第五章 分类

张静
（Jingzhang@ecust.edu.cn）

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

分类 vs. 数值预测

- 分类:
  - 预测分类标号
  - 在分类属性中的训练样本集和值(类标号)的基础上分类数据(建立模型)并使用它分类新数据
- 数值预测:
  - 为连续值函数建模,预测未知的或缺省值
- 典型应用
  - 信誉证实
  - 目标市场
  - 医学诊断
  - 性能预测

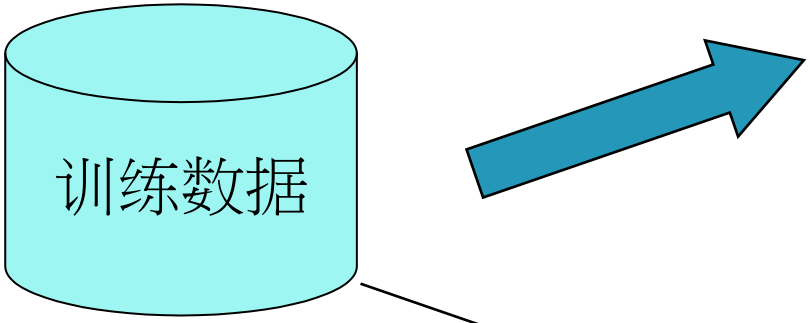
分类—两阶段过程

- 学习阶段 -- 模型构造（Model construction）：描述预先定义好的类别
  - 每个元组/样本被假定为从属于一个预定义的类别，即类标号属性（class label attribute）
  - 用于构造模型的元组集合被称之为训练集合（training set）
  - 模型可以有多种表示方法，诸如分类规则，决策树，或者数学公式等

分类—两阶段过程

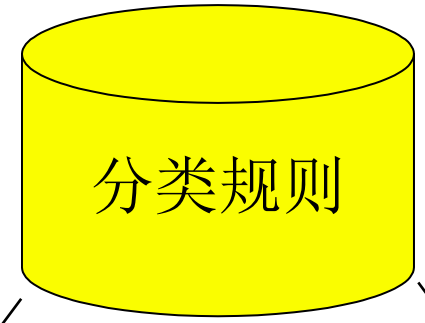
- 分类阶段 -- 使用模型进行分类：在将来对未知对象进行分类
  - 估算模型的准确度
    - 各元组的已知分类标签和从模型中获得的标签进行比较
    - 准确度即样本集合中能够被该模型正确分类的元组的百分比
    - 测试集合（Test set）和训练集合（training set）独立,否则无法正确衡量模型的准确度
  - 如果准确度是可接受的，则可以使用该模型对未被分类的数据元组进行分类

分类过程：构造模型



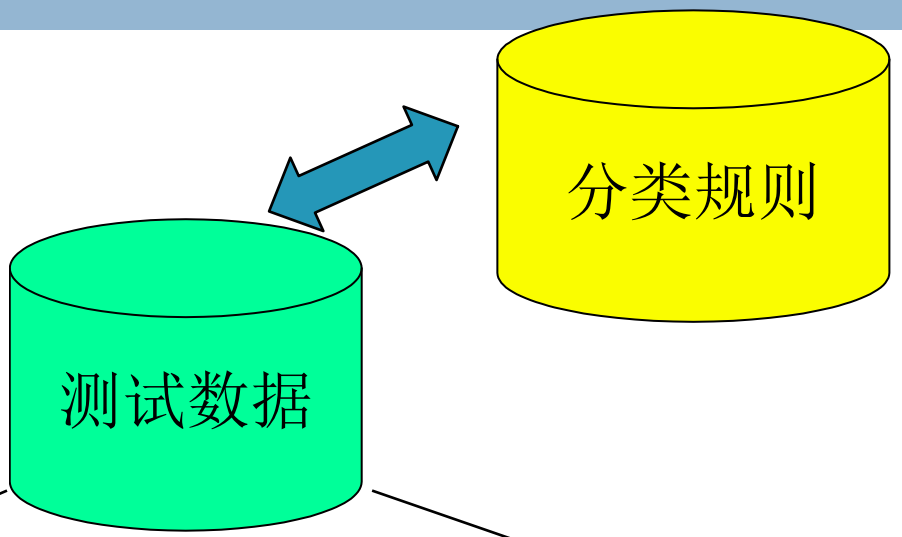
分类算法

NAME	RANK	YEARS	Bonus
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no



IF rank = ‘professor’
OR years > 6
THEN Bonus = ‘yes’

分类过程：使用模型进行预测



NAME	RANK	YEARS	Bonus
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

(Jeff, Professor, 4)

Bonus?

Yes

有监督学习vs无监督学习
Supervised vs. Unsupervised Learning

- 有监督学习（Supervised learning (classification)）
  - 有监督（Supervision）：训练数据均含有一个字段，该字段用于表明各个元组所属的类别
  - 新数据在训练集的基础上进行分类
- 无监督学习（Unsupervised learning (clustering)）
  - 训练数据集合中并不含有一个表征各个元组类别的字段（训练集的类标号未知）
  - 给定一个度量或者观测值集，意在确定数据中类或聚类的存在

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

决策树归纳

- 决策树
  - 一个类似于流程图的树结构
  - 内部节点表示一个属性上的测试
  - 每个分支代表一个测试的输出
  - 叶结点代表类或类分布

决策树生成

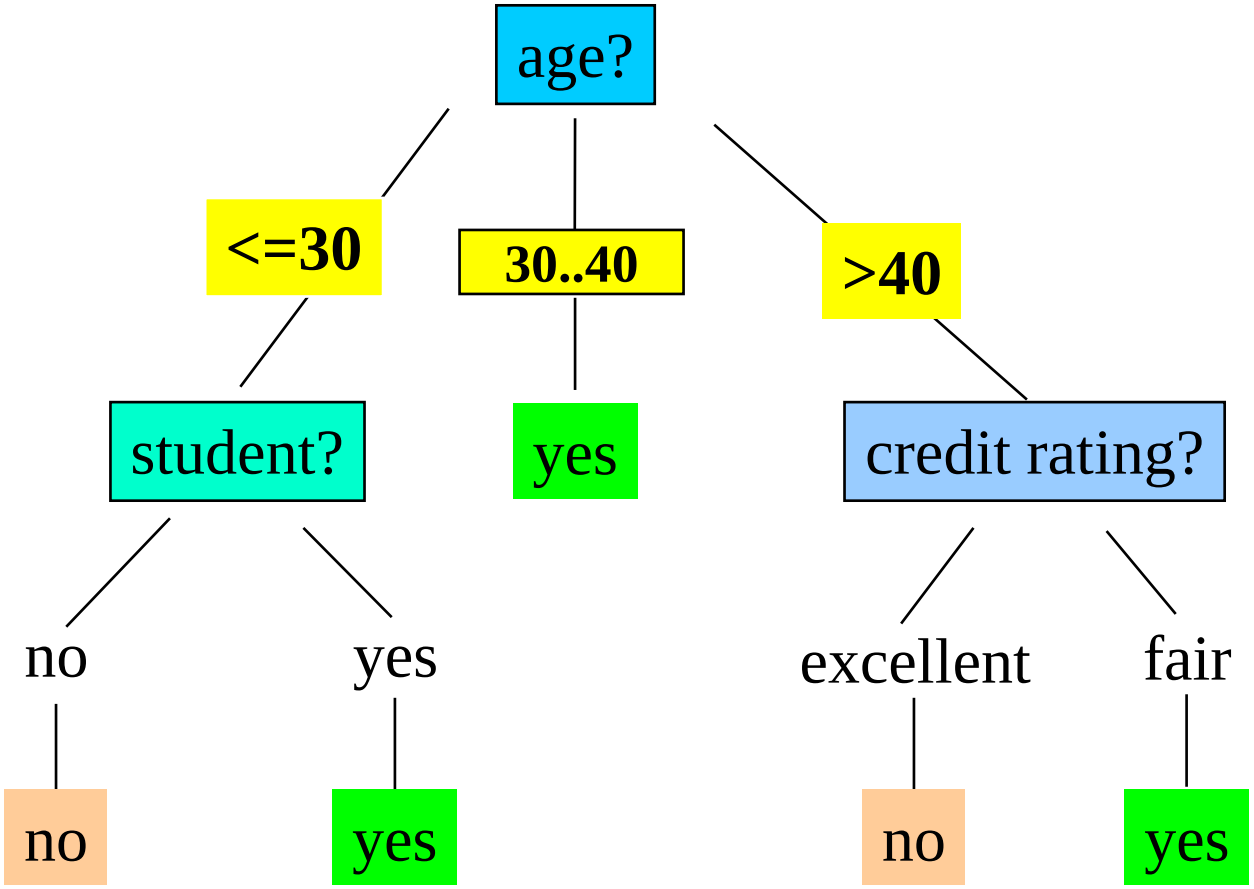
- 决策树的生成包括两个过程
  - 树的构建
    - 首先所有的训练样本都在根结点
    - 基于所选的属性循环的划分样本
  - 树剪枝
    - 识别和删除那些反映噪声或离群点的分支
- 决策树的使用:为一个未知的样本分类
  - 在决策树上测试样本的属性值

