



数据挖掘

Data Mining

规则和最近邻分类器

- 主讲人：丁兆云



数据挖掘

Data Mining

规则和最近邻分类器

- 主讲人：丁兆云



内容提纲

基于规则的分类
急切学习与惰性学习
最近邻分类器



基于规则的分类



1 基于规则的分类

- 使用一组 “if...then...” 规则进行分类

- 规则: $(Condition) \rightarrow y$

- 其中

- $Condition$ 是属性测试的合取
- y 是类标号

- 左部: 规则的前件或前提 (Rule antecedent)

- 右部: 规则的结论 (Rule consequent)

- 分类规则的例子:

- $(Blood\ Type=Warm) \wedge (Lay\ Eggs=Yes) \rightarrow Birds$
- $(Taxable\ Income < 50K) \wedge (Refund=Yes) \rightarrow Evade = No$

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	否	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲑鱼	冷血	鳞片	否	是	否	否	否	鱼类
鲸	恒温	毛发	是	是	否	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	是	是	是	爬行类
蝙蝠	恒温	毛发	否	否	否	是	是	哺乳类
鸽子	恒温	羽毛	是	否	否	是	否	鸟类
猫	恒温	软毛	否	否	否	否	否	哺乳类
虹鳟	冷血	鳞片	是	是	是	否	否	鱼类
美洲鳄	冷血	鳞片	否	否	半	否	否	爬行类
企鹅	恒温	羽毛	否	否	否	否	是	鸟类
豪猪	恒温	刚毛	是	否	否	否	否	哺乳类
鳗鲡	冷血	鳞片	否	是	半	否	是	鱼类
蝾螈	冷血	无	否	否	否	否	否	两栖类



1.1 基于规则的分类: 例

■ 脊椎动物数据集

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	否	爬行类
鲤鱼	冷血	鳞片	是	是	否	否	是	鱼类
鲸	恒温	毛发	否	否	否	是	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	是	是	否	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
鸽子	恒温	羽毛	否	否	是	否	否	鸟类
猫	恒温	软毛	是	否	否	否	否	哺乳类
虹鱥	冷血	鳞片	是	是	否	否	否	鱼类
美洲鳄	冷血	鳞片	否	半	半	否	是	爬行类
企鹅	恒温	羽毛	否	半	半	否	是	鸟类
豪猪	恒温	刚毛	是	否	否	否	否	哺乳类
鳗鲡	冷血	鳞片	否	是	否	否	是	鱼类
蝾螈	冷血	无	否	半	否	否	是	两栖类



1.1 基于规则的分类: 例

- 规则 r 覆盖 实例 x (记录), 如果该实例的属性满足规则 r 的条件

r_1 : (胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类

r_2 : (胎生 = 否) \wedge (水生动物 = 是) \rightarrow 鱼类

r_3 : (胎生 = 是) \wedge (体温 = 恒温) \rightarrow 哺乳类

r_4 : (胎生 = 否) \wedge (飞行动物 = 否) \rightarrow 爬行类

r_5 : (水生动物 = 半) \rightarrow 两栖类

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
鹰	恒温	羽毛	否	否	是	是	否	?
灰熊	恒温	软毛	是	否	否	是	是	?

- 规则 r_1 覆盖 “鹰” \Rightarrow 鸟类



根据规则集，灰熊属于什么类别

A 鸟

$r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$

B 鱼

$r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$

C 哺乳

$r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$

D 爬行

$r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$

$r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$



1.1 基于规则的分类: 例

- 规则 r 覆盖 实例 x (记录), 如果该实例的属性满足规则 r 的条件

r_1 : (胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类

r_2 : (胎生 = 否) \wedge (水生动物 = 是) \rightarrow 鱼类

r_3 : (胎生 = 是) \wedge (体温 = 恒温) \rightarrow 哺乳类

r_4 : (胎生 = 否) \wedge (飞行动物 = 否) \rightarrow 爬行类

r_5 : (水生动物 = 半) \rightarrow 两栖类

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
鹰	恒温	羽毛	否	否	是	是	否	?
灰熊	恒温	软毛	是	否	否	是	是	?

- 规则 r_1 覆盖 “鹰” \Rightarrow 鸟类
- 规则 r_3 覆盖 “灰熊” \Rightarrow 哺乳类



1.2 规则的质量

- 用覆盖率和准确率度量
- 规则的覆盖率 (Coverage)：
 - 满足规则前件的记录所占的比例
- 规则的准确率 (Accuracy)：
 - 在满足规则前件的记录中，满足规则后件的记录所占的比例
- 规则: $(\text{Status}=\text{Single}) \rightarrow \text{No}$
 $\text{Coverage} = 40\%, \text{ Accuracy} = 50\%$

Tid	Refund	Marital Status	Taxable Income	Class
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



1.3 如何用规则分类

■ 一组规则

- $r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$
- $r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$
- $r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$
- $r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$
- $r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$

■ 待分类记录

名称	体温	胎生	飞行动物	水生动物	类
狐猴	恒温	是	否	否	?
海龟	冷血	否	否	半水生	?
狗鲨	冷血	是	否	是	?

- 狐猴触发规则 r_3 , 它分到哺乳类
- 海龟触发规则 r_4 和 r_5 ----冲突
- 狗鲨未触发任何规则



1.4 规则分类的特征

- 互斥规则集
 - 每个记录最多被一个规则覆盖
 - 如果规则都是相互独立的，分类器包含互斥规则
 - 如果规则集不是互斥的
 - 一个记录可能被多个规则触发
 - 如何处理？
 - 有序规则集
 - 基于规则的序 vs 基于类的序
 - 无序规则集
 - 在无序规则方案中，允许一条记录触发多条规则，规则被触发时视为**对其相应类的一次投票**，然后计算不同类的票数（可以使用加权方式）来决定记录的类所属。
- $r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$
 $r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$
 $r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$
 $r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$
 $r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$
- 海龟触发规则 r_4 和 r_5 ——冲突



1.4 规则分类的特征(续)

- 穷举规则集
 - 每个记录至少被一个规则覆盖
 - 如果规则集涵盖了属性值的所有可能组合，则规则集具有穷举覆盖
- 如果规则集不是穷举的
 - 一个记录可能不被任何规则触发
 - 如何处理?
 - 使用缺省类



规则分类具有下列哪些特征

- A 互斥规则集
- B 非互斥规则集
- C 穷举规则集
- D 非穷举规则集

提交



1.4.1有序规则集

- 根据规则优先权将规则排序定秩 (rank)
 - 有序规则集又成决策表 (decision list)
- 对记录进行分类时
 - 由被触发的，具有最高秩的规则确定记录的类标号
 - 如果没有规则被触发，则指派到缺省类

r_1 : (胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类
 r_2 : (胎生 = 否) \wedge (水生动物 = 是) \rightarrow 鱼类
 r_3 : (胎生 = 是) \wedge (体温 = 恒温) \rightarrow 哺乳类
 r_4 : (胎生 = 否) \wedge (飞行动物 = 否) \rightarrow 爬行类
 r_5 : (水生动物 = 半) \rightarrow 两栖类

名称	体温	胎生	飞行动物	水生动物	类
海龟	冷血	否	否	半水生	?



1.4.2 规则定序方案

- 基于规则的序
 - 根据规则的质量排序（覆盖率(coverage)和准确率(accuracy)）
- 基于类的序
 - 属于同一类的规则放在一起
 - 基于类信息（如类的分布、重要性）对每类规则排序

基于规则的排序

(表皮覆盖=羽毛, 飞行动物=是) \Rightarrow 鸟类
(体温=恒温, 胎生=是) \Rightarrow 哺乳类
(体温=恒温, 胎生=否) \Rightarrow 鸟类
(水生动物=半) \Rightarrow 两栖类
(表皮覆盖=鳞片, 水生动物=否) \Rightarrow 爬行类
(表皮覆盖=鳞片, 水生动物=是) \Rightarrow 鱼类
(表皮覆盖=无) \Rightarrow 两栖类

基于类的排序

(表皮覆盖=羽毛, 飞行动物=是) \Rightarrow 鸟类
(体温=恒温, 胎生=否) \Rightarrow 鸟类
(体温=恒温, 胎生=是) \Rightarrow 哺乳类
(水生动物=半) \Rightarrow 两栖类
(表皮覆盖=无) ==> 两栖类
(表皮覆盖=鳞片, 水生动物=否) \Rightarrow 爬行类
(表皮覆盖=鳞片, 水生动物=是) \Rightarrow 鱼类



1.5如何建立基于规则的分类器

- 直接方法:
 - 直接由数据提取规则
 - 例如: **RIPPER, CN2, Holte's 1R**

- 间接方法:
 - 由其他分类模型提取规则(例如, 从决策树等).
 - 例如: **C4.5rules**



1.5.1 直接方法：顺序覆盖

- 基本思想

- 依次对每个类建立一个或多个规则

- 对第*i*类建立规则

- 第*i*类记录为正例，其余为负例

- 建立一个第*i*类的规则*r*，尽可能地覆盖正例，而不覆盖负例（即构建一个正例的规则）

- 删除*r*覆盖的所有记录，在剩余数据集上学习下一个规则，直到所有第*i*类记录都被删除

$r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$

$r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$

$r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$

$r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$

$r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$



1.5.1 直接方法：顺序覆盖

顺序覆盖 (sequential covering) 算法

- 1: 令 E 是训练记录, A 是属性—值对的集合 $\{(A_j, v_j)\}$

2: 令 Y_o 是类的有序集 $\{y_1, y_2, \dots, y_k\}$

3: 令 $R = \{\}$ 是初始规则列表

4: for 每个类 $y \in Y_o - \{y_k\}$ do

5: while 终止条件不满足 do

6: $r \leftarrow \text{Learn-One-Rule}(E, A, y)$

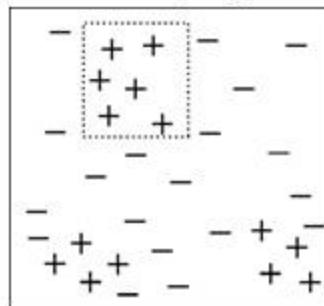
7: 从 E 中删除被 r 覆盖的训练记录

8: 追加 r 到规则列表尾部: $R \leftarrow R \cup r$

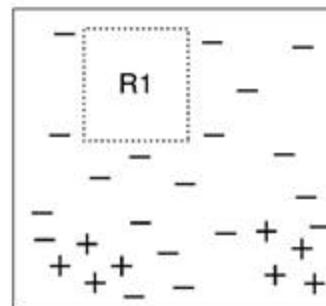
9: end while

10: end for

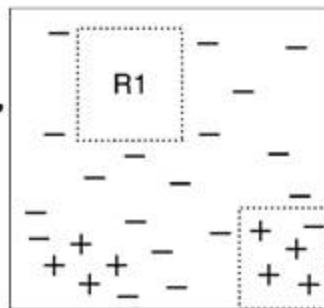
(a) Original data



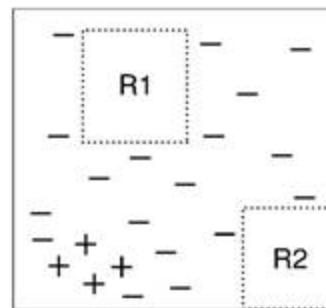
(b) Step 1



(c) Step 2



(c) Step 3



11: 把默认规则 $\{\} \rightarrow y_k$ 插入到规则列表R尾部



1.5.2 删除实例

$r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$

$r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$

$r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$

$r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$

$r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲑鱼	冷血	鳞片	否	是	否	否	否	鱼类
鲸	恒温	毛发	是	是	否	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
鸽子	恒温	羽毛	否	否	是	是	否	鸟类
猫	恒温	软毛	是	否	否	是	否	哺乳类
虹鳟	冷血	鳞片	是	是	否	否	否	鱼类
美洲鳄	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
豪猪	恒温	刚毛	是	否	否	是	是	哺乳类
鳗鲡	冷血	鳞片	否	是	否	否	否	鱼类
蝾螈	冷血	无	否	半	否	否	是	两栖类



