



本课件仅用于教学使用。未经许可，任何单位、组织和个人不得将课件用于该课程教学之外的用途(包括但不限于盈利等)，也不得上传至可公开访问的网络环境

1

新媒体大数据分析

New Media Big Data Analysis

第三章 数据建模

黄振亚，朱孟潇，张凯

课程主页：

<http://staff.ustc.edu.cn/~huangzhy/Course/NM2024.html>

助教：陈宗阳

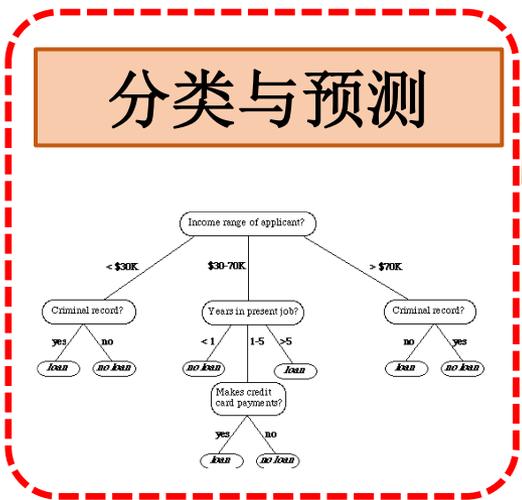
bigdata_2024@163.com

11/26/2024



数据建模基础

数据挖掘——四个任务有哪些常用方法？



关联分析

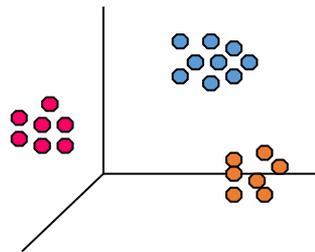


数据

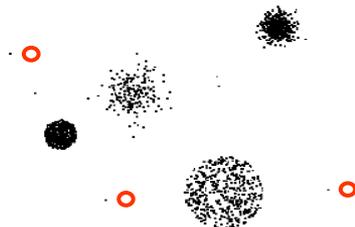
	T		H		P	
	L	H	L	H	L	H
J	-6.0	8.8	60	100	986	1044
F	-2.8	10.9	48	100	973	1025
M	-5.6	17.7	34	100	976	1037
A	-1.2	22.2	27	100	996	1036
M	-0.8	27.8	25	100	1003	1034
J	5.2	29.1	26	100	998	1030
J	9.8	30.6	23	99	997	1027
A	5.6	26.1	31	100	992	1029
S	5.2	24.8	35	100	998	1028
O	-0.4	21.3	42	100	990	1031
N	-7.6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

table 17a
2010 monthly weather variation, Cambridge (UK)

聚类



异常检测

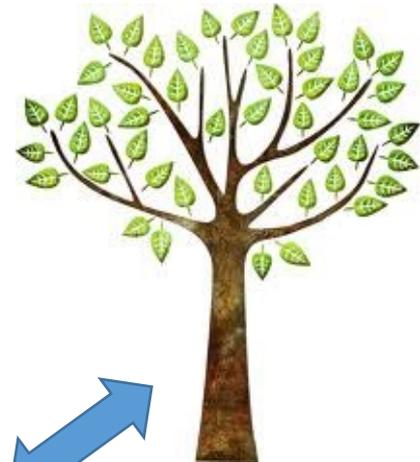




分类：决策树

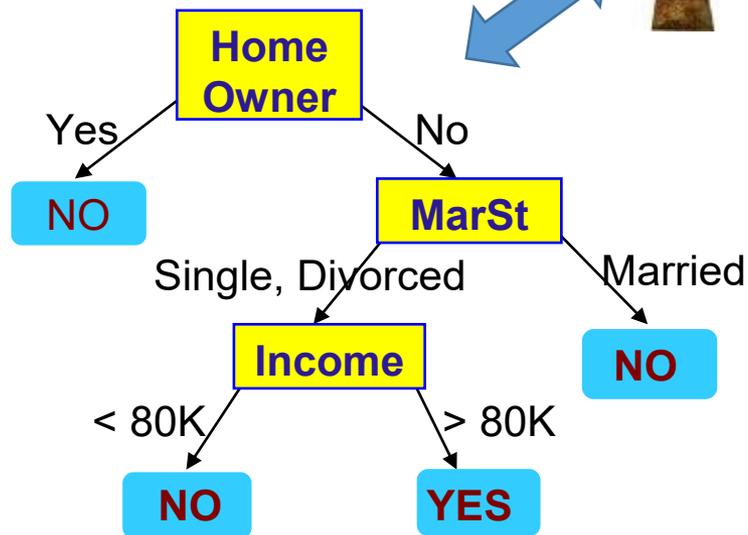
什么是决策树

- 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

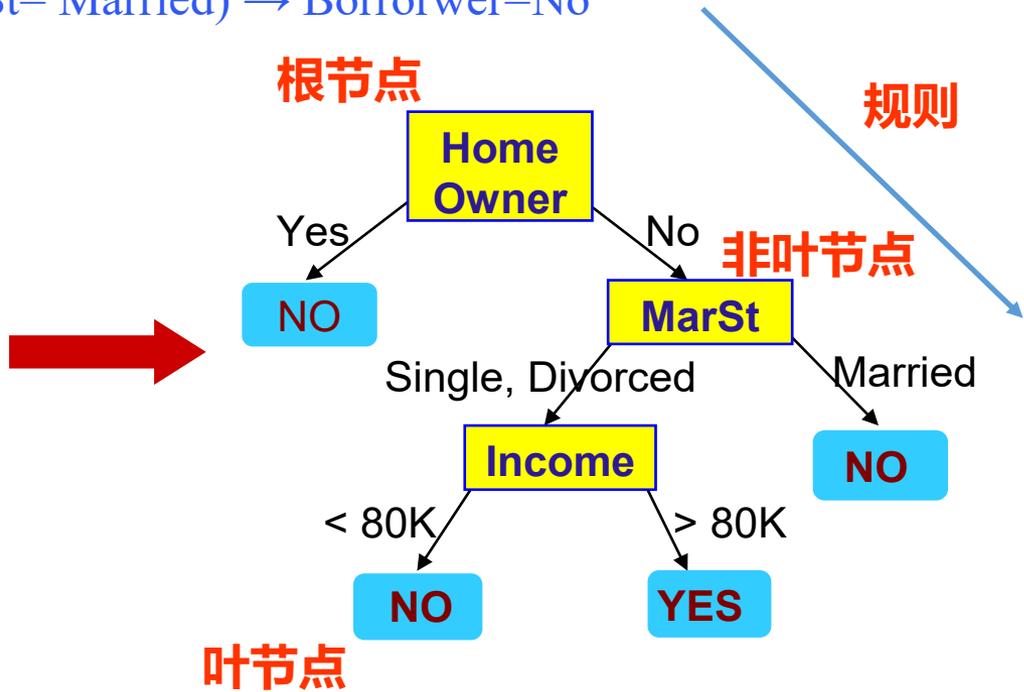


分类：决策树

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
 - $(HomOwn = No) \wedge (MarSt = Married) \rightarrow Borrower = No$

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



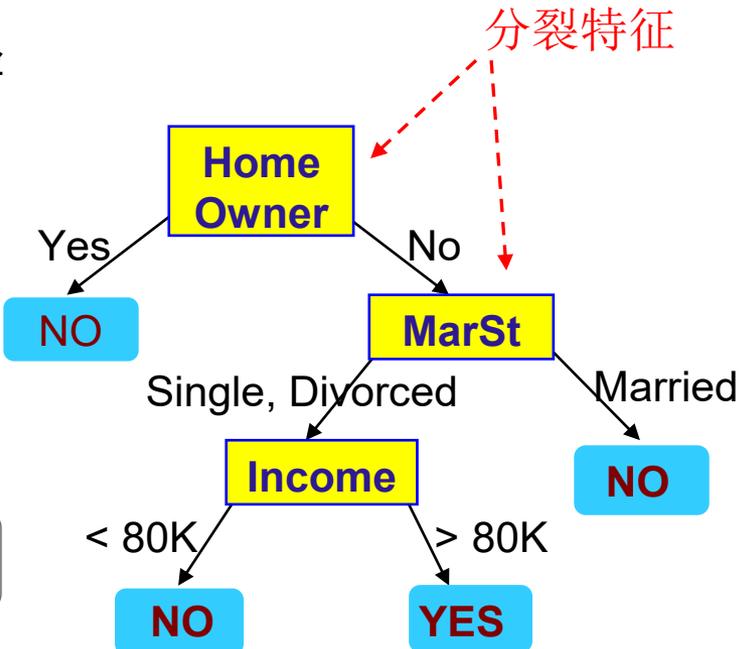
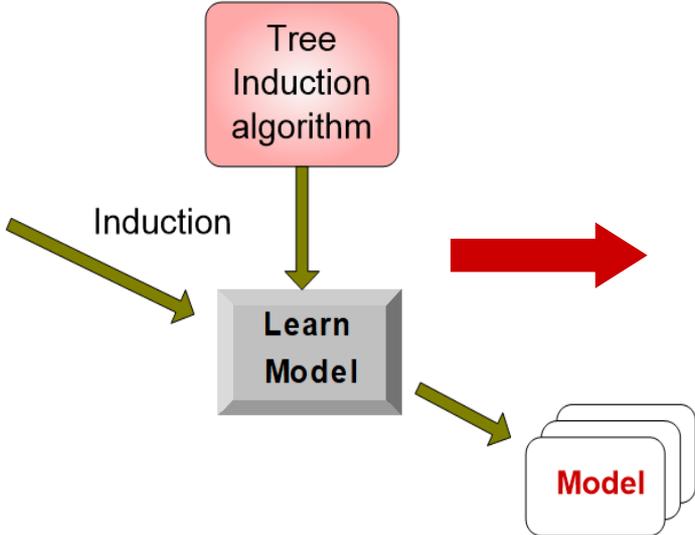


