　P(C2) = 5/6

Gini = 1 – (1/6)^2 – (5/6)^2 = 0.278

C1	**2**
C2	**4**

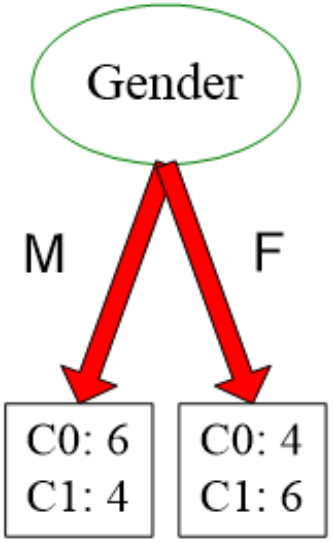
P(C1) = 2/6　　P(C2) = 4/6

Gini = 1 – (2/6)^2 – (4/6)^2 = 0.444

决策树特征选择：基尼指数

- 特征选择准则三：基尼指数
  - 当一个特征节点p 分裂成 k 个子节点（如两个子节点）

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$



    - n表示父节点的样本个数
    - n_i 表示子节点i的样本个数
  - 选择准则：选择最大的GINI 对应的特征m

基尼指数在CART, SLIQ, SPRINT等决策树算法中被应用

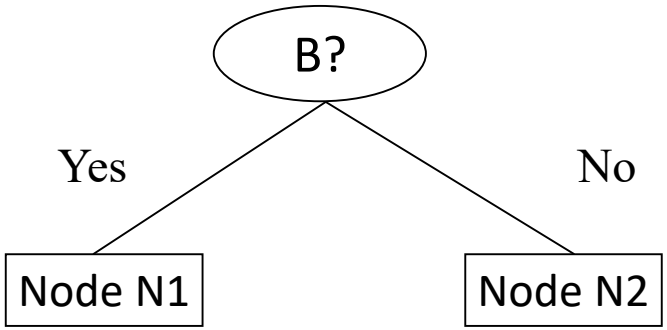
决策树特征选择：基尼指数

- 基尼指数计算示例

分裂前：

	Parent
C1	**6**
C2	**6**
Gini = 0.500	

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$



$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

	N1	N2
C1	**5**	**2**
C2	**1**	**4**
Gini=0.361		

Gini(N1)
= 1 – $(5/6)^2$ – $(1/6)^2$
= 0.278

Gini(N2)
= 1 – $(2/6)^2$ – $(4/6)^2$
= 0.444

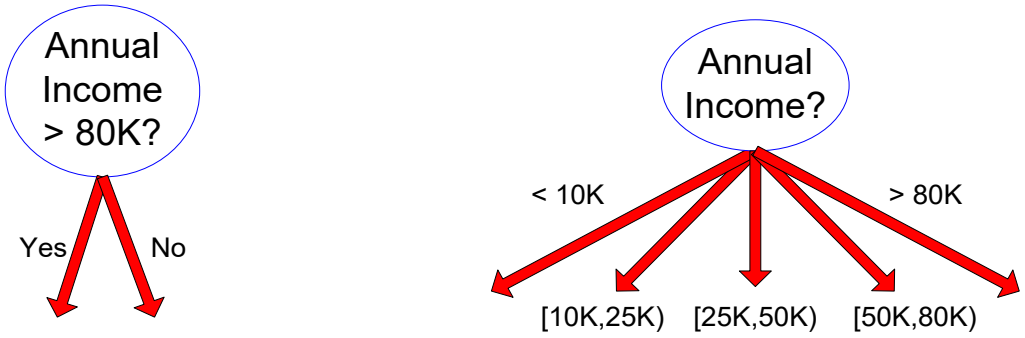
分裂后：

Gini(Children)
= 6/12 * 0.278 +
6/12 * 0.444
= 0.361

决策树特征选择

- 决策树特征选择：连续属性的分裂
  - 将连续属性进行离散化
    - 最简单：人为划分一次，如设定收入的阈值
    - 通过等间隔分段、等频率分段(百分位数)或聚类找到划分位置
    - 利用算法二分离散化考虑所有情况，找出最好的划分

回顾第二章知识：基于熵的数据离散化



(i) Binary split　　(ii) Multi-way split

决策树生成过程

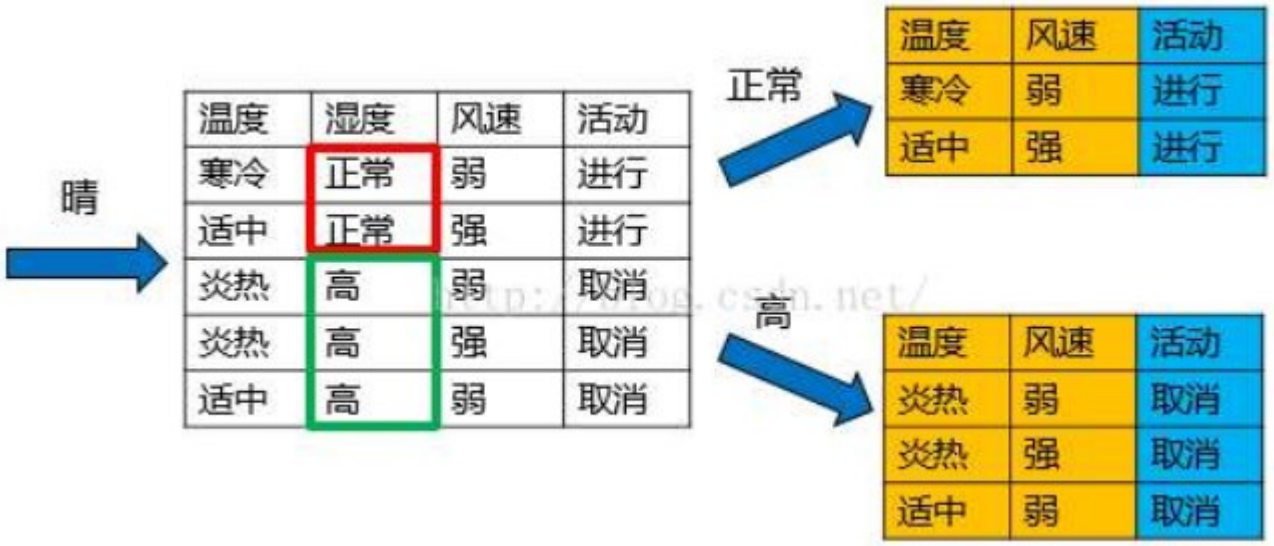
2. 生成决策树

- 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C4.5算法为例，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
  - 计算当前节点各个特征的信息熵，并进而计算得到该特征对应的信息增益率
  - 基于最大信息增益率的特征，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采用信息增益，而CART采用基尼指数

决策树算法有很多，如ID3, C4.5, CART等，核心区别在特征选择准则不同，具体算法请大家课后查阅学习

决策树生成过程

- **2. 生成决策树**：树停止分裂条件
  - 停止分裂直到所有节点属于同一类
  - 停止分裂当所有记录有相同的属性值
  - 早停策略



按湿度特征划分，发现两个节点均属于同一类，即停止

决策树剪枝

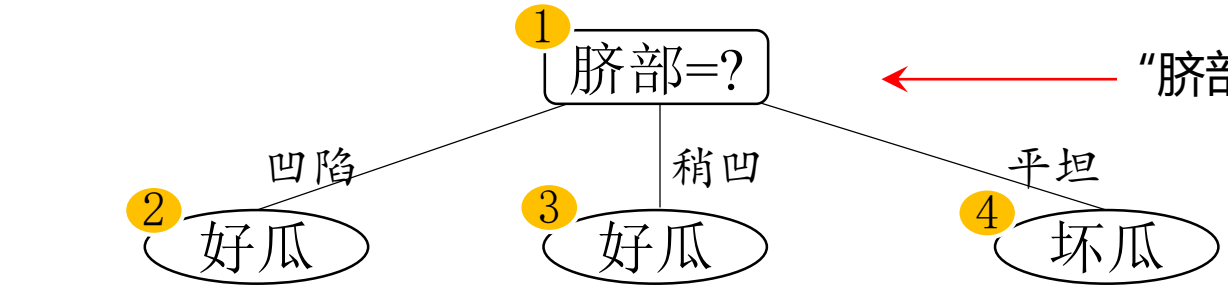
3. 决策树剪枝

- 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”
    - 如果训练集效果好，而测试集效果不好，说明出现“过拟合”
- 过拟合出现的原因：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
- 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”
- 包括两种策略：预剪枝、后剪枝

决策树剪枝

3. 决策树剪枝：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
  - 通过提前停止生成分枝对决策树进行剪枝，可以利用信息增益等测度来对分枝生成情况（优劣）进行评估



“脐部=?”

验证集精度

划分前: 42.9%

划分后: 71.4%

预剪枝决策: 划分

“色泽=?”

验证集精度

划分前: 71.4%

划分后: 57.1%

预剪枝决策: 禁止划分

“根蒂=?”