1.5.2 删除实例

$r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$

$r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$

$r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$

$r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$

$r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲑鱼	冷血	鳞片	否	是	否	否	否	鱼类
鲸	恒温	毛发	是	是	否	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
蝙蝠	恒温	毛发	是	否	否	是	是	哺乳类
猫	恒温	软毛	是	否	否	是	否	哺乳类
虹鳟	冷血	鳞片	是	是	否	否	否	鱼类
美洲鳄	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	是	鸟类
豪猪	恒温	刚毛	是	否	否	是	否	哺乳类
鳗鲡	冷血	鳞片	否	是	否	否	是	鱼类
蝾螈	冷血	无	否	半	否	否	是	两栖类



1.5.2 删除实例

$r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$

$r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$

$r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$

$r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$

$r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲑鱼	冷血	鳞片	否	是	否	否	否	鱼类
鲸	恒温	毛发	是	是	否	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
猫	恒温	软毛	是	否	否	是	否	哺乳类
虹鱥	冷血	鳞片	是	是	否	否	否	鱼类
美洲鳄	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
豪猪	恒温	刚毛	是	否	否	是	是	哺乳类
鳗鲡	冷血	鳞片	否	是	否	否	否	鱼类
蝾螈	冷血	无	否	半	否	21	是	两栖类



1.5.2 删除实例

$r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$

$r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$

$r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$

$r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$

$r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲸	恒温	毛发	是	是	否	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
猫	恒温	软毛	是	否	否	是	否	哺乳类
虹鱈	冷血	鳞片	是	是	否	否	否	鱼类
美洲鳄	冷血	鳞片	否	半	否	是	是	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
豪猪	恒温	刚毛	是	否	否	是	是	哺乳类
蝾螈	冷血	无	否	半	否	22	是	两栖类



1.5.2 删除实例

r_1 : (胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类

r_2 : (胎生 = 否) \wedge (水生动物 = 是) \rightarrow 鱼类

r_3 : (胎生 = 是) \wedge (体温 = 恒温) \rightarrow 哺乳类

r_4 : (胎生 = 否) \wedge (飞行动物 = 否) \rightarrow 爬行类

r_5 : (水生动物 = 半) \rightarrow 两栖类

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲸	恒温	毛发	是	是	否	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
猫	恒温	软毛	是	否	否	是	否	哺乳类
虹鱥	冷血	鳞片	是	是	否	否	否	鱼类
美洲鳄	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
豪猪	恒温	刚毛	是	否	否	是	是	哺乳类
蝾螈	冷血	无	否	半	否	23	是	两栖类



1.5.2 删除实例

$r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$

$r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$

$r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$

$r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$

$r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
虹鳟	冷血	鳞片	是	是	否	否	否	鱼类
美洲鳄	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
蝾螈	冷血	无	否	半	否	24	是	两栖类



1.5.2 删除实例

r_1 : (胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类

r_2 : (胎生 = 否) \wedge (水生动物 = 是) \rightarrow 鱼类

r_3 : (胎生 = 是) \wedge (体温 = 恒温) \rightarrow 哺乳类

r_4 : (胎生 = 否) \wedge (飞行动物 = 否) \rightarrow 爬行类

r_5 : (水生动物 = 半) \rightarrow 两栖类

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
虹鳟	冷血	鳞片	是	是	否	否	否	鱼类
美洲鳄	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
蝾螈	冷血	无	否	半	否	是	是	两栖类



1.5.2 删除实例

r_1 : (胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类

r_2 : (胎生 = 否) \wedge (水生动物 = 是) \rightarrow 鱼类

r_3 : (胎生 = 是) \wedge (体温 = 恒温) \rightarrow 哺乳类

r_4 : (胎生 = 否) \wedge (飞行动物 = 否) \rightarrow 爬行类

r_5 : (水生动物 = 半) \rightarrow 两栖类

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
青蛙	冷血	无	否	半	否	是	是	两栖类
虹鳟	冷血	鳞片	是	是	否	否	否	鱼类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
蝾螈	冷血	无	否	半	否	26	是	两栖类



1.5.2 删除实例

r_1 : (胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类

r_2 : (胎生 = 否) \wedge (水生动物 = 是) \rightarrow 鱼类

r_3 : (胎生 = 是) \wedge (体温 = 恒温) \rightarrow 哺乳类

r_4 : (胎生 = 否) \wedge (飞行动物 = 否) \rightarrow 爬行类

r_5 : (水生动物 = 半) \rightarrow 两栖类

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
青蛙	冷血	无	否	半	否	是	是	两栖类
虹鳟	冷血	鳞片	是	是	否	否	否	鱼类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
蝾螈	冷血	无	否	半	否	27	是	两栖类

删除负实例

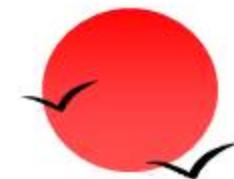


1.5.2 删除实例

- r_1 : (胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类
- r_2 : (胎生 = 否) \wedge (水生动物 = 是) \rightarrow 鱼类
- r_3 : (胎生 = 是) \wedge (体温 = 恒温) \rightarrow 哺乳类
- r_4 : (胎生 = 否) \wedge (飞行动物 = 否) \rightarrow 爬行类
- r_5 : (水生动物 = 半) \rightarrow 两栖类

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
青蛙	冷血	无	否	半	否	是	是	两栖类
虹鱥	冷血	鳞片	是	是	否	否	否	鱼类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
蝾螈	冷血	无	否	半	否	28	是	两栖类

删除负实例



1.5.2 删除实例

- r_1 : (胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类
- r_2 : (胎生 = 否) \wedge (水生动物 = 是) \rightarrow 鱼类
- r_3 : (胎生 = 是) \wedge (体温 = 恒温) \rightarrow 哺乳类
- r_4 : (胎生 = 否) \wedge (飞行动物 = 否) \rightarrow 爬行类
- r_5 : (水生动物 = 半) \rightarrow 两栖类

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
虹鳟	冷血	鳞片	是	是	否	否	否	鱼类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类

删除负实例



1.5.2 删除实例

- $r_1: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$
- $r_2: (\text{胎生} = \text{否}) \wedge (\text{水生动物} = \text{是}) \rightarrow \text{鱼类}$
- $r_3: (\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$
- $r_4: (\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{否}) \rightarrow \text{爬行类}$
- $r_5: (\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$

提问：该规则集
是穷举规则还是
非穷举规则？

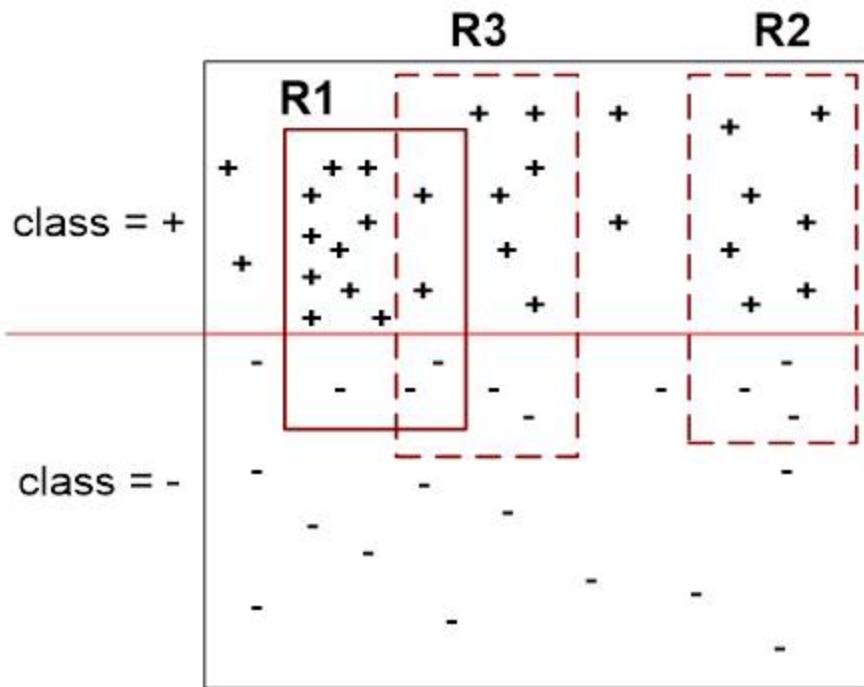
名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
虹鳟	冷血	鳞片	是	是	否	否	否	鱼类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类

删除负实例



1.5.2 删除实例

- 为什么要删除实例?
 - 否则,下一个规则将与前面的规则相同 (规则可能重复)
 - 为什么删除正实例?
 - 防止高估后面规则的准确率
 - 确保下一个规则不同
 - 为什么删除负实例?
 - 防止过拟合错误训练集
 - 防止低估后面规则的准确率
 - 比较图中的规则 R2 和 R3



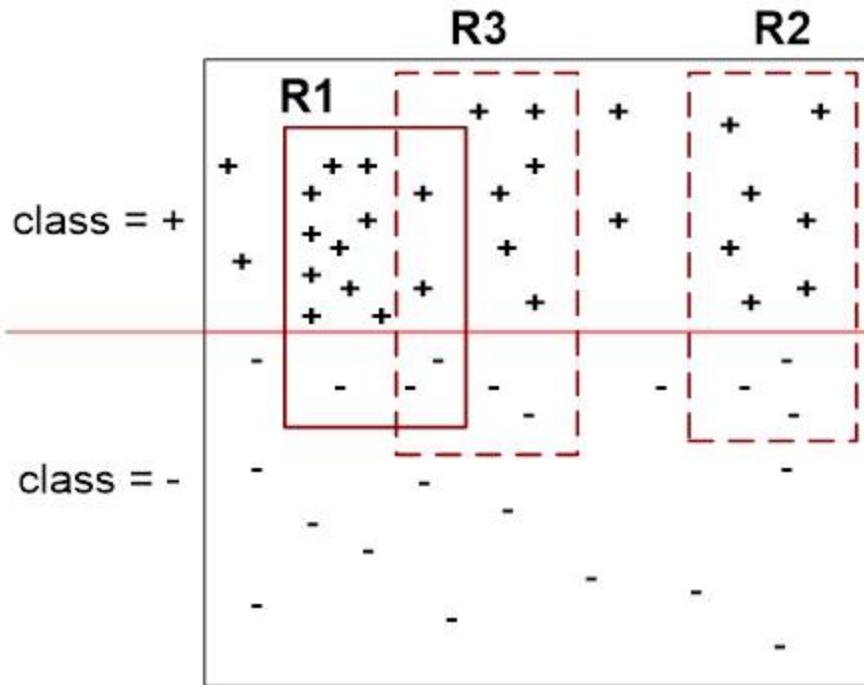


1.5.2 删除实例

- R1: $12/15=80\%$
- R2: $7/10=70\%$
- R3: $8/12=66.7\%$

- 1) 产生R1 (第一步)
- 2) 产生R2? R3?
 - R1 U R2: $19/25=76\%$
 - R1 U R3:

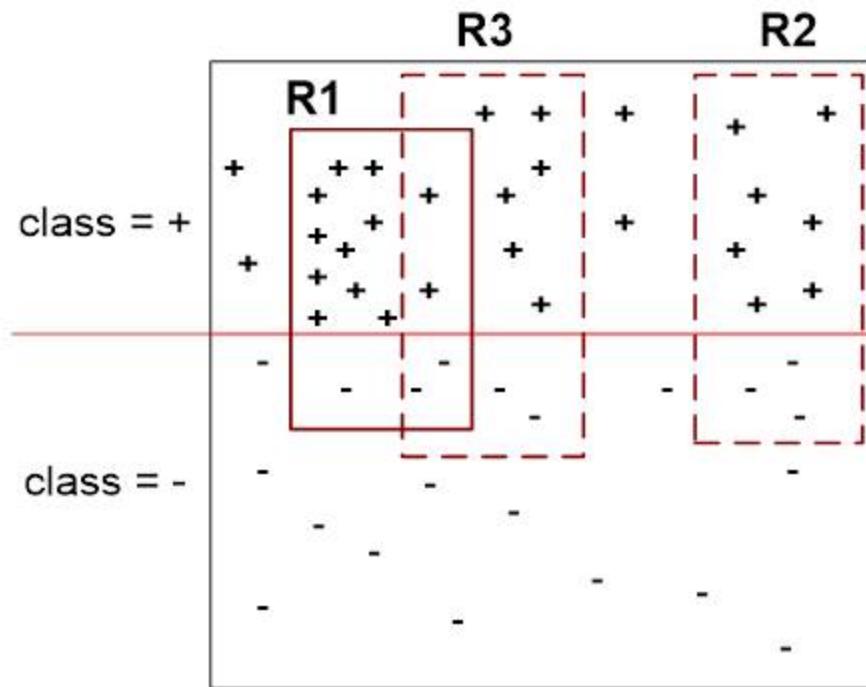
- 3) 产生R? (第二步)





- R1: $12/15=80\%$
- R2: $7/10=70\%$
- R3: $8/12=66.7\%$

- 1) 产生R1 (第一步)
- 2) 产生R2? R3?
 - R1 U R2: $19/25=76\%$
 - R1 U R3: [填空1]



正常使用填空题需3.0以上版本雨课堂

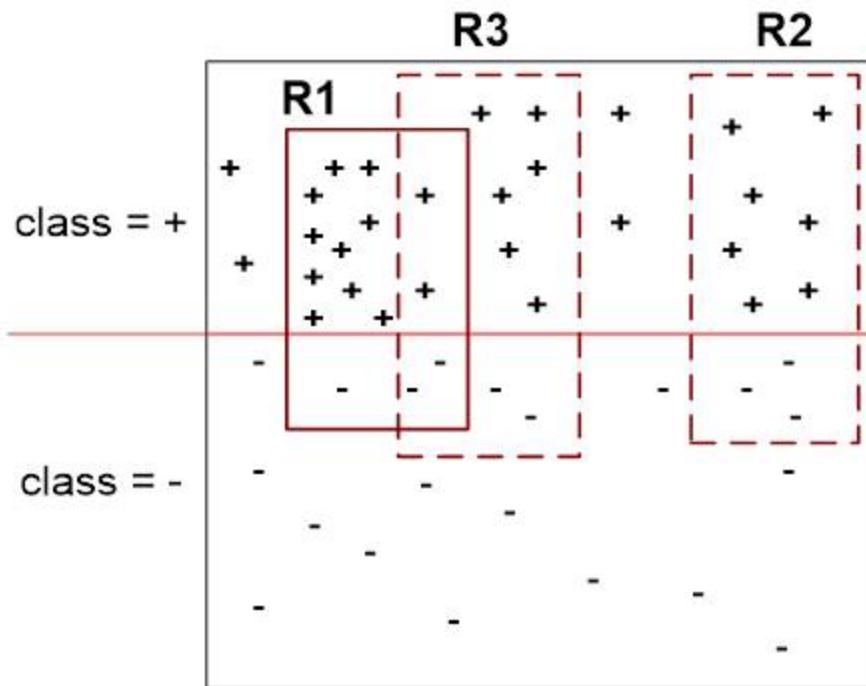


1.5.2 删除实例

- R1: $12/15=80\%$
- R2: $7/10=70\%$
- R3: $8/12=66.7\%$

- 1) 产生R1 (第一步)
- 2) 产生R2? R3?
 - R1 U R2: $19/25=76\%$
 - R1 U R3: $18/23=78.3\%$

- 3) 产生R3 (第二步, $6/8=75\%$)





1.5.3 Learn-One-Rule

- 规则增长
- 规则评估
- 停止准则
- 规则剪枝
- 顺序覆盖 (sequential covering) 算法
 - 1: 令 E 是训练记录, A 是属性—值对的集合 $\{(A_j, v_j)\}$
 - 2: 令 Y_o 是类的有序集 $\{y_1, y_2, \dots, y_k\}$
 - 3: 令 $R = \{\}$ 是初始规则列表
 - 4: **for** 每个类 $y \in Y_o - \{y_k\}$ **do**
 - 5: **while** 终止条件不满足 **do**
 - 6: $r \leftarrow \text{Learn-One-Rule}(E, A, y)$
 - 7: 从 E 中删除被 r 覆盖的训练记录
 - 8: 追加 r 到规则列表尾部: $R \leftarrow R \cup r$
 - 9: **end while**
 - 10: **end for**
 - 11: 把默认规则 $\{\} \rightarrow y_k$ 插入到规则列表 R 尾部



1.5.3 规则增长

- 两种策略

- 一般到特殊（通常采用的策略）

- 从初始规则 $r: \{\} \rightarrow y$ 开始
 - 反复加入合取项，得到更特殊的规则，直到不能再加入

- 特殊到一般（适用于小样本情况）

- 随机地选择一个正例作为初始规则
 - 反复删除合取项，得到更一般的规则，直到不能再删除

(胎生 = 否) \rightarrow 鸟类



(胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类

(胎生 = 否) \wedge (飞行动物 = 是) \wedge
(体温 = 恒温) \rightarrow 鸟类



(胎生 = 否) \wedge (飞行动物 = 是) \rightarrow 鸟类



1.5.3 规则增长

- 两种策略

- 一般到特殊（通常采用的策略）

- 从初始规则 $r: \{\} \rightarrow y$ 开始

$(\text{胎生} = \text{否}) \rightarrow \text{鸟类}$



$(\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$

- 反复加入合取项，得到更特殊的规则，直到不能再加入

- 特殊到一般（适用于小样本情况）

- 随机地选择一个正例作为初始规则

$(\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{鸟类}$



- 反复删除合取项，得到更一般的规则，直到不能再删除

$(\text{胎生} = \text{否}) \wedge (\text{飞行动物} = \text{是}) \rightarrow \text{鸟类}$

- 问题

- 加入/删除合取项有多种选择，如何选择？

- 何时停止加入/删除合取项？（准确率100%）

→ 需要评估标准



1.5.3 规则增长（一般到特殊）

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	https://blog.csdn.net/guozhihang	稍凹	硬滑	否

- 规则r后件为（好瓜=是），前件从空开始，先依次添加一个(属性，值)，计算覆盖的部分记录编号及它的分类准确率。



1.5.3 规则增长（一般到特殊）

- 可以看到，(纹理=清晰)的正确率最高，因此首先在规则r的前件中添加(纹理=清晰)，接着在(纹理=清晰)覆盖的记录中，继续规则r前件中(属性，值)添加。

属性-值	覆盖的记录编号	准确率
色泽=青绿	1,4,6,10,13,17	1/2
色泽=乌黑	2,3,7,8,9,15	2/3
根蒂=蜷缩	1,2,3,4,5,12,16,17	5/8
敲声=浊响	1,3,5,6,7,8,12,13,15,16	3/5
纹理=清晰	1,2,3,4,5,6,8,10,15	7/9
脐部=凹陷	1,2,3,4,5,13,14,17	5/8



1.5.3 规则增长（一般到特殊）

- 可以看到，在(纹理=清晰)覆盖的记录中，**(根蒂=蜷缩)**与**(脐部=凹陷)**覆盖记录的准确率都达到了100%，可以任选一个(属性，值)，这里可以选择**(根蒂=蜷缩)**，此时也达到了(属性，值)添加的终止条件，故在类**(好瓜=是)**中，函数生成了第一条规则。

属性-值	覆盖的记录编号	准确率
色泽=青绿	1,4,6,10	3/4
色泽=乌黑	2,3,8,15	3/4
根蒂=蜷缩	1,2,3,4,5	5/5
敲声=浊响	1,3,5,6,8,15	5/6
脐部=凹陷	1,2,3,4,5	5/5

- $\{(纹理=清晰) \wedge (根蒂=蜷缩)\} \rightarrow (好瓜=是)$



1.5.3 规则增长（一般到特殊）

- 从一般到特殊的规则生成策略中，每次只考虑一个最优的(属性，值)
- 这显得过于贪心，容易陷入局部最优麻烦
- 为了缓解该问题，可以采用一种“**集束搜索(Beam search)**”的方式
 - 具体做法为：每次选择添加的(属性，值)时，可以保留前k个最优的(属性，值)，而不是只选择最优的那个，然后对这k个最优的(属性，值)继续进行下一轮的(属性，值)添加。



1.5.4 Learn-One-Rule

- 规则增长
- 规则评估
- 停止准则
- 规则剪枝
- 顺序覆盖 (sequential covering) 算法
 - 1: 令 E 是训练记录, A 是属性—值对的集合 $\{(A_j, v_j)\}$
 - 2: 令 Y_o 是类的有序集 $\{y_1, y_2, \dots, y_k\}$
 - 3: 令 $R = \{\}$ 是初始规则列表
 - 4: **for** 每个类 $y \in Y_o - \{y_k\}$ **do**
 - 5: **while** 终止条件不满足 **do**
 - 6: $r \leftarrow \text{Learn-One-Rule}(E, A, y)$
 - 7: 从 E 中删除被 r 覆盖的训练记录
 - 8: 追加 r 到规则列表尾部: $R \leftarrow R \cup r$
 - 9: **end while**
 - 10: **end for**
 - 11: 把默认规则 $\{\} \rightarrow y_k$ 插入到规则列表 R 尾部



1.5.4 规则评估

- 常用的度量
 - 准确率
 - 似然比
 - Laplace
 - FOIL信息增益

例如考虑一个训练集，它包含60个正例和100个反例，现有两个候选规则：

- r1: 覆盖50个正例和5个反例
- r2: 覆盖2个正例和0个反例



1.5.4 规则评估(续)

- 准确率

- $\blacksquare \text{ Accuracy} = \frac{n_c}{n}$ $\text{Acc}(r_1): 90.9\%$

- $\blacksquare n$: 被规则覆盖的实例数
- $\blacksquare n_c$: 被规则正确分类的实例数

$\text{Acc}(r_2): ?$

例如考虑一个训练集，它包含60个正例和100个反例，现有两个候选规则：

r_1 : 覆盖50个正例和5个反例

r_2 : 覆盖2个正例和0个反例



■ 准确率

$$\text{Accuracy} = \frac{n_c}{n}$$

Acc(r₁): 90.9%

- n: 被规则覆盖的实例数
- n_c: 被规则正确分类的实例数

Acc(r₂)等于多少?