训练数据集合

其中：
Buys_computer
是类别标签

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

输出：一个针对“*buys_computer*”的决策树



决策树归纳算法

- 基本算法 (贪心算法)
  - 按照自顶向下递归划分的方法构造决策树（top-down recursive divide-and-conquer manner）
  - 最开始，所有的训练样本均在根结点
  - 样本基于被选择的属性被递归划分(所有属性都是分类的,若属性值连续，则首先要进行离散化处理)
  - 测试属性的选择基于启发式规则或者统计度量（例如信息增益（information gain））
- 终止划分的条件
  - 给定结点的所有样本属于同一类
  - 没有更多的属性用于再次划分
  - 没有剩余的样本

决策树归纳算法

- **ID3算法**
  - 树以代表训练样本的单个结点开始，如果所有样本都属于同一个类，则该结点成为树叶，并用该类标号。
  - 否则，算法使用称为信息增益的基于熵的度量作为启发信息，选择能够最好地将样本分类的属性，使该属性成为该结点的“测试”或“判定”属性。
  - 对测试属性的每个已知值，创建一个分枝，并据此划分样本。
  - 算法使用同样的过程，递归地形成每个划分上的样本决策树。一旦一个属性出现在一个结点上，该结点的任何后代均不该考虑该属性。

决策树归纳算法

- 递归划分步骤仅当满足下列条件之一时停止：
  - 给定结点的所有样本属于同一类。
  - 没有剩余属性可以用来进一步划分样本。在此情况下，使用多数表决。这涉及将给定的结点转换成树叶，并用样本中多数所在的类标记它。
  - 分枝测试属性的某一值下没有样本。则以样本中的多数类创建一个树叶。

属性选择度量

- 属性选择度量
  - 是一种选择分裂准则，将给定的类标记的训练元组的数据划分D“最好”地分成个体类的启发式方法。
- 三种流行的属性选择度量
  - 信息增益
  - 增益率
  - Gini指数

信息增益

- 信息增益
  - 选择具有最高信息增益（或最大熵压缩）的属性作为当前节点的测试属性
  - 该方法使得对一个对象分类所需的信息量最小，并确保找到一棵简单的（但不必是最简单）树。
  - 所有的属性值被假定为分类的
  - 修正后可以用于连续属性

熵（Entropy）和信息增益

- 选择具有最大信息增益的属性。
- 设S是s个数据样本的集合。
- 假定类标号属性具有m个不同的值，定义m个不同类C_i (i= 1,...,m)。设s_i 是类C_i 中的样本数。
- 对一个给定的样本分类所需的期望信息如下：

$$I(s_1, s_2, ..., s_m) = \sum_{i=1}^{m} p_i \log_2(p_i)$$

- 其中，p_i 是任意样本属于C_i 的概率，并用s_i/s 估计。

熵（Entropy）和信息增益

- 设属性A共有v个不同值$\{a_1,a_2,\ldots,a_v\}$，可以用属性A将S划分为v个子集$\{S_1,\ldots,S_v\}$；其中，S_i包含S中在属性A上具有值a_i的那些样本。
- 如果A被选作测试属性，则这些子集对应于由包含集合S的结点生长出来的分枝。设s_{ij}是子集S_i中类C_i的样本数，根据A划分子集的熵或期望信息由下式给出。

$$E(A)=\sum_{j=1}^{v}\frac{S_{1j}+\ldots+S_{mj}}{s}I(S_{1j},\ldots,S_{mj})$$

- 项$\frac{S_{1j}+\cdots+S_{mj}}{s}$为第j个子集的权，并且等于子集（即，A值为$a_i$的样本集合）中的样本个数除以S中的样本总数。熵值越小，子集划分的纯度越高。对于给定的子集S_i，

$$\mathrm{I}(s_{1j},s_{2j},\ldots,s_{mj})=-\sum_{i=1}^{m}p_{ij}\log_2(p_{ij})$$

- 其中，$p_{ij}=\frac{s_{ij}}{|S_j|}$是$S_i$中的样本属于$C_i$的概率。
- 属性A的信息增益（Information gain）可如下计算：

$$\mathrm{Gain(A)}=\mathrm{I}(s_1,s_2,\ldots,s_m)-\mathrm{E(A)}$$

通过计算信息增益来选择属性

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- I(p, n) = I(9, 5) =0.940
- Compute the entropy for age:

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

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

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$\frac{5}{14} I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes's and 3 no's. Hence

$$Gain(age) = I(p,n) - E(age) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

为连续值属性计算信息增益

- 假定属性A 为连续值属性
- 确定A的最佳分裂点
  - 将A的值按递增序排序
  - 典型地，每对相邻值的中点看做可能的分裂点
    - $(a_i+a_{i+1})/2$ 是a_i 和 a_{i+1}之间的中点
  - A具有最小期望信息需求的点选作A的分裂点
- 分裂:
  - D1 是满足A ≤ split-point的元组集合, 而 D2 是满足 A > split-point的元组的集合

增益率

- 信息增益度量偏向于选择具有大量值的属性
- **C4.5 (ID3的改进) 采用增益率来克服这个问题 (信息增益的规范化)**

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

GainRatio(A) = Gain(A)/SplitInfo(A)

- **Ex.** $SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$

gain_ratio(income) = 0.029/1.557 = 0.019

- 具有最大增益率的属性被选作分裂属性

Gini 指数（Gini Index）

- 数据集合**D**包含来自**n**类的样例，**gini**指数度量数据划分或训练元组集**D**的不纯度, ***gini*(D)** 定义如下，

$$gini(D)=1-\sum_{i=1}^{n} p_i^2$$

其中p_j 是**D**中元素属于类 **j**的概率

- 如果数据集合**D** 被属性**A**分裂成两个子集D_1 和 D_2, ***gini*** 指数 ***gini*(D)** 定义为

$$gini_A(D)=\frac{|D_1|}{|D|}gini(D_1)+\frac{|D_2|}{|D|}gini(D_2)$$

- 不纯度降低:

$$\Delta gini(A)=gini(D)-gini_A(D)$$

- 具有最小$gini_{split}(D)$ (或最大不纯度降低) 的属性被选作分裂结点 (需要为每一个属性列举所有的分裂结点)

Gini 指数

- Ex. D 中有 9 个元组属于类 buys_computer = “yes”， 5 个元组属于类 “no”

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- 假定属性income 从D中划分10个元组到D_1中: {low, medium} ， 其余4 个元组分到 D_2中

$$gini_{income \in \{low,medium\}}(D) = \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_2)$$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right)$$

$$= 0.443$$

$$= Gini_{income \in \{high\}}(D).$$

同理可得，$Gini_{\{low,high\}}$ is 0.458; $Gini_{\{medium,high\}}$ is 0.450. 因此，属性income的最好二元划分在 {low,medium} (and {high}) 上，因为它有最小化的Gini 指数

属性选择度量的比较

- 三种度量一般情况下都能够得到较好的结果，但是
  - **Information gain:**
    - 偏向多值属性
  - **Gain ratio:**
    - 倾向于不平衡分裂，其中一个划分比其他划分小得多
  - **Gini index:**
    - 偏向于多值属性
    - 当类的数量很大时会有困难
    - 倾向于导致相等大小的划分和纯度

过拟合（Overfitting）和树剪枝

- 过拟合（Overfitting）：产生的决策树过分适应训练数据
  - 由于噪声或者离群点，使得很多分支反映的是训练数据中的异常。
  - 太依赖于训练数据集合，对于未使用的例子预测结果精度很差。
- 避免Overfitting的两种方法
  - 先剪枝: 通过提早停止树的构造而对树“剪枝”
    - 难以选择一个合适的阈值
  - 后剪枝: 它由完全生长的树剪去分支。
    - 后剪枝所需的计算比先剪枝多，但是通常可以产生更优的树

可伸缩决策树归纳方法

- **SLIQ(EDBT'96 —Mehta et al.)**
  - 为每个属性创建索引并只将类列表和目前的属性列表放入内存
- **SPRINT(VLDB'96 —J. Shafer et al.)**
  - 构造一个“属性列表”数据结构存放类和RID信息
- **PUBLIC(VLDB'98 —Rastogi& Shim)**
  - 把树分裂和树剪枝集成起来：早点停止树的增长
- **RainForest(VLDB'98 —Gehrke, Ramakrishnan& Ganti)**
  - 把“可伸缩”从决定树质量的标准中分离出来
  - 创建AVC-集(属性－值和类标号)

分类

- 基本概念
- 决策树归纳
- **贝叶斯分类方法**
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

贝叶斯分类

- 贝叶斯分类
  - 统计学习分类方法
  - 预测类成员关系的可能性
- 朴素贝叶斯分类
  - 类条件独立
    - 假定一个属性值对给定类的影响独立于其他属性的值。
- 贝叶斯信念网络
  - 可以表示属性子集间的依赖

