商务智能
Business Intelligence

分类：基本概念、决策树、贝叶斯方法、模型评价

朱卫平 博士
计算机学院
武汉大学

分类

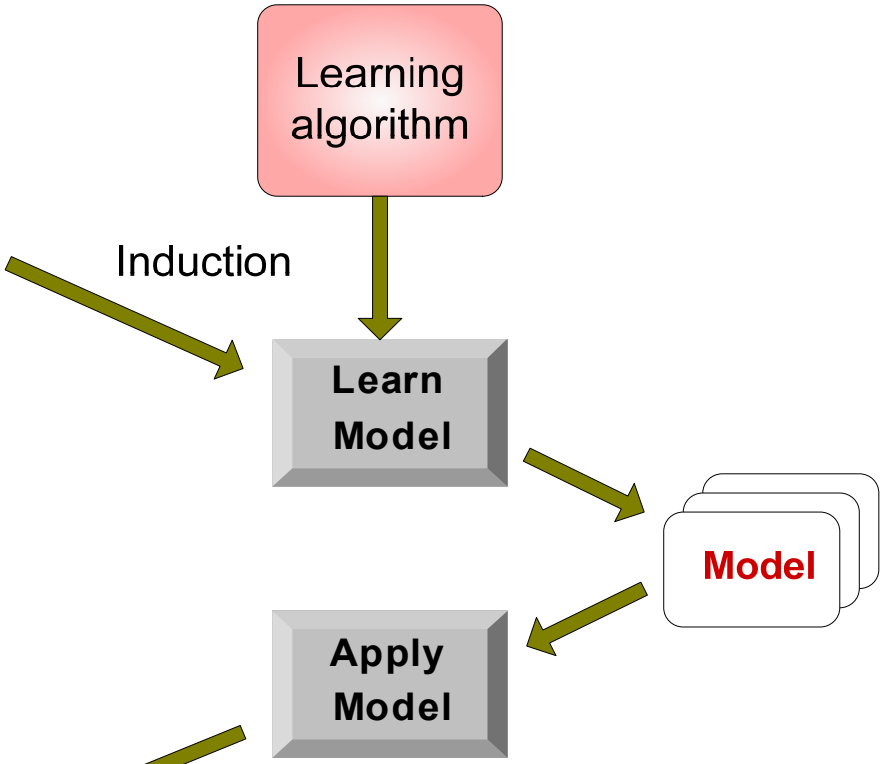
- 分类是利用一个分类函数（分类模型、分类器），该模型能把数据库中的数据映射到一个给定类别中。

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

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



决策树中Entropy的计算

- 给定结点t的 Entropy值计算 :

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

($p(j \mid t)$ 是在结点t中，类j发生的概率).

  - 当类分布均衡时， Entropy值达到最大值 ($\log n_c$)
  - 相反当只有一个类时，值达到最小值0
  - Entropy 与 GINI相似

计算 Entropy的例子

$$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

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

不纯性的测量: 信息增益

■ Information Gain:

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

n_i = 孩子结点 i的记录数,

n = 结点 p的记录数.

– 在 ID3 和 C4.5中使用

基于信息增益的划分

■ 增益率（Gain Ratio）：

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

- 熵和Gini指标等不纯性趋向于有利于具有大量不同值的属性!
  - 如：利用雇员id产生更纯的划分，但它却毫无用处
- 每个划分相关联的记录数太少，将不能做出可靠的预测
- 解决该问题的策略有两种：
  - 限制测试条件只能是二元划分
  - 使用增益率。K越大Split Info越大增益率越小

不纯性的测量: Classification Error

■ 给定结点t的 Classification Error值计算 :