验证集精度

划分前: 71.4%

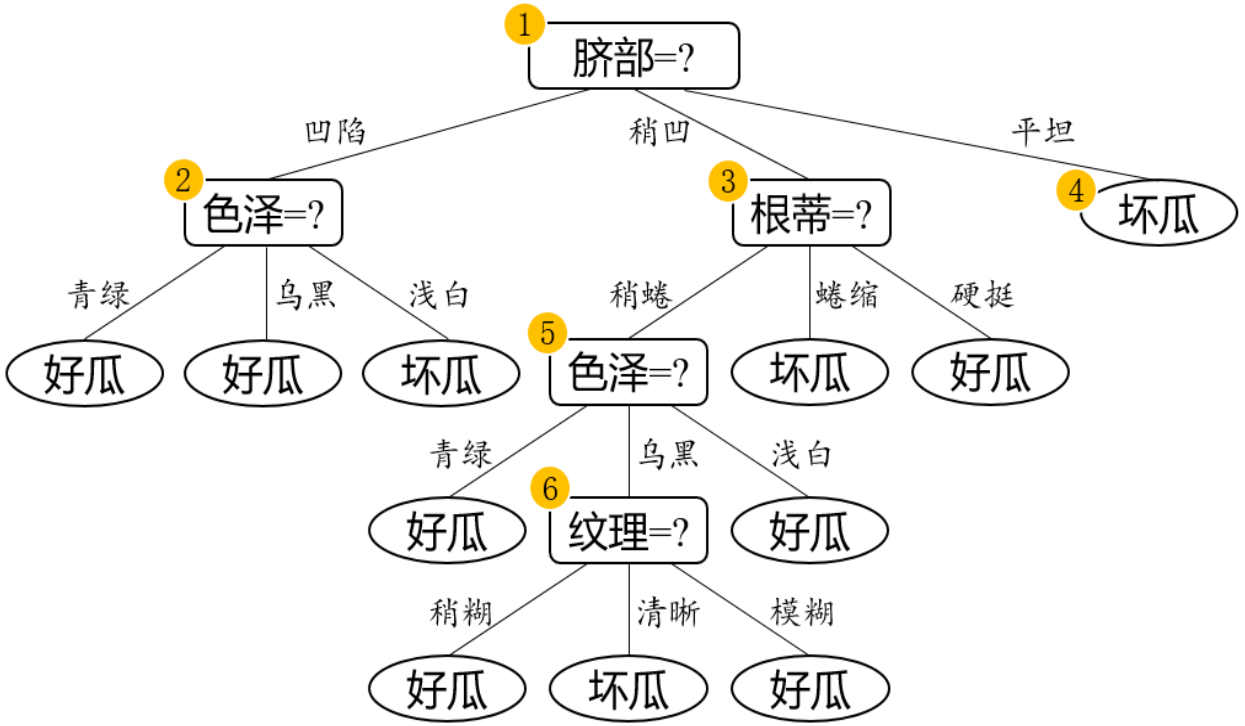
划分后: 71.4%

预剪枝决策: 禁止划分

决策树剪枝

- **3. 决策树剪枝：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

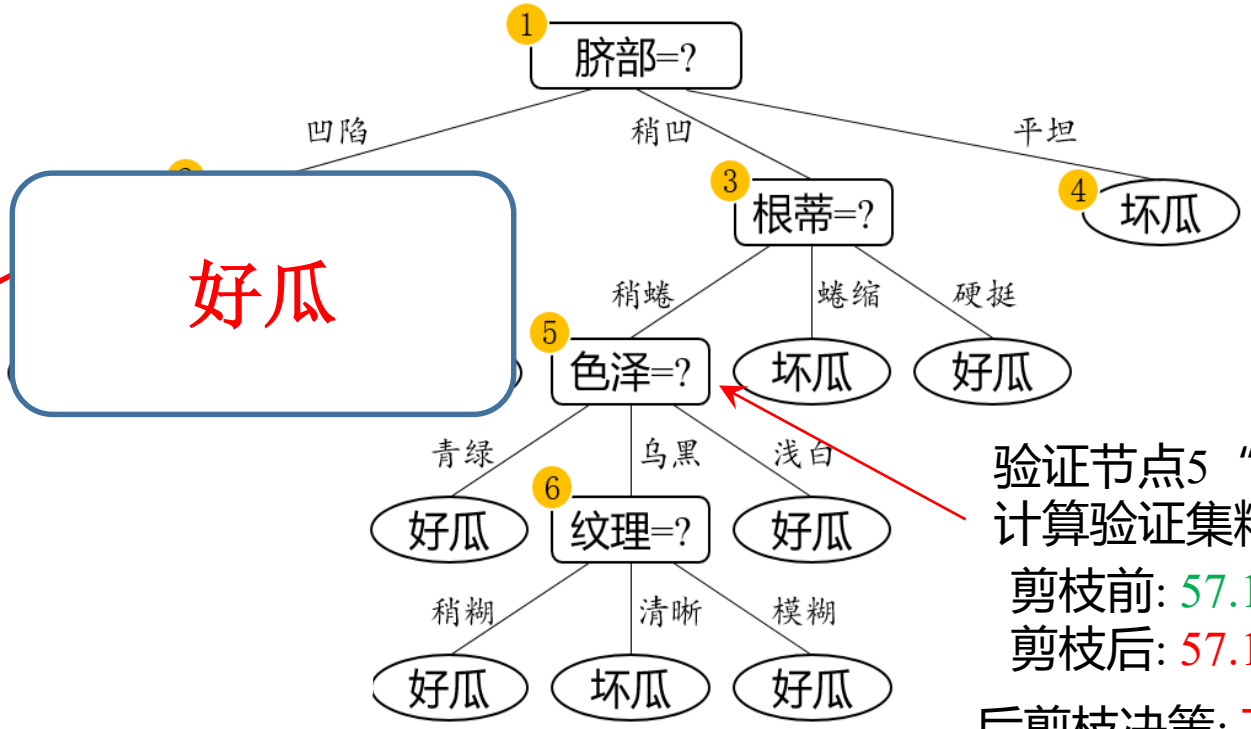
剪枝前



决策树剪枝

- 3. 决策树剪枝：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



验证节点2“色泽”计算验证集精度

剪枝前: 57.1%

剪枝后: 71.4%

后剪枝决策: 剪枝

验证节点5“色泽”计算验证集精度

剪枝前: 57.1%

剪枝后: 57.1%

后剪枝决策: 不剪枝

决策树剪枝

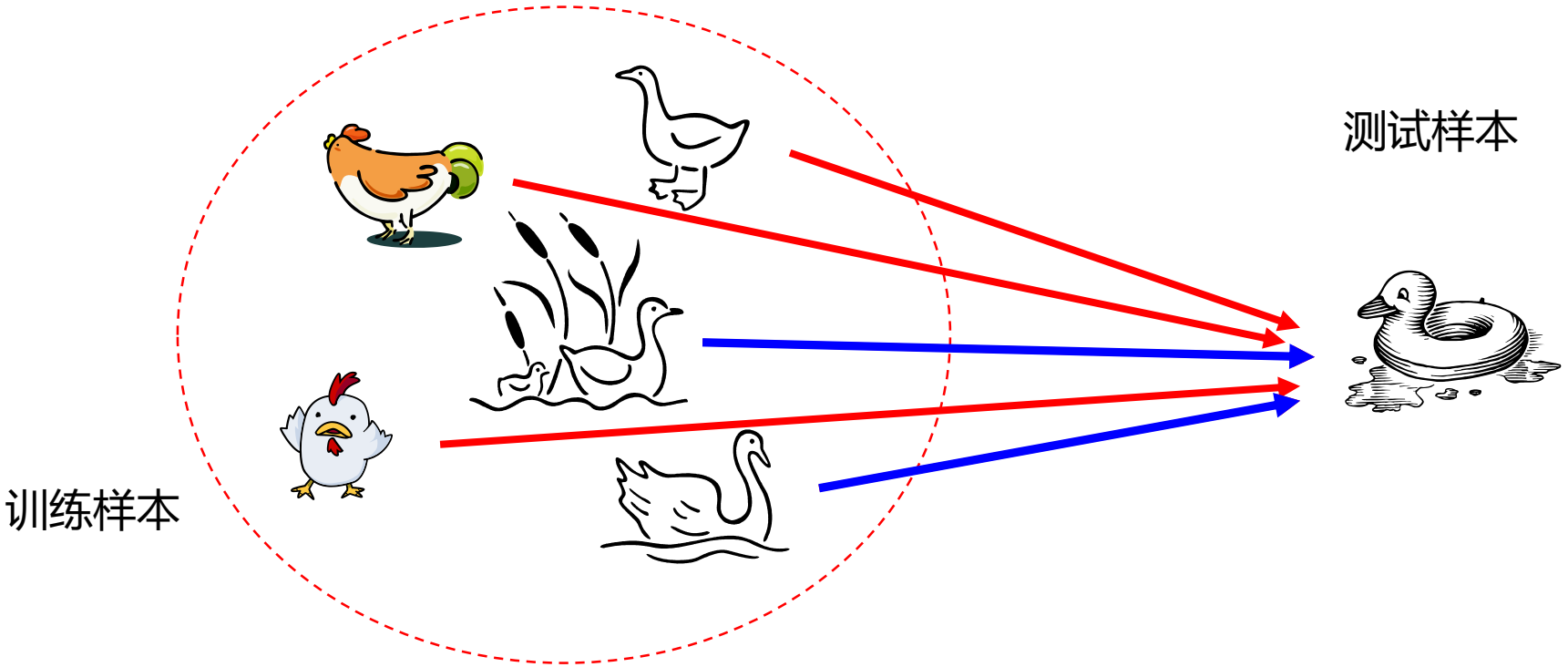
- **3. 决策树剪枝**：比较两种剪枝策略
  - 从过程上看，后剪枝方法经过了“构建”到“剪枝”这样的过程，显然它要比事前剪枝需要更多的计算时间
  - 对应的，后剪枝可以获得更可靠的决策树
  - 实际使用时：先剪枝可以与后剪枝方法相结合，从而构成一个混合的剪枝方法

分类与预测

- 有监督学习：分类与预测
- 常用方法
  - 规则方法
  - 决策树
  - 最近邻方法
  - 支持向量机（SVM）
  - 神经网络
  - 集成方法
- 类不平衡问题
- 分类的评价指标

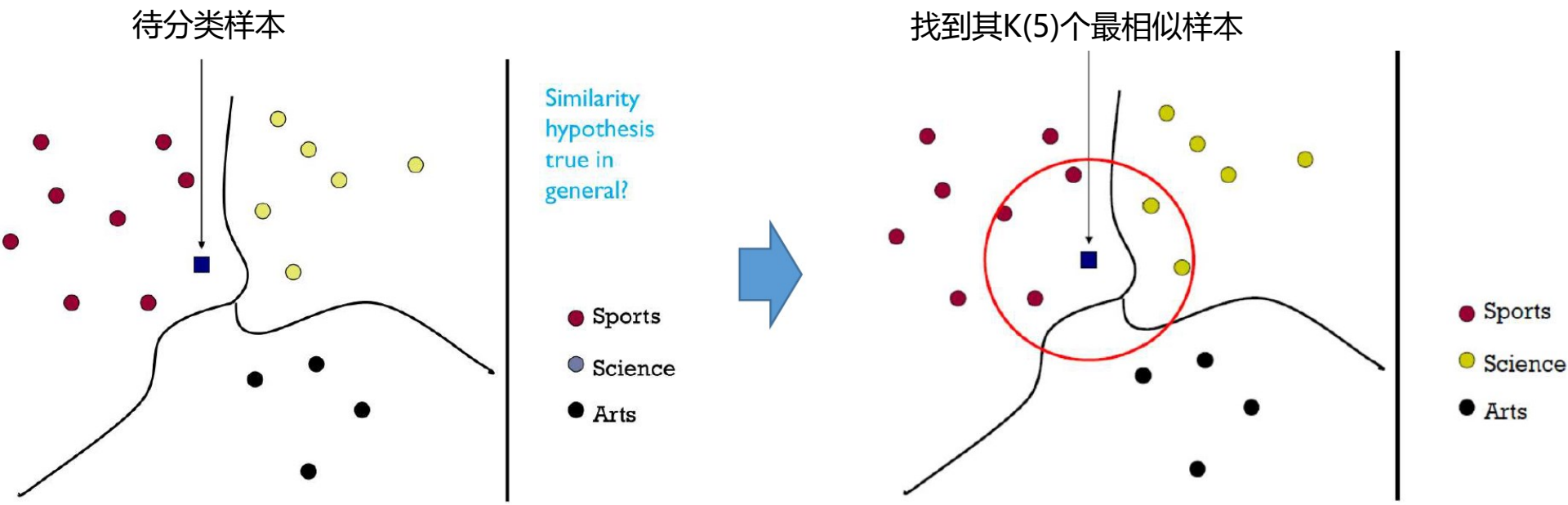
分类：K近邻方法

- 分类——K近邻方法
  - 问题：判断测试样本是什么动物？



分类：K近邻方法

- 分类——K近邻方法
  - 对数据空间内的样本，可提出相似样本假设
    - 表征上相近的样本应该属于同一个类别
  - K近邻思想：用K个最相似样本的类别来预测未知样本的类别(投票方法)
  - 核心问题：距离度量、K的取值



K近邻方法：距离度量

- K近邻方法核心问题：距离度量
  - K近邻分类的效果严重依赖于距离度量
  - 对于高维空间而言，最基本的度量方式为欧式距离

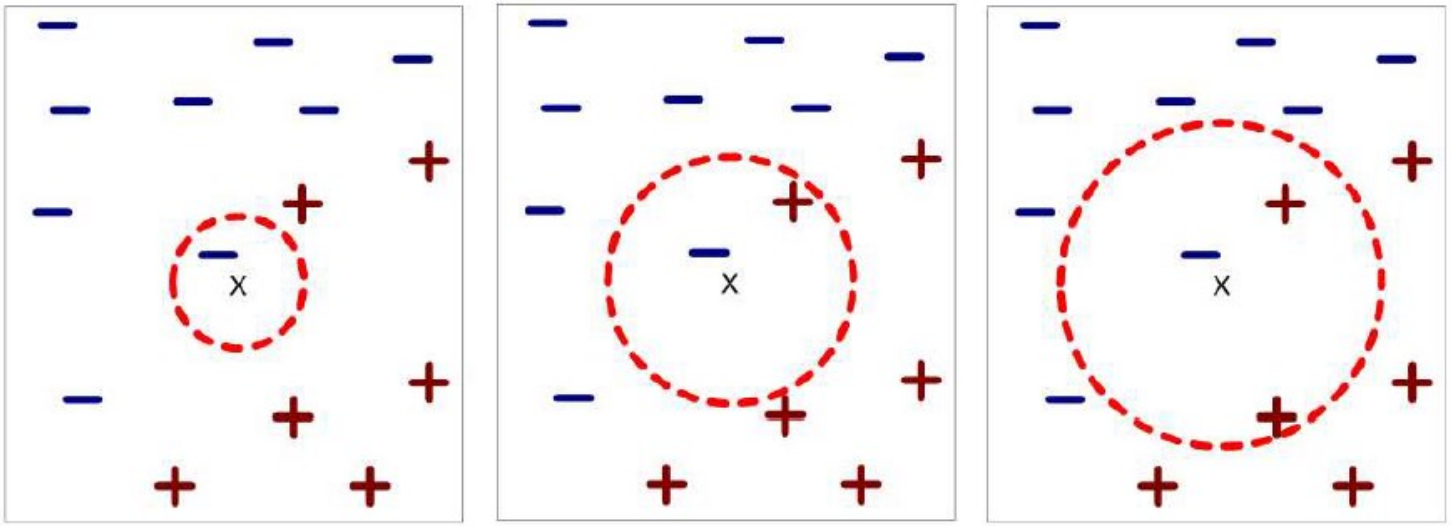
$$d(x, x') = \sqrt{\sum_i (x_i - x'_i)^2}$$