贝叶斯定理

- 给定训练数据 X, 假设H的后验概率, P(H | X), 遵循贝叶斯定理

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

- 上述公式可以被非正式地表示为

posteriori = likelihood x prior/evidence

- 预测 X 属于 C_i，当且仅当对于所有的k个类，概率 $P(C_i | X)$ 的值在所有$P(C_k | X)$当中最大
- 实践的困难：需要一些概率的初始化知识，大的计算开销

朴素贝叶斯分类

- 朴素假设: 属性独立
  - $P(x_1,\ldots,x_k \mid C) = P(x_1 \mid C)\cdot\ldots\cdot P(x_k \mid C)$
- 如果第i个属性是分类属性:
  - $P(x_i \mid C)$ 被评估为类C中第i个属性的值为x_i的样本的相对频率
- 如果第i个属性是连续属性:
  - $P(x_i \mid C)$ 通过一个高斯密度函数来评估

朴素贝叶斯分类器

- 令 D 是元组和它们相关类标号的训练数据集合，每一个元组用一个n-D 属性矢量$\mathbf{X} = (x_1, x_2, \ldots, x_n)$表示
- 假定有*m* 个类 $C_1, C_2, \ldots, C_m$.
- 分类就是得到最大后验概率，即, 最大 $P(C_i \mid \mathbf{X})$
- 可以从贝叶斯定理得到 $$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$
- 因为 P(X) 对于所有类是常量, 所以仅有 $P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$

需要被最大化。

一个朴素贝叶斯分类的例子
训练集如下

Class:
C1:buys_computer= ‘yes’
C2:buys_computer= ‘no’

Data sample
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
30…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 0.0.667 =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

Therefore, X belongs to class "buys_computer=yes"

避免零概率值问题

- 朴素贝叶斯分类需要每一个条件概率都必须非零. 否则预测的概率将为零

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

- 例如： 假定数据集合有1000 个元组, income=low (0), income= medium (990), and income = high (10),
- 用拉普拉斯校准（ Laplacian correction） (or 拉普拉斯估值法)
  - *为每一类增加 1 个元组*
    - Prob(income = low) = 1/1003
    - Prob(income = medium) = 991/1003
    - Prob(income = high) = 11/1003
  - 这些校准过的概率估计与对应的未校准的估计很接近，但是避免了零概率值。

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

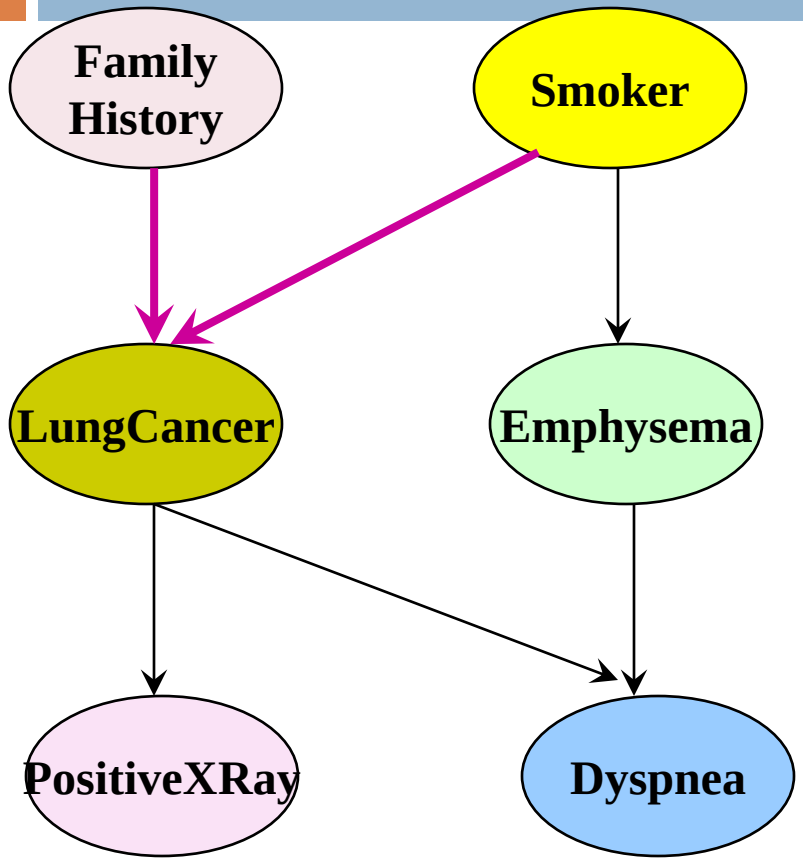
有关独立性假设

- 独立性假设使得朴素贝叶斯分类成为可能
- 当独立性假设满足时生成最优分类器
- 但是实践中很少满足，因为属性（变量）通常时相关的
- 试着克服这些限制：
  - 贝叶斯信念网络，联合属性的贝叶斯推理和因果关系
  - 决策树，在一个时刻只推理一个属性，首先考虑最重要的属性

贝叶斯信念网络

- 贝叶斯信念网络（贝叶斯网络，信念网络，概率网络）
  - 表示一组变量的联合概率分布，它通过一组条件概率来指定一组条件独立性假定
  - 可表述变量的一个子集上的条件独立性假定
    - 比朴素贝叶斯分类在限制条件上更为宽松也更为实用，同时又比在所有变量中计算条件依赖更可行。
  - 提供一种因果关系的图形，可以在其上进行学习，表示变量之间的因果关系
  - 条件独立性假设是贝叶斯网络进行定量推理的理论基础
- 贝叶斯信念网络的组成
  - 有向无环图
  - 条件概率表

贝叶斯信念网络: 一个例子



贝叶斯信念网络

Bayesian Belief Networks

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

变量 LungCancer的条件概率表:
表明了该变量父结点可能组合的条件概率

$$P(z1,\ldots,zn) = \prod_{i=1}^{n} P(z_i \mid Parents(Z_i))$$

贝叶斯信念网络

- 贝叶斯信念网允许变量的子集间定义类条件独立性
- 提供因果关系的图形
- 贝叶斯信念网络的学习
  - 结构学习
    - 网络拓扑结构的学习
  - 参数学习
    - 已知网络结构对网络中每个分量的局部条件概率分布的学习

	网络结构已知	网络结构未知
数据完备	概率参数学习：简单统计估计，MLE方法，贝叶斯方法	找最优网络结构：MDL、BDe等评分标准，启发式搜索、模拟退火搜索等
数据不完备	找最优概率参数：EM算法、基于梯度的方法、高斯算法等	既要找最佳结构，又要找最优参数：有结构EM算法，混合模型等

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

使用 IF-THEN 规则分类

- 用IF-THEN 规则的形式表示知识
  - 例如：IF *age* = youth AND *student* = yes THEN *buys_computer* = yes
  - IF部分称作规则前件或前提；then部分是规则的结论
- 规则的评价: 覆盖率和准确率
  - n_{covers} = 规则R覆盖的元组数
  - $n_{correct}$ = R正确分类的元组数
  - coverage(R) = n_{covers} / |D| /* D: 训练数据集*/
  - accuracy(R) = $n_{correct}$ / n_{covers}

使用 IF-THEN 规则分类

- 如果多个规则被触发，则需要冲突解决策略
  - 规模序（Size ordering）：将最高优先权赋予具有“最苛刻”要求的触发规则(即, 具有最多属性测试的触发规则)
  - 规则序（Rule ordering）：预先确定规则的优先次序
    - 基于规则排序（Rule-based ordering ）(decision list): 根据规则质量的度量（准确率、覆盖率等）或领域专家建议将规则组织成一个优先权列表
    - 基于类排序（Class-based ordering）：类按“重要性”递减排序或误分类代价排序

从决策树中抽取分类规则

- 以**IF-THEN**的形式表示知识
- 每条从根到叶的路径均为一个独立的规则
- 沿着给定路径上的每个属性 - 值对形成规则前件的一个合取项
- 叶节点保留分类预测
- 规则易于为人们所理解
- 例子

IF *age* = "<=30" ***AND*** *student* = "*no*" **THEN** *buys_computer* = "*no*"

IF *age* = "<=30" ***AND*** *student* = "*yes*" **THEN** *buys_computer* = "*yes*"

IF *age* = "31…40" **THEN** *buys_computer* = "*yes*"

IF *age* = ">40" ***AND*** *credit_rating* = "*excellent*" **THEN** *buys_computer* = "*yes*"

IF *age* = "<=30" ***AND*** *credit_rating* = "*fair*" **THEN** *buys_computer* = "*no*"