$$Error(t) = 1 - \max_i P(i \mid t)$$

- 当类分布均衡时， error值达到最大值 (1 - 1/n_c)
- 相反当只有一个类时， error值达到最小值0

计算Classification Error的例子

$$Error(t) = 1 - \max_i P(i \mid t)$$

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

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

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

P(C1) = 1/6 P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

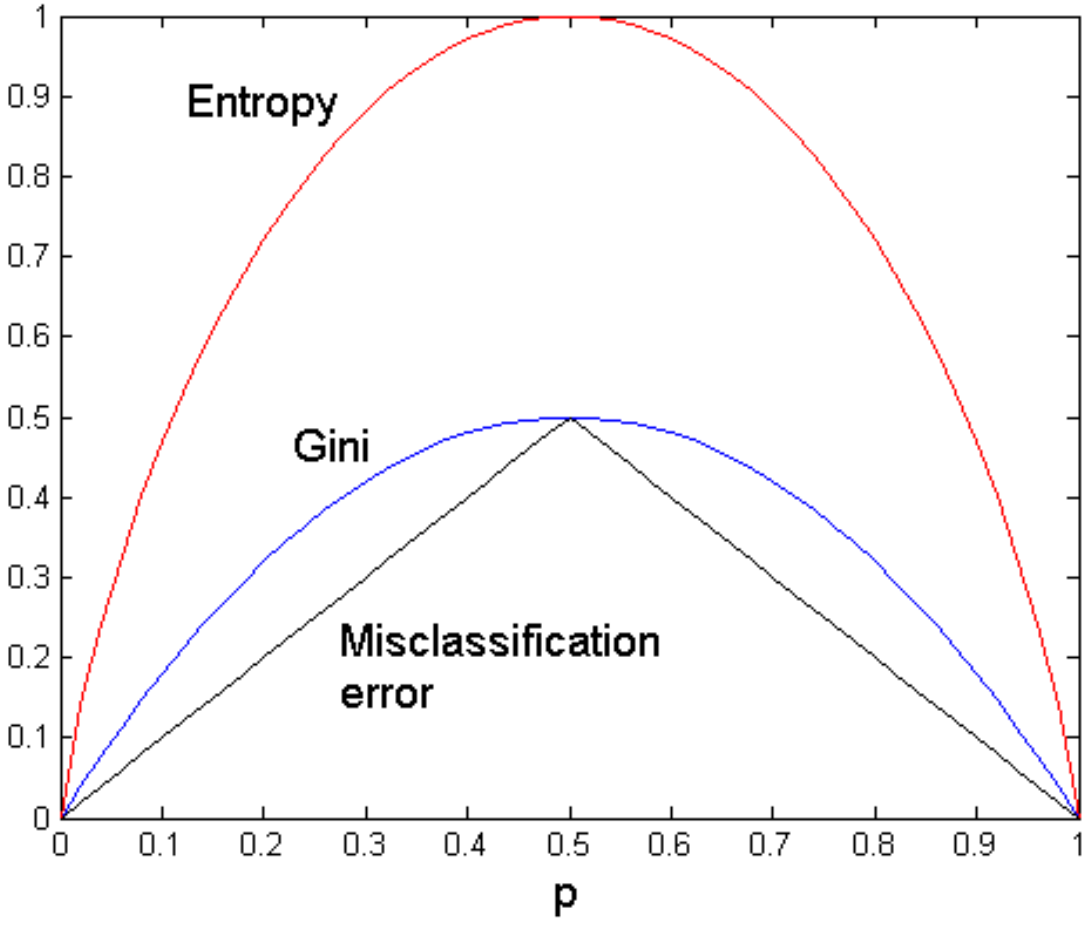
C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

不纯性度量之间的比较

二元分类问题:



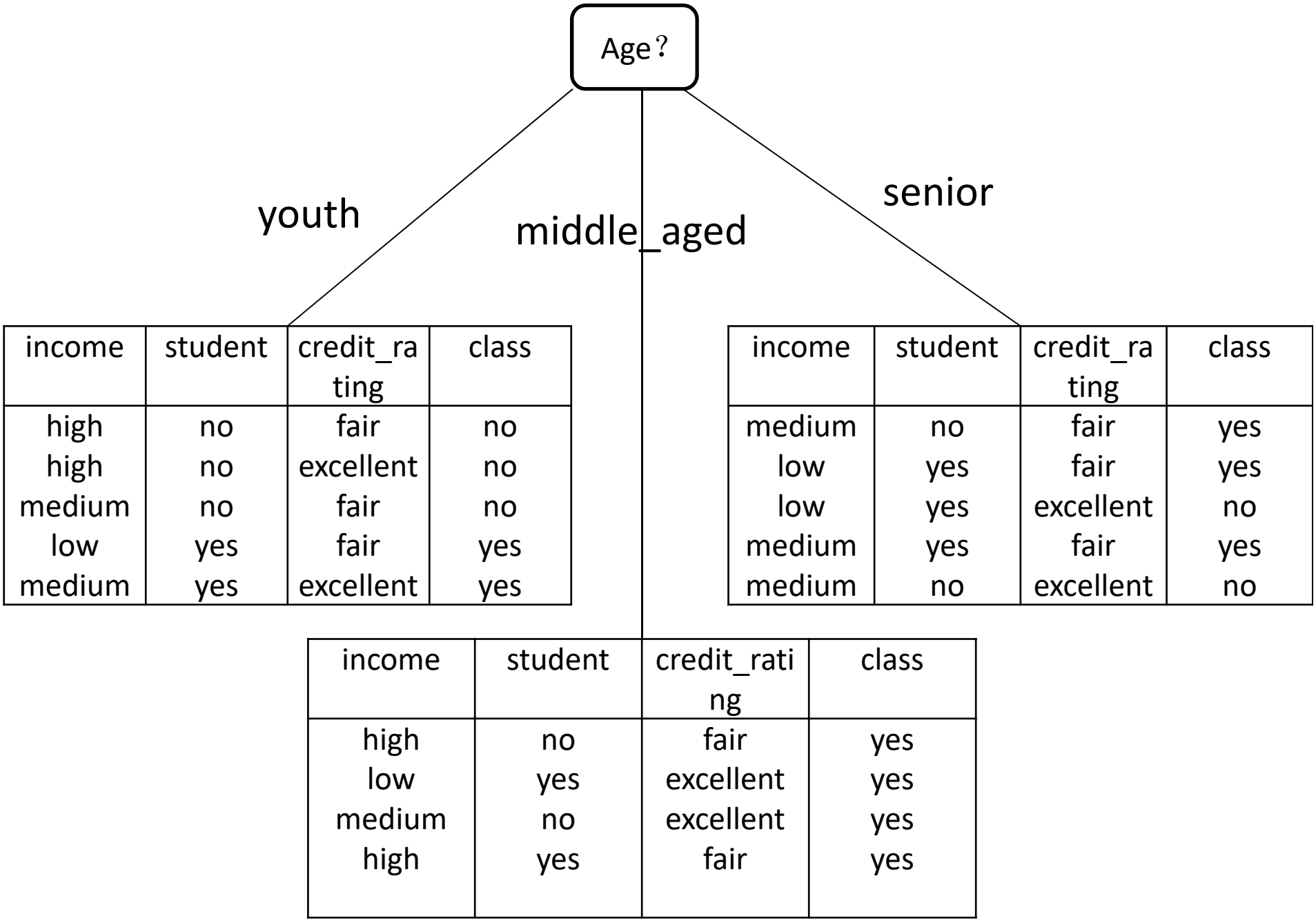
课堂练习：对下列数据集进行二元分类

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

课堂练习：对下列数据集进行二元分类

ALLElectronics 顾客数据库标记类的训练元组

RID	age	income	student	credit_rating	Class:buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no



income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

决策树

- 决策树归纳的设计问题
  - 如何分裂训练记录
    - 怎样为不同类型的属性指定测试条件?
    - 怎样评估每种测试条件?
  - 如何停止分裂过程

停止分裂过程

- 当所有的记录属于同一类时，停止分裂
- 当所有的记录都有相同的属性时，停止分裂
- 提前终止树的生长

三种著名的决策树

- Cart：基本的决策树算法

- Id3：利用增益比不纯性，树采用二叉树，停止准则为当所有的记录属于同一类时，停止分裂，或当所有的记录都有相同的属性时，停止分裂

- C4.5：id3的改进版本，也是最流行的分类数算法。采用多重分支和剪枝技术。

决策树

- 特点:
  - 决策树是一种构建分类模型的非参数方法
  - 不需要昂贵的的计算代价
  - 决策树相对容易解释
  - 决策树是学习离散值函数的典型代表
  - 决策数对于噪声的干扰具有相当好的鲁棒性
  - 冗余属性不会对决策树的准确率造成不利影响
  - 数据碎片问题。随着数的生长，可能导致叶结点记录数太少，对于叶结点代表的类，不能做出具有统计意义的判决
  - 子树可能在决策树中重复多次。使决策树过于复杂

贝叶斯分类

- 一种统计分类器: 进行概率性预测，也就是预测元组属于某类的概率
- 理论基础: 基于贝叶斯定理（Bayes' Theorem）
- 性能: 朴素贝叶斯分类法（*naïve Bayesian classifier*）可以媲美于决策树和特定的神经网络
- 增量式: 每一个训练实例会逐渐增加（或减少）某一假设的概率，也就是说观察数据可以与先验知识合并

贝叶斯理论: 基础

- 全概率公式: $P(B)=\sum_{i=1}^{M} P(B|A_i)P(A_i)$
- 贝叶斯定理: $P(H|\mathbf{X})=\frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}=P(\mathbf{X}|H)\times P(H)/P(\mathbf{X})$
  - **X** 是待测试的数据元组，类标识未知
  - 令 H 是一个假设（*hypothesis*）：X 属于类 C
  - 问题在于确定后验概率 P(H|**X**)：当观测到X时假设H成立的概率
  - P(H) (先验概率):
    - E.g., 任意顾客将购买计算机的概率，不管他们的年龄、收入等
  - P(X): X被观测到的概率
  - P(**X**|H) (可能性，likelihood): 给定假设H成立，观测数据被观察到的概率
    - E.g., 假设已知X将会购买计算机，X的年龄在31到40之间，中等收入的概率

朴素贝叶斯分类

- 令 D 是训练元组和他们所关联类标号的集合。每个元组表示为一个n维属性向量 **X** = (x_1, x_2, …, x_n)
- 假设有 *m* 个类 C_1, C_2, …, C_m.
- 分类就是要求出最大的后验概率 P(C_i|**X**)
- 使用贝叶斯定理有：

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- 可以用以下关系表示 posteriori = likelihood x prior/evidence
- 预测 **X** 属于 C_i iff P(C_i|**X**) =max(P(C_k|X)), for all *k* classes
- 实践困难: 需要许多概率的初始知识 , 将需要较大的计算耗费
- 由于 P(X) 对所有类是常数，只需要计算下式的最大值