  - 离散0/1向量，则可使用汉明距离（Hamming）代替
  - 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等

回顾第二章：数据集成

K近邻方法：K的取值

- K近邻方法核心问题：K的取值
  - K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，容易受噪声干扰；
    - K太大，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

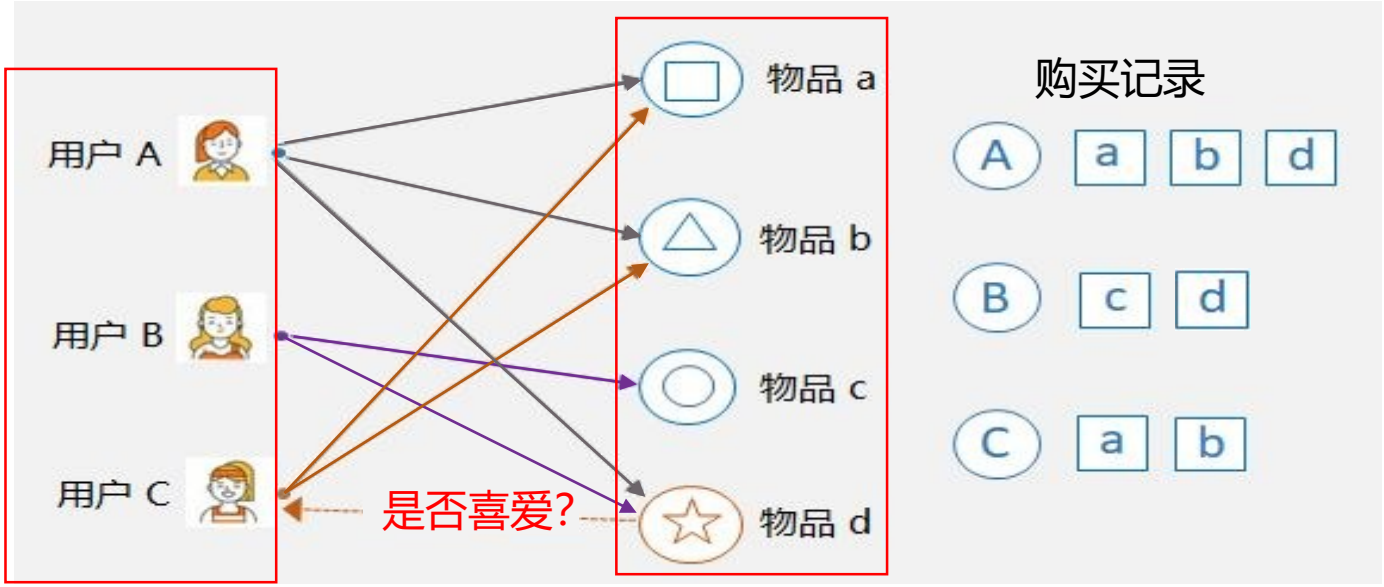
不同的K值，结果不同

分类：K近邻方法

- K近邻方法的特点
  - K近邻方法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）
  - K近邻方法是一种消极学习，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
  - K近邻方法基于局部信息进行判别，受噪声影响很大
  - K近邻方法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
    - （回顾第二章：数据集成）已知：小明(160,60000)；小王(160,59000)；小李(170，60000)。小明与谁的体型更相似？

K近邻方法实例

- K近邻思想的应用实例：推荐系统的UCF、ICF模型



UCF: 基于K个相似用户对物品的评分

- 用户A、B购买过物品d，且与C相似，可用他们对d的平均喜爱程度作为C对d的喜爱程度

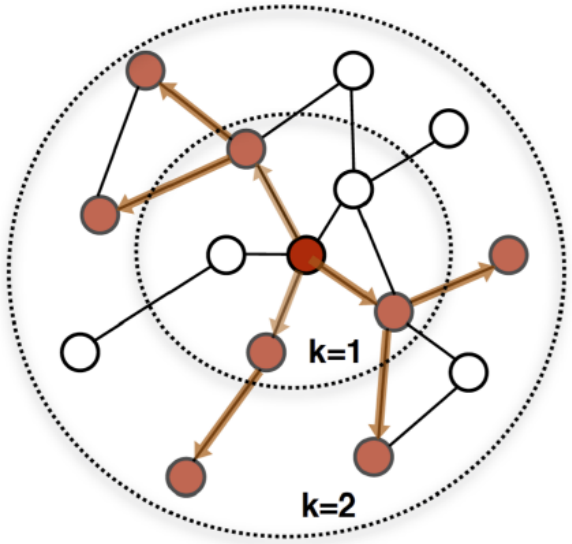
ICF: 基于用户对K个相似物品的评分

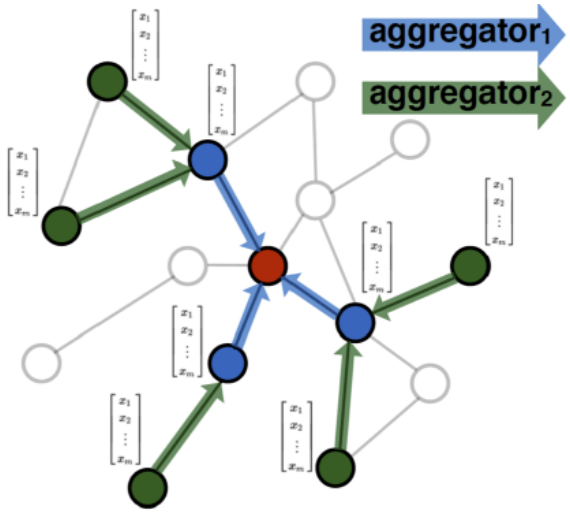
- C购买过a,b，若物品d与a,b相似，可用C对他们的平均喜爱程度作为对物品d的喜爱程度

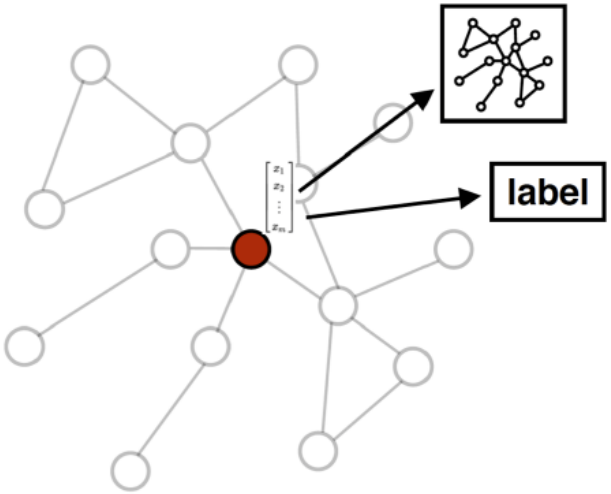
- Sarwar, Badrul, et al. Item-based collaborative filtering recommendation algorithms. WWW 2001.
- Amazon recommendations: Item-to-item collaborative filtering. IEEE Internet computing, 2003

K近邻方法实例

- K近邻思想的应用实例：图神经网络中的近邻
  - 基本思想：将K个邻居节点的信息传播到当前节点
  - 距离度量：基于注意力机制计算(GAT模型)







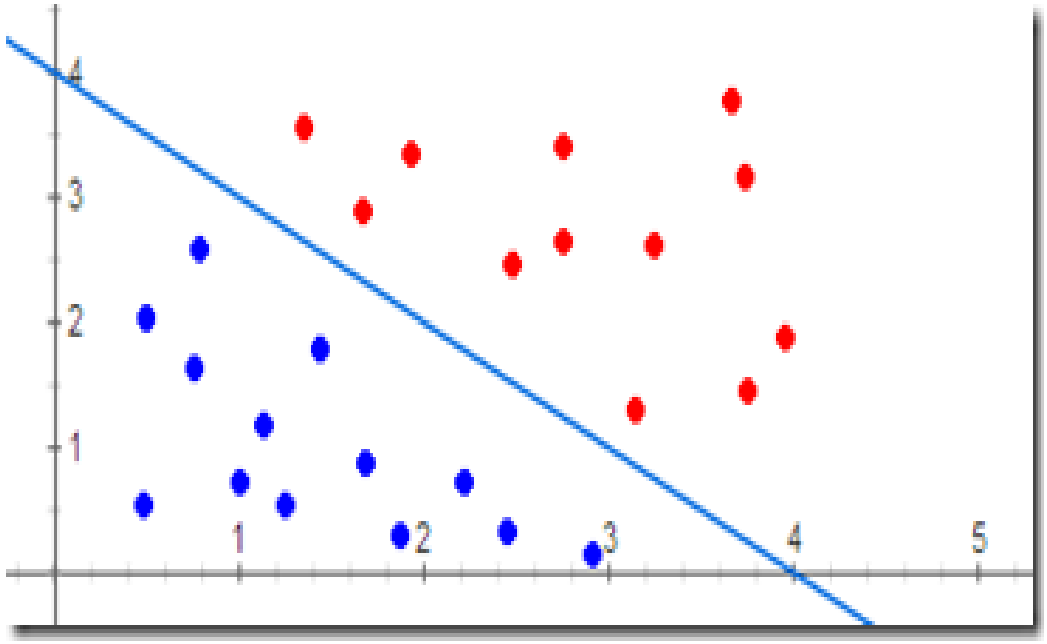
➢ Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." NeurIPS 2017

分类与预测

- 有监督学习：分类与预测
- 常用方法
  - 规则方法
  - 决策树
  - 最近邻方法
  - 支持向量机（SVM）
  - 神经网络
  - 集成方法
- 类不平衡问题
- 分类的评价指标

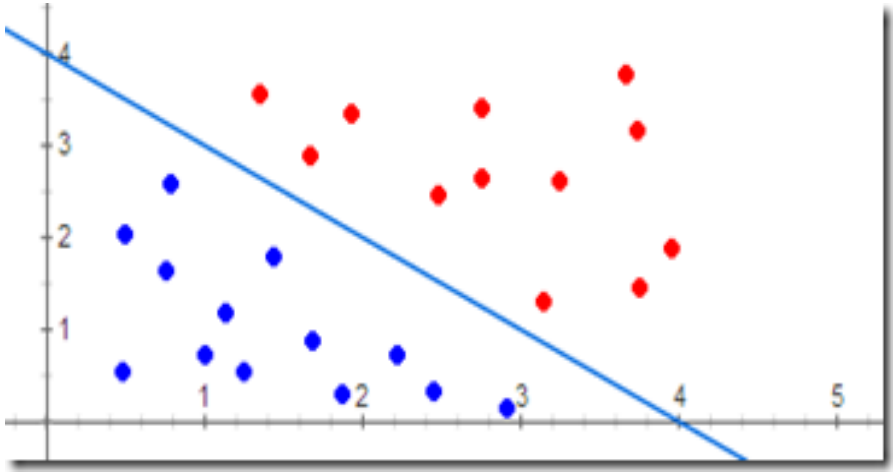
分类：感知机

- 分类——感知机（perceptron）
  - 目标：寻找一条直线S（高维时是超平面）划分不同类别的数据
  - 输入：样本的特征向量X={x}, $x \in R^d$
  - 输出：样本类别y $\in \{-1, +1\}$

