人工智能

目录

CONTENTS

1.1 回归分析

- 回归分析（regression analysis)
  - 是确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法
- 为什么使用回归分析：从实际应用上来看
  - 更好地了解：更好地了解该现象并有可能基于对该现象的了解来影响政策的制定以及决定采取何种相应措施。例：了解某些特定濒危鸟类的主要栖息地特征（例如：降水、食物源、植被、天敌），以协助通过立法来保护该物种。

1.1 回归分析

- 为什么使用回归分析：从实际应用上来看（续）
  - 建模预测：对某种现象建模以预测其他地点或其他时间的数值，例：如果已知人口增长情况和典型的天气状况，预计明年的用电量将会是多少?
  - 探索检验：假设根据以往数据探索即将发生事件，例：公安部门对城市各个住宅区的犯罪活动进行建模，以更好地了解犯罪活动并希望实施可能阻止犯罪活动的策略。

1.1 回归分析

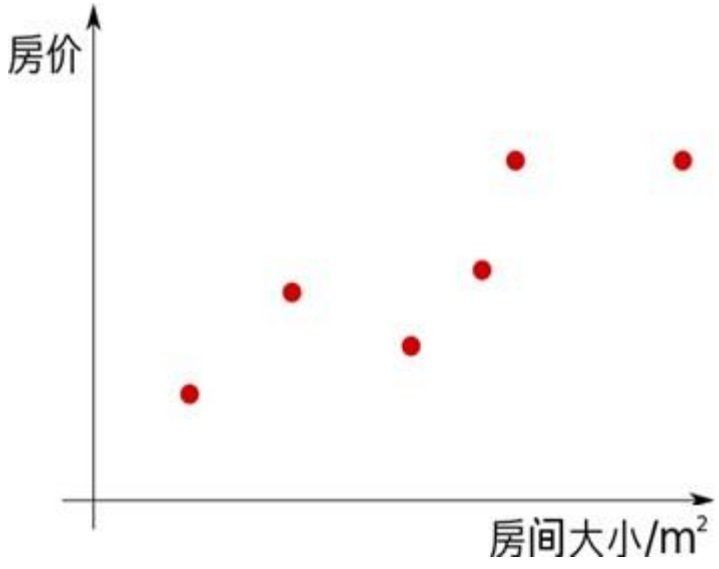
- 为什么使用回归分析：从算法功能上来看
  - 挑选与因变量相关的自变量；
  - 描述因变量与自变量之间的关系强度；
  - 生成模型，通过自变量来预测因变量；
  - 根据模型，通过因变量，来控制自变量。
- 回归的三个度量：
  - 自变量的个数，因变量的类型以及回归线的形状

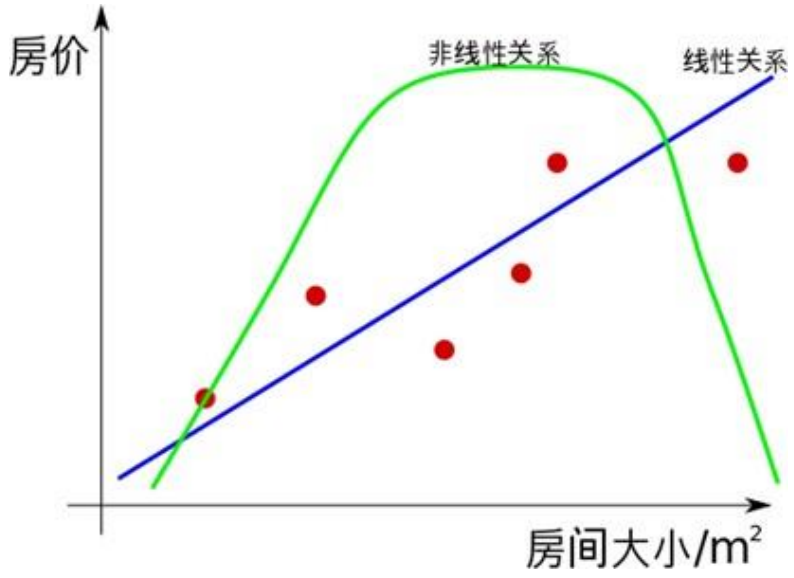
1.1 回归分析

- 回归分析分类
  - 按照自变量的个数：一元回归和多元回归分析；
  - 按照自变量和因变量之间的关系类型：线性回归分析和非线性回归分析；
  - 按照回归线形状：线性回归和非线性回归等。
- 常见回归的模型
  - 线性回归
  - 逻辑回归
  - softmax回归

1.2 线性回归

- 回归分析常用于分析两个变量X和Y 之间的关系。 比如 X = 房子大小 和 Y = 房价 之间的关系， X=(公园人流量，公园门票票价) 与 Y=(公园收入) 之间的关系等等