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

朴素贝叶斯分类

- 类条件独立假设: 各属性之间是条件独立的 (i.e., 各属性之间不存在相互关系). 那么进行化简:

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times ... \times P(x_n|C_i)$$

- 这可以极大的减少计算开销: 只需计算单个属性的类分布
- If A_k 是标称的, $P(x_k|C_i)$ 等于类为C_i 的元组中 A_k 值为 x_k 的元组个数除以 $|C_{i,D}|$ (D中属于C_i 的元组个数)
- If A_k 是连续值属性， $P(x_k|C_i)$ 通常用高斯分布计算

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$P(X_K | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

朴素贝叶斯分类：训练集

Class:

C1:buys_computer = ‘yes’

C2:buys_computer = ‘no’

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

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

朴素贝叶斯分类: 例子

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

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes")
  = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no")
  = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

因此，**X 属于类("buys_computer = yes")**

避免0概率问题

- Naïve Bayesian 需要每个条件概率非零，否则总概率将为0

$$P(X_K \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- Ex. 假定数据集有 1000 元组, income=low (0), income= medium (990), income = high (10)
- 使用拉普拉斯校准（**Laplacian correction** ）
  - 每个类别的元组加1

    Prob(income = low) = 1/1003

    Prob(income = medium) = 991/1003

    Prob(income = high) = 11/1003
  - “校准的” 概率估计和 “未校准的” 的概率估计很接近，但避免了0概率问题

朴素贝叶斯分类: 分析

- 优势
  - 易于实现
  - 在大量的情况下有较好的结果
- 不足
  - 类条件独立假设假设会引起准确度损失
  - 在实际中，各个属性间可能存在依赖性
    - E.g., hospitals: patients: Profile: age, family history, etc. Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.

评估分类器性能的度量

	预测的类			
实际的类		Class=Yes	Class=No	合计
	Class=Yes	TP (True Positive)	FN (False Negative)	P
	Class=No	FP (False Positive)	TN (True Negative)	N

混淆矩阵

$$\text{准确率}\ (\text{Accuracy}) = \frac{TP+TN}{TP+TN+FP+FN} = \frac{TP+TN}{P+N}$$

$$\text{错误率}(\text{error rate}) = 1 - \text{Accuracy}$$

混淆矩阵-例子

类	buyscomputer=yes	buyscomputer=no	合计	识别率（%）
buys_computer=yes	6954	46	7000	
buys_computer=no	412	2588	3000	
合计	7366	2634	10000	95.42

类buys_computer=yes和buys_computer=no的混淆矩阵

类不平衡问题-准确率的缺点

- 在类分布相对平衡时有效
- 考虑2类问题
  - 类0的样本数 = 9990
  - 类1的样本数 = 10

- 如果模型预测所有的样本为类0, 准确率为 9990/10000 = 99.9 %
  - 准确率的值具有欺骗性
  - 模型并没有分对类1的任何样本

类不平衡问题-准确率的缺点

■ 灵敏性（sensitivity）和特效性（specificity）

$$\text{sensitivity} = \frac{\text{TP}}{\text{P}}$$ 正确识别的正元组的百分比

$$\text{specificity} = \frac{\text{TN}}{\text{N}}$$ 正确识别的负元组的百分比

■考虑2类问题，类0的样本数 = 9990, 类1的样本数 = 10,如果模型预测所有的样本为类0, 灵敏性和特效性是多少？

类不平衡问题-准确率的缺点

类	yes	no	合计	识别率（%）
yes	90	210	300	30.00
no	140	9560	9700	98.56
合计	230	9770	10000	96.50

类cancer=yes和cancer=no的混淆矩阵

灵敏性和特效性是多少？

精度和召回率

- 精度和召回率

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{P} \qquad \text{sensitivity} = \frac{TP}{P}$$

- 精度表示的是标记为某一类的元组是正确的概率
- 召回率表示的是实际是某一类元组的元组被正确标记的概率
- 两者存在逆关系，可能降低其中一个提高另一个
- 两者通常同时使用，用固定召回率（如0.75）比较精度