- A 90%
- B 100%

例如考虑一个训练集，它包含60个正例和100个反例，现有两个候选规则：

r1: 覆盖50个正例和5个反例

r2: 覆盖2个正例和0个反例

提交



1.5.4 规则评估(续)

- 准确率

- $\text{Accuracy} = \frac{n_c}{n}$ $\text{Acc}(r_1): 90.9\%$

$$\text{Acc}(r_2): 100\%$$

- n : 被规则覆盖的实例数
- n_c : 被规则正确分类的实例数
- 问题: 准确率高的规则可能覆盖率太低

例如考虑一个训练集, 它包含60个正例和100个反例, 现有两个候选规则:

r_1 : 覆盖50个正例和5个反例

r_2 : 覆盖2个正例和0个反例



1.5.4 规则评估(续)

- 似然比 LRS (越高越好)

- k 是类的个数
- f_i 是被规则覆盖的类 i 的样本的观测频度
- e_i 是规则作随机猜测的期望频度

$$R = 2 \sum_{i=1}^k f_i \log(f_i/e_i)$$

例如考虑一个训练集，它包含 60 个正例和 100 个反例，现有两个候选规则：

r1：覆盖 50 个正例和 5 个反例

r2：覆盖 2 个正例和 0 个反例

简单理解就是当前规则分类效果比随机效果越高，说明规则越好

$$LRS(r_1) = 2 \times \left[50 \times \log_2 \frac{50}{55 \times 60 / 160} + 5 \times \log_2 \frac{5}{55 \times 100 / 160} \right] = 99.99$$



■ 似然比LRS（越高越好）

- k 是类的个数
- f_i 是被规则覆盖的类*i*的样本的观测频度
- e_i 是规则作随机猜测的期望频度

$$R = 2 \sum_{i=1}^k f_i \log(f_i/e_i)$$

例如考虑一个训练集，它包含60个正例和100个反例，现有两个候选规则：

r1: 覆盖50个正例和5个反例

r2: 覆盖2个正例和0个反例

简单理解就是当前规则分类效果比随机效果越高，说明规则越好

$$LRS(r_1) = 2 \times \left[50 \times \log_2 \frac{50}{55 \times 60 / 160} + 5 \times \log_2 \frac{5}{55 \times 100 / 160} \right] = 99.99$$

规则r2的似然比LRS= [填空1]

正常使用填空题需3.0以上版本雨课堂



1.5.4 规则评估(续)

- 似然比LRS (越高越好)

- k 是类的个数
- f_i 是被规则覆盖的类*i*的样本的观测频度
- e_i 是规则作随机猜测的期望频度

$$R = 2 \sum_{i=1}^k f_i \log(f_i/e_i)$$

例如考虑一个训练集，它包含60个正例和100个反例，现有两个候选规则：

r1: 覆盖50个正例和5个反例

r2: 覆盖2个正例和0个反例

简单理解就是当前规则分类效果比随机效果越高，说明规则越好

$$LRS(r_1) = 2 \times \left[50 \times \log_2 \frac{50}{55 \times 60 / 160} + 5 \times \log_2 \frac{5}{55 \times 100 / 160} \right] = 99.99$$

$$LRS(r_2) = 2 \times \left[2 \times \log_2 \frac{2}{2 \times 60 / 160} + 0 \times \log_2 \frac{0}{2 \times 100 / 160} \right] = 5.66$$



1.5.4 规则评估(续)

- Laplace估计

- k 是类的个数
- n_+ 是被规则覆盖的正例数
- n 是被规则覆盖的样例数
- p_+ 是正例的先验概率

$$Laplace = \frac{n_+ + 1}{n + k}$$

例如考虑一个训练集，它包含60个正例和100个反例，现有两个候选规则：

r1: 覆盖50个正例和5个反例

r2: 覆盖2个正例和0个反例

$$Laplace(r_1) = \frac{50+1}{55+2} = 0.8947$$

- 准确率

- Accuracy = $\frac{n_c}{n}$  简单理解：Laplace估计即为准确率的平滑



Laplace估计

- k 是类的个数
- n_+ 是被规则覆盖的正例数
- n 是被规则覆盖的样例数
- p_+ 是正例的先验概率

$$Laplace = \frac{n_+ + 1}{n + k}$$

例如考虑一个训练集，它包含60个正例和100个反例，现有两个候选规则：

r1: 覆盖50个正例和5个反例

r2: 覆盖2个正例和0个反例

$$Laplace(r_1) = \frac{50+1}{55+2} = 0.8947$$

规则r2的Laplace估计= [填空1]

准确率

- Accuracy = $\frac{n_c}{n}$

简单理解：Laplace估计即为准确率的平滑



1.5.4 规则评估(续)

- Laplace估计

- k 是类的个数
- n_+ 是被规则覆盖的正例数
- n 是被规则覆盖的样例数
- p_+ 是正例的先验概率

$$Laplace = \frac{n_+ + 1}{n + k}$$

例如考虑一个训练集，它包含60个正例和100个反例，现有两个候选规则：

r1: 覆盖50个正例和5个反例

r2: 覆盖2个正例和0个反例

$$Laplace(r_1) = \frac{50+1}{55+2} = 0.8947$$

$$Laplace(r_2) = \frac{2+1}{2+2} = 0.75$$

- 准确率

- Accuracy = $\frac{n_c}{n}$  简单理解：Laplace估计即为准确率的平滑



1.5.4 规则评估(续)

■ FOIL信息增益 类似决策树的信息增益

- 假设规则 $r:A \rightarrow$ 覆盖 n_{0+} 个正例和 n_{0-} 个反例，增加新的合取项B后，扩展的规则 $r:B \rightarrow$ 覆盖 n_{1+} 个正例和 n_{1-} 个反例，此时扩展规则后FOIL信息增益定义为： 

$$FOIL(r) = n_{1+} \times \left[\log_2 \frac{n_{1+}}{n_{1+} + n_{1-}} - \log_2 \frac{n_{0+}}{n_{0+} + n_{0-}} \right]$$

- 该度量倾向于选择那些高覆盖率和高准确率的规则

例如考虑一个训练集，它包含60个正例和100个反例，现有两个候选规则：

- r_1 : 覆盖50个正例和5个反例
- r_2 : 覆盖2个正例和0个反例

$$FOIL(r_1) = 50 \times \left[\log_2 \frac{50}{50+5} - \log_2 \frac{60}{60+100} \right] = 63.87$$



FOIL信息增益 类似决策树的信息增益

- 假设规则 $r: A \rightarrow$ 覆盖 n_{0+} 个正例和 n_{0-} 个反例，增加新的合取项 B 后，扩展的规则 $r: B \rightarrow$ 覆盖 n_{1+} 个正例和 n_{1-} 个反例，此时扩展规则后 FOIL 信息增益定义为：

使用规则1 **使用规则2**

$$FOIL(r) = n_{1+} \times \left[\log_2 \frac{n_{1+}}{n_{1+} + n_{1-}} - \log_2 \frac{n_{0+}}{n_{0+} + n_{0-}} \right]$$

- 该度量倾向于选择那些高覆盖率和高准确率的规则

例如考虑一个训练集，它包含 60 个正例和 100 个反例，现有两个候选规则：

- r_1 : 覆盖 50 个正例和 5 个反例
- r_2 : 覆盖 2 个正例和 0 个反例

$$FOIL(r_1) = 50 \times \left[\log_2 \frac{50}{50+5} - \log_2 \frac{60}{60+100} \right] = 63.87$$

正常使用填空题需 3.0 以上版本雨课堂

规则 r_2 的 FOIL 信息增益 = [填空1]

作答



1.5.4 规则评估(续)

FOIL信息增益 → 类似决策树的信息增益

- 假设规则 $r: A \rightarrow$ 覆盖 n_{0+} 个正例和 n_{0-} 个反例，增加新的合取项 B 后，扩展的规则 $r: B \rightarrow$ 覆盖 n_{1+} 个正例和 n_{1-} 个反例，此时扩展规则后 FOIL 信息增益定义为：

使用规则1 **使用规则2**

$$FOIL(r) = n_{1+} \times \left[\log_2 \frac{n_{1+}}{n_{1+} + n_{1-}} - \log_2 \frac{n_{0+}}{n_{0+} + n_{0-}} \right]$$

- 该度量倾向于选择那些高覆盖率和高准确率的规则

例如考虑一个训练集，它包含 60 个正例和 100 个反例，现有两个候选规则：

- r_1 : 覆盖 50 个正例和 5 个反例
- r_2 : 覆盖 2 个正例和 0 个反例

$$FOIL(r_1) = 50 \times \left[\log_2 \frac{50}{50+5} - \log_2 \frac{60}{60+100} \right] = 63.87$$

$$FOIL(r_2) = 2 \times \left[\log_2 \frac{2}{2} - \log_2 \frac{60}{60+100} \right] = 2.83$$



1.5.5 Learn-One-Rule

- 规则增长
- 规则评估
- 停止准则
- 规则剪枝

- 顺序覆盖 (sequential covering) 算法

- 1: 令 E 是训练记录, A 是属性—值对的集合 $\{(A_j, v_j)\}$
- 2: 令 Y_o 是类的有序集 $\{y_1, y_2, \dots, y_k\}$
- 3: 令 $R = \{\}$ 是初始规则列表
- 4: **for** 每个类 $y \in Y_o - \{y_k\}$ **do**
- 5: **while** 终止条件不满足 **do**
- 6: $r \leftarrow \text{Learn-One-Rule}(E, A, y)$
- 7: 从 E 中删除被 r 覆盖的训练记录
- 8: 追加 r 到规则列表尾部: $R \leftarrow R \cup r$
- 9: **end while**
- 10: **end for**
- 11: 把默认规则 $\{\} \rightarrow y_k$ 插入到规则列表 R 尾部



1.5.5 停止条件与规则剪枝

- 停止条件
 - 计算增益
 - 如果增益不显著, 则丢弃新规则
- 规则剪枝
 - 类似于决策树后剪枝
 - 降低错误剪枝:
 - 删除规则中的合取项
 - 比较剪枝前后的错误率
 - 如果降低了错误率, 则剪掉该合取项



1.6如何建立基于规则的分类器

- 直接方法:
 - 直接由数据提取规则
 - 例如: **RIPPER, CN2, Holte's 1R**

- 间接方法:
 - 由其他分类模型提取规则(例如, 从决策树等).
 - 例如: **C4.5rules**



1.6 直接方法: RIPPER

- 对于**2类**问题,选定一个类为正类, 另一个为负类
 - 从正类学习规则
 - 负类时缺省类
- 多类问题
 - 按类的大小(属于特定类的实例所占的比例)对诸类排序
 - 从最小的类开始学习规则, 其余类都看做负类
 - 对次小类学习规则, 如此下去



1.6 直接方法: RIPPER(续)

- 规则增长:
 - 由空规则开始
 - 只要能够提高FOIL信息增益就增加一个合取项
 - 当规则不再覆盖负实例时就停止（该规则都是正类）
 - 剪枝
 - 剪枝度量:
$$v = (p-n)/(p+n)$$
 - p : 验证集中被规则覆盖的正实例数
 - n : 验证集中被规则覆盖的负实例数
 - 剪枝方法:
 - 如果剪掉合取项可以提高 v 就剪