使用顺序覆盖算法的规则归纳

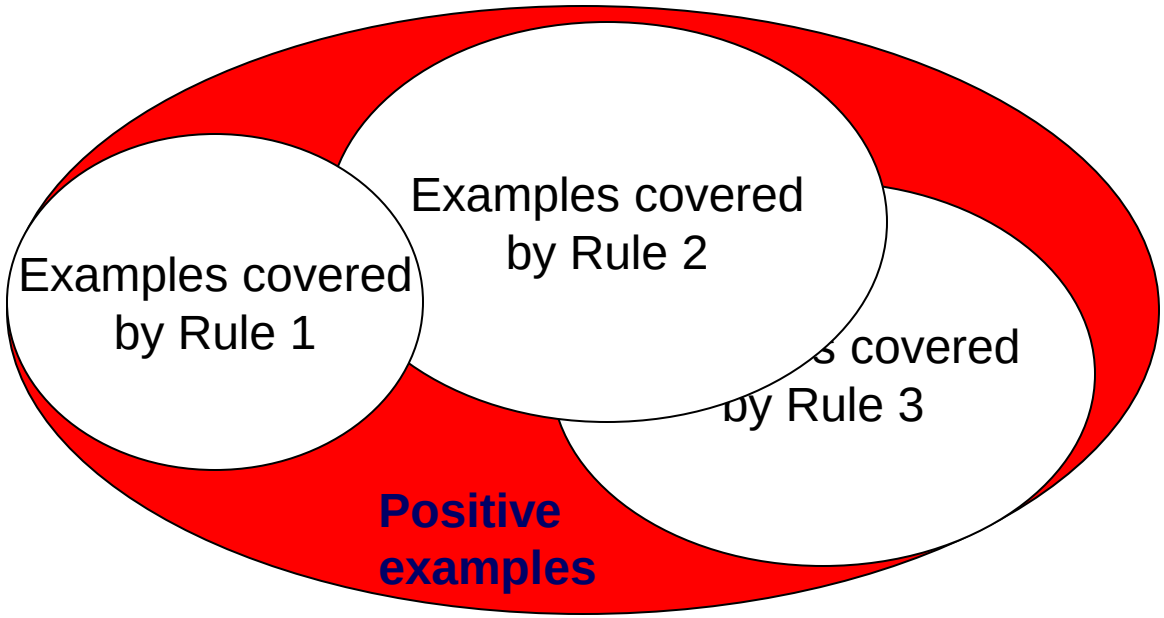
- 顺序覆盖算法：直接从训练数据抽取规则
- 典型的顺序覆盖算法: **FOIL, AQ, CN2, RIPPER**
- 顺序地学习规则，给定类C_i的每个规则理想地覆盖该类的许多元组，并且希望不覆盖其他类的元组
- 步骤:
  - 一次学习一个规则
  - 每当学习一个规则，就删除该规则覆盖的元组
  - 对剩下的元组重复该过程直到满足终止条件，即没有更多的训练样本或者当返回规则的质量低于用户设定的阈值。
- 决策树归纳: 同时学习一组规则

顺序覆盖算法

while (enough target tuples left)

generate a rule

remove positive target tuples satisfying this rule



如何学习规则

- 束搜索（beam search）
  - 从空规则开始，然后逐渐地向它添加属性测试
  - 选择属性的方法—贪心的深度优先策略
    - 每当面临添加一个新的属性测试到当前规则时，它根据训练样本选择最能提高规则质量属性的测试
  - 规则质量度量
    - 熵：对数据集D的元组分类所需要的期望信息
      - 偏向于覆盖单个类大量元组和少量其他类的元组的条件
    - 信息增益：
      - 偏向于具有高准确率并且覆盖许多正元组的规则
    - 考虑覆盖率的统计检验：将规则覆盖的元组的观测类分布与规则做随机预测产生的期望类分布进行比较
      - 有助于识别具有显著覆盖率的规则

规则剪枝

- 使用上述方法产生的规则，可能过分拟合训练数据，而对于测试数据集可能效果没有那么好，为此可以对规则进行适当的剪枝
- **FOIL**使用的剪枝法

$$FOIL\_\mathrm{Pr}une(R) = \frac{pos - neg}{pos + neg}$$

  - 其中**pos**和**neg**分别为规则**R**覆盖的正元组和负元组数。这个值将随**R**在剪枝集上的准确率增加。
  - 如果**R**剪枝后的**FOIL_Prune**值较高，则对**R**剪枝。
  - 剪枝从最近添加的合取项开始，只要剪枝导致改进，则一次减去一个合取项

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

分类的数学模型

- 分类:
  - 预测类标号
- E.g., Personal homepage classification
  - $x_i = (x_1, x_2, x_3, \ldots)$, $y_i = +1$ or -1
  - x_1 : # of word "homepage"
  - x_2 : # of word "welcome"
- Mathematically
  - $x \in X = \Re^n$, $y \in Y = \{+1, -1\}$
  - We want a function f: X → Y

通过后向传播分类

- 后向传播（Backpropagation）：一种神经网络学习方法
- 由心理学家和神经学家开发和测试神经的计算模拟
- 神经网络：连接输入/输出单元的集合，其中每一个连接都有一个权重与之相关联
- 在学习阶段，通过调整权重使得能够正确地预测输入元组的类标号来学习网络。
- 由于单元之间的连接，神经网络学习又称连接者学习（connectionist learning）。

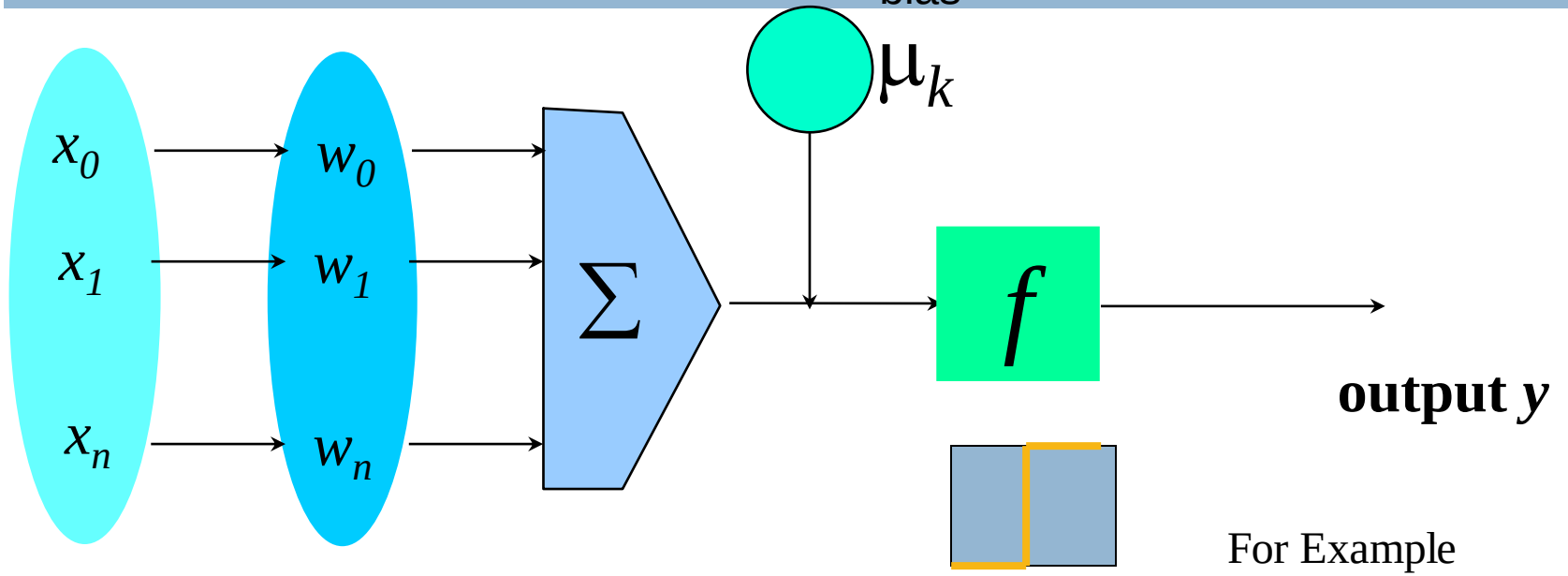
神经网络作为分类器

- 缺点
  - 训练时间过长
  - 需要大量的参数，且这些参数的值大部分依靠经验获得, 如网络拓扑或结构
  - 可解释性差: 很难解释网络中学习的权重和“隐藏单元”的符号含义