精度和召回率

类	yes	no	合计	
yes	90	210	300	
no	140	9560	9700	
合计	230	9770	10000	

类cancer=yes和cancer=no的混淆矩阵

精度和召回率是多少？

F度量和F_β度量

- 精度和召回率

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

$$F_\beta = \frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

  - F度量赋予精度和召回率相同的权重
  - F_β度量赋予召回率权重是精度的β倍

度量小结

准确率 $\text{accuracy} = \frac{TP + TN}{P + N}$

精度 $\text{precision} = \frac{TP}{TP + FP}$

错误率 $\text{error rate} = \frac{FP + FN}{P + N}$

召回率 $\text{recall} = \frac{TP}{P}$

灵敏性 $\text{sensitivity} = \frac{TP}{P}$

$F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

特效性 $\text{specificity} = \frac{TN}{N}$

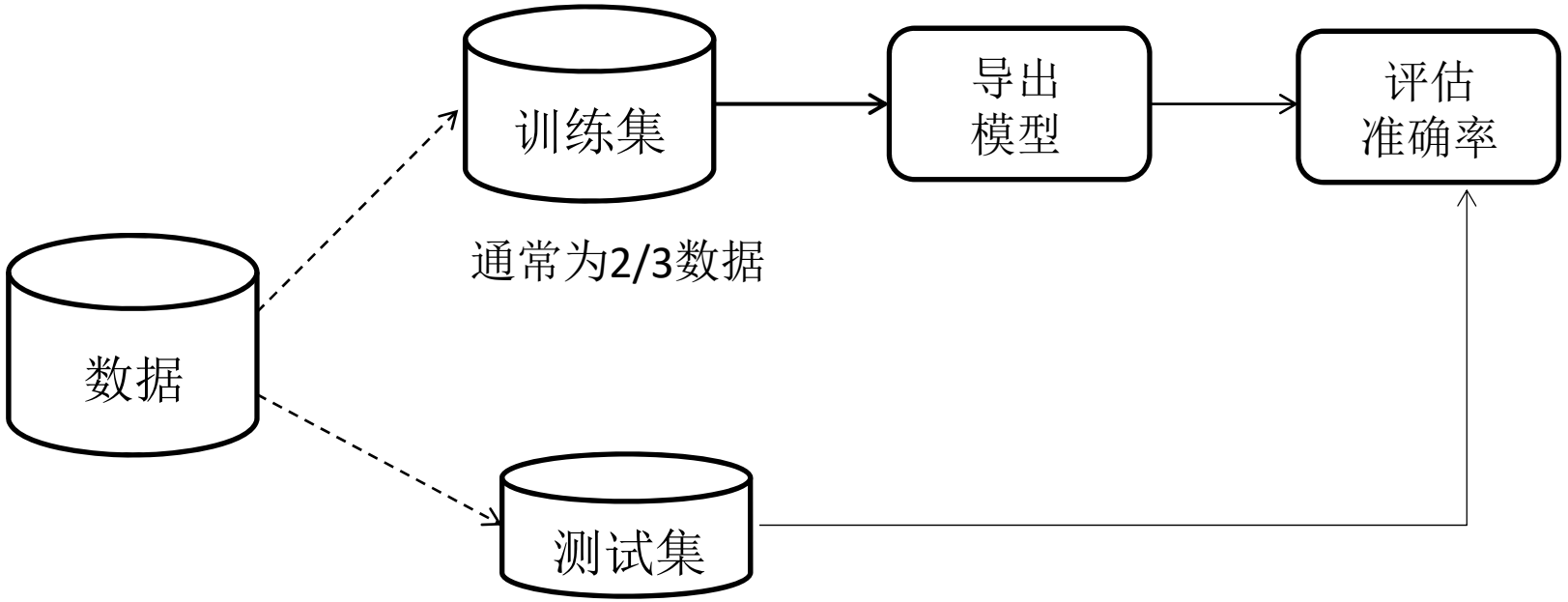
$F_\beta = \frac{(1 + \beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$

其他问题

当每个元组可以属于多个类是怎么办？

检验方法

■ 保持（hold out） 随机固定划分数据集



■ 随机二次抽样：保持法重复k次取平均值

检验方法

- 交叉验证
  - K折交叉验证
  - 留一法 （特殊的K折交叉验证，K设置为元组数，每次只给检验集留一个样本）
- 自助法
  - .632自助法： d个样本有放回的抽样d次作为训练集，允许多次选择同一样本
  - 63.2%的原数据元组将会出现在训练集中。为什么是63.2%?

ROC (Receiver Operating Characteristic)

- ROC曲线是一种比较不同分类器的可视化方法。它显示分类器真正率（TPR）和假正率（FPR）之间的关系。
- 真正率 TPR = TP / P （灵敏度）
- 假正率 FPR = FP / N （1-特效型）

ROC (Receiver Operating Characteristic)