分类：决策树

- 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

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



分裂特征: Home Owner, MarSt, Income

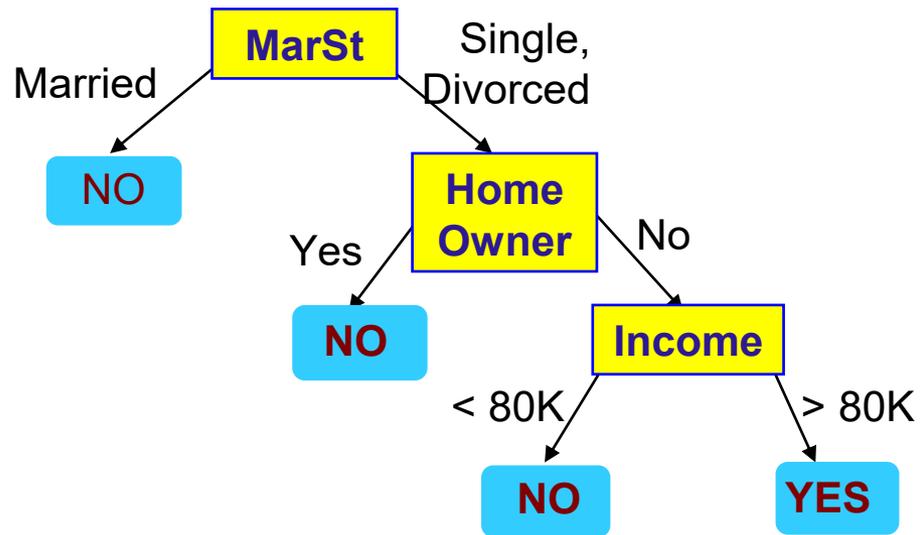
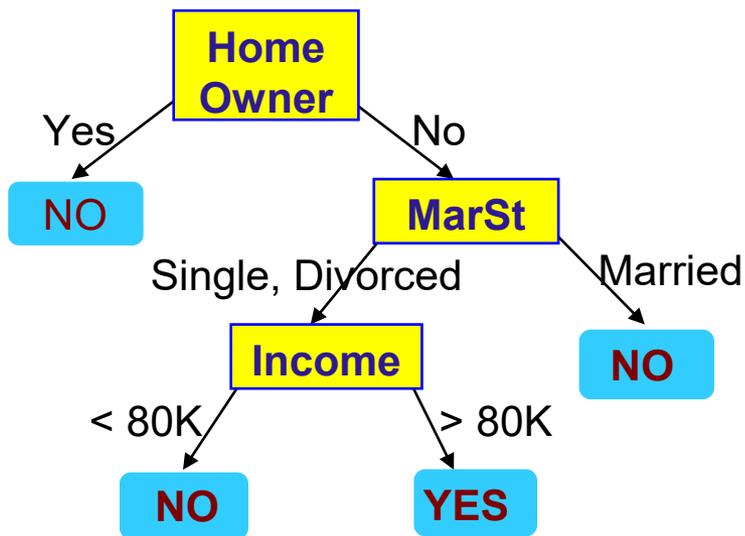
生成模型: 决策树



分类：决策树

是否有其他决策树?

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



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

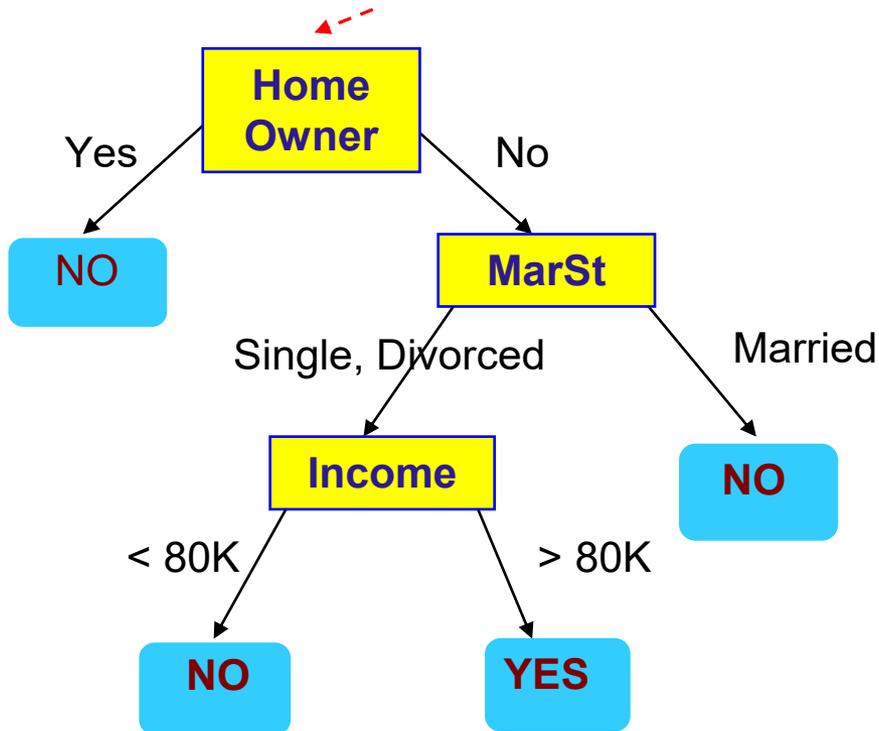


分类：决策树

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

从树根开始



测试数据（预测标签）

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

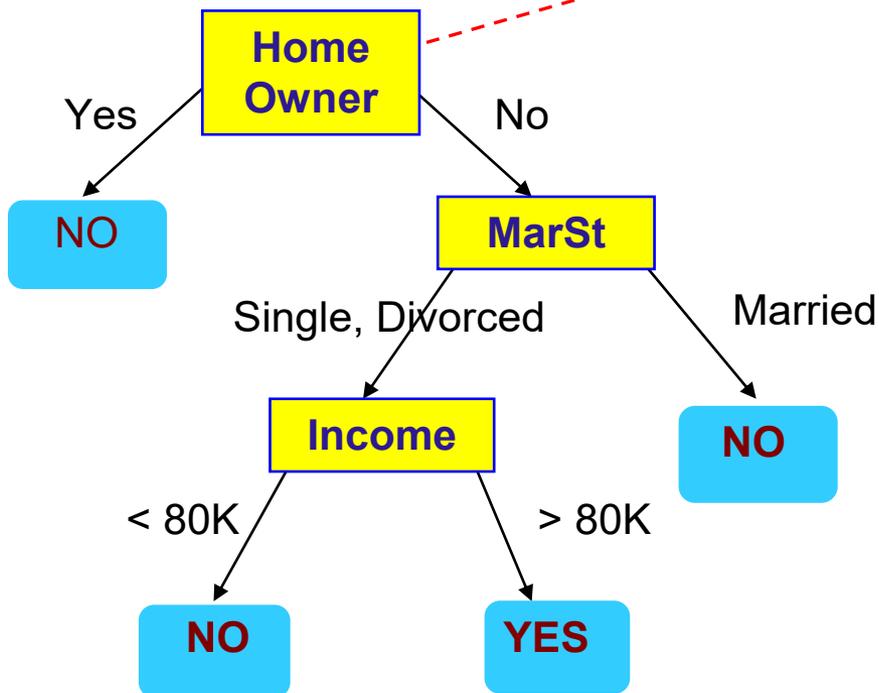


分类：决策树

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	?