1.2 线性回归

- 线性回归的特点
  - 因变量是连续的，自变量（单个或多个）可以是连续的也可以是离散的，回归线的性质是线性的。
  - 线性回归使用最佳的拟合直线（回归模型），建立因变量（Y）和一个或多个自变量 (X) 之间的联系。即：

$$Y=a+b*X + e$$

注：a 表示截距，b 表示直线的倾斜率，e 是误差项。

1.2 线性回归

- 回归过程
  - 已知N组数据，数据的特征描述为X，用X1,X2，... ,Xj 去描述特征值里面分量，假设这些数据分布特点成线性：

估计值：$Yi' = a+b*X$

真实值：$Yi = a+b*X + e$

误差项：$e = Yi - Yi' = Y-(a+b*X)$

求得最优a、b值，即：使误差项 e 的平方和最小（最小二乘法）

1.2 线性回归

- 最小二乘法 — 确定回归系数
  - 误差平方和 = $\sum_{i=1}^{N}\left[y_i-(a+bx_i)\right]^2$

$$\frac{\partial}{\partial a}\sum_{i=1}^{N}\left[y_i-(a+bx_i)\right]^2=-2\sum_{i=1}^{N}(y_i-a-bx_i)=0,$$

$$\frac{\partial}{\partial b}\sum_{i=1}^{N}[y_i-(a+bx_i)]^2=-2\sum_{i=1}^{N}[y_i-(a+bx_i)]x_i=0$$

$$\hat{a}=\frac{(\sum x_i^2)(\sum y_i)-(\sum x_i)(\sum x_i y_i)}{N(\sum x_i^2)-(\sum x_i)^2}$$

$$\hat{b}=\frac{N(\sum x_i y_i)-(\sum x_i)(\sum y_i)}{N(\sum x_i^2)-(\sum x_i)^2}$$

明确了：

◆ **Yi'= a+b*X**

实现了：

◆ 可以根据**Xi**预测**Yi**

◆ 可以根据**Yi**控制**Xi**

1.2 线性回归

- 最小二乘法 – 确定相关系数r

$$r = \frac{\sum(x-\overline{x})(y-\overline{y})}{\sqrt{\sum(x-\overline{x})^2 \sum(y-\overline{y})^2}}$$

  - 当|r|=1时，表示两变量为完全线性相关
  - 当r=0时，表示两变量间无线性相关关系
  - 当0<|r|<1时，|r|越接近1，两变量间线性关系越密切；|r|越接近于0，两变量的线性相关越弱
- 可以判断自变量与因变量的相关性，可以挑选最相关的自变量

1.2 线性回归

回归分析步骤：

- 判断并构造预测函数/回归模型（Y’）
- 构造损失函数（误差e）
- 使损失函数最小，最小二乘法获得回归系数（a,b）
- 分析相关参数及结果（r/分类结果）

1.2 线性回归

- *Liner_ regression.example*

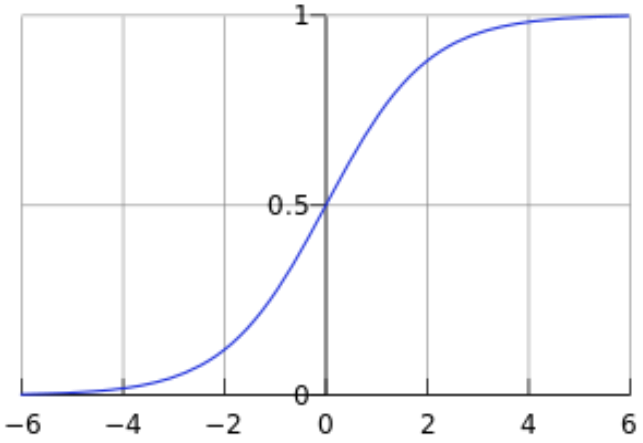
1.3 逻辑回归

- 逻辑回归的特点
  - 用来计算“事件=Success”和“事件=Failure”的概率。当因变量的类型属于二元（1 / 0，真/假，是/否）变量时，则使用逻辑回归。
- 逻辑回归适用的问题
  - 二分类问题
- 思考：
  - 多分类问题是否能采用逻辑回归?

1.3 逻辑回归

- 逻辑回归实现过程：
  - 构造预测函数/回归模型

Logistic 回归模型中的因变量的只有 1-0（如是和否、发生和不发生）两种取值。假设在 p 个独立自变量 $x_1, x_2, \cdots, x_p$ 作用下，记 y 取 1 的概率是 $p = P(y = 1|X)$，取 0 概率是 $1-p$，取 1 和取 0 的概率之比为 $\frac{p}{1-p}$，称为事件的优势比（odds），对 odds 取自然对数即得 Logistic 变换 $\text{Logit}(p) = \ln\left(\frac{p}{1-p}\right)$。令 $\text{Logit}(p) = \ln\left(\frac{p}{1-p}\right) = z$，则 $p = \frac{1}{1+e^{-z}}$ 即为 Logistic 函数