1.6 直接方法: RIPPER(续)

- 建立规则集:
 - 使用顺序覆盖算法
 - 找出覆盖当前正实例的最佳规则
 - 删除被规则覆盖的所有正实例和负实例
 - 当一个规则加入规则集时, 计算新的描述长度
 - 当新的描述长度比已经得到的描述长度多 d 位时, 就停止增加新规则
 - 当在确认集上的错误率超过50%时, 停止增加新规则



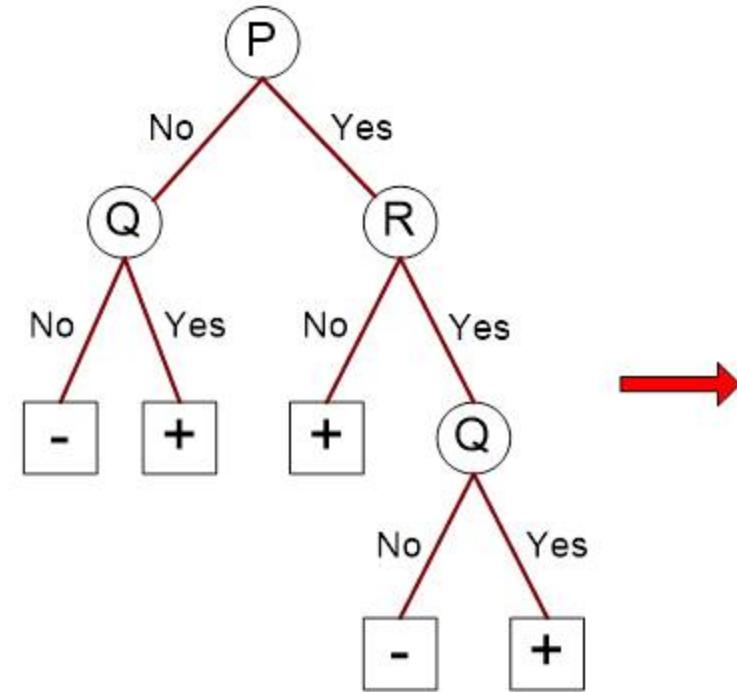
1.6 直接方法: RIPPER(续)

- 优化规则集:
 - 对规则集 R 中的每个规则 r
 - 考虑2个替换的规则:
 - 替换规则(r^*): 重新新增规则
 - 编辑的规则(r'): 把一个新的合取项增加到规则 r
 - 比较替换前后的规则集
 - 选择最小化**描述长度(MDL)**的规则集
 - 对剩下的正实例, 重复规则产生和优化



1.7 规则提取的间接方法

- 决策树从根结点到叶结点的每一条路径都可以表示为一个分类规则
 - 路径中的测试条件构成规则前件的合取项，叶结点的类标号赋给规则后件



Rule Set

```
r1: (P=No,Q=No) ==> -
r2: (P=No,Q=Yes) ==> +
r3: (P=Yes,R=No) ==> +
r4: (P=Yes,R=Yes,Q=No) ==> -
r5: (P=Yes,R=Yes,Q=Yes) ==> +
```



1.8 规则分类的特点

- 优点：

- 表达能力与决策树一样高
- 容易解释
- 容易产生
- 能够快速对新实例分类
- 性能可与决策树相媲美

- 缺点：

- 规则库难以维护
- 规则匹配计算量大
- 模型缺乏泛化能力



2 急切学习与惰性学习



急切学习 vs 惰性学习

- 急切学习 (Eager Learner)
 - 两步过程: (1) 归纳 (2) 演绎
- 惰性学习 (Lazy Learner)
 - 把训练数据建模过程推迟到需要对样本分类时
 - 例子
 - Rote-learner (死记硬背)
 - 记住所有的训练数据, 仅当记录的属性值与一个训练记录完全匹配才对它分类
 - 最近邻 (Nearest neighbor)
 - 使用“最近”的 k 个点(最近邻)进行分类



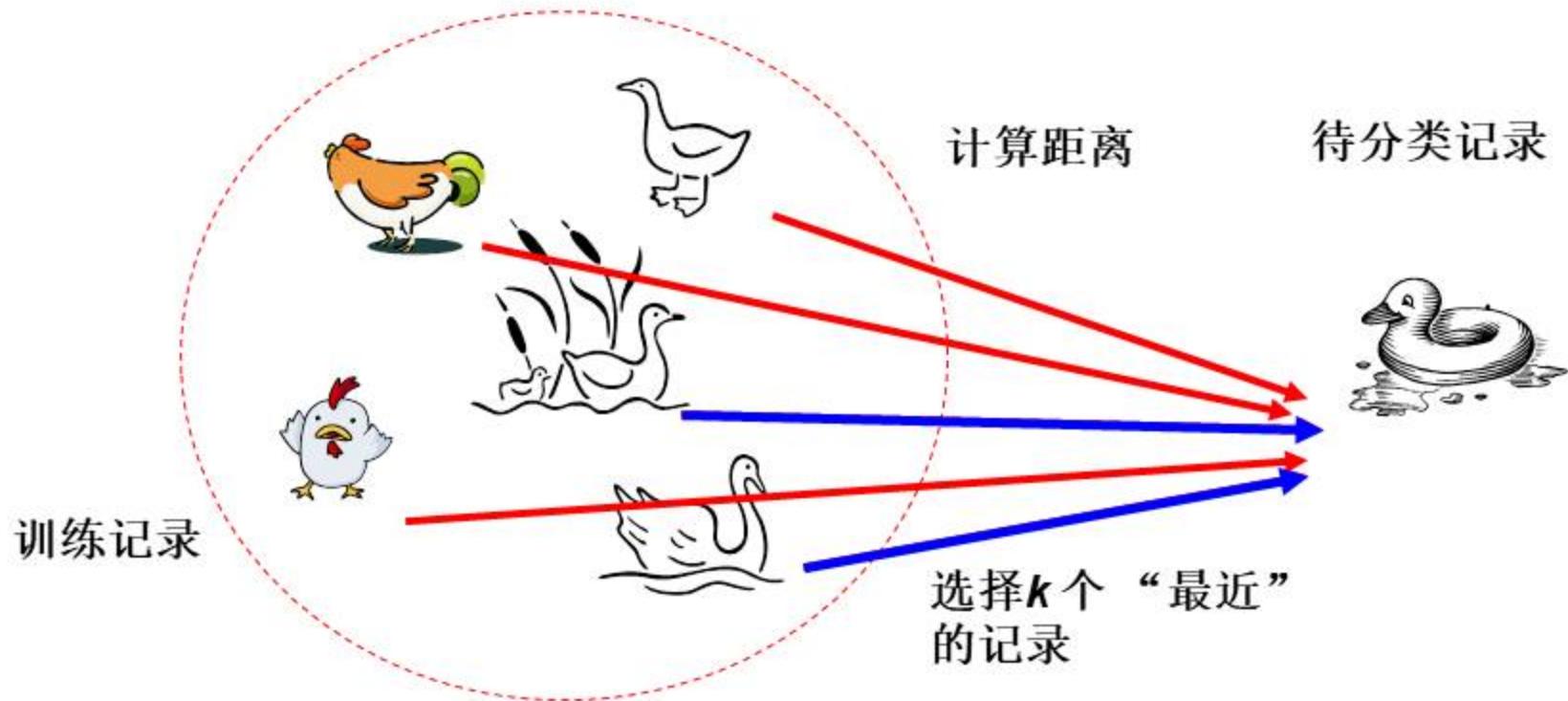
3 最近邻分类器knn



3最近邻分类器

- 基本思想:

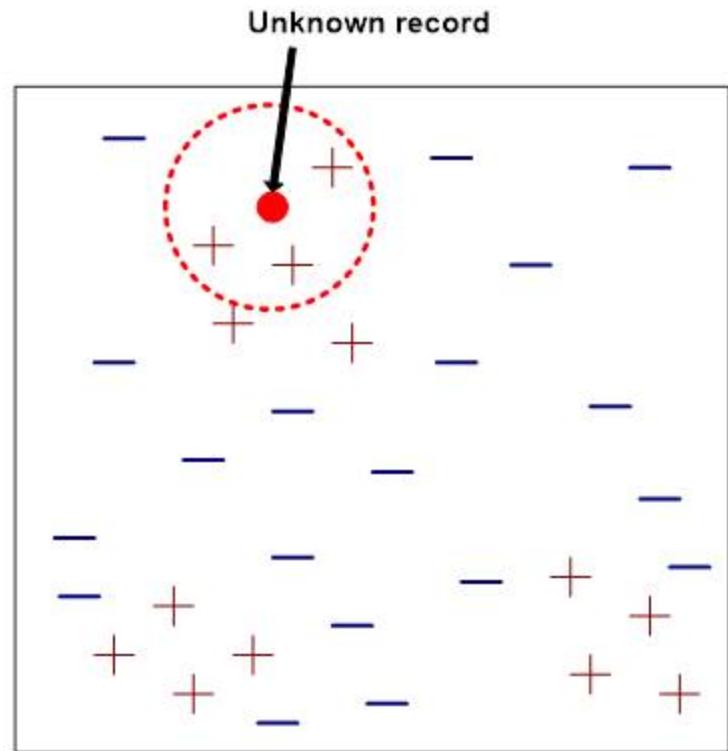
- If it walks like a duck, quacks like a duck, then it's probably a duck





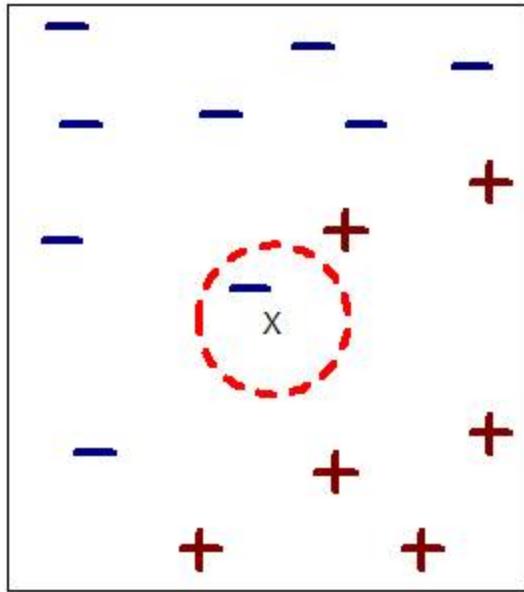
3最近邻分类器

- 要求
 - 存放训练记录
 - 计算记录间距离的度量
 - k 值, 最近邻数
- 对未知记录分类:
 - 计算域各训练记录的距离
 - 找出 k 个最近邻
 - 使用最近邻的类标号决定未知记录的类标号(例如, 多数表决)