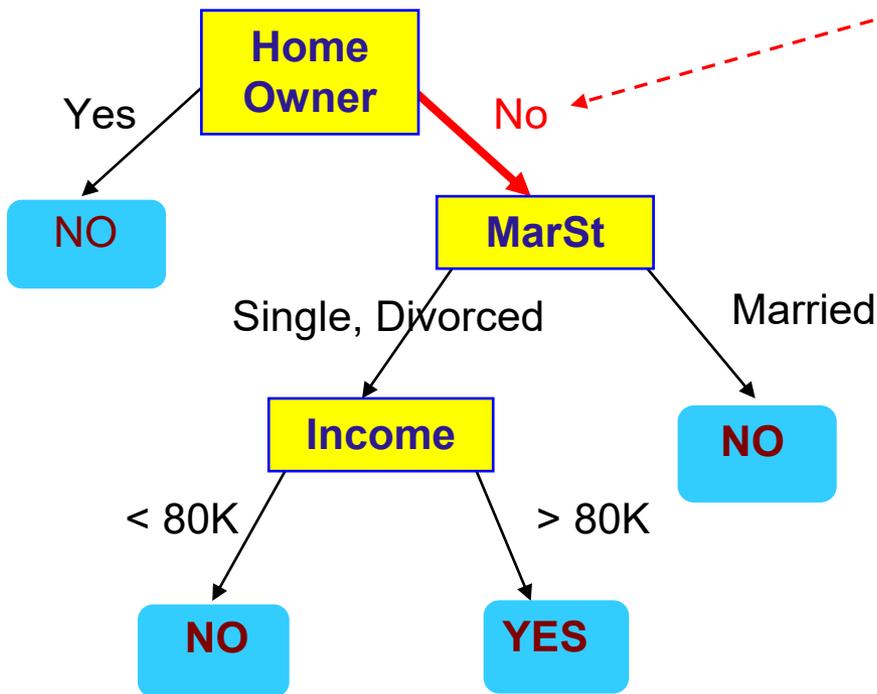


分类：决策树

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	?



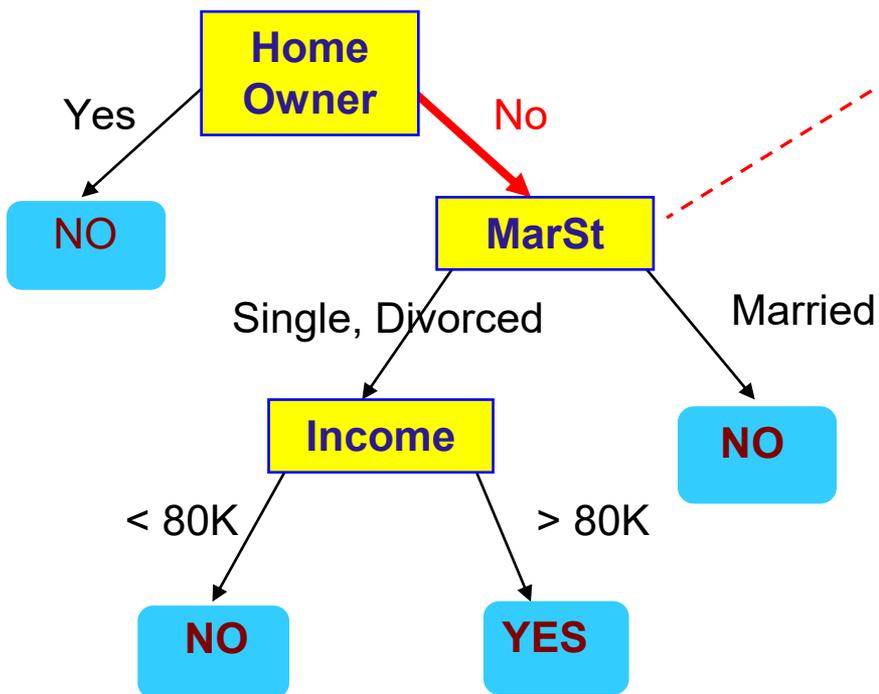


分类：决策树

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	?



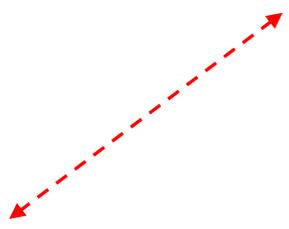
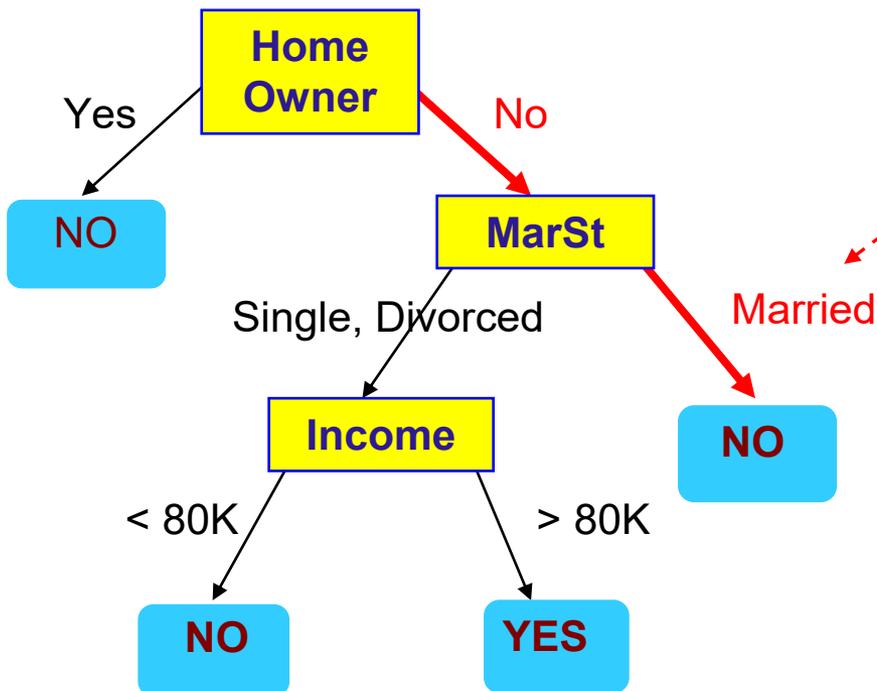


分类：决策树

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	?