神经网络作为分类器

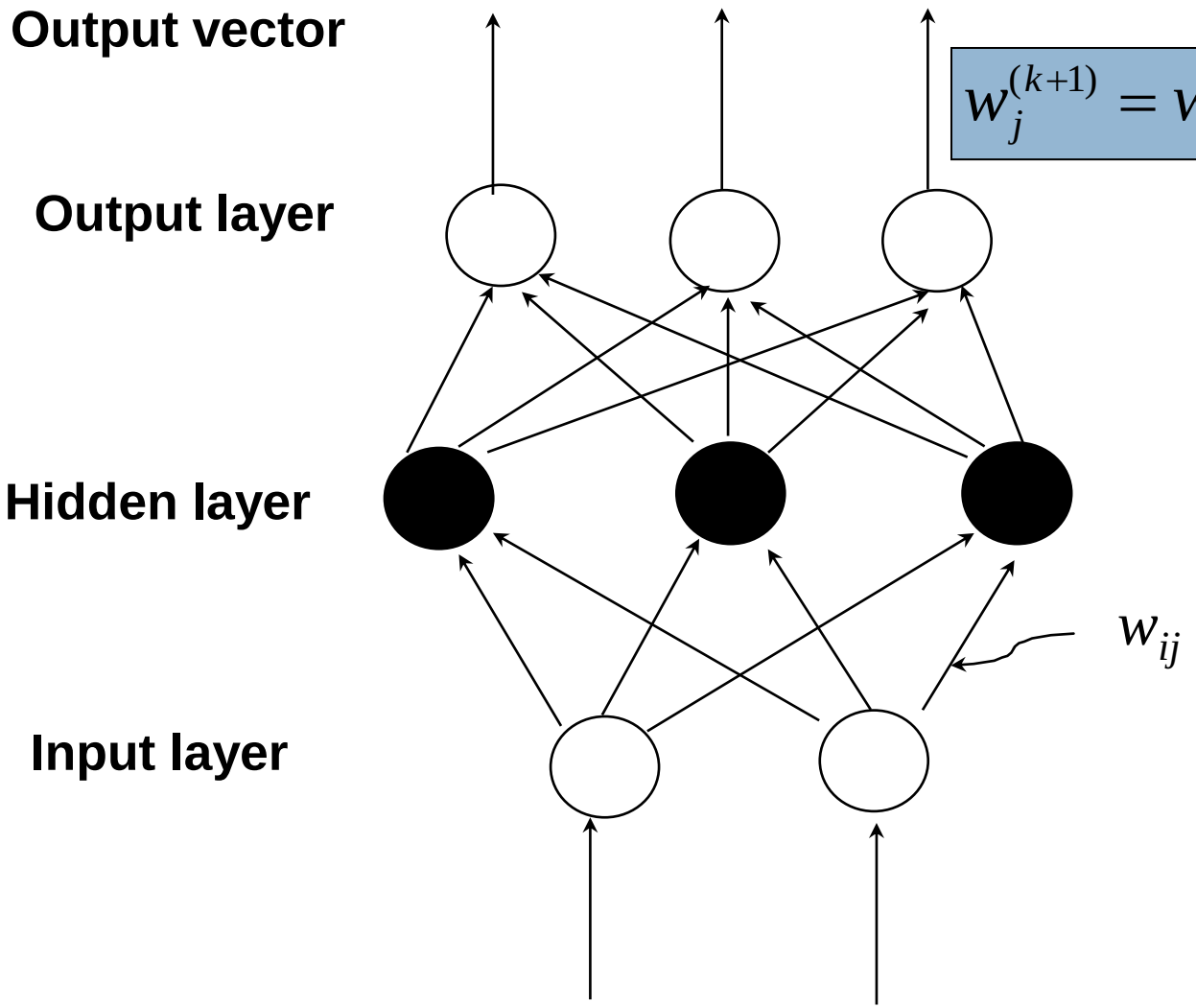
- 优点
  - 对噪音数据的高承受能力
  - 对未经训练的数据的模式分类能力
  - 适合连续值的输入和输出
  - 已经成功地用于广泛的现实世界数据
  - 算法是固有并行的，可使用并行技术加快计算过程
  - 最近已经开发了一些从训练过的神经网络提取规则的技术

神经网络



- The *n*-dimensional input vector **x** is mapped into variable y by means of the scalar product and a nonlinear function mapping

多层前向反馈神经网络



$$w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}$$

多层神经网络是如何工作的

- 神经网络的输入对应于每个训练元组测量的属性
- 输入同时提供给称作“输入层”的单元
- 这些输入通过输入层，然后加权同时提供给称作隐藏层的“类神经元”第二层。
- 隐藏层的数目是任意的，尽管通常只有一层
- 最后一个隐藏层的加权输出作为构成输出层的单元的输入。输出层发布给定给定元组的网络预测。
- 如果其权重都不回送到输入单元，或前一层的输出单元，则该网络是前向反馈的。
- 从统计观点来看，神经网络实现的是非线性回归：给定足够多的隐藏单元和足够多的训练样本，多层前馈网络可以逼近任何函数

定义网络拓扑

- 确定网络拓扑结构: 输入层的单元数, 隐藏层的层数 (if > 1), 每个隐藏层的单元数, 以及输出层的单元数
- 对训练元组中每个属性的测量输入值进行规范化，使之落入 [0.0—1.0]之间
- 每个域值一个输入单元, 每个单元初始化为 0
- 输出： 对于分类，并且超过两个类，每个类一个输出单元
- 一旦网络经过训练且准确率不能接受，通常用不同的网络拓扑或不同的初始权重集合，重复训练过程

后向传播

- 后向传播迭代地处理训练元组数据集，将每个元组的网络预测与实际已知的目标值比较。
- 对每个训练样本，修改权重使网络预测和实际目标值之间均方误差最小。
- 修改“后向”进行: 由输出层，经由每个隐藏层，到第一个隐藏层，因此称作“后向传播”
- 步骤
  - 初始化网络的所有权重（为很小的随机数）和偏倚
  - 向前传播输入(通过激励函数)
  - 向后传播误差(通过更新权重和误差)
  - 终止条件(当错误非常小等)

后向传播和可解释性

- 知识的表示
  - 提取隐藏在训练后的神经网络中的知识，并用符号解释这些知识的研究
- 从网络提取规则: 网络剪枝
  - 通过剪去对训练后的网络影响最小的加权链简化网络结构
  - 一旦训练后的网络已剪枝，某些方法将进行链、单元或活跃值聚类
  - 研究输入值和活跃值的集合，导出描述输入和隐藏单元层联系的规则。
- 灵敏度分析: 用于评估一个给定的输入变量对网络输出的影响。从这种分析得到的知识是形如“IF X 减少5% THEN Y 增加8%”的规则。

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

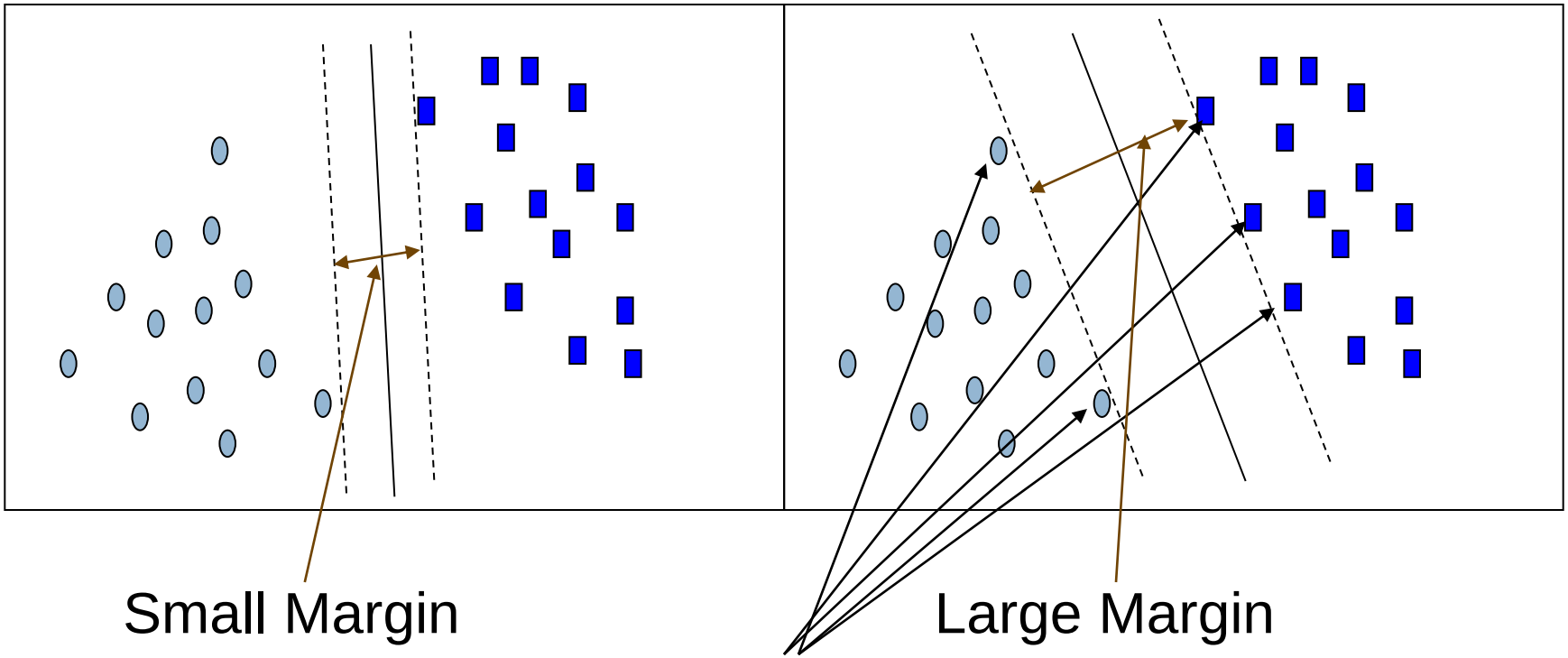
SVM—支持向量机

- 一种线性和非线性数据的有前途的新分类方法
- 它使用一种非线性映射将原有训练数据映射到高维
- 在新的维上，它搜索线性最佳分离超平面（即决策边界）
- 使用一个适当的对足够高维的非线性映射，两类的数据总可以被超平面分开。
- SVM 使用支持向量（“基本”训练元组）和边缘（由支持向量定义）发现该超平面