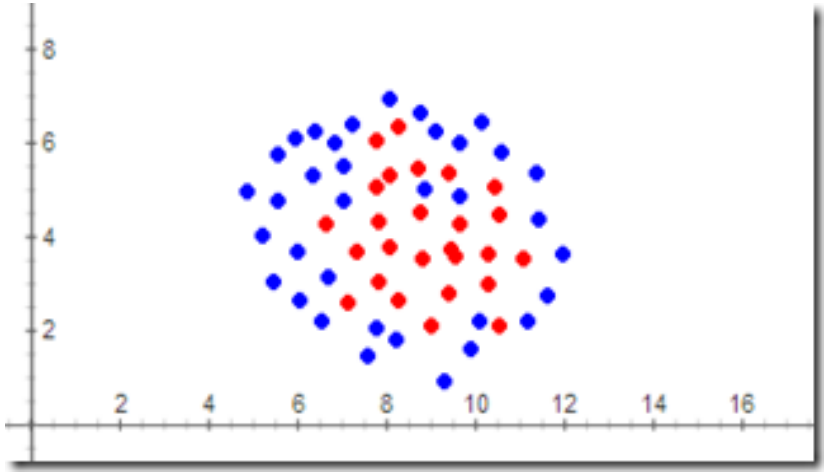


分类：感知机

- 分类——感知机（perceptron）
  - 1957年由Rosenblatt提出，是神经网络与支持向量机的基础
  - 感知机的前提：样本空间线性可分
    - 左例中，可以用一条直线将+1类和-1类完美分开，称这个样本空间是线性可分的
    - 右例的样本是线性不可分的，感知机不能处理这种情况

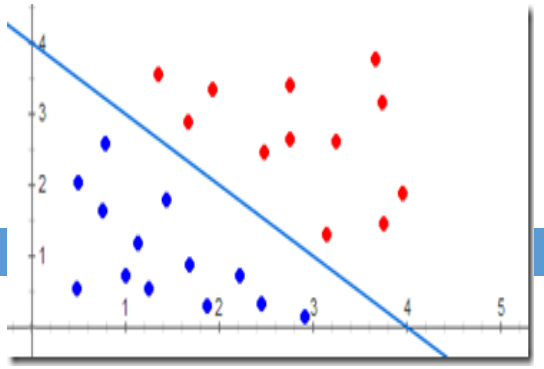


线性可分



线性不可分

分类：感知机



- 感知机的基本概念
  - 模型：符号函数 $f(x) = sign(w \cdot x + b) = \begin{cases} +1, & w \cdot x + b \geq 0 \\ -1, & w \cdot x + b < 0 \end{cases}$
  - 分界超平面 S： $w \cdot x + b = 0$
  - 学习目标：最小化 误分类点 到 超平面 的 总距离
    - 点(x_0, y_0)到超平面S: $w \cdot x + b = 0$的距离$\frac{1}{\|w\|} |w \cdot x_0 + b|$
    - 误分类点(x_i, y_i)到超平面S: $w \cdot x_i + b = 0$的距离为 $-\frac{1}{\|w\|} y_i (w \cdot x_i + b)$

推导过程省略

x_i错分时，若y_i为-1，则计算的$(w \cdot x_i + b) > 0$
若y_i为+1，则计算的$(w \cdot x_i + b) < 0$

  - 损失函数：$\underset{w,b}{\operatorname{argmin}} L(w, b) = -\sum_{x_i \in M} y_i (w \cdot x_i + b)$，
  - 学习策略：找到参数w和b，使得损失函数最小

它是连续可导的，这就使得我们比较容易求得其最小值

分类：感知机

- 感知机学习算法：梯度下降 — 课后学习

$$\min_{w,b} L(w,b) = -\sum_{x_i \in M} y_i (w \cdot x_i + b)$$

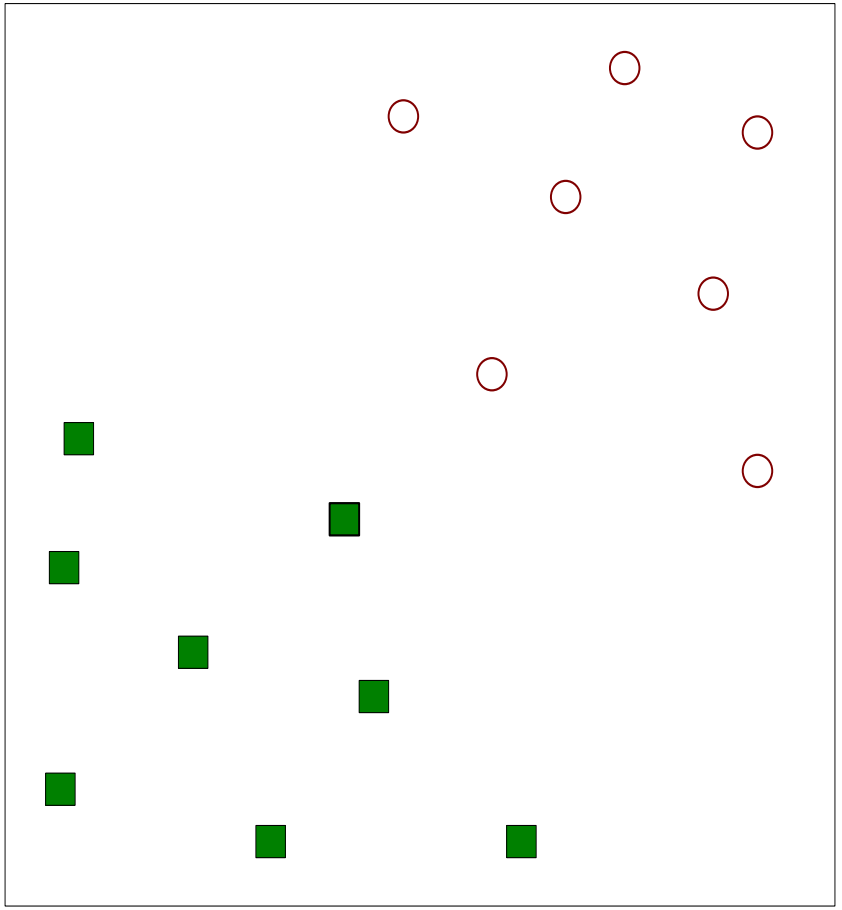
  - 随机初始化w_0和b_0
  - 梯度下降不断地极小化损失函数
    - 每次随机选取一个误分类点对w和b进行更新。
    - 设误分类点为(x_i, y_i)，那么损失函数$L(w,b)$的梯度为：

$$\nabla_w L(w,b) = -y_i \cdot x_i$$
$$\nabla_b L(w,b) = -y_i$$

    - 接下来对w, b进行更新$w \leftarrow w + \gamma y_i \cdot x_i, b \leftarrow b + \gamma y_i$，其中$\gamma (0 \le \gamma \le 1)$为步长，也、称为学习速率（learning rate）。
    - 通过迭代，直到损失函数为0 (无误分点)