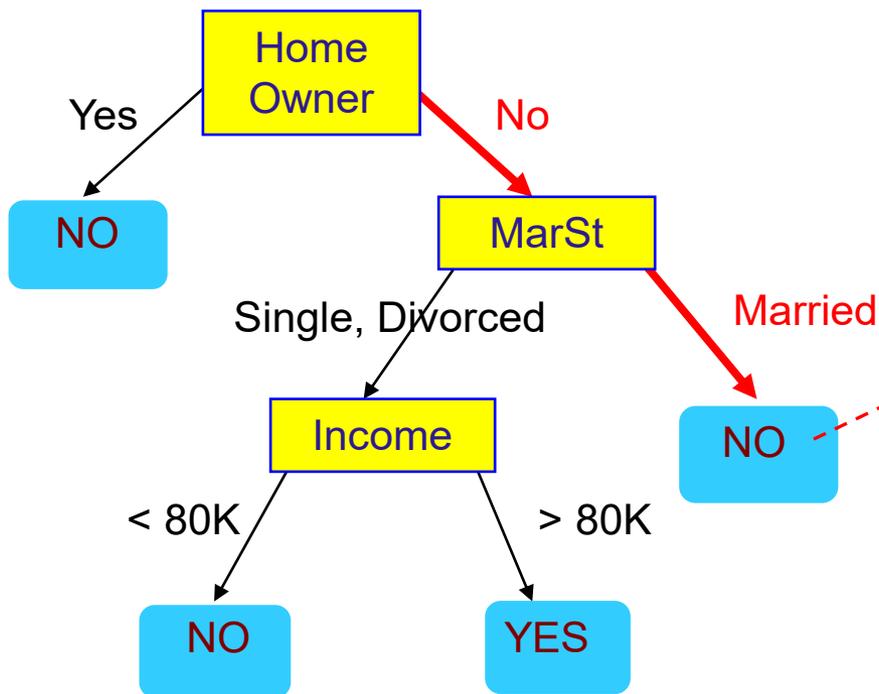


分类：决策树

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

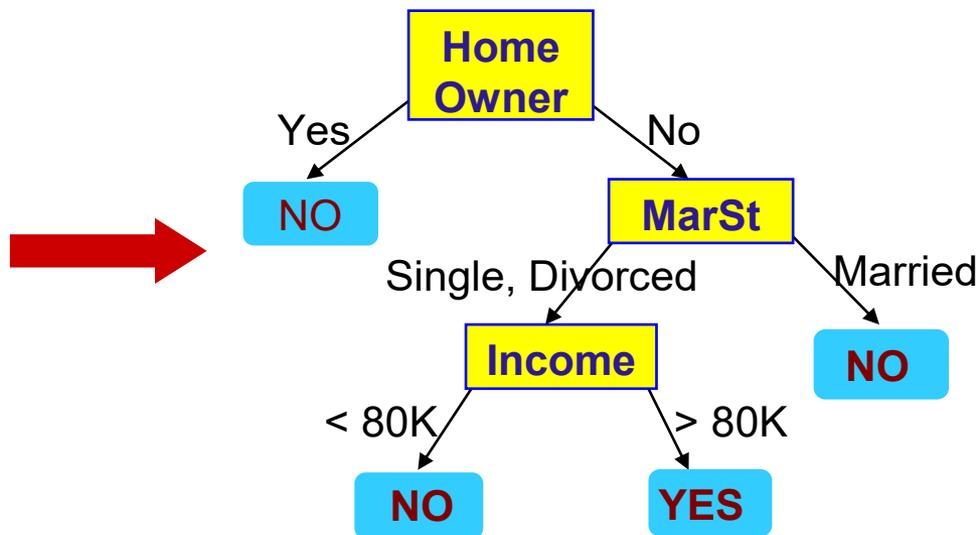


分类：决策树

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





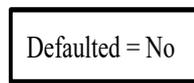
分类：决策树

1. 特征选择

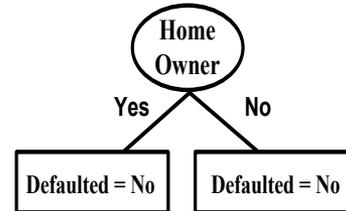
- 决策树的基本思想：特征分裂
- 初始节点包含所有的数据样本，我们希望这些样本能划分到同一个类里，如defaulted=No
- 但往往不成立，因此通过选择特征和取值，将样本集合不断划分

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

初始

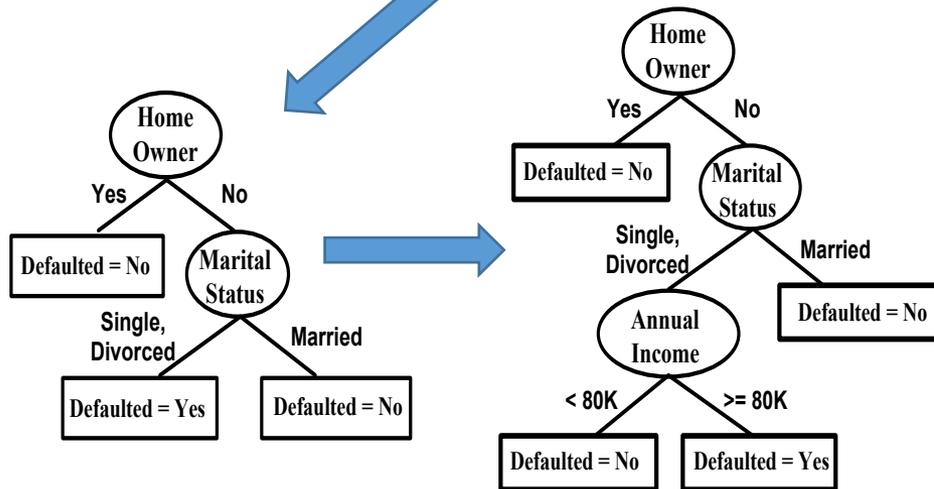


选择特征：Home Owner



(a)

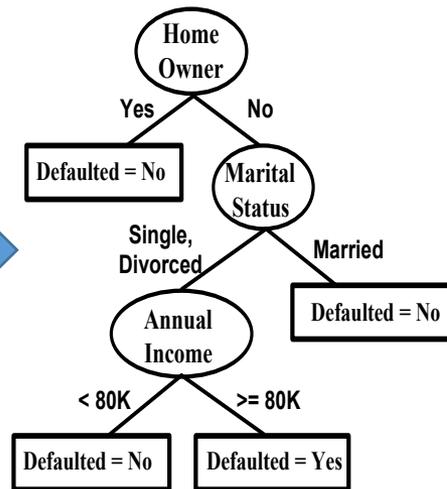
(b)



选择特征：
MarSt

(c)

选择特征：
Annual
Income



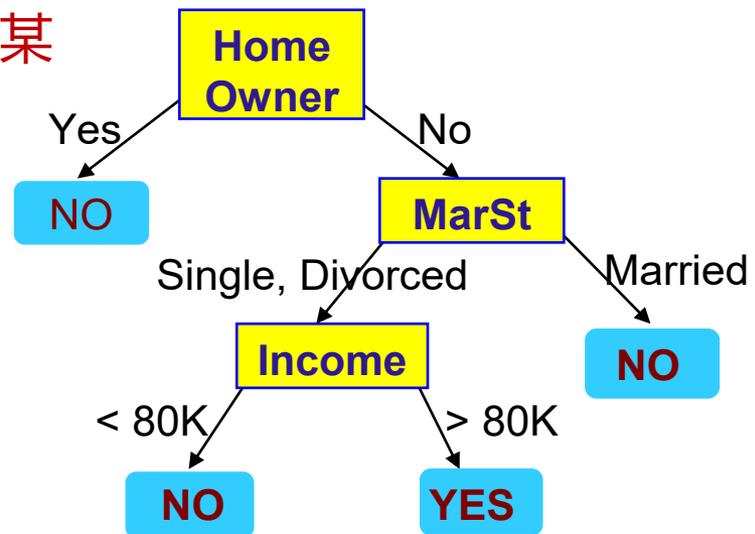
(d)



分类：决策树

- 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

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

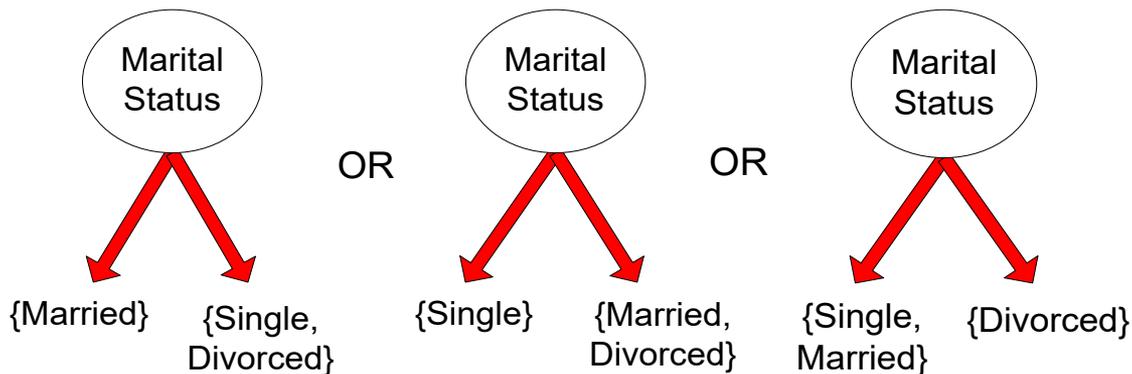
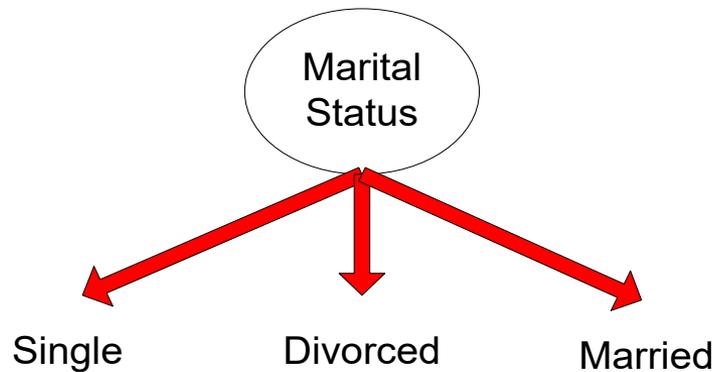




分类：决策树

1. 特征选择：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

40

- 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

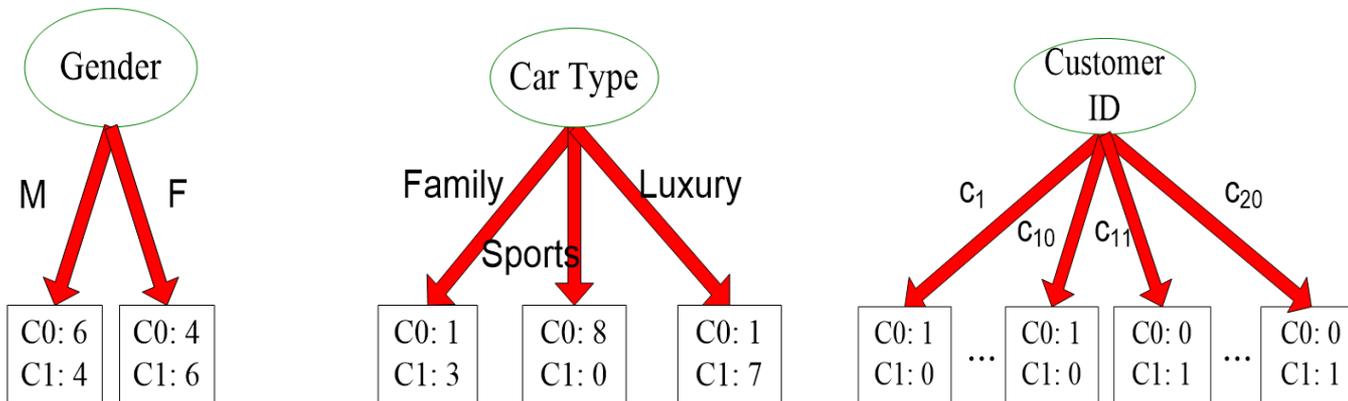
问题1: 如何选择分裂特征?