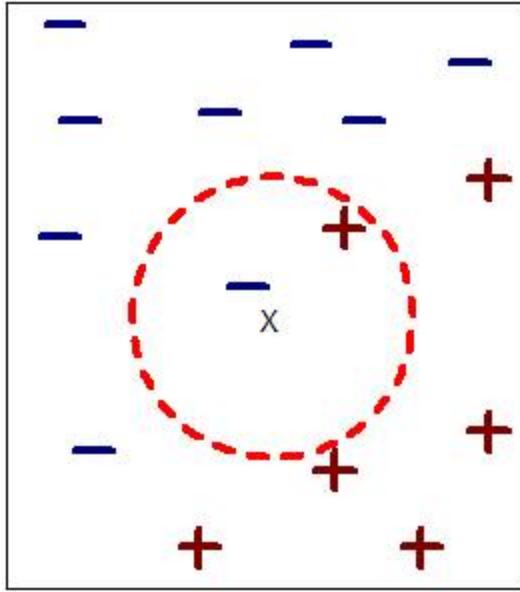




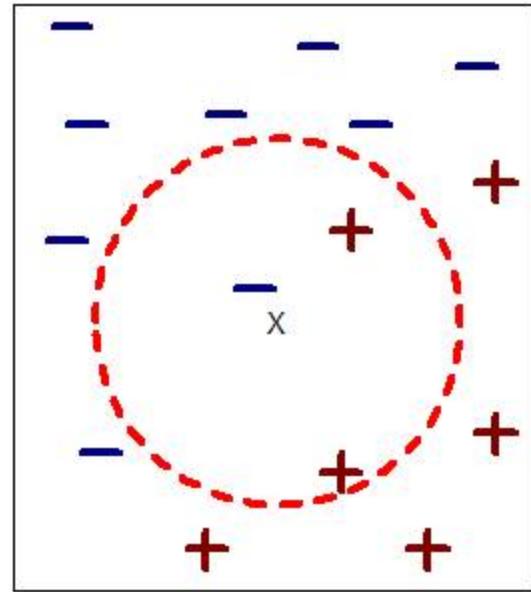
3.1最近邻定义



(a) 1-nearest neighbor



(b) 2-nearest neighbor



(c) 3-nearest neighbor

记录 x 的 k -最近邻是与 x 之间距离最小的 k 个训练数据点



3.2 k -最近邻分类算法

- k -最近邻分类算法

- 1: 令 k 是最近邻数目, D 是训练样例的集合
- 2: for 每个测试样例 $z = (x', y')$ do
- 3: 计算 z 和每个样例 $(x, y) \in D$ 之间的距离 $d(x', x)$
- 4: 选择离 z 最近的 k 个训练样例的集合 $D_z \subseteq D$
- 5: $y' = \operatorname{argmax}_v \sum_{(x_i, y_i) \in D_z} I(v = y_i)$
- 6: end for

- 距离加权表决

$$y' = \operatorname{argmax}_v \sum_{(x_i, y_i) \in D_z} w_i \times I(v = y_i)$$



3.2 k-最近邻分类算法

- **k -最近邻分类算法**

- 1: 令 k 是最近邻数目, D 是训练样例的集合

- 2: **for** 每个测试样例 $z = (\mathbf{x}', y')$ **do**

- 3: 计算 z 和每个样例 $(\mathbf{x}, y) \in D$ 之间的距离 $d(\mathbf{x}', \mathbf{x})$

计算开销大

- 4: 选择离 z 最近的 k 个训练样例的集合 $D_z \subseteq D$

- 5: $y' = \operatorname{argmax}_v \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)$

- 6: **end for**

- 距离加权表决

$$y' = \operatorname{argmax}_v \sum_{(\mathbf{x}_i, y_i) \in D_z} w_i \times I(v = y_i)$$



3.2 k-最近邻分类算法

- **k -最近邻分类算法**

- 1: 令 k 是最近邻数目, D 是训练样例的集合

- 2: **for** 每个测试样例 $z = (\mathbf{x}', y')$ **do**

- 3: 计算 z 和每个样例 $(\mathbf{x}, y) \in D$ 之间的距离 $d(\mathbf{x}', \mathbf{x})$

计算开销大

- 4: 选择离 z 最近的 k 个训练样例的集合 $D_z \subseteq D$

- 5: $y' = \operatorname{argmax}_v \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)$

- 两种特殊的**数据**

- 6: **end for**

- 距离加权表决

$$y' = \operatorname{argmax}_v \sum_{(\mathbf{x}_i, y_i) \in D_z} w_i \times I(v = y_i)$$

结构 提前对训练集进行优化存储

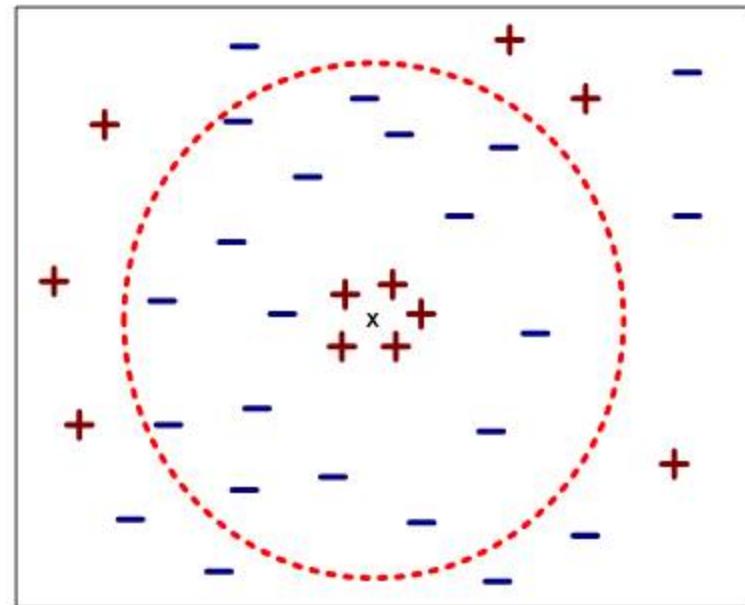
□ Kd-Tree

□ Kd-Ball



3.3 k -最近邻注意的问题

- k 值的选择：
 - 如果 k 太小，则对噪声点敏感
 - 如果 k 太大，邻域可能包含很多其他类的点
- 定标问题（规范化）
 - 属性可能需要规范化，防止距离度量被具有很大值域的属性所左右





3.4 k -NN的特点

- k -NN的特点

- 是一种基于实例的学习
 - 需要一个邻近性度量来确定实例间的相似性或距离
- 不需要建立模型，但分类一个测试样例开销很大
 - 需要计算域所有训练实例之间的距离
- 基于局部信息进行预测，对噪声非常敏感
- 最近邻分类器可以生成任意形状的决策边界
 - 决策树和基于规则的分类器通常是直线决策边界
- 需要适当的邻近性度量和数据预处理
 - 防止邻近性度量被某个属性左右



Knn分类属于下面哪一类

- A 急切学习
- B 惰性学习



knn分类编程实践

- <https://scikit-learn.org/stable/modules/neighbors.html>
- 同学们可以尝试利用python读入本地iris数据集，来完成knn分类，分析其分类效果

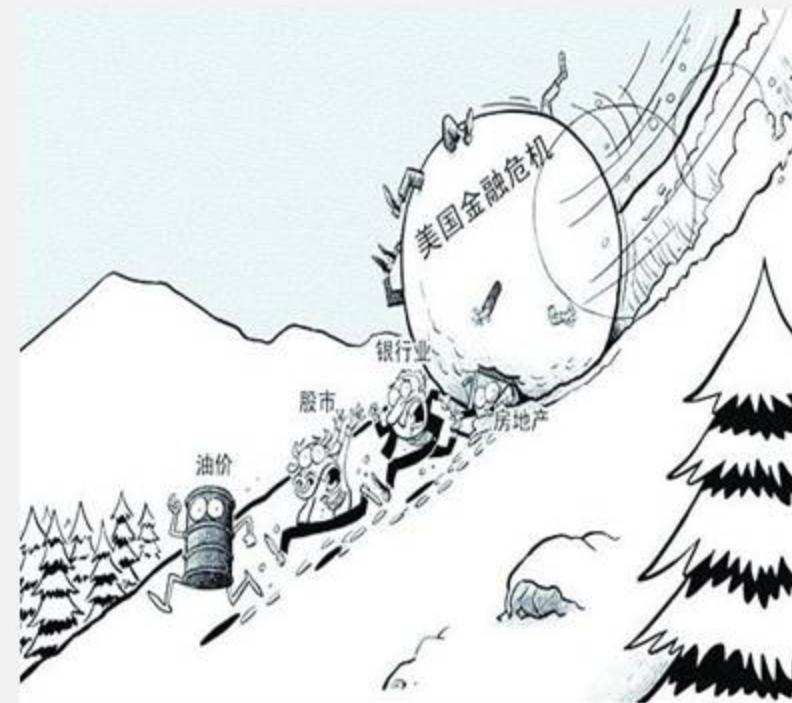


第10次课后作业

- 第十次课后作业-在educoder平台上完成作业
- <https://www.educoder.net/shixuns/kf8zsnhe/challenges>
- <https://www.educoder.net/shixuns/f8nxhame/challenges>
- <https://www.educoder.net/shixuns/azc7klio/challenges>
- <https://www.educoder.net/shixuns/u9ntcv68/challenges>
- <https://www.educoder.net/shixuns/u9ntcv68/challenges>
- <https://www.educoder.net/shixuns/ru69ogj3/challenges>

提交作业截至时间： **2020年3月24日**

美国次贷危机 (subprime crisis) 是因次级抵押贷款机构破产、投资基金被迫关闭、股市剧烈震荡引起的全球性金融风暴。致使世界主要金融市场出现流动性不足危机。



赛题简介

The screenshot shows a competition page on the DataCastle platform. At the top left is the DataCastle logo and name. The top right features a user icon, language selection (English), and navigation links for Home, All Competitions, DC Ranking, and Help. Below the header, a breadcrumb navigation shows '首页 > 竞赛 > 竞赛详情'. The main content area has a banner for '融360' (Rong360.com) on the left. The competition title is '用户贷款风险预测' (User Loan Risk Prediction), categorized as '算法竞赛' (Algorithm Competition). A large circular badge on the right indicates a total prize of '¥ 60000'. Below the badge, statistics are listed: 参赛队伍: 3243, 参赛人数: 5135, and 作品提交数: 10059. At the bottom, four buttons are visible: '下载数据' (Download Data), '提交结果' (Submit Results), '我的排名' (My Ranking), and '邀请好友' (Invite Friends).