分类：支持向量机

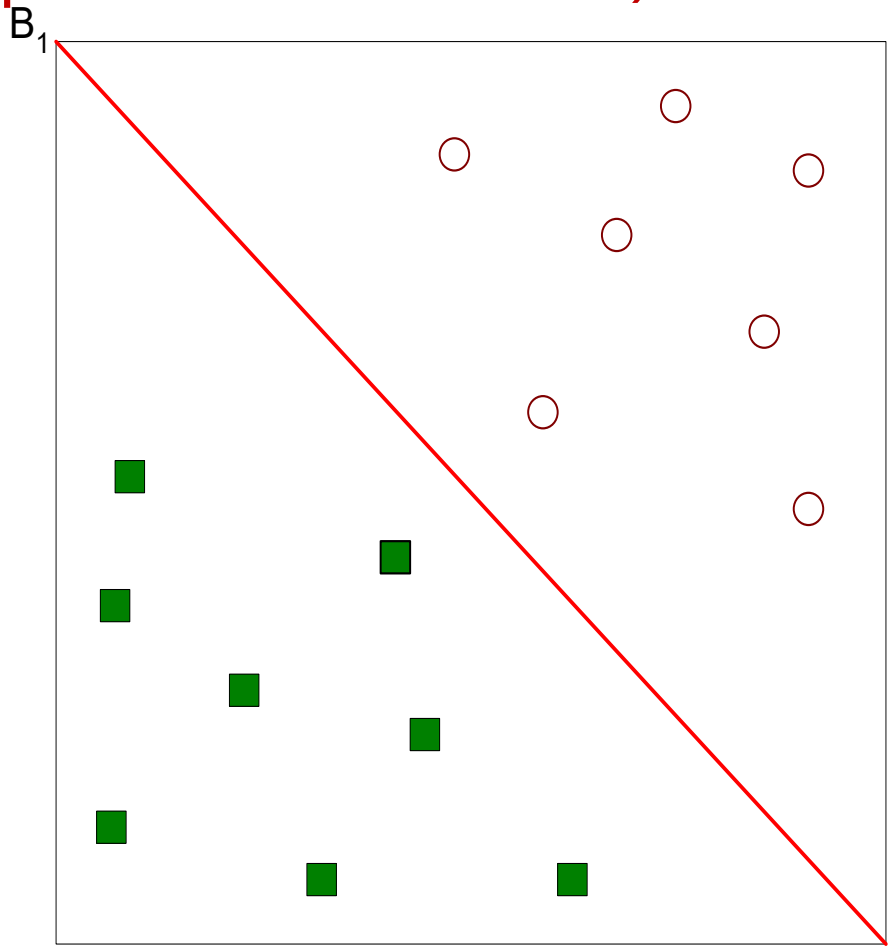
- 分类——支持向量机（Support Vector Machine）



分类：支持向量机

- 分类——支持向量机（Support Vector Machine）

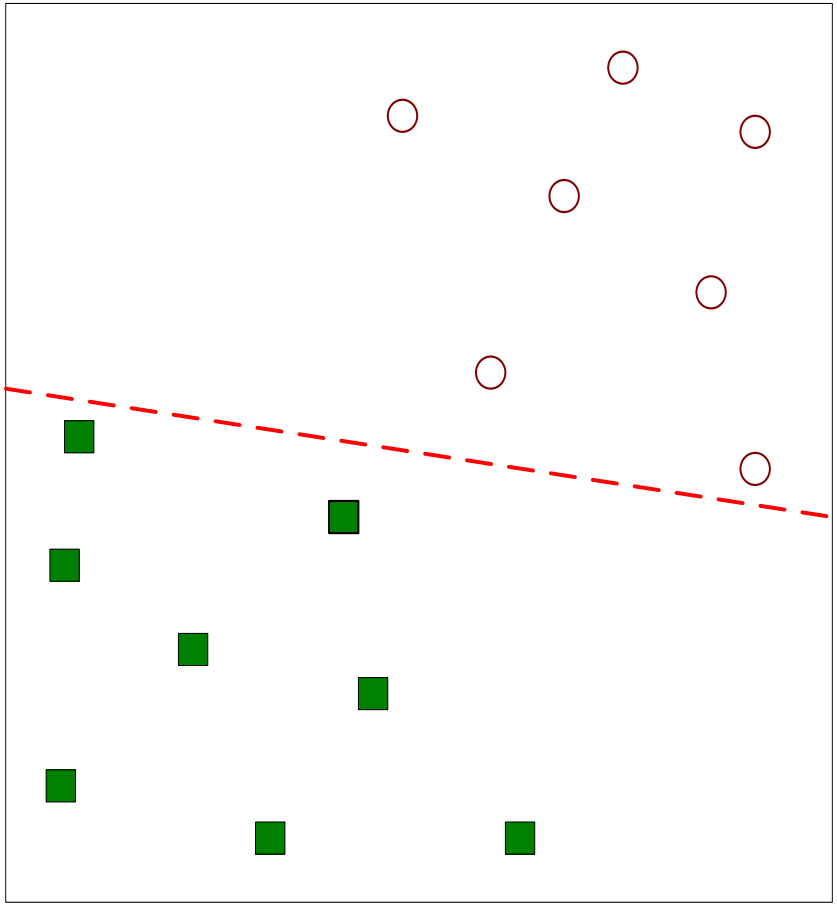
一个可行解



分类：支持向量机

- 分类——支持向量机（Support Vector Machine）

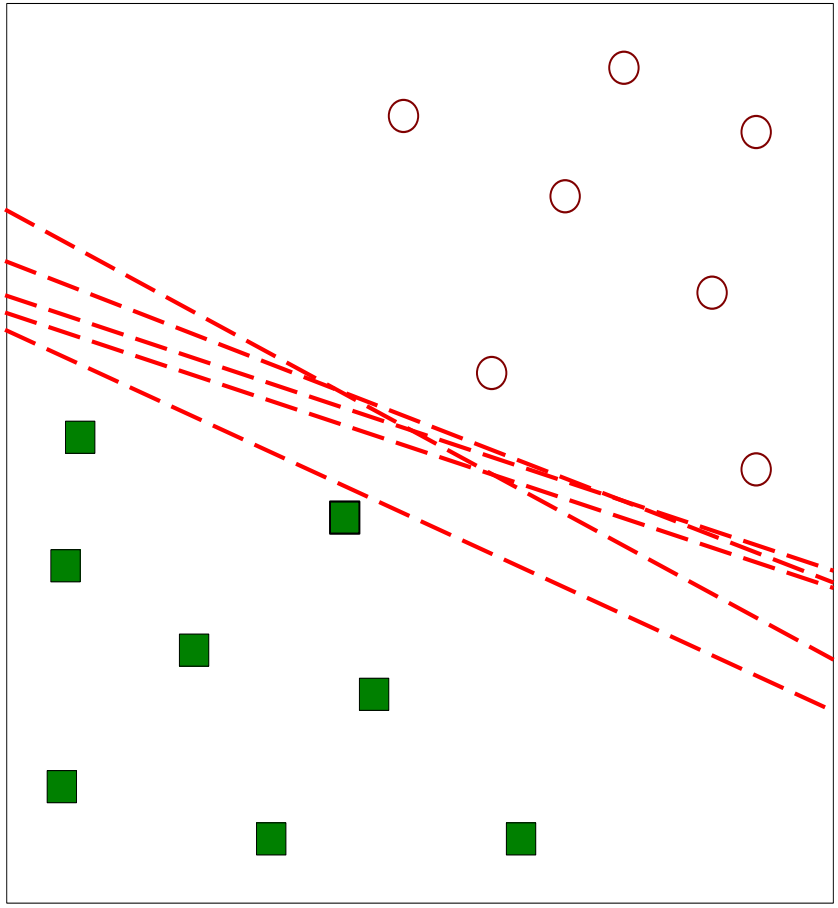
另一个可行解



分类：支持向量机

- 分类——支持向量机（Support Vector Machine）

其他可行解



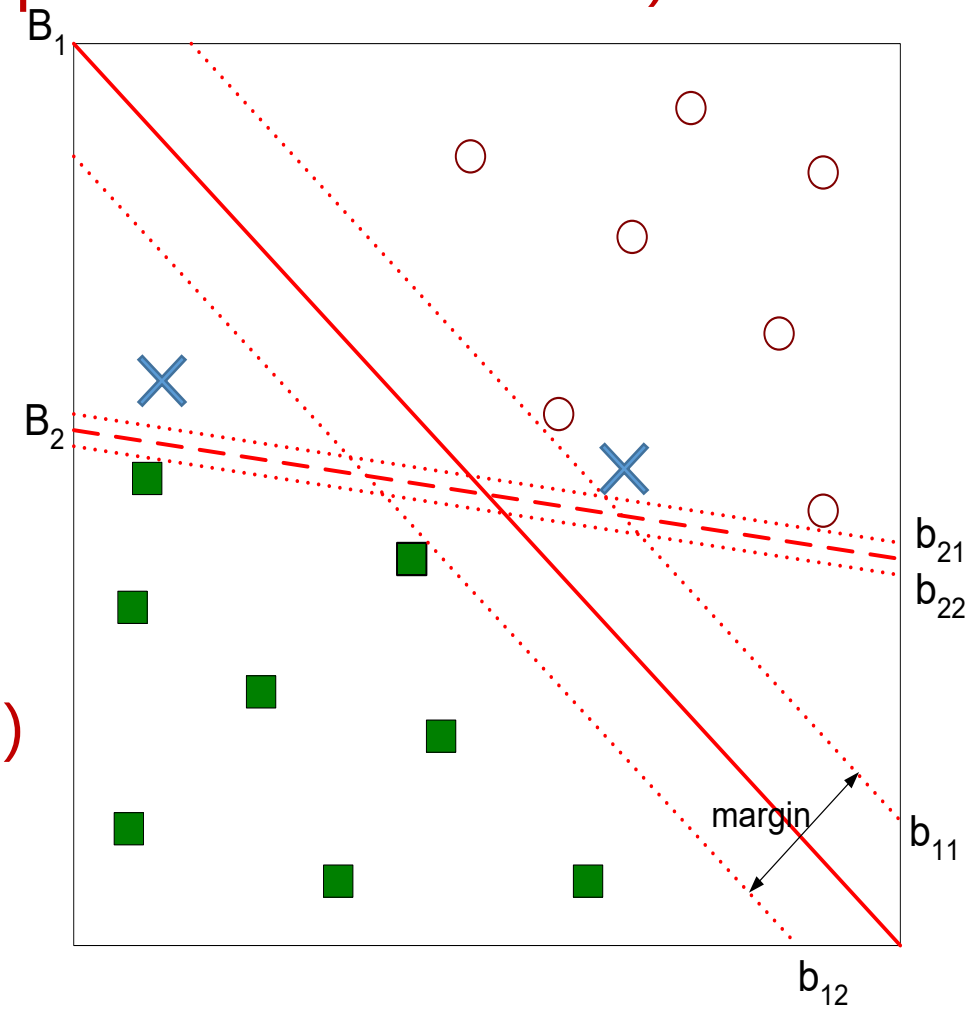
分类：支持向量机

- 分类——支持向量机（Support Vector Machine）

B1与B2，哪个更好？

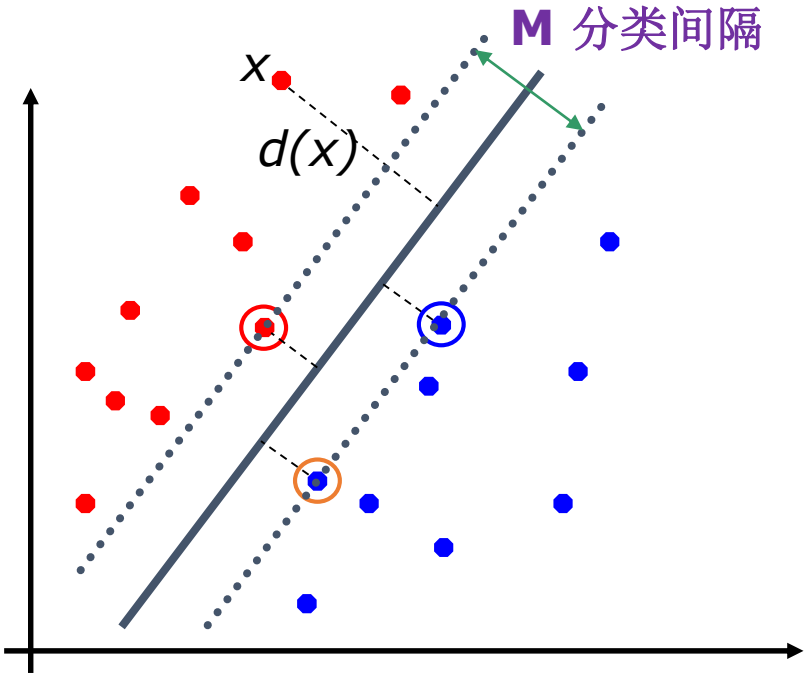
B1 保证分类正确（区分）

分类间隔大 （更容易区分）



分类：支持向量机

- 分类——支持向量机
  - 因此，应该选择“正中”的最大间隔超平面（分类间隔最大）
    - 容忍性好，泛化能力强
    - 在线性可分的条件下，符合这样条件的超平面“存在且唯一”
  - 问题：如何找到最优的超平面？
  - 最大化分类间隔

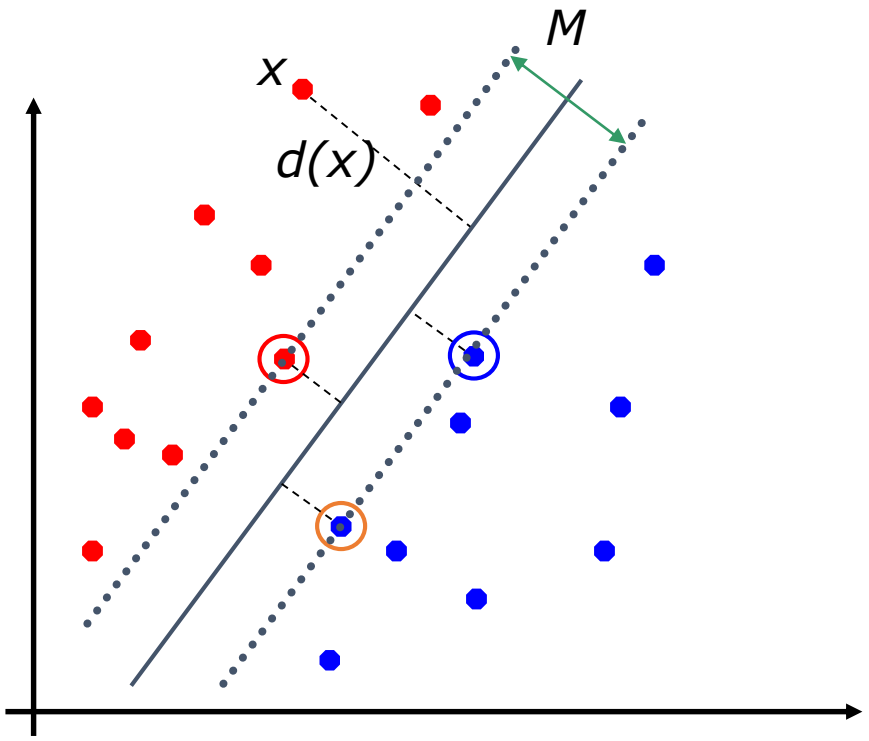


分类：支持向量机

- 分类——支持向量机
  - 理解：最大化分类间隔
    - 区分能力更强
    - 概率的角度: 最难的点置信度最大
    - 即使我们在选边界的时候犯了小错误，使得边界有偏移，仍然有很大概率保证可以正确分类绝大多数样本
    - 很容易实现交叉验证，因为边界只与极少数的样本点有关
    - 有一定的理论支撑
    - 实验结果验证了其有效性