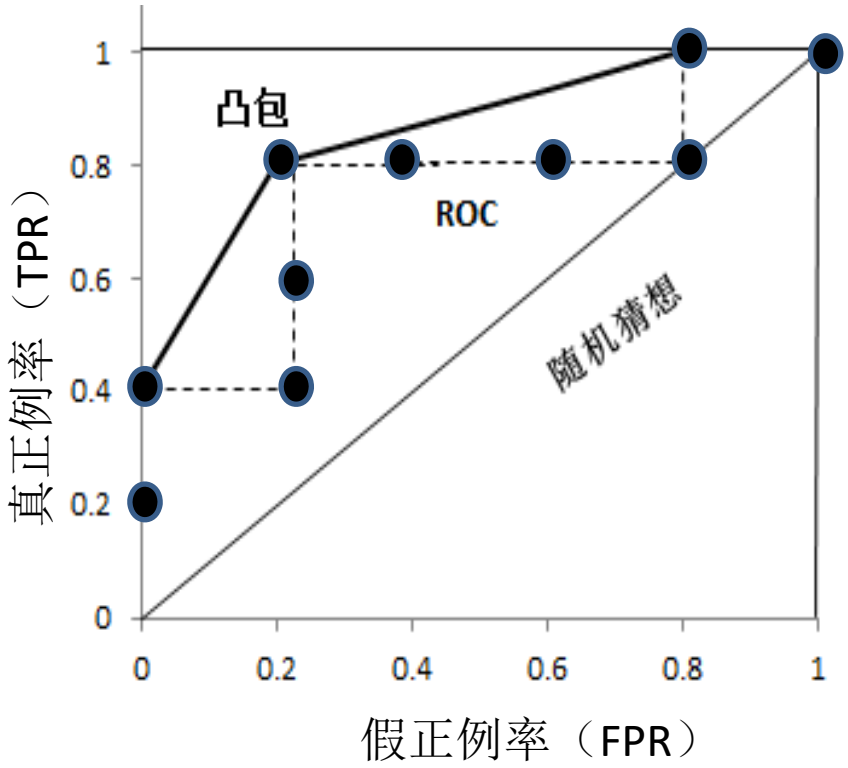
- ROC曲线显示分类器真正率（TPR）和假正率（FPR）之间的关系。
- 首先形成以下图表

元组编号	类	概率	TP	FP	TN	FN	TPR	FPR
1	P	0.90	1	0	5	4	0.2	0.0
2	P	0.80	2	0	5	3	0.4	0.0
3	N	0.70	2	1	4	3	0.4	0.2
4	P	0.60	2	1	4	2	0.6	0.2
5	P	0.55	4	1	4	1	0.8	0.2
6	N	0.54	4	2	3	1	0.8	0.4
7	N	0.53	4	3	2	1	0.8	0.6
8	N	0.51	4	4	1	1	0.8	0.8
9	P	0.50	5	4	1	0	1.0	0.8
10	N	0.40	5	5	0	0	1.0	1.0

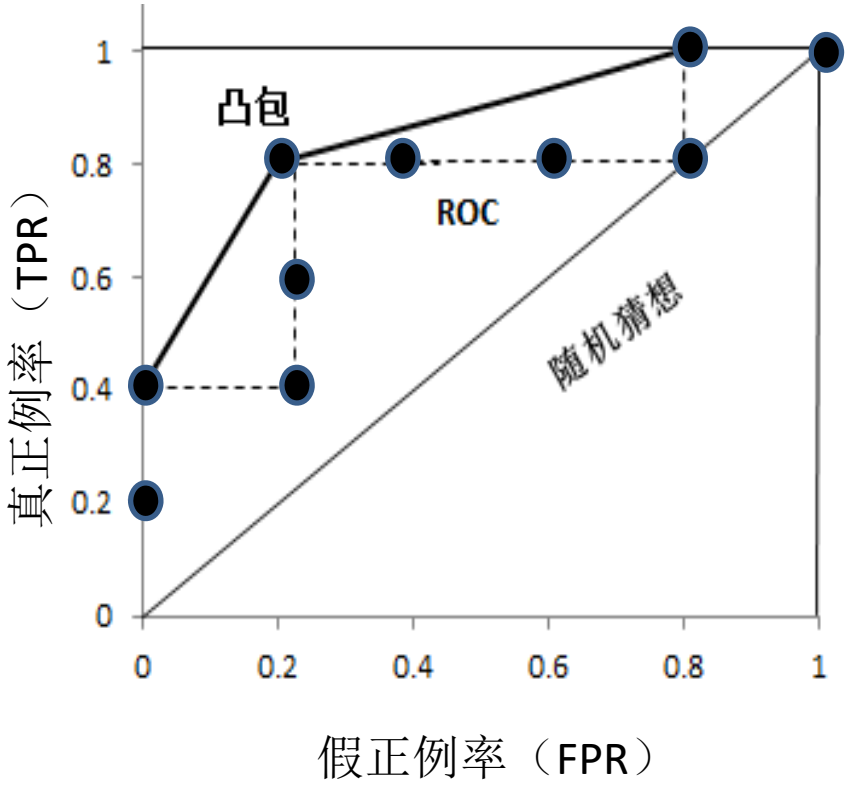
元组按递减得分排序，其中得分是概率分类器返回的值，以此计算第三列中数据为阈值时的TP, FP, TN, FN等数值，然后计算TPR和FPR