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

42

□ 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

43

▣ 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
 - $p(j|t)$ 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

44

- 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

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$Entropy = - 0 \log 0 - 1 \log 1 = - 0 - 0 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

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

C1	2
C2	4

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

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



决策树特征选择：信息增益

特征选择准则一：信息增益

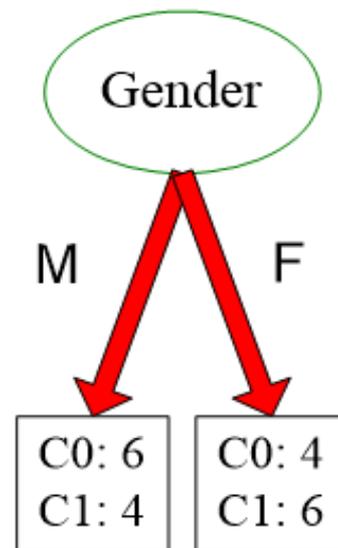
信息增益: 按某个特征划分之后，数据不确定性降低的程度

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

- 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

46

特征选择准则一：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

特征选择准则一：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

$Entropy(B_{Yes})$

特征A	Yes	No
特征B	No	No
类别	C0	C1

$Entropy(B_{No})$

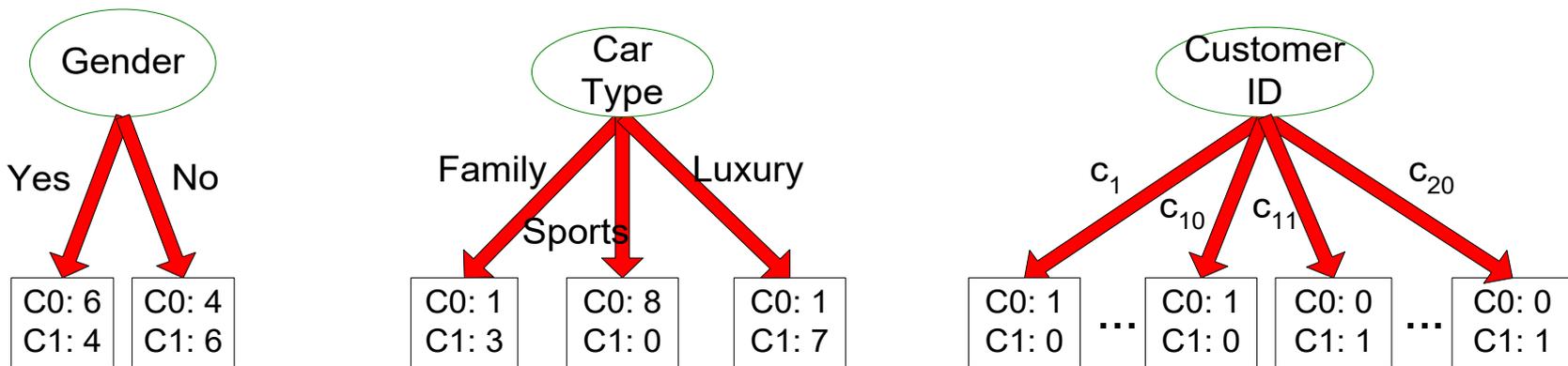
$$M = \frac{|B_{Yes}|}{n} Entropy(B_{Yes}) + \frac{|B_{No}|}{n} Entropy(B_{No})$$

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



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

49

特征选择准则二：信息增益率

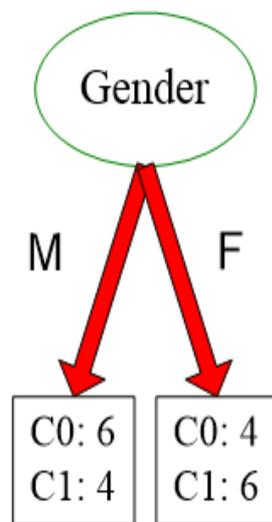
- 信息增益率(Gain ratio): 综合考虑划分结果信息增益和划分数量的信息

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

- 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

50

- 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

51

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：

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

- $p(j|t)$ 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

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

C1	0
C2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$Gini = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

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

C1	2
C2	4

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

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



决策树特征选择：基尼指数

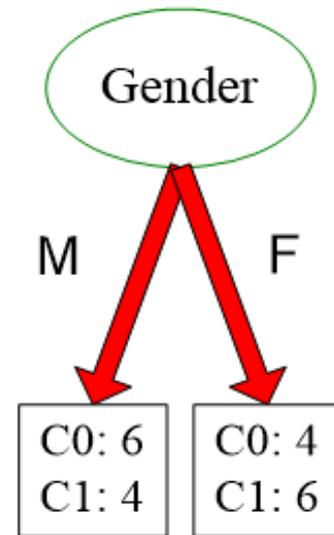
特征选择准则三：基尼指数

- 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



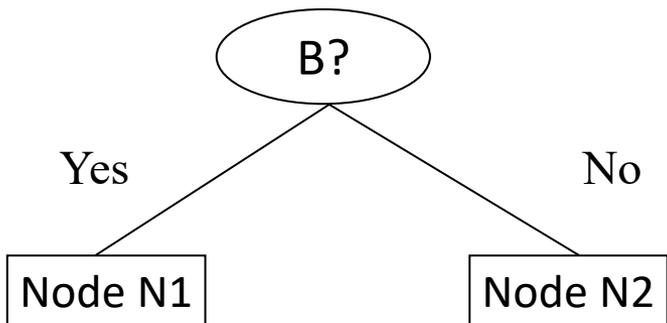
决策树特征选择：基尼指数

基尼指数计算示例

分裂前:

	Parent
C1	6
C2	6
Gini = 0.500	

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



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

	N1	N2
C1	5	2
C2	1	4
Gini=0.361		

$$\begin{aligned} Gini(N1) &= 1 - (5/6)^2 - (1/6)^2 \\ &= 0.278 \end{aligned}$$

$$\begin{aligned} Gini(N2) &= 1 - (2/6)^2 - (4/6)^2 \\ &= 0.444 \end{aligned}$$



分裂后:

$$\begin{aligned} Gini(Children) &= 6/12 * 0.278 + \\ &\quad 6/12 * 0.444 \\ &= 0.361 \end{aligned}$$



决策树特征选择

55

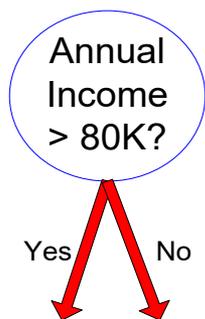
决策树特征选择：连续属性的分裂

将连续属性进行离散化

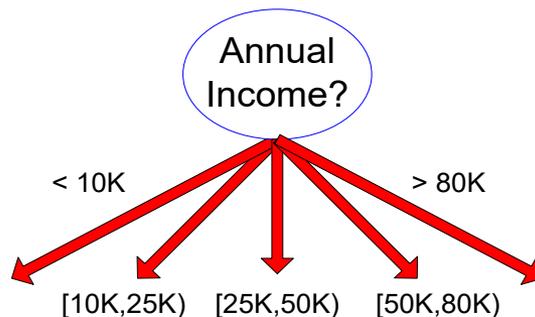
- 最简单：人为划分一次，如设定收入的阈值
- 通过等间隔分段、等频率分段(百分位数)或聚类找到划分位置
- 利用算法二分离散化考虑所有情况，找出最好的划分