首页 全部竞赛 DC榜 帮助 English

首页 > 竞赛 > 竞赛详情

用户贷款风险预测 算法竞赛

融360 Rong360.com

Data Castle Competition

¥ 60000 1st Prize

参赛队伍: 3243
参赛人数: 5135
作品提交数: 10059

下载数据 提交结果 我的排名 邀请好友

数据来源：融360与平台上的金融机构合作，提供了近7万贷款用户的数据信息
数据内容：用户基本身份信息、消费行为、银行还款等数据信息

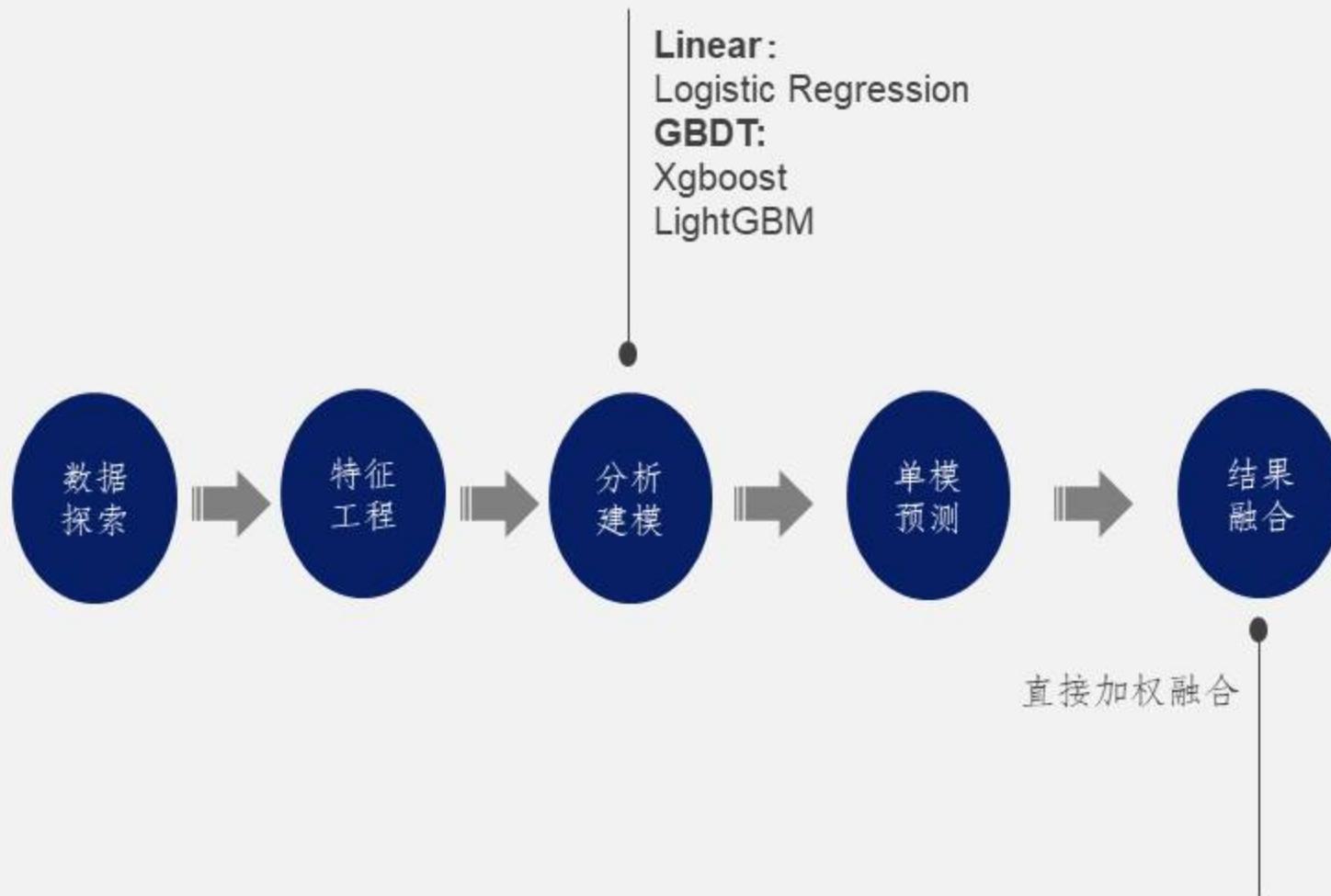


采用 Kolmogorov-Smirnov (KS) 统计量值来衡量预测结果。KS 是风险评分领域常用的评价指标，KS 越高表明模型对正负样本的区分能力越强。其计算方法为：

假设 $f(s|P)$ 为正样本预测值的累计分布函数 (cdf)， $f(s|N)$ 为负样本在预测值上的累计分布函数，则有

$$KS = \max_s \{|f(s|P) - f(s|N)|\}$$

KS 统计量基于累计分布函数，用以检验两个经验分布是否不同或一个经验分布与另一个理想分布是否不同。



Part**2**

数据探索

赛题数据集：

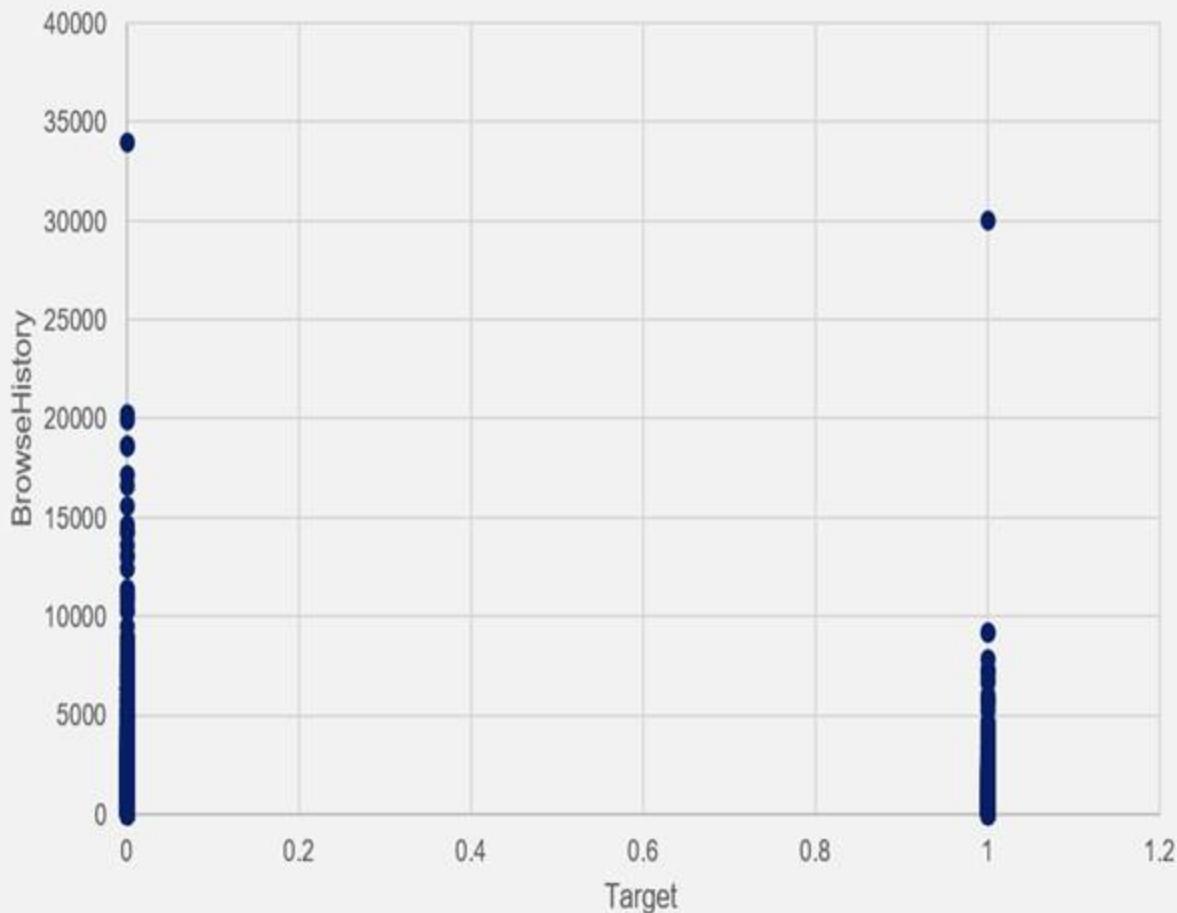
名称	修改日期	类型	大小
bank_detail_test.txt	2016/11/8 13:40	文本文档	11,651 KB
bank_detail_train.txt	2016/11/8 13:40	文本文档	184,883 KB
bill_detail_test.txt	2016/11/8 13:41	文本文档	48,259 KB
bill_detail_train.txt	2016/11/8 13:41	文本文档	269,973 KB
browse_history_test.txt	2016/11/8 13:47	文本文档	126,669 KB
browse_history_train.txt	2016/11/8 13:46	文本文档	525,827 KB
loan_time_test.txt	2016/11/8 13:47	文本文档	245 KB
loan_time_train.txt	2016/11/8 13:47	文本文档	967 KB
overdue_train.txt	2016/11/8 13:47	文本文档	478 KB
user_info_test.txt	2016/11/8 13:47	文本文档	231 KB
user_info_train.txt	2016/11/8 13:47	文本文档	913 KB
usersID_test.txt	2016/11/8 13:47	文本文档	96 KB

并非每一位用户都有非常完整的记录，如有些用户并没有信用卡账单记录，有些用户没有银行流水记录；脱敏处理：(a) 隐藏了用户id信息(b) 用户属性信息数字化(c) 时间戳和金额值函数变换。

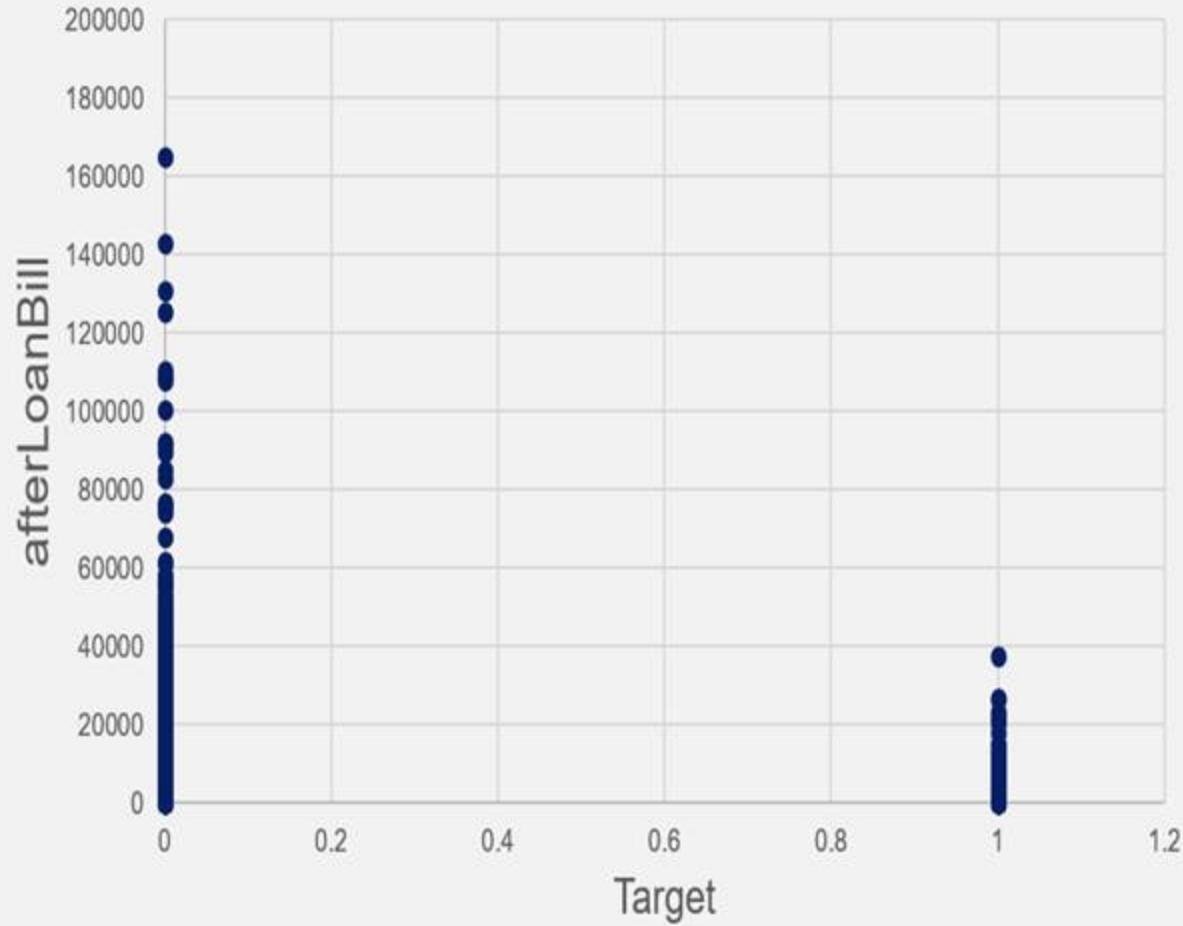
信用卡账单记录 bill_detail.txt
共15个字段

字段	注释
用户 id	整数
账单时间戳	整数 0 表示未知
银行 id	枚举类型
上期账单金额	浮点数
上期还款金额	浮点数
信用卡额度	浮点数
本期账单余额	浮点数
本期账单最低还款额	浮点数
消费笔数	整数
本期账单金额	浮点数
调整金额	浮点数
循环利息	浮点数
可用余额	浮点数
预借现金额度	浮点数
还款状态	枚举值