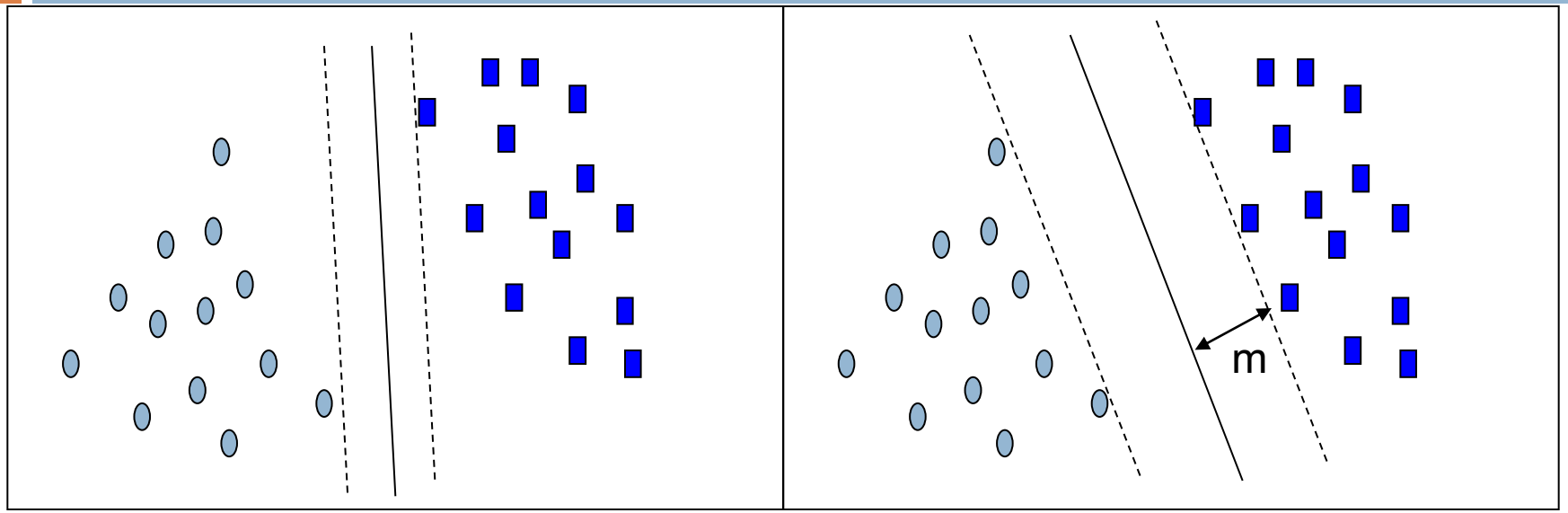
SVM—历史和应用

- Vapnik 和他的同事(1992)—基础理论来自于Vapnik & Chervonenkis'在六十年代关于统计学习理论的研究
- 特征: 虽然训练时间长，但对复杂的非线性决策边界的建模能力是高度准确的，且不容易过分拟合。
- 用于: 分类和数值预测
- 应用:
  - 手写数字识别，对象识别，语音识别，以及基准时间序列预测检验

SVM —边界和支持向量



SVM—当数据是线性可分的



Let data D be ($\mathbf{X}_1$, y_1), …, ($\mathbf{X}_{|D|}$, $y_{|D|}$), where $\mathbf{X}_i$ is the set of training tuples associated with the class labels y_i

There are infinite lines (hyperplanes) separating the two classes but we want to find the best one (the one that minimizes classification error on unseen data)

SVM searches for the hyperplane with the largest margin, i.e., **maximum marginal hyperplane** (MMH)

SVM—线性可分

- 分离超平面可以记作

  $$W \bullet X + b = 0$$

  其中 $W=\{w_1, w_2, \ldots, w_n\}$ 是权重向量，b是一个标量（偏倚）

- 对于2-D 的训练元组，分离超平面可以写作

  $$w_0 + w_1 x_1 + w_2 x_2 = 0$$

- 分离超平面定义边界的两边:

  $$H_1: w_0 + w_1 x_1 + w_2 x_2 \geq 1 \quad \text{for } y_i = +1, \text{ and}$$

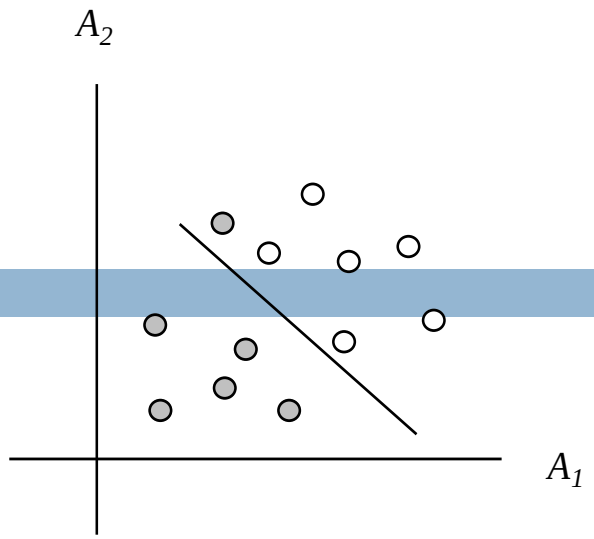
  $$H_2: w_0 + w_1 x_1 + w_2 x_2 \leq -1 \text{ for } y_i = -1$$

- 落在超平面H_1 或 H_2 (即定义边缘的两侧) 上的训练元组称为支持向量

为什么SVM对高维数据有效?

- 学习后的分类器的复杂度由支持向量数而不是由数据的维数刻画
- 支持向量是基本或临界的训练元组-----它们离决策边界(MMH)最近
- 如果删除所有其他元组并重复训练，则将发现相同的分离超平面
- 找到的支持向量数可以用来计算SVM分类器的期望误差率的上界，这独立于数据的维数。
- 具有少量支持向量的SVM可以具有很好的推广性能，即使数据的维度很高时也如此

SVM—线性不可分



- 转换原始输入数据到一个更高维的空间

Example 6.8 Nonlinear transformation of original input data into a higher dimensional space. Consider the following example. A 3D input vector $\mathbf{X} = (x_1, x_2, x_3)$ is mapped into a 6D space Z using the mappings $\phi_1(X) = x_1, \phi_2(X) = x_2, \phi_3(X) = x_3$, $\phi_4(X) = (x_1)^2, \phi_5(X) = x_1x_2$, and $\phi_6(X) = x_1x_3$. A decision hyperplane in the new space is $d(\mathbf{Z}) = \mathbf{WZ} + b$, where $\mathbf{W}$ and $\mathbf{Z}$ are vectors. This is linear. We solve for $\mathbf{W}$ and b and then substitute back so that we see that the linear decision hyperplane in the new ($\mathbf{Z}$) space corresponds to a nonlinear second order polynomial in the original 3-D input space,

$$\begin{aligned} d(Z) &= w_1x_1 + w_2x_2 + w_3x_3 + w_4(x_1)^2 + w_5x_1x_2 + w_6x_1x_3 + b \\ &= w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5 + w_6z_6 + b \end{aligned}$$

■

- 在新的空间搜索一个线性分割超平面

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

惰性学习法 vs. 急切学习法

- 惰性 vs. 急切学习
  - 惰性学习(例如：基于实例的学习): 简单存储训练数据(或只是稍加处理) 并且一直等到给定一个检验元组。
  - 急切学习(前面我们讲过的所有方法): 给定训练元组的集合，在收到新的测试数据进行分类之前先构造一个分类器模型
- 惰性学习: 花费极少的时间用来训练，更多的时间用来预测。
- 准确性
  - 惰性学习法具有丰富的假设空间，因为它用很多的局部线性函数去构造目标函数内在的全局近似
  - 急切学习: 必须用单个假设覆盖所有的实例空间

惰性学习法

- 惰性学习法
  - 存放所有的训练样本,并且直到新的样本需要分类时才建立分类
- 典型的方法
  - **k-**最近邻分类
    - 把训练样本作为欧氏空间的点存放
  - 基于案例的推理
    - 使用符号描述和基于知识的推论