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

56

□ 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

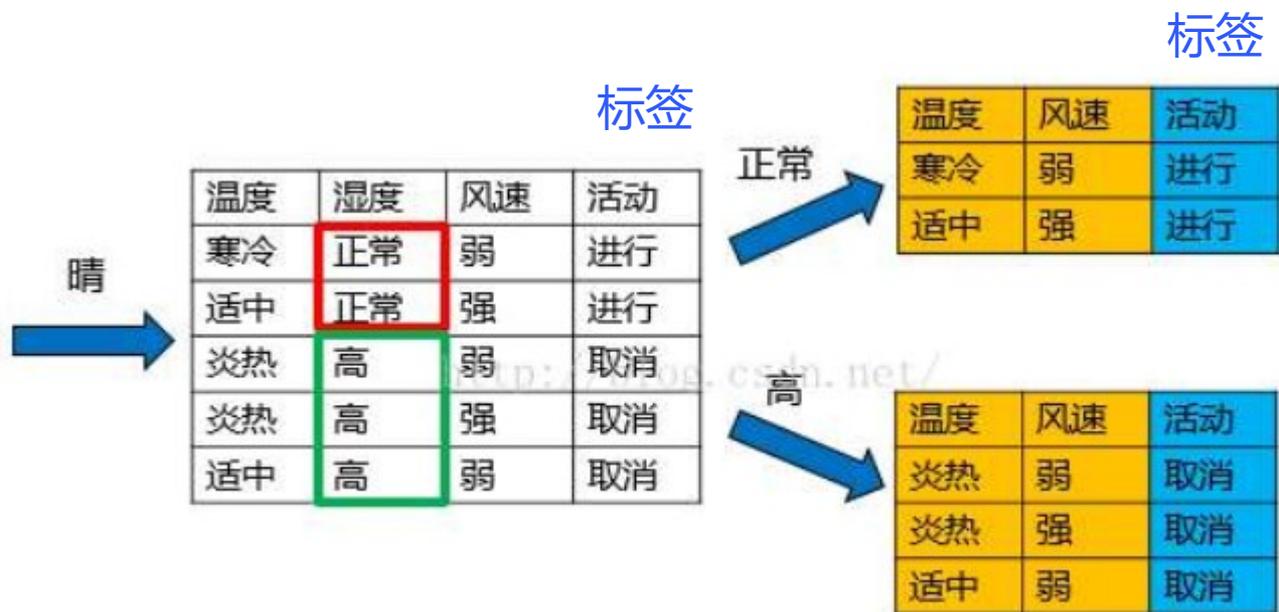
决策树算法有很多，如ID3, C4.5, CART等，核心区别在特征选择准则不同，具体算法请大家课后查阅学习



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

58

□ 3. 决策树剪枝

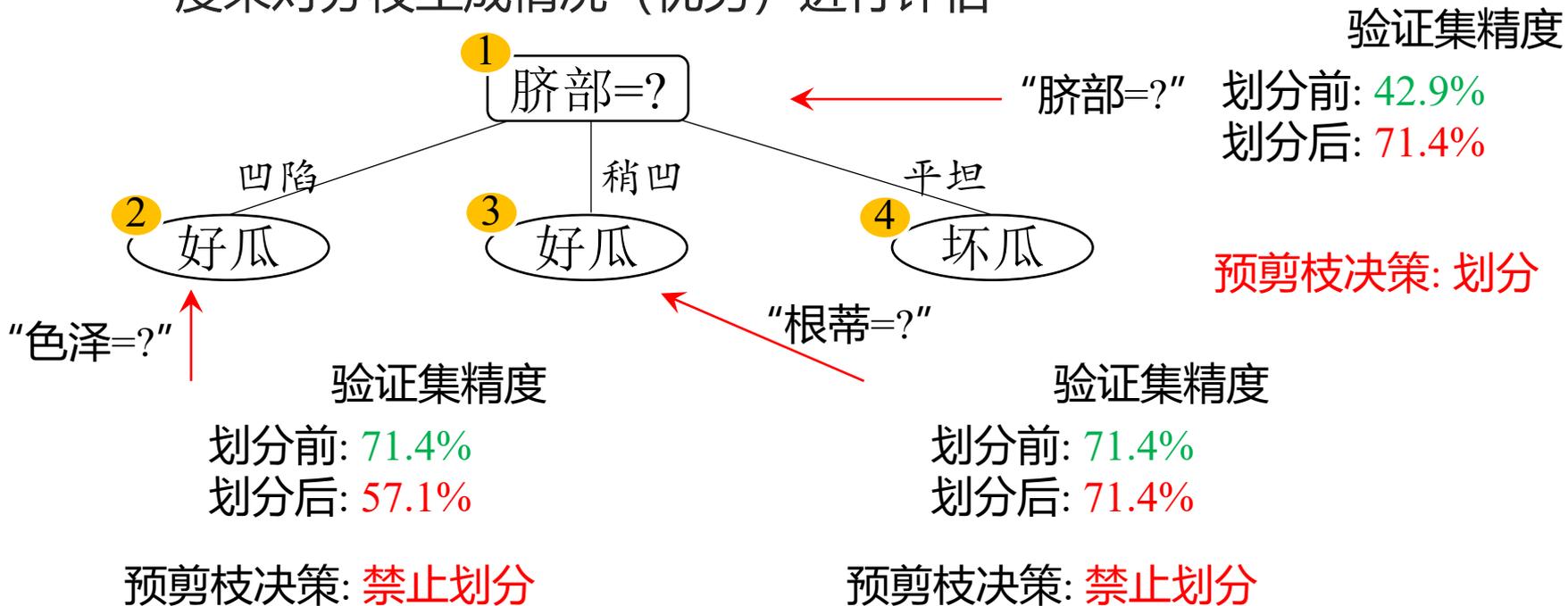
- 在生成决策树之后，我们还将根据实际情况，对决策树进行剪枝
 - 剪枝的原因在于训练过程的“过拟合”问题
 - 如果训练集与测试集效果都不好，说明出现“欠拟合”
 - 如果训练集效果好，而测试集效果不好，说明出现“过拟合”
- 过拟合出现的原因：训练过程中过度迁就训练数据特性，而导致构造出过于复杂、过于细枝末节的决策树，泛化能力较差
- 解决这一问题的办法在于对已生成的决策树进行简化，即“剪枝”
- 包括两种策略：预剪枝、后剪枝