1.3 逻辑回归

- 逻辑回归实现过程：
  - 构造预测函数/回归模型

    边界函数　　输入x分类结果为类别1和类别0的概率

$$\theta_0+\theta_1x_1+,\ldots,+\theta_nx_n=\sum_{i=1}^{n}\theta_ix_i=\theta^Tx$$

$$P(y=1\mid x;\theta)=h_\theta(x)$$

$$h_\theta(x)=g(\theta^Tx)=\frac{1}{1+e^{-\theta^Tx}}$$

$$P(y=0\mid x;\theta)=1-h_\theta(x)$$

1.3 逻辑回归

- 逻辑回归实现过程：
  - 构造损失函数

$$P(y \mid x;\theta) = (h_\theta(x))^y (1-h_\theta(x))^{1-y}$$

$$L(\theta) = \prod_{i=1}^{m} P(y_i \mid x_i;\theta) = \prod_{i=1}^{m} (h_\theta(x_i))^{y_i} (1-h_\theta(x_i))^{1-y_i}$$

$$l(\theta) = \log L(\theta) = \sum_{i=1}^{m} \left(y_i \log h_\theta(x_i) + (1-y_i) \log(1-h_\theta(x_i)) \right)$$

1.3 逻辑回归

假设我们有 m 个独立的训练样本$(x^1,y^1)...(x^j,y^j)...(x^m,y^m)$，$y^j=\{0,1\}$，那每一个观察到的样本$(x^j,y^j)$出现的概率是：

$$P = h_\theta(x^j)^{y^j} * (1-h_\theta(x^j))^{1-y^j}$$

当然，这个概率公式是有巧妙之处的，正是因为$y^j=\{0,1\}$，所以当$y^j=1$时，P=$h_\theta(x^j)$，而$h_\theta(x^j)$正是$y^j=1$的概率；当$y^j=0$时，$P=1-h_\theta(x^j)$。于是似然函数为：

$$L(\theta)=\prod_{j=1}^{m} P=\prod_{j=1}^{m} h_\theta(x^j)^{y^j} * (1-h_\theta(x^j))^{1-y^j}$$

在逻辑回归算法中，采用了对数似然函数，只需要将上面的函数取对数即可：

$$J(\theta)=\sum_{j=1}^{m} y^j * \log(h_\theta(x^j)) + (1-y^j) * \log(1-(h_\theta(x^j))$$

1.3 逻辑回归

- 逻辑回归实现过程：
  - 使损失函数最小，获得回归系数（按照求导数思想）

我们先对J(θ)求导，θ是一个向量，我们分别对每一维的θ进行求导：

$$\frac{\partial J(\theta)}{\partial \theta_i} = \sum_{j=1}^{m} ((h_\theta(x^j) - y^j)x_i^j$$

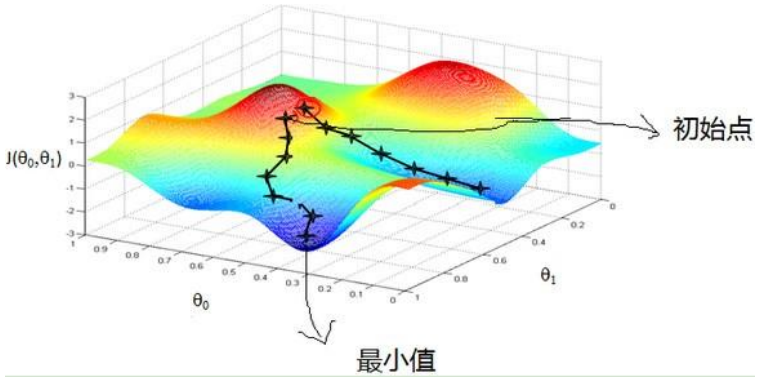
（m 表示数据集个数，n 表示特征个数，x_i^j表示第几个特征，j 表示第几组数据）

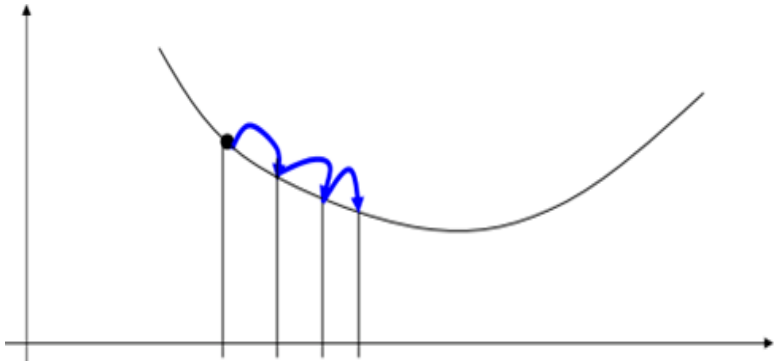
但此时发现$\frac{\partial J(\theta)}{\partial \theta_i} = 0$是无法求解的，所以按照似然估计求解θ行不通，这里就通过**梯度下降法**求解。

为什么无法求解？

1.3 逻辑回归

- 逻辑回归