ROC

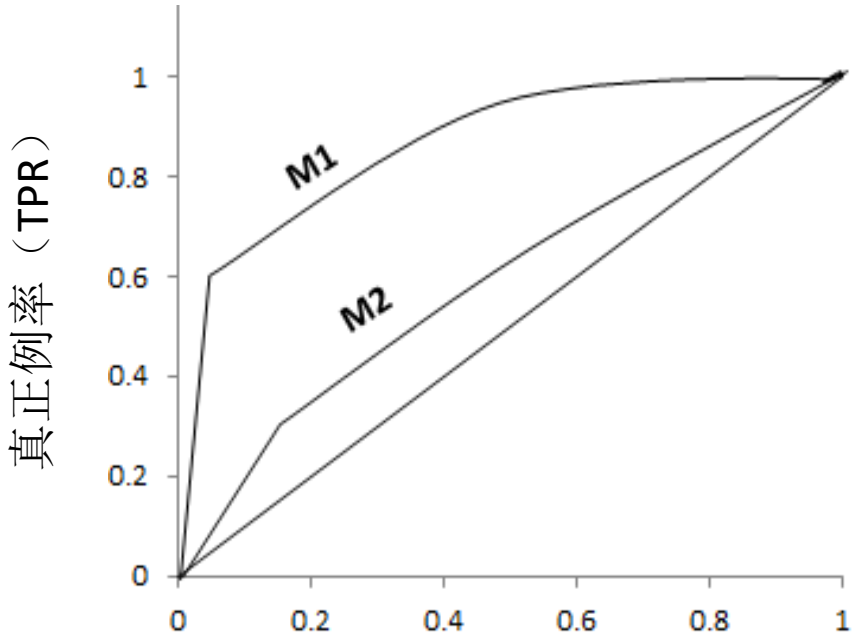
元组编号	类	概率	TP	FP	TN	FN	TPR	FPR
1	P	0.90	1	0	5	4	0.2	0.0
2	P	0.80	2	0	5	3	0.4	0.0
3	N	0.70	2	1	4	3	0.4	0.2
4	P	0.60	2	1	4	2	0.6	0.2
5	P	0.55	4	1	4	1	0.8	0.2
6	N	0.54	4	2	3	1	0.8	0.4
7	N	0.53	4	3	2	1	0.8	0.6
8	N	0.51	4	4	1	1	0.8	0.8
9	P	0.50	5	4	1	0	1.0	0.8
10	N	0.40	5	5	0	0	1.0	1.0