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



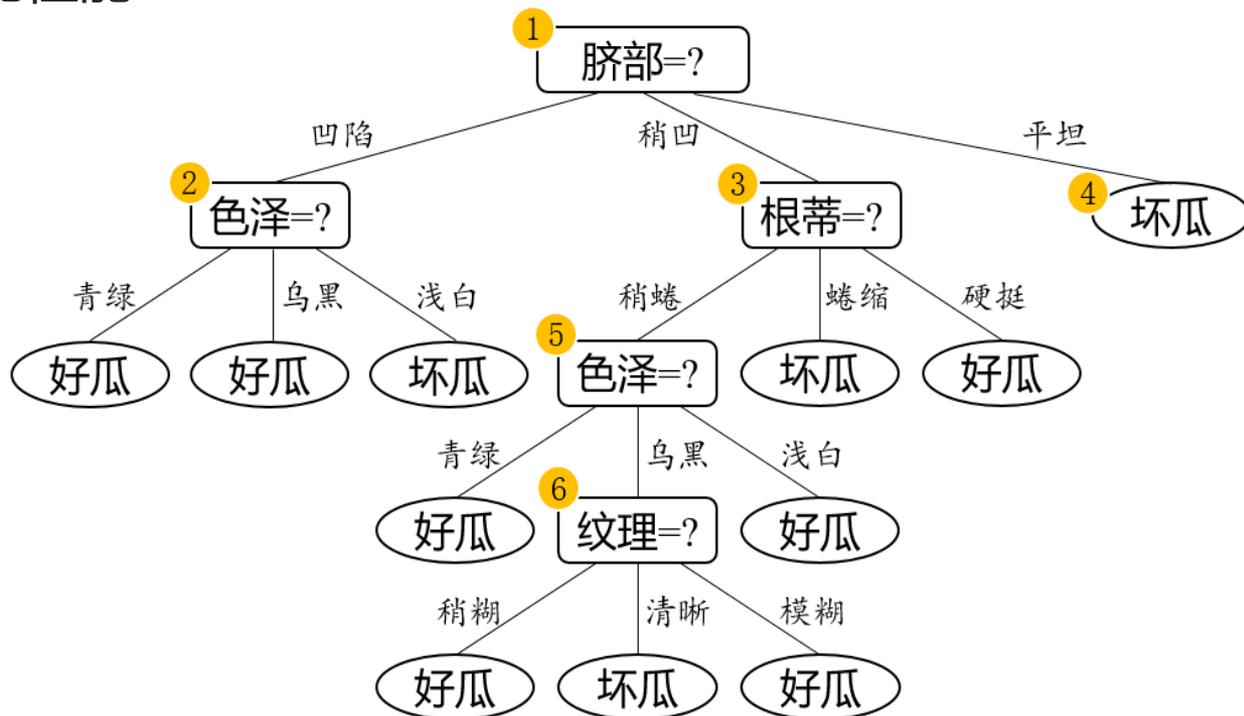


决策树剪枝

3. 决策树剪枝：后剪枝

- 在生成决策树之后再行剪枝，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



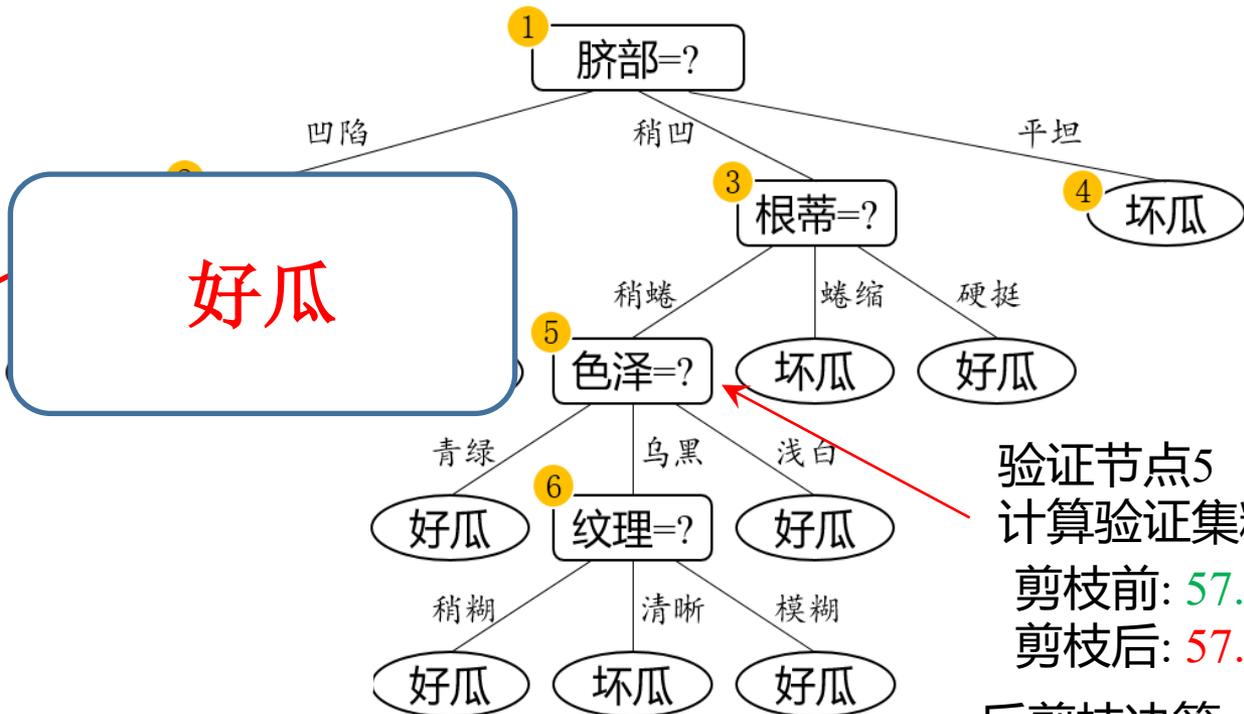


决策树剪枝

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

62

- **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

63

- 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)
 - 集成方法
- 类不平衡问题
- 分类的评价指标

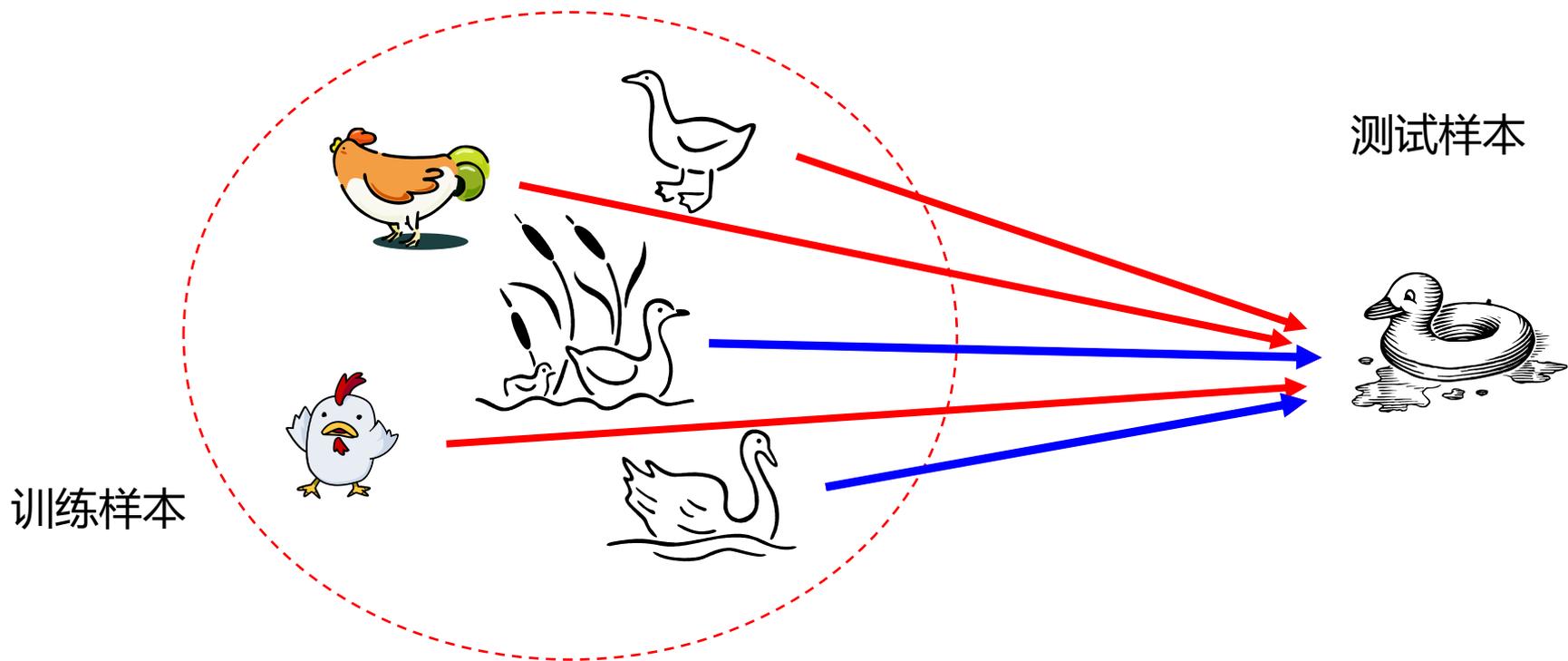


分类：K近邻方法

64

□ 分类——K近邻方法

□ 问题：判断测试样本是什么动物？





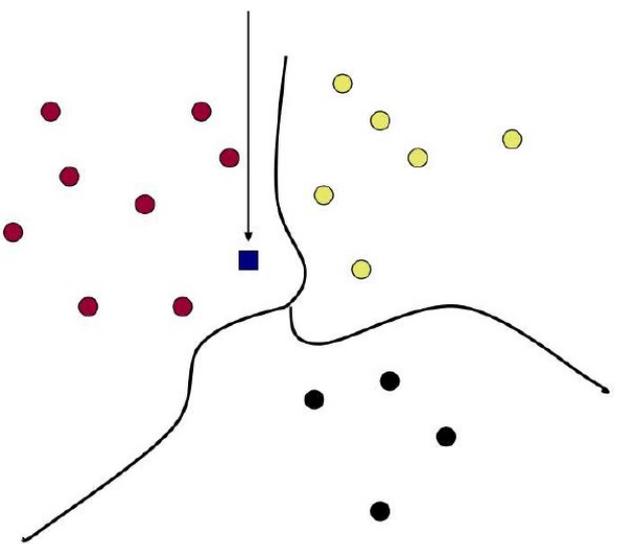
分类：K近邻方法

65

□ 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

待分类样本

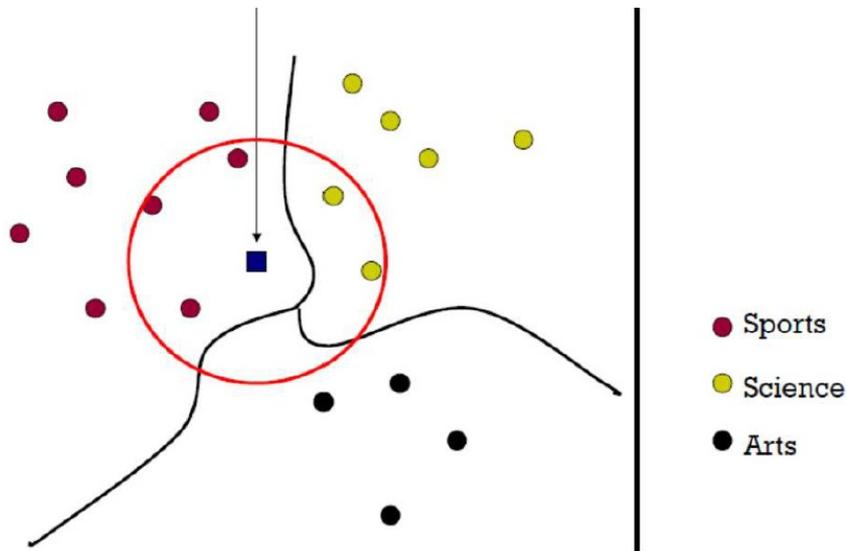


Similarity hypothesis true in general?