保证分类正确性—当前

保证分类质量—未来

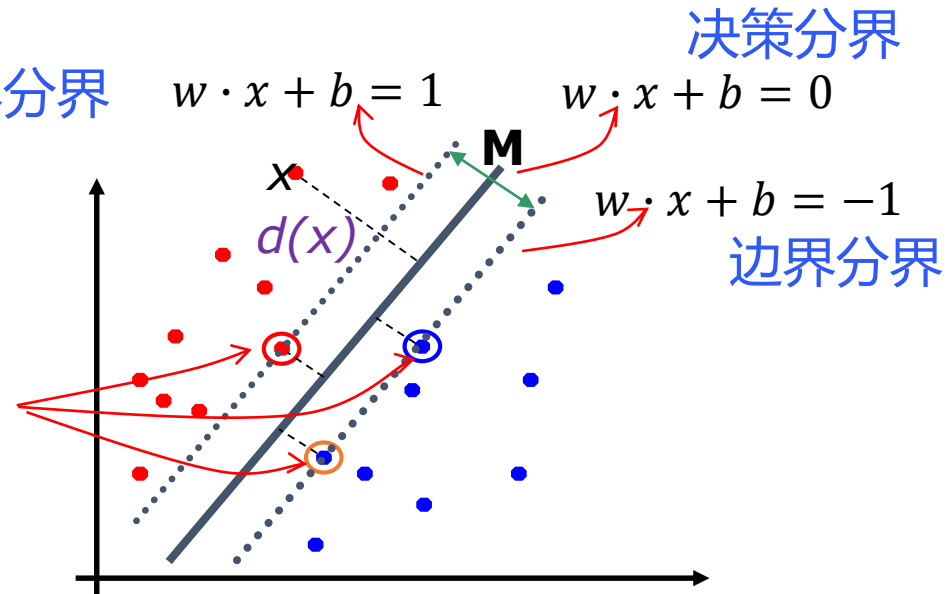


分类：支持向量机

- 支持向量机的基本概念
  - 模型：符号函数$y = sign(w \cdot x + b) = \begin{cases} +1, w \cdot x + b \geq 0 \\ -1, w \cdot x + b < 0 \end{cases}$
  - 决策分界面(Decision Boundary)：$w \cdot x + b = 0$
  - 边界分界面(Margin Boundary)：$w \cdot x + b = \pm 1$
  - 支持向量(Support Vectors)：满足$w \cdot x + b = \pm 1$的样本

两个边界之间的距离**d(x)**

支持向量



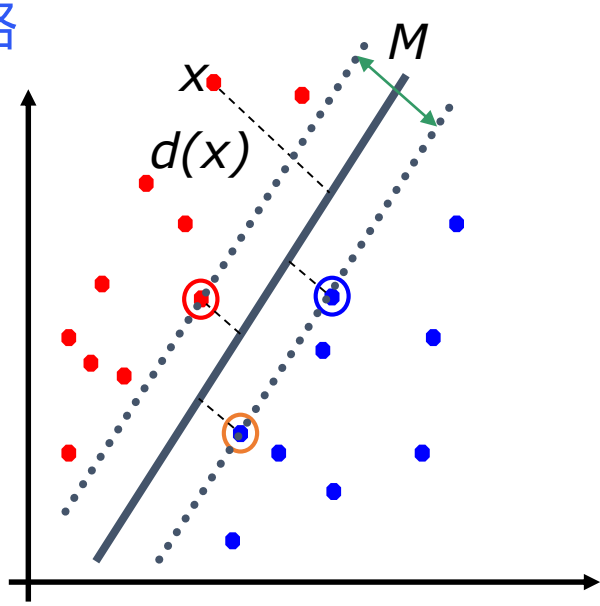
分类：支持向量机

- 支持向量机
  - **两个边界之间的距离d(x)：支持向量到决策分界面的距离**
    - 点x_i到平面$w \cdot x + b = 0$的距离为$\frac{1}{\|w\|} y_i(w \cdot x_i + b)$
    - 支持向量到决策平面的距离为$\frac{1}{\|w\|}$

最大化

- 两个边界之间的距离：$\frac{2}{\|w\|}$　推导过程省略
  - 目标函数：$\text{argmax}\, L(w,b) = \frac{2}{\|w\|}$
  - 学习策略：找到参数w和b，使得目标最大

$$\underset{w,b}{\text{argmax}}\, L(w,b) = \frac{2}{\|w\|}$$
$$s.t. \quad y_i(\mathrm{w}^T \cdot x_i + b) \geq 1, \qquad \mathrm{i} = 1,2,\ldots,\mathrm{m}$$



分类：支持向量机

- 支持向量机学习算法
  - 优化任务转化

$$\underset{w,b}{argmax}\ L(w,b)=\frac{2}{\|w\|}$$
$$s.t.\quad y_i(\mathrm{w}^T\cdot x_i+b)\geq 1,\qquad \mathrm{i}=1,2,\ldots,\mathrm{m}$$

优化目标：平方和系数都是为了求导方便

等价

$$\underset{w,b}{\mathrm{argmin}}\ L(w,b)=\frac{1}{2}\|w\|^2$$
$$s.t.\quad y_i(\mathrm{w}^T\cdot x_i+b)\geq 1,\qquad \mathrm{i}=1,2,\ldots,\mathrm{m}$$

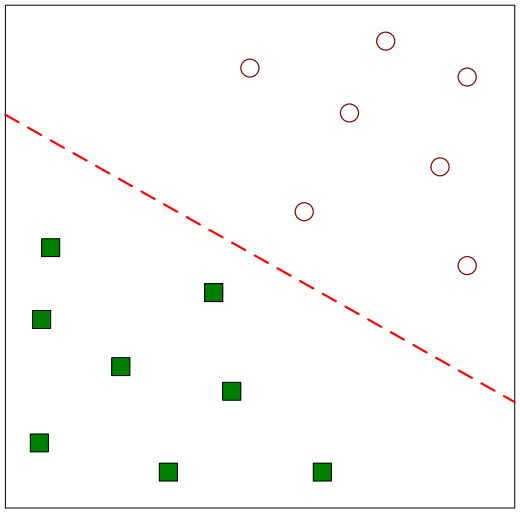
  - (课后学习)注意到，该形式符合凸二次规划问题特征，可以借助拉格朗日对偶性，通过求解对偶问题加以求解
  - (课后学习)具体而言，求解方式可采用序列最小优化算法（SMO）

```
from sklearn.svm import LinearSVC
```

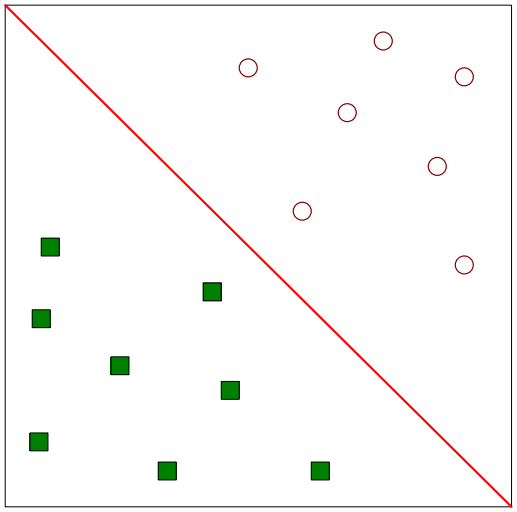
分类：支持向量机

- 感知机与支持向量机对比
  - 相同点：均采用模型$f(x) = sign(w \cdot x + b)$
  - 不同点：采用不同的优化目标

感知机



SVM



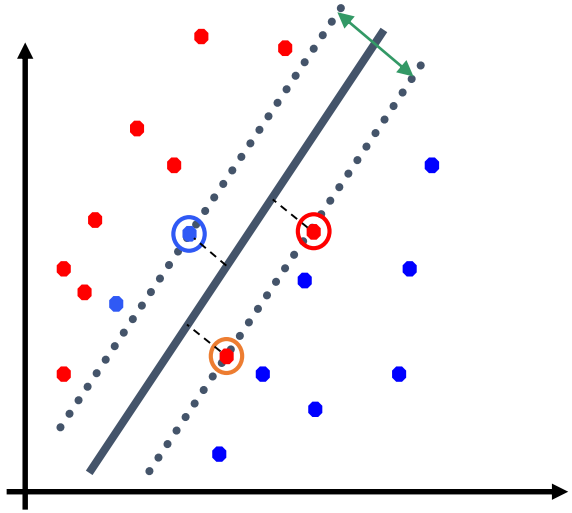
优化目标：

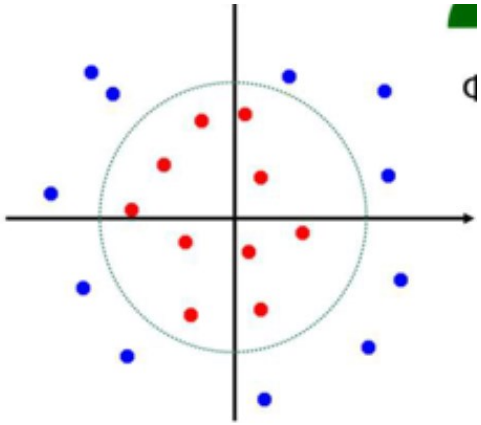
$$\min_{w,b} L(w,b) = -\sum_{x_i \in M} y_i(w \cdot x_i + b)$$

$$\min_{w,b} L(w,b) = \frac{1}{2}\|w\|^2$$
$$s.t.\, y_i(w^T x_i + b) \geq 1$$

SVM总结

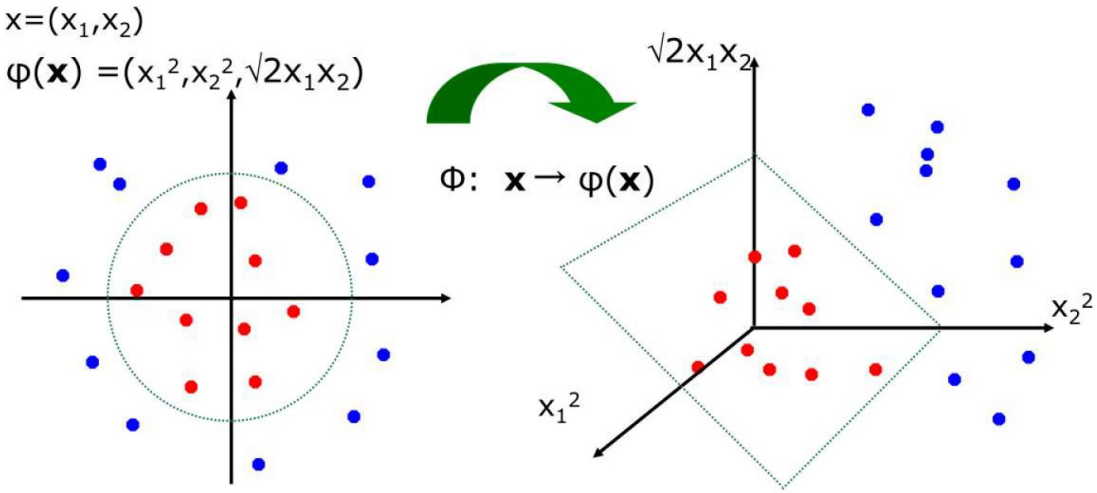
- 支持向量机
  - 优点：强分类器
    - 区分分类结果
    - 最大化间隔：更容易区分分类结果
    - 有数学理论保证
    - 只有支持向量在影响，优化简单
  - 缺点： Hard Margin SVM
    - 只能处理线性可分问题
    - 线性不可分
      - 软间隔：soft margin SVM（课后学习）
    - 线性完全不可分





线性不可分问题

- 线性不可分问题
  - 如果数据集线性不可分，不存在这样一个超平面，怎么办？
    - 解决方法：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
  - **核函数：**
    - 核函数的目的，在于将高维空间下的SVM求解时需要的内积运算转化为低维空间下的核函数计算，从而避免可能遇到的“维度灾难”问题