用户浏览历史记录



放款后信用卡账单记录



Part 3 特征工程

“ 数据和特征决定了机器学习的上限，
而模型和算法只是逼近这个上限。 ”



特征工程

用户基本特征: 描述用户的基本属性信息

性别, 职业, 教育程度, 婚姻状态, 户口类型

银行流水特征: 描述用户的所有银行消费信息

账户流通金额, 交易频繁程度, 收入支出比, 平均每天收入金额, 工资与非工资收入

用户浏览特征: 描述用户对本身消费行为的关注情况

识别用户的责任意识

274组浏览数据——浏览行为

信用卡消费特征: 描述用户的信用消费情况

统计特征

信用卡记录频率；
银行个数；
信用卡额度
上期账单金额；
上期还款金额；
.....

业务特征

上期账单金额大于上期还款金额；
总体额度趋势情况；
轻度拖欠期数；
轻度拖欠金额；
重度拖欠期数；
重度拖欠金额；
重度拖欠期数占比
.....

交叉特征: 描述基本特征之间的交叉关系

上期账单金额

上期还款金额

信用卡额度

本期账单金额

.....

上期账单金额减上期还款金额, $>0, <0, =0$
上期账单金额减信用卡额度, $>0, <0, =0$
上期账单金额减本期账单金额, $>0, <0, =0$
.....

共12个交叉特征

构造交叉特征群, 进行特征重要性排序

时间特征: 描述用户相关信息间的时间特性

统计特性

信用卡记录频率
时间跨度
独立时间记录个数
.....

相对贷款放款时间交叉特性

放款前最近记录时间戳
放款后最近记录时间戳
放款前记录数
放款后记录数
.....

特征工程

放款特征: 描述用户放款前后特征对比

银行流水

放款前后记录个数
放款前后流水额度比
.....

浏览记录

放款前后记录个数
放款前一个月记录个数
.....

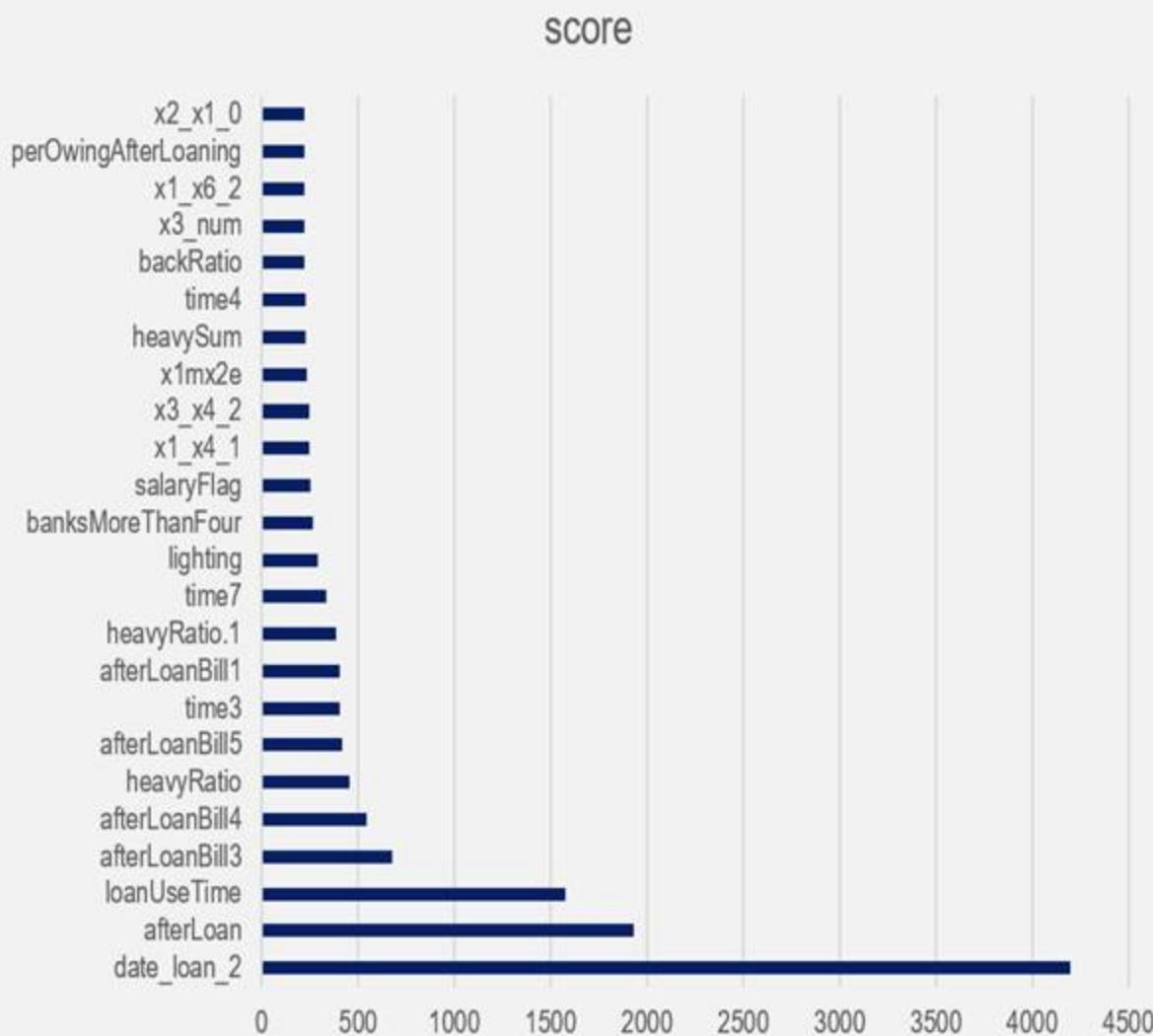
信用卡账单

放款前后记录个数
放款前后还款比例
放款前后信用卡支出均额
.....

特征工程

清洗异常样本（僵尸用户）

归一化
离散化
构造哑变量
One-Hot编码
缺失值填补
.....



Part **4**

分析建模

$$Y = a + bX_1 + cX_2 + \dots + nX_n$$

广义线性模型



多重线性回归

Logistic
Regression

Poisson
Regression

负二项回归

Logistic Regression

```
#对数据进行划分并且进行训练
train = loan_data.iloc[0: 55596, :]
test = loan_data.iloc[55596:, :]
train_X, test_X, train_y, test_y = train_test_split(train, target, test_size = 0.2, random_state = 0)
train_y = train_y['label']
test_y = test_y['label']
lr_model = LogisticRegression(C = 1.0, penalty = 'l2')
lr_model.fit(train_X, train_y)
#验证集进行预测
pred_test = lr_model.predict(test_X)
#对预测结果进行评估
print classification_report(test_y, pred_test)

#对测试集生成结果并存储为csv格式
pred = lr_model.predict_proba(test)
result = pd.DataFrame(pred)
result.index = test.index
result.columns = ['0', 'probability']
result.drop('0', axis = 1, inplace = True)
print result.head(5)
result.to_csv(self.result)
```

0.43263 82th

Random Forest

GBDT (Gradient Boosting Decision Tree) :

Xgboost

LightGBM



Xgboost:

```
def pipeline(iteration, random_seed, gamma, max_depth, lambd, subsample, colsample_bytree, min_child_weight):  
    params={  
        'booster': 'gbtree',  
        'objective': 'binary:logistic',  
        'scale_pos_weight': float(len(train_y)-sum(train_y))/float(sum(train_y)),  
        'eval_metric': 'auc',  
        'gamma': gamma,  
        'max_depth': max_depth,  
        'lambda': lambd,  
        'subsample': subsample,  
        'colsample_bytree': colsample_bytree,  
        'min_child_weight': min_child_weight,  
        'eta': 0.4,  
        'seed': random_seed  
    }  
    if max_depth==4:  
        nround = 800  
    elif max_depth==5:  
        nround = 700  
    else:  
        nround = 600  
    watchlist = [(dtrain, 'train')]  
    model = xgb.train(params, dtrain, num_boost_round=nround, evals=watchlist)
```

LightGBM:

```
for x_index, y_index in kf.split(train_x, train_y):
    x_train, x_val = train_x[x_index], train_x[y_index]
    y_train, y_val = train_y[x_index], train_y[y_index]
    lgb_train = lgb.Dataset(x_train, y_train)
    lgb_eval = lgb.Dataset(x_val, y_val, reference=lgb_train)
    params = {
        'task': 'train',
        'boosting_type': 'gbdt',
        'objective': 'binary',
        #      'metric': 'auc',
        'num_leaves': 16,
        'learning_rate': 0.1,
        'feature_fraction': 0.8,
        'bagging_fraction': 0.8,
        'bagging_freq': 2,
        'bagging_seed': 10,
        'lambda_l1': 20,
        'lambda_l2': 150,
        'verbose': 0,
        'is_unbalance': True
    }
    model = lgb.train(params, lgb_train, num_boost_round=1500, valid_sets=[lgb_train, lgb_eval],
                      feval=ks_lgb, verbose_eval=100, early_stopping_rounds=300)
    pred_eval = model.predict(x_val, num_iteration=model.best_iteration)
    a = ks_score(y_val, pred_eval)
    auc = roc_auc_score(y_val, pred_eval)
    print(a, auc)
```

Xgboost: 0.45732 19th

LightGBM: 0.45844 17th

模型加权融合:

$0.5 * \text{Xgboost} + 0.5 * \text{LightGBM} = 0.46437$ 11th

$0.4 * \text{Xgboost} + 0.6 * \text{LightGBM} = 0.46711$ 7th

Part 5

总结展望

总结展望

排名	排名变化	团队logo	队名	最高得分	提交次数	最后提交时间	创新应用简介
7	-		NUDT丁兆云DM课程...	0.46711	13	2017-11-21 16:38	未提交
1	+/-		龙樱	0.47461	44	2017-02-20 14:28	已提交
2	+/-		Say what all late	0.47350	25	2017-02-19 13:36	未提交
3	+/-		swthekingc4da4	0.47192	34	2017-02-20 15:45	已提交
4	+/-		夜月唱情歌	0.47028	68	2017-02-20 23:09	已提交
5	+/-		PHM-LAB-第一为何这么	0.47010	110	2017-02-20 22:47	已提交
6	+/-		好奇怪bd34b	0.46979	69	2017-02-15 09:34	未提交
7	+/-		NUDT丁兆云DM课程刘麦	0.46711	13	2017-11-21 16:38	未提交



Any Questions?

谢谢！