- Sports
- Science
- Arts



找到其K(5)个最相似样本



- Sports
- Science
- Arts



K近邻方法：距离度量

66

- 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

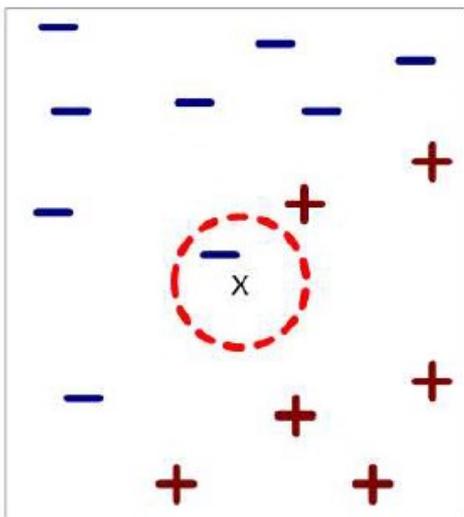
回顾第二章：数据集成



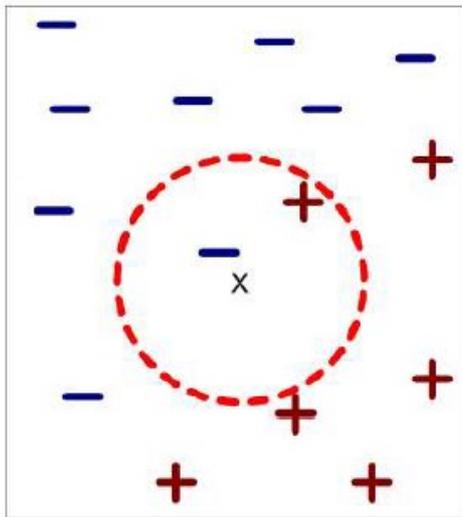
K近邻方法：K的取值

67

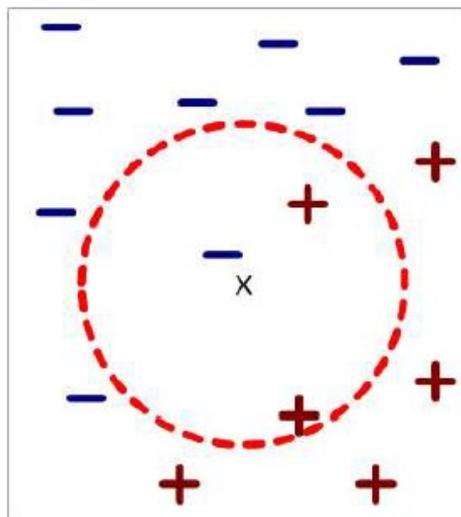
- K近邻方法核心问题：K的取值
 - K近邻分类的效果同样严重依赖于K的取值（即邻居的数量）
 - K太小，容易受噪声干扰；
 - K太大，可能导致错误涵盖其他类别样本



(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

68

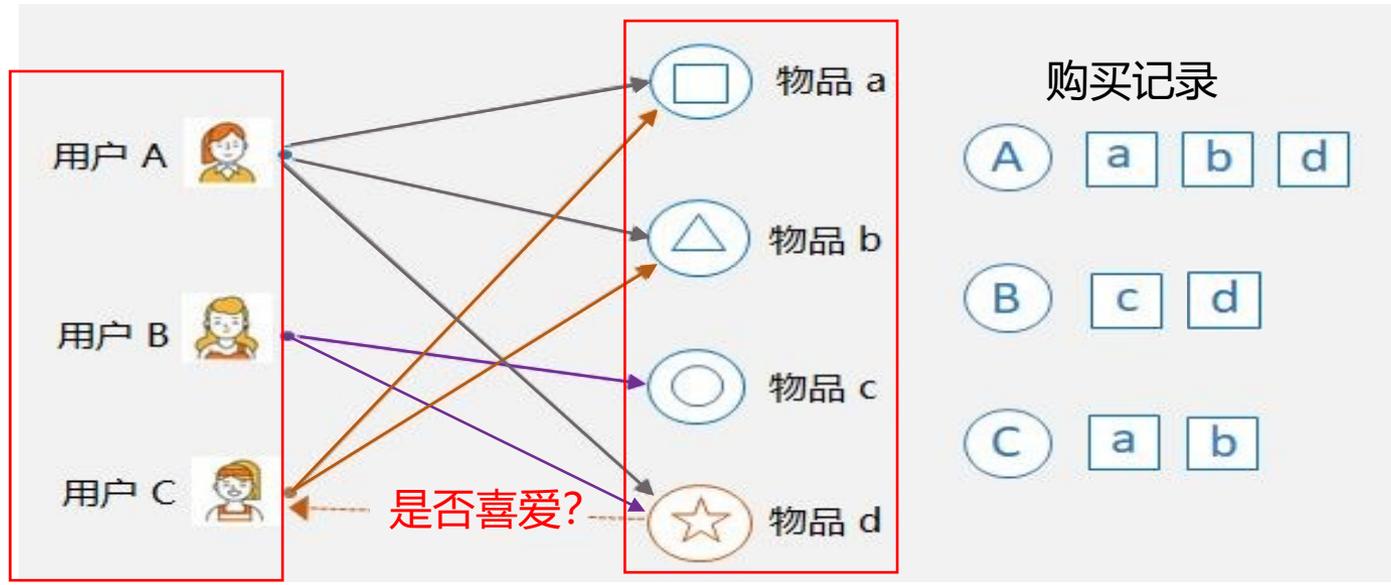
□ K近邻方法的特点

- 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

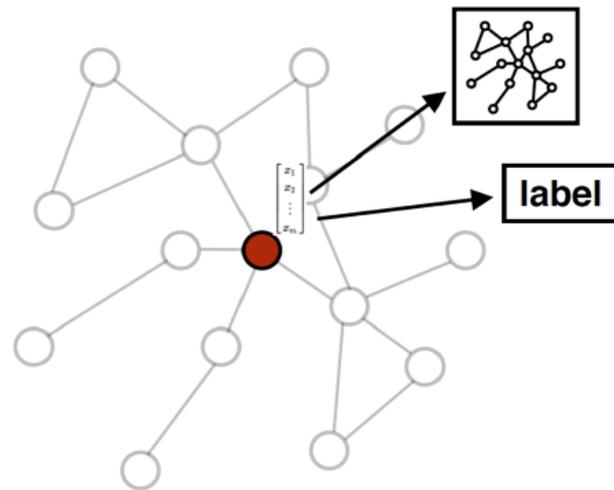
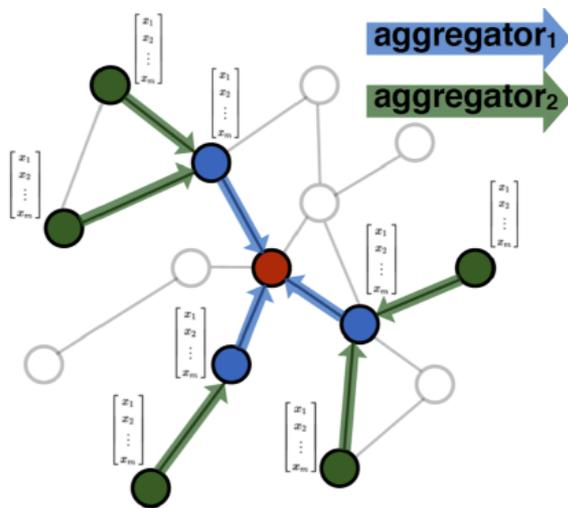
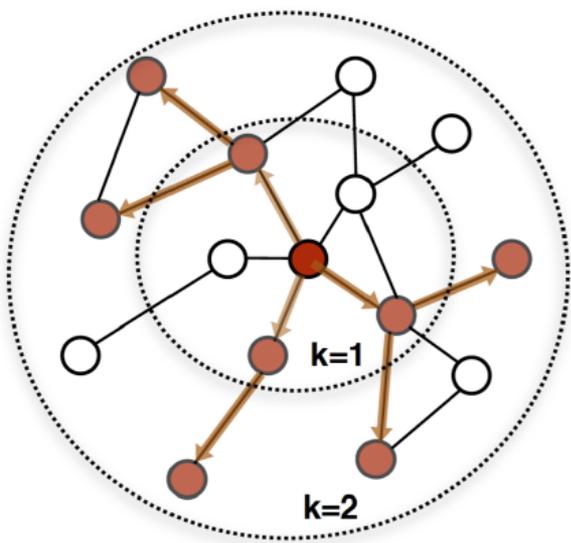
➤ Sarwar, Badrul, et al. Item-based collaborative filtering recommendation algorithms. WWW 2001.
➤ Amazon recommendations: Item-to-item collaborative filtering. IEEE Internet computing, 2003



K近邻方法实例

70

- 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, Zhitaoying, and Jure Leskovec. "Inductive representation learning on large graphs." NeurIPS 2017