核函数

- 线性不可分问题与核函数——（课后学习）
  - 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	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\delta^2}\right)$	$\delta > 0$为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\delta}\right)$	$\delta > 0$
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
  - 常见挑选核函数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

分类与预测

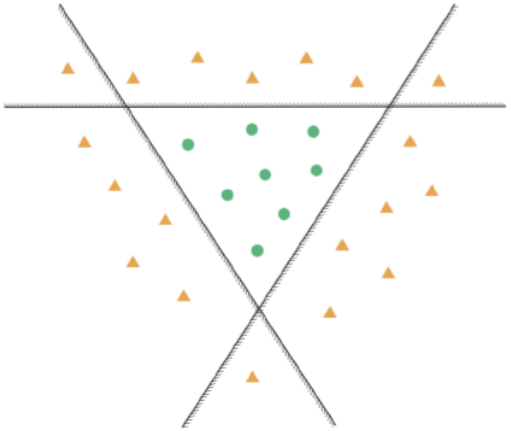
- 有监督学习：分类与预测
- 常用方法
  - 规则方法
  - 决策树
  - 最近邻方法
  - 感知机，支持向量机（SVM）
  - 神经网络
  - 集成方法
- 分类的评价指标
- 类不平衡问题

分类：集成学习

- 分类——集成学习
  - 思想：集成多个模型的能力，得到比单一模型更好的效果
  - 为什么能够提升效果？
    - 增强模型的表达能力
      - 单个感知机无法正确分类数据能用三个感知机完成
    - 降低误差
      - 假设单个分类器误差p， 有T个独立的分类器采用投票进行预测，得到集成模型 H
      - 集成分类器误差为

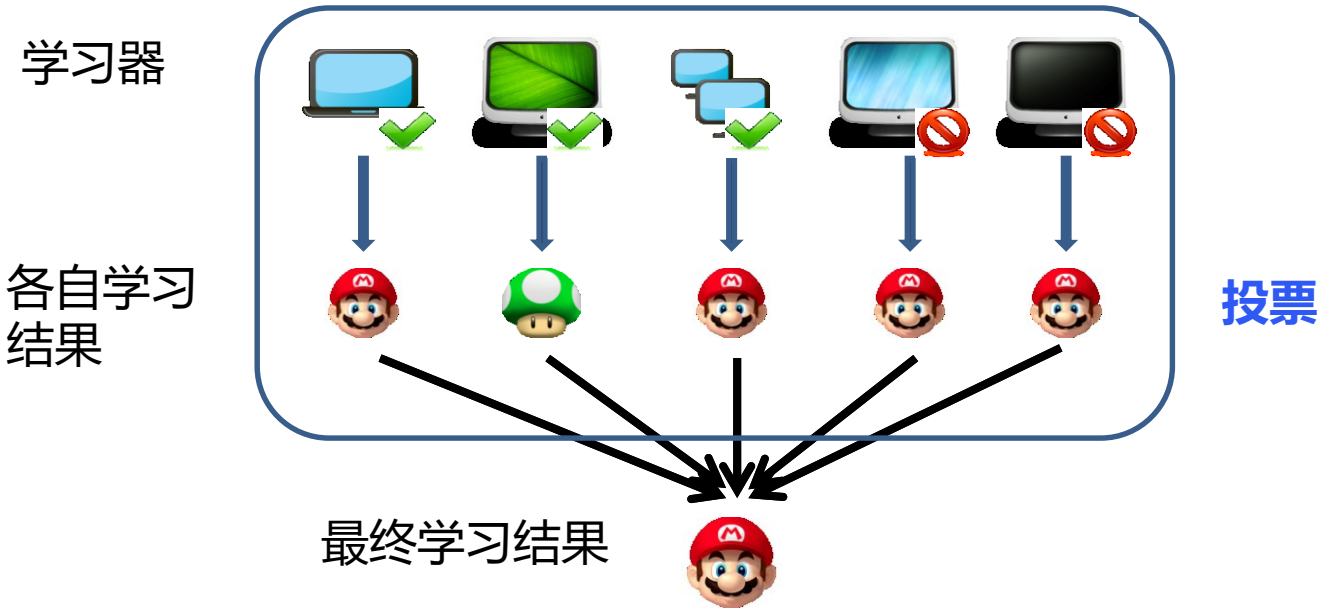
$$Error_H = \sum_{k \le \frac{T}{2}} C_T^k \cdot p^{T-k} \cdot (1-p)^k$$

      - $T = 5, p = 0.1$时， $Error_H < 0.01$



分类：集成学习

- 分类——集成学习
  - 集成过程



  - 此类方法经常用在数据挖掘竞赛中(如KDD CUP，CCF-BDCI)
    - KDD Cup2021 MAG240M-LSC赛道第一名集成了30个模型
    - KDD Cup2021 WikiKG90M-LSC赛道第三名集成了15个模型

分类：集成学习

- 常见集成学习方法
  - Bagging（Bootstrap Aggregating）
    - 对样本或特征随机取样，学习产生多个独立的模型，然后平均所有模型的预测值
    - 主要减小方差
    - 典型代表：随机森林
  - Boosting
    - 串行训练多个模型，后面的模型是基于前面模型的训练结果（误差）
    - 主要减小偏差
    - 典型代表： AdaBoost

随机森林

- 集成学习算法：随机森林
  - 最典型的Bagging算法：算法思想
  - 随机：每棵树，保证各棵树之间的独立性，采用两到三层的随机性
    - 随机有放回的抽取样本作为训练集
    - 随机选取m个特征作为树节点的划分特征
    - 随机选择特征取值进行分割 (不遍历特征所有取值)

样本和特征尽可能不同

  - 森林：多颗决策树集成
    - 假设使用三棵决策树组合成随机森林，每各不相同且预测结果相互独立，每棵树的预测错误率为 40%。那么两棵树以及上预测错误的概率下降为：三三棵全部错误+两棵树错误一个正确= $0.4^3 + 3 * 0.4^2 * (1 - 0.4) = 0.352$

sklearn.ensemble.RandomForestClassifier

AdaBoost

- 集成学习算法：AdaBoost
  - 最有代表性的Boosting算法
  - 算法思想：利用同一训练样本的不同加权版本，训练一组弱分类器，然后把这些弱分类器以加权的形式集成起来，形成一个最终的强分类器：
    - 在每一步迭代过程中，会给训练集中的样本赋予一个权重$w_1, w_2 \ldots, w_n$
    - 样本的初始权重都一样，设置为$\frac{1}{n}$；
    - 在每一步迭代过程中，
      - 被当前弱分类器分错的样本的权重会相应得到提高
      - 被当前弱分类器分对的样本的权重则会相应降低；
    - 弱分类器的权重则根据当前分类器的加权错误率来确定。

```
from sklearn.ensemble import AdaBoostClassifier
```

分类与预测

- 有监督学习：分类与预测
- 常用方法
  - 规则方法
  - 决策树
  - 贝叶斯方法
  - 最近邻方法
  - 支持向量机（SVM）
  - 神经网络
  - 集成方法
- 分类的评价指标
- 类不平衡问题

分类模型的评价

- 如何评价分类模型的效果？——以二分类为例
  - 基本概念
    - T/F：True or False，表示二分类结果的正确与否
    - P/N：Positive or Negative，表示算法对样本的判断
  - 四种简写的含义：
    - 真正(True Positive,TP): 样本为正例，预测为正，(正确)
    - 假负(False Negative,FN): 样本为正例，预测为负，(错误)
    - 假正(False Positive, FP): 样本为负例，预测为正，(错误)
    - 真负(True Negative, TP): 样本为负例，预测为负，(正确)

分类模型的评价

- 如何评价分类模型的效果？——指标Accuracy
  - 通常用混淆矩阵 表示：TP、 FN、 FP、 TP

		PREDICTED（预测） CLASS	
		Class=Yes	Class=No
ACTUAL（真实） CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- 评价指标1： $\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$

分类模型的评价

- 如何评价分类模型的效果？——指标Accuracy
  - 举例：假设一共有200个测试数据样本，以下是使用分类模型得到的分类结果，请问*Accuracy*是多少？

		PREDICTED（预测） CLASS	
		Class=Yes	Class=No
ACTUAL（真实）CLASS	Class=Yes	60 (TP)	40 (FN)
	Class=No	20 (FP)	80 (TN)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{60 + 80}{60 + 80 + 20 + 40} = 0.7$$

分类模型的评价

- 如何评价分类模型的效果？— Accuracy的局限性
  - 样本不均衡时，评估结果可能不合理
  - 例子：考虑2分类问题
    - 假设10000个数据样本中，类别0的样本数为 9990，类别1的样本数为 10
    - 假设模型将所有样本均预测为0
    - 计算$\text{Accuracy} = 9990/10000 = 99.9\ \%$

Accuracy很高，表明模型很好?

结论：模型不好

原因：10个正例均预测错误

分析：这个例子中，显然我们关注类别为1的样本，但Accuracy被类别为0的样本影响了

Count	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

分类模型的评价

- 如何评价分类模型的效果？——分类问题的常用指标
  - 准确率(查准率)Precision(p) $= \frac{a}{a+c} = \frac{TP}{TP+FP}$：预测为Yes中正确的比例
    - **正确预测的个体总数 / 预测出的个体总数**
  - 召回率(查全率) Recall(r) $= \frac{a}{a+b} = \frac{TP}{TP+FN}$: 真实为Yes中被预测正确的比例
    - **正确预测的个体总数 / 测试集中存在的个体总数**

Count		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

分类模型的评价

- 分类模型的其他指标——分类问题的常用指标
  - F值 $= \frac{2PR}{P+R} = \frac{2a}{2a+b+c}$：正确率和召回率的调和平均值
  - 意义：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
      - 牺牲准确率，保证较高召回率

Count		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

分类模型的评价

- 分类问题的常用指标——**课堂练习**
  - 在一次垃圾邮件检测中，使用某分类模型认为有100篇邮件是垃圾邮件，后经过专家判定，其中真是垃圾邮件的为60篇，其余的40篇为误分类，那么请问本次分类的准确率Precision就等于_____。
  - 假如专家发现邮件样本集里还有90篇垃圾邮件，由于各种原因而未被检出（漏检），那么按照上述公式，本次分类的查全率Recall就等于_____，F1值等于_____。

$$\text{Precision(P)} = \frac{a}{a+c}$$

$$\text{Recall(R)} = \frac{a}{a+b}$$

$$\text{F值} = \frac{2PR}{P+R} = \frac{2a}{2a+b+c}$$

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

分类模型的评价

- 分类模型的其他指标—ROC与AUC
  - ROC(Receiver Operating Characteristic)与AUC(Area Under the ROC curve)
    - 背景：发展于20世纪50年代的信号检测理论，用于分析噪声信号
    - 两者的关系：ROC曲线的面积就是AUC
  - 基本概念
    - 真正例率TPR(True Positive Rate) = TP/(TP+FN)
      - **预测为正且实际为正的样本占所有正样本的比例**
    - 假正例率FPR(False Positive Rate)=FP/(TN+FP)
      - **预测为正但实际为负的样本占所有负样本的比例**