ROC



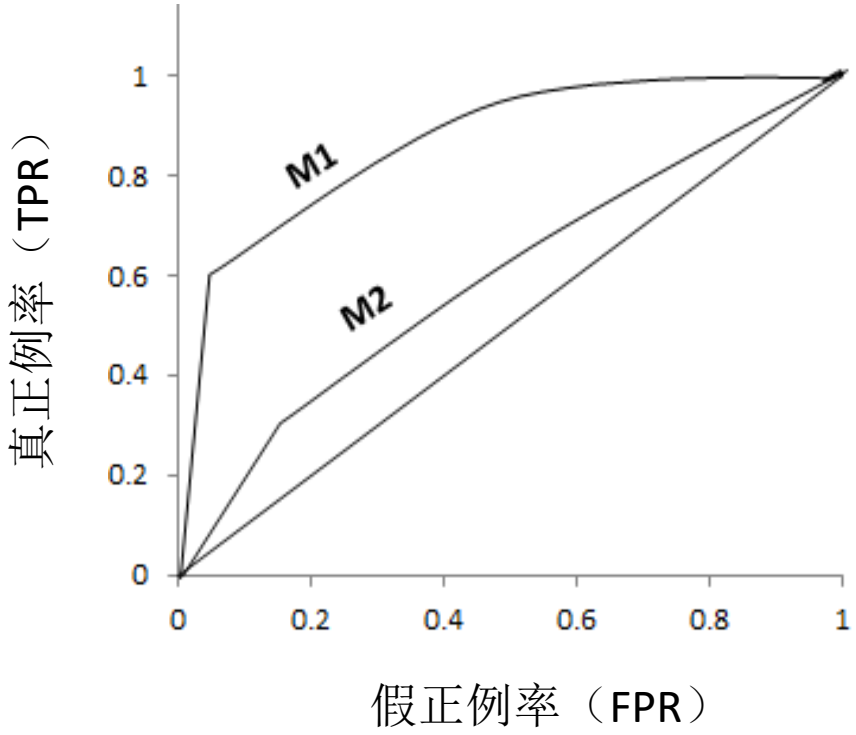
ROC 曲线



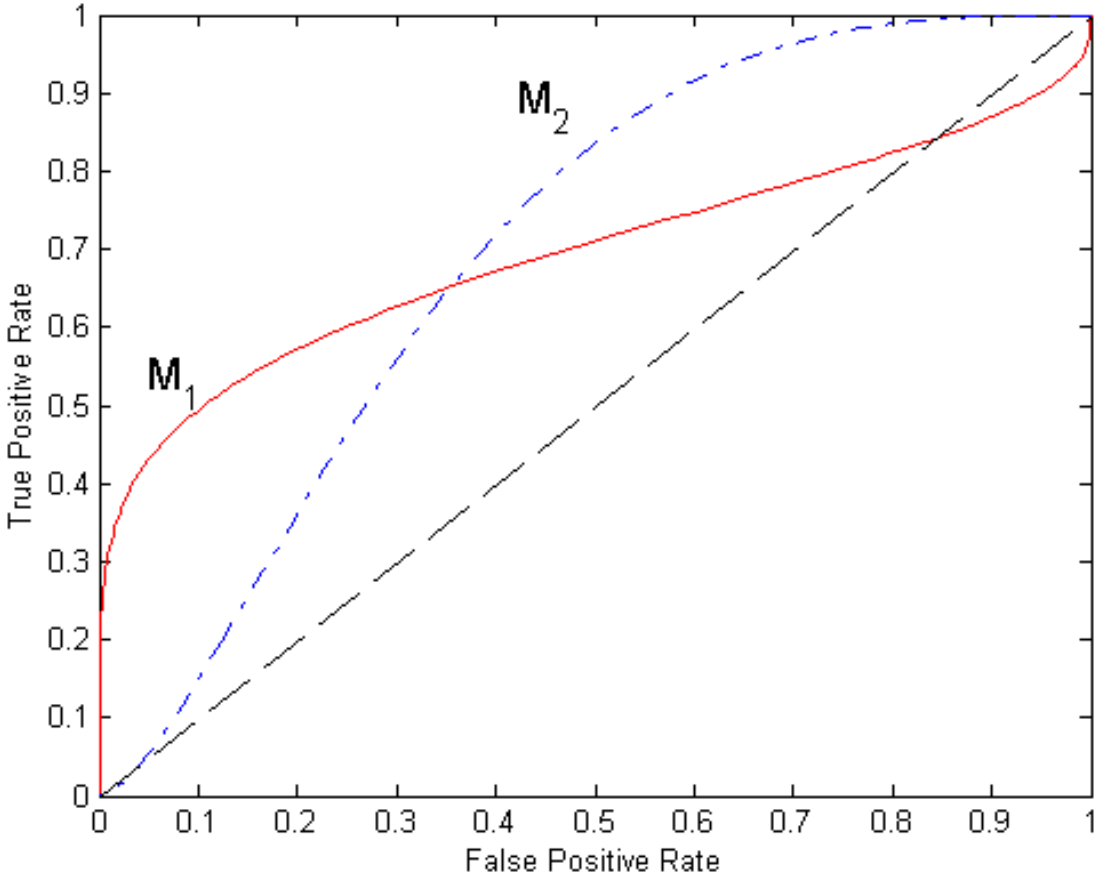
两个分类模型M1和M2的ROC曲线（凸包）

ROC (Receiver Operating Characteristic)

- ROC 曲线上有几个关键点，它们有公认的解释：
  - （TPR=0，FPR=0）：把每个实例都预测为负类的模型
  - （TPR=1，FPR=1）：把每个实例都预测为正类的模型
  - （TPR=1，FPR=0）：理想模型



使用ROC曲线比较模型

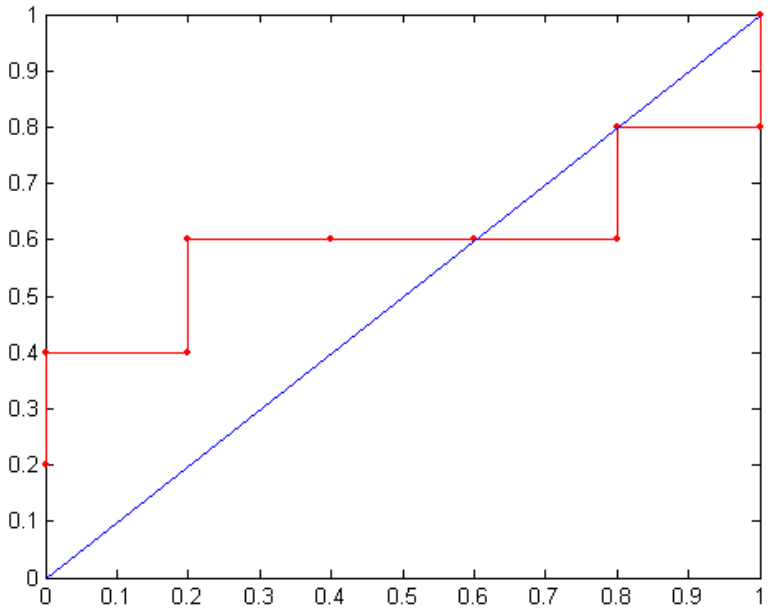


- 没有哪个模型能够压倒对方
  - FRR<0.36, M_1 较好
  - FRR>0.36, M_2较好
- ROC曲线下方的面积
  - 理想情况:
    - 面积= 1
  - 随机猜测:
    - 面积 = 0.5

ROC曲线练习

	Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=		0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
	TP	5	4	4	3	3	3	3	2	2	1	0
	FP	5	5	4	4	3	2	1	1	0	0	0
	TN	0	0	1	1	2	3	4	4	5	5	5
	FN	0	1	1	2	2	2	2	3	3	4	5
→	TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
→	FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC 曲线:



Thank You!

Q&A