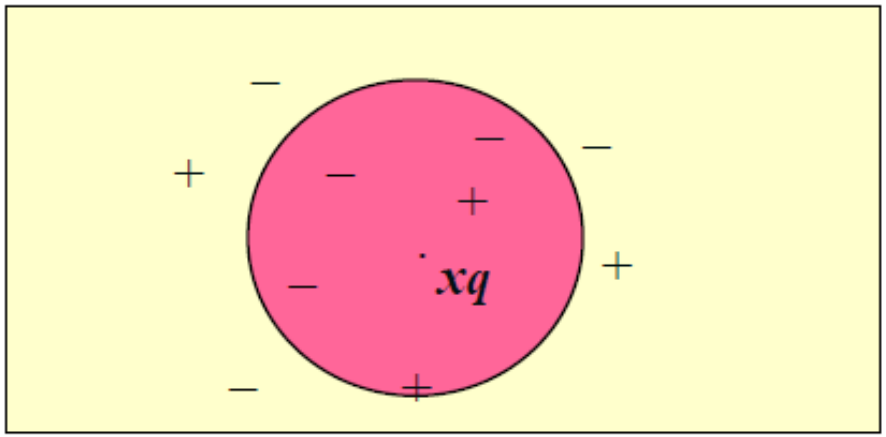
k-最近邻分类

- 基于类比学习
- 所有样本用n维数值属性描述，对应于n维空间的点
- 最近的邻居是用欧几里德距离定义的
  - 两个点$X = (x_1, x_2, \ldots, x_n)$和$Y = (y_1, y_2, \ldots, y_n)$的欧几里德距离定义如下：

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

k-最近邻分类

- K-最近邻分类法搜索模式空间，找出最接近未知样本的k个训练样本。这k个训练样本是未知样本的k个“近邻”
- 未知样本被分类到k个最近邻者中最公共的类。



有关KNN算法的讨论

- 最近邻分类可以用于预测，返回给定未知样本的实数值预测
  - 分类返回未知样本的*k*个最临近者实数值的平均值
- 距离权重的最近邻算法
  - 根据其与查询点x_q之间的距离，给*k*个近邻分配不同的权重

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

  - 给更近的邻居以更大的权重
- 通过求最近邻的平均值可以平滑噪音
- 不相关属性可能会影响最近邻分类法的准确率
  - 为此，需要对最近不相关数据进行剪枝

基于案例的推理(CBR)

- **CBR:** 使用一个问题解的数据库来求解新问题
  - 与**kNN**的相同点：消极评估+ 分析相似实例
  - 与**kNN**的不同点：实例不是“欧氏空间中的点”，而是复杂的符号描述
- 方法
  - 实例可以用丰富的符号描述表示（例如，功能图）
  - 检索到的多个相似案例可以被合并
  - 案例检索、基于知识的推理和问题求解间是紧密耦合在一起的

基于案例的推理(CBR)

- 研究课题
  - 找到好的相似度度量
  - 为索引训练案例，选择显著的特征和开发有效的索引技术
    - 基于句法相似性度量的索引，失败时，回溯搜索另外的实例以适应现有的案例

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- **其它分类方法**
- 数值预测
- 模型评估与选择
- 小结

其它的分类方法

- 遗传算法
- 粗糙集方法
- 模糊集方法

遗传算法

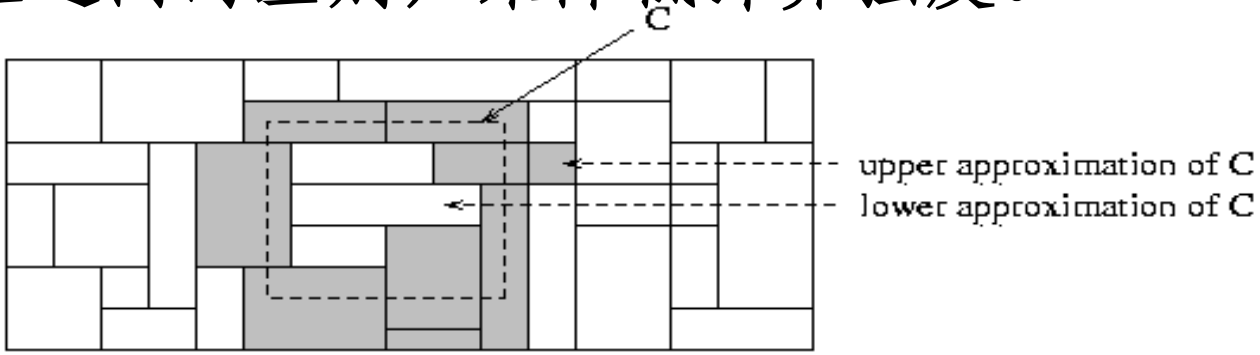
- 遗传算法：基于类似于生物进化的思想
- 遗传学习
  - 创建一个由随机产生的规则组成的初始群体
  - 每个规则用一个二进位串表示
    - 例如： IF $A_1$and Not A_2 then C_2可以用“100”编码
    - 其中最左边的两个二进位分别表示属性A_1和A_2，而最右边的二进位表示类。
    - 如果一个属性具有k（k>2）个值，则可以用k个二进位对该属性值编码。类可用类似的形式编码。

遗传算法

- 根据适者生存的原则,形成由当前群体中最合适的规则组成的新的群体,以及这些规则的后代
  - 规则的拟合度（fitness）用它对训练本集的分类准确率评估
- 通过交叉和变异来产生后代
  - 交叉：来自规则对的子串交换，形成新的规则对
  - 变异：规则串中随机选择的位被反转

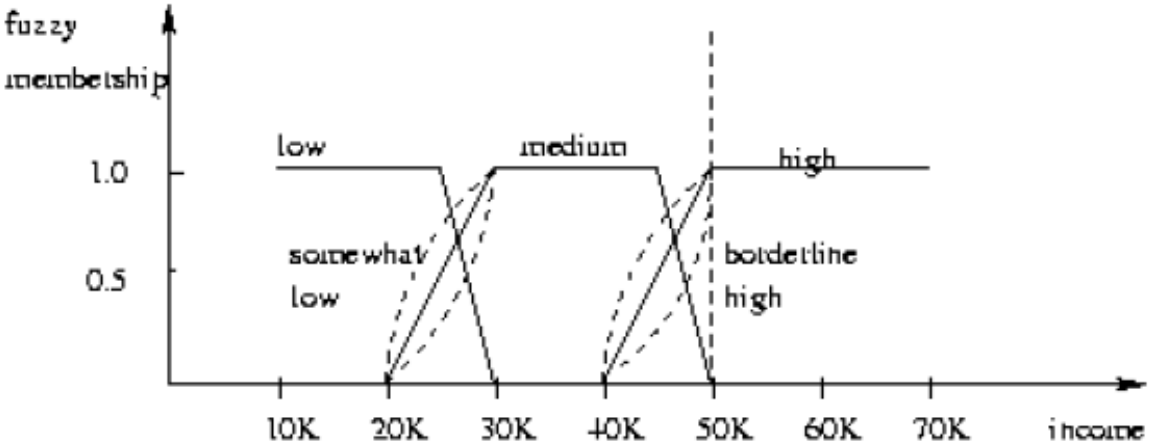
粗糙集方法

- 粗糙集用于近似地或“粗糙地”定义等价类。
- 给定类C的粗糙集通过两个集合近似：C的下近似（一定属于C）和C的上近似（不可能认为不属于C）
- 找到可以描述给定数据集中所有概念的最小属性子集（归约集）问题是NP困难的。但是可以用识别矩阵discernibility matrix （存放每对数据元组的属性值之间的差别）来降低计算强度。



模糊集方法

- 基于规则的分类的缺点
  - 对于连续属性，有陡峭的截断。
    - 例如：**IF (years_employed>=2)and(income>=50k) then credit="approved"**



- 模糊逻辑使用0.0到1.0之间的真值表示一个特定的值是一个给定类成员的隶属程度

模糊集方法

- 模糊逻辑的使用
  - 真值被转换成模糊值
    - **e.g.** 收入被映射到一个离散的分类**{low, medium, high}**，然后使用模糊逻辑，允许对每个类定义“模糊”阈值或边界。
  - 对于给定的新样本，可以使用多个模糊规则。每个可用的规则为分类的成员关系贡献一票。通常，对每个预测分类的真值进行求和，并组合这些和。

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

数值预测

- 数值预测
  - 首先,建立一个模型
  - 其次,使用模型预测未知值
- 数值预测的主要方法是回归
  - 线性回归和多元回归
  - 非线性回归

线性回归

- 线性回归是最简单的回归形式，采用直线建模。双变量回归将一个随机变量Y（称作响应变量）视为另一个随机变量X（称为预测变量）的线性函数，即 $Y = \alpha + \beta X$
  - 其中，Y 的方差为常数，α和β是回归系数，分别表示直线在Y 轴的截断和斜率。
  - 这些系数可以用最小二乘法求解，这使得实际数据与该直线的估计之间误差最小。给定s个样本或形如(x_1,y_1), $(x_2,y_2),...,(x_s,y_s)$的数据点，回归系数α和β可以用下列公式计算。

$$\beta = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s}(x_i - \bar{x})^2} \qquad \alpha = \bar{y} - \beta\bar{x}$$

  - $\bar{x}$是$x_1, x_2, ..., x_s$ 的均值，而 $\bar{y}$是$y_1, y_2, ..., y_s$ 的均值

多元线性回归与非线性回归

- 多元线性回归
  - 是线性回归的扩展，涉及多个预测变量。响应变量**Y** 可以是一个多维特征向量的线性函数。基于两个预测属性或变量X_1 和X_2的多元线性回归模型可以表示为：
    - $Y = \alpha + \beta_1 X_1 + \beta_2 X_2$
      - α，$\beta 1$和$\beta 2$可以用最小二乘法求解
- 非线性回归
  - 在基本线性模型上添加多项式项建模，通过对变量进行变换，将非线性模型转换成线性的，然后用最小二乘方法求解。
    - 例： $y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$
    - 转换成： $y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

评估分类模型

- 准确率
- 速度
  - 构造模型的时间
  - 使用模型的时间
- 鲁棒性
  - 处理噪声和缺失值的能力
- 可伸缩性
  - 涉及给定大量数据，有效地构造模型的能力
- 可解释性
  - 涉及到学习模型提供的理解和洞察的水平

分类器评估度量: 准确率和误差率

混淆矩阵:

Actual class\Predicted class	C_1	$\sim C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\sim C_1$	**False Positives (FP)**	**True Negatives (TN)**

分类器的正确率或识别率：被正确分类的测试元组占测试元组总数的百分比

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

错误率: *1 – accuracy*, or

$$error\ rate = \frac{FP + FN}{TP + TN + FP + FN}$$

分类器评估度量: 混淆矩阵的例子

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total	Recognition(%)
buy_computer = yes	**6954**	**46**	7000	99.34
buy_computer = no	**412**	**2588**	3000	86.27
Total	7366	2634	10000	95.42

- *给定 m 个类, 混淆矩阵至少是m*m的表，* $CM_{i,j}$ *表示类i用分类器分到类j的元组数。*
- 附加的行或列，可以提供每个类的合计或识别率

分类器评估度量: 灵敏性和特效性

- 类不平衡问题:
  - 某一个类可能比较稀少,比如: 诈骗检测
  - 显著多数的负类和少数的正类
- 灵敏性: 真正（识别）率,

$$sensitivity = \frac{TP}{P}$$

特效性: 真负（识别）率,

$$specificity = \frac{TN}{N}$$

分类器评估度量: 准确率和召回率

- **Precision:** 正确性 – what % of tuples that the classifier labeled as positive are actually positive?

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

- **Recall:** 完整性 – what % of positive tuples did the classifier label as positive?

$$recall = \frac{TP}{TP + FN}$$

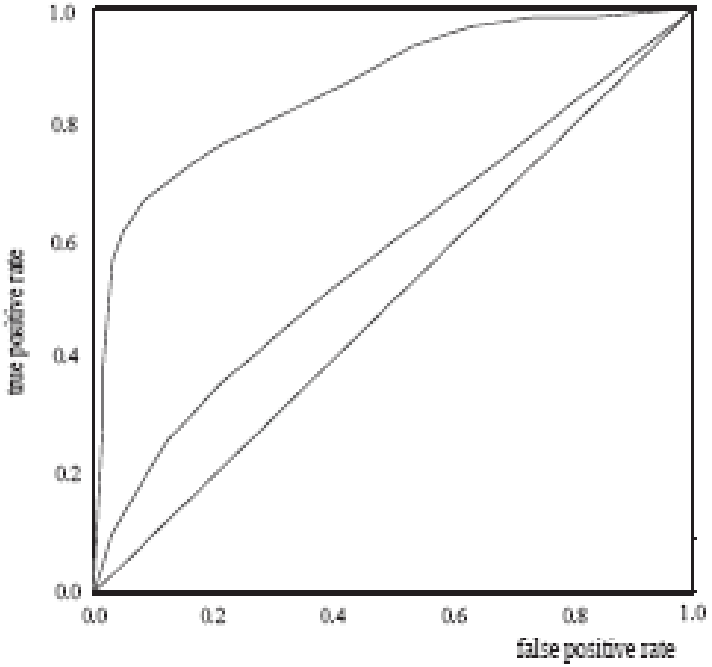
- Perfect score is 1.0
- precision & recall之间是相反的关系

分类器评估度量：准确率和召回率

- 调和值
  - Precision和Recall经常一起使用，用固定的Precision比较Recall，或者用固定的Recall比较Precision，也可以把它们组合成一个度量使用。
  - F度量： $F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$
  - F_β 度量： $F_\beta = \frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$

模型选择: ROC 曲线

- **ROC** (Receiver Operating Characteristics) curves: for visual comparison of classification models
- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve is a measure of the accuracy of the model
- Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model



- Vertical axis represents the true positive rate
- Horizontal axis rep. the false positive rate
- The plot also shows a diagonal line
- A model with perfect accuracy will have an area of 1.0

分类器评估度量: 例子

Actual class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 *sensitivity*
cancer = no	**140**	**9560**	9700	98.56 *specificity*
Total	230	9770	10000	96.40 *accuracy*

Precision = 90/230 = 39.13%; *Recall* = 90/300 = 30.00%

评估分类法的准确率

- 保持(holdout)
  - 使用两个独立的数据集,例如,训练集(2/3),测试集(1/3)
  - 估计是悲观的，因为只用一部分初始数据导出模型
  - 用于具有大数量样本的数据集
  - 随机子抽样（random subsampling）：保持方法的变形，把保持方法重复k次。
- 交叉确认（cross-validation）
  - 把数据集分成k个互不相交的子样本集
  - 训练和检验进行k次
  - 使用k-1子样本集作为训练数据,一个子样本作为测试数据---k-折交叉确认
  - 适用于具有中等大小的数据集

评估分类法的准确率

- 自助（bootstrapping）和留一（leave-one-out）
  - “自助”使用一致的、带放回的选样，选取给定的训练实例
  - “留一”为k-折交叉确认的特殊情况，其中k设置为初始元组数，每次只给检验集留出一个样本。
  - 适用于小数量的数据

分类

- 基本概念
- 决策树归纳
- 贝叶斯分类方法
- 贝叶斯信念网络
- 基于规则的分类
- 用后向传播分类
- 支持向量机
- 惰性学习法
- 其它分类方法
- 数值预测
- 模型评估与选择
- 小结

小结

- 分类和数值预测是两种数据分析形式，可以用来提取模型，描述重要数据类或预测未来的数据趋势。
- 分类预测分类标号，数值预测建立连续值函数模型。
- 分类方法
  - 决策树分类
  - 贝叶斯分类
  - 后向传播分类
  - 支持向量机
  - 惰性学习法
  - 。。。
- 分类的评估方法

参考文献

- C. Apteand S. Weiss. Data mining with decision trees and decision rules. Future Generation Computer Systems, 13, 1997.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth International Group, 1984.
- P. K. Chan and S. J. Stolfo. Learning arbiter and combiner trees from partitioned data for scaling machine learning. In Proc. 1st Int. Conf. Knowledge Discovery and Data Mining (KDD'95), pages 39-44, Montreal, Canada, August 1995.
- U. M. Fayyad. Branching on attribute values in decision tree generation. In Proc. 1994 AAAI Conf., pages 601-606, AAAI Press, 1994.
- J. Gehrke, R. Ramakrishnan, and V. Ganti. Rainforest: A framework for fast decision tree construction of large datasets. In Proc. 1998 Int. Conf. Very Large Data Bases, pages 416-427, New York, NY, August 1998.
- M. Kamber, L. Winstone, W. Gong, S. Cheng, and J. Han. Generalization and decision tree induction: Efficient classifcation in data mining. In Proc. 1997 Int. Workshop Research Issues on Data Engineering (RIDE'97), pages 111-120, Birmingham, England, April 1997.

习题一

- 设**status**是类标号属性，给定一个数据元组，他在属性**department,age**和**salary**的值分别为“**Systems**”，“**26…30**”和“**46K…50K**”。该元组**status**的朴素贝叶斯分类是什么？

department	status	age	salary
Sales	Senior	31…35	46K…50K
Sales	Junior	26…30	26K…30K
Sales	Junior	31…35	31K…35K
Systems	Junior	21…25	46K…50K
Systems	Senior	31…35	66K…70K
Systems	Junior	26…30	46K…50K
Systems	Senior	41…45	66K…70K
Marketing	Senior	36…40	46K…50K
Marketing	Junior	31…35	41K…45K
Secretary	Senior	46…50	36K…40K
Secretary	Junior	26…30	26K…30K

习题二

- 右表给出学生的期中和期末考试成绩。
  - 绘数据图。X和Y看上去具有线性关系么?
  - 使用最小二乘法，求由学生的期中成绩预测学生的期末成绩的方程式。
  - 预测其中成绩为86分的学生的期末成绩。

X（期中考试）	Y（期末考试）
72	84
50	63
81	77
74	78
94	90
86	75
59	49
83	79
65	77
33	52
88	74
81	90

思考题

- 比较急切分类（如决策树、贝叶斯、神经网络）相对于惰性分类（如k最近邻、基于案例推理）的优缺点。

END