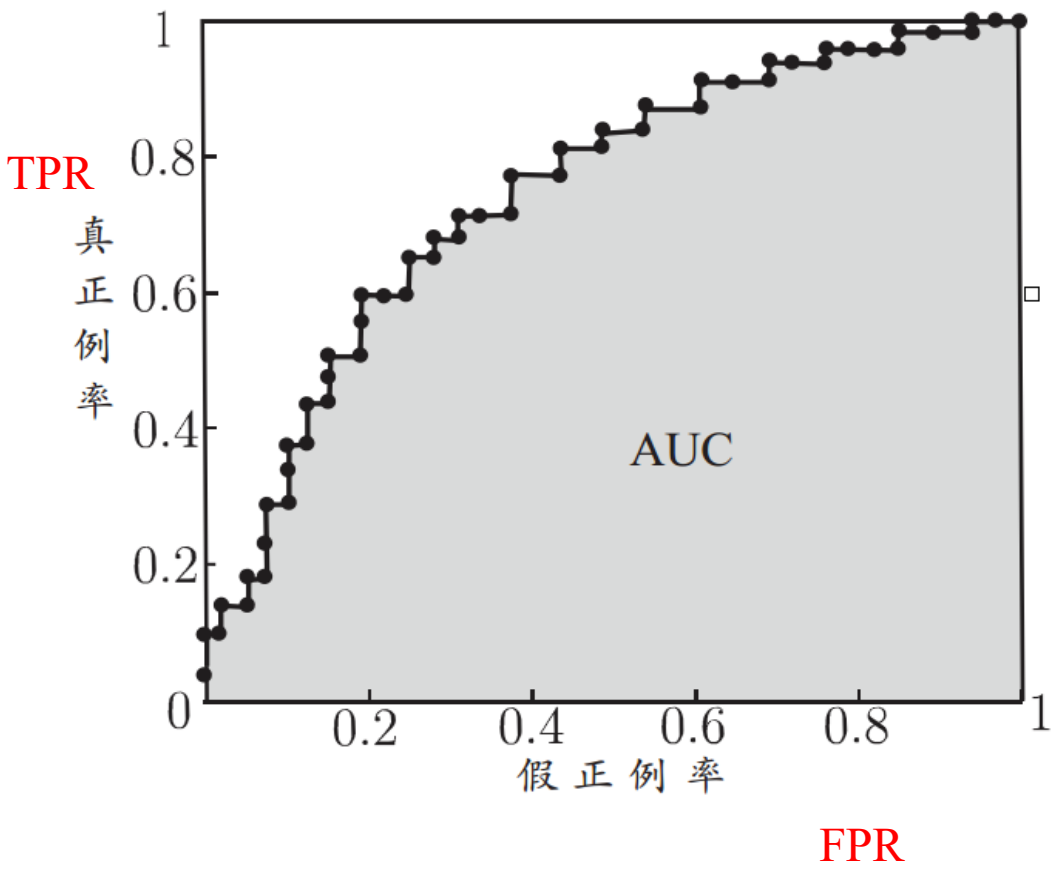
		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

分类模型的评价

- ROC曲线的绘制方法
  - 1. 测试集中m^+个正例和m^-个负例，根据学习器预测为正例的概率大小对测试样本进行排序
  - 2. 选择最低的预测概率为阈值，将该样本及预测概率高于它的样本全分为正类，计算TPR(=1)和FPR(=1)
  - 3. 选择下一个样本，将该样本及预测概率高于它的样本全分为正类，预测概率低于它的样本分为负类，计算TPR和FPR
    - 记当前(TPR, FPR) 为(x, y)
    - 当前样本若为**真正例**，则下一标记点的坐标为$(x, y - \frac{1}{m^+})$
    - 当前样本若为**假正例**，则下一标记点的坐标为$(x - \frac{1}{m^-}, y)$

分类模型的评价

- ROC曲线的绘制方法
  - 4. 重复步骤3，直到预测值最高的样本被选择
  - 5. 用线段连接相邻点

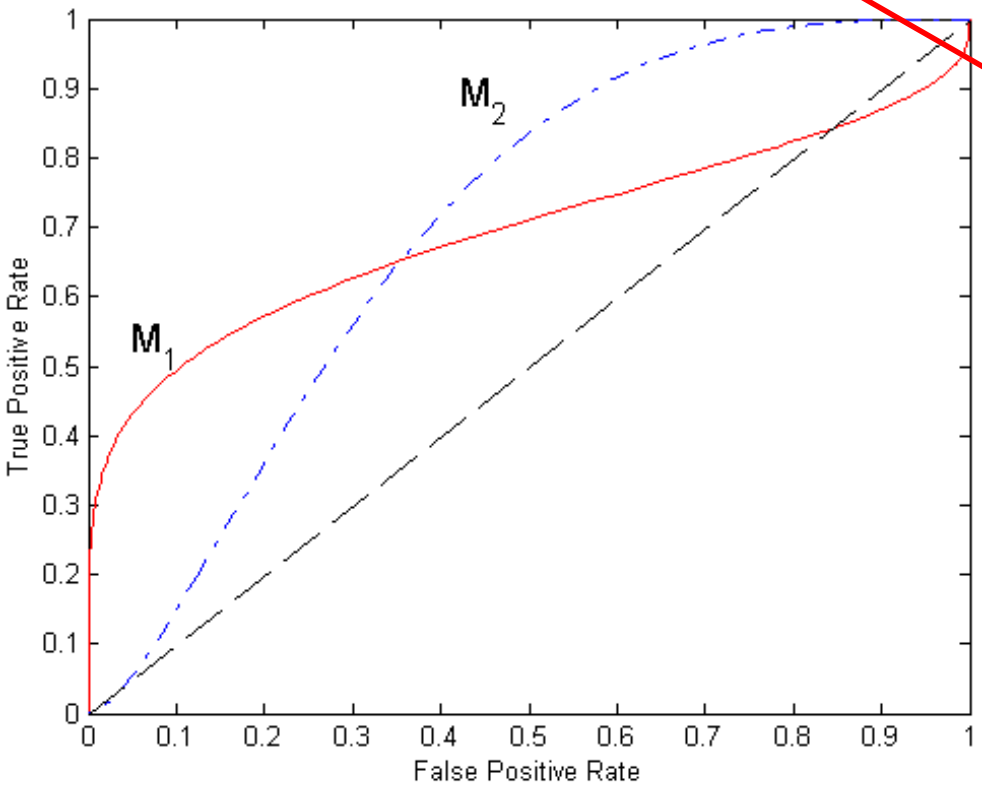


特点：

- 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）

分类模型的评价

- 如何基于ROC曲线比较模型好坏？
  - 一般而言，模型A的ROC曲线将模型B的完全包住，则模型A更好
  - 但往往并不会完全包住



对比ROC曲线发现，两个模型都不是一直表现得好

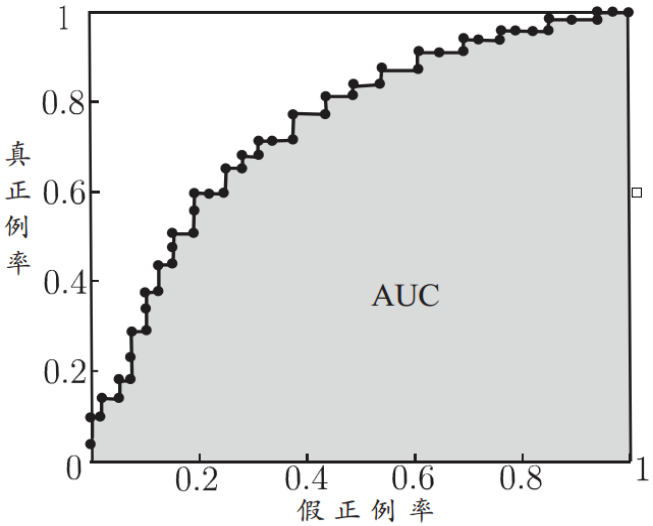
- M_1在FPR较小时表现好
- M_2在FPR较大时表现好

分类模型的评价

- ROC不能量化——AUC量化指标
  - AUC定义为ROC曲线的面积，可以直接计算如下
  - 假设ROC曲线由$\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m), x_1 = 0, x_m = 1\}$的点按序连接而形成，则AUC为：

$$AUC = \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1} - x_i) \cdot (y_{i+1} + y_i)$$

AUC越大，结果越好



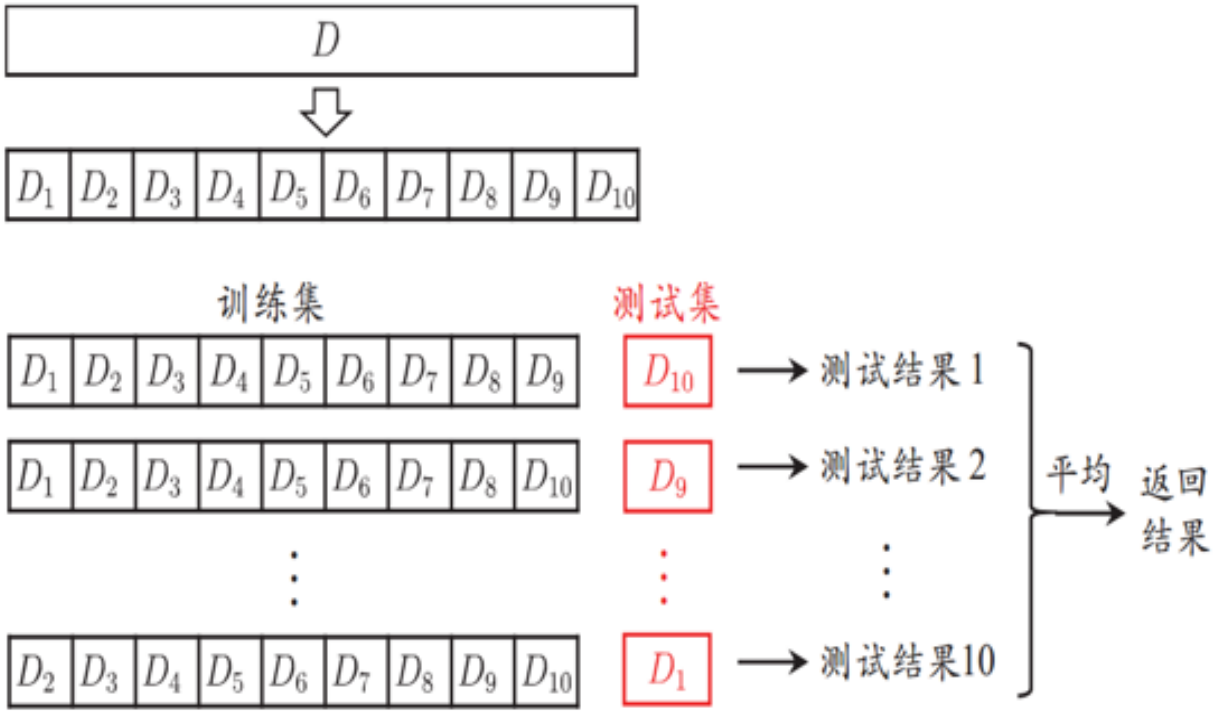
AUC衡量了样本预测的排序质量

分类模型的评价

- 如何评价分类模型的效果？—分类模型的比较方法
  - 关于性能比较:
    - 测试性能并不等于泛化性能
    - 测试性能随着测试集的变化而变化
    - 很多机器学习算法本身有一定的随机性
      - 例如，对两个模型:
        - M1: accuracy = 85%, 在30个样本上测试
        - M2: accuracy = 75%, 在5000个样本上测试
  - 常常进行假设检验，判断不同模型的性能差别是否具有统计意义
    - 假设检验为学习器性能比较提供了重要依据，基于其结果我们可以推断出：若在测试集上观察到学习器A比B好，则A的泛化性能是否在统计意义上优于B，以及这个结论的把握有多大。

分类模型的验证

- 分类模型的验证方法 —— 增加结果的可靠性
  - 交叉验证法（Cross Validation）：将数据集分层采样划分为k个大小相似的互斥子集，每次用k-1个子集的并集作为训练集，余下的子集作为测试集，最终返回k个测试结果的均值，k最常用的取值是10.



10折交叉验证

分类与预测

- 有监督学习：分类与预测
- 常用方法
  - 规则方法
  - 决策树
  - 贝叶斯方法
  - 最近邻方法
  - 支持向量机（SVM）
  - 神经网络
  - 集成方法
- 分类的评价指标
- 类不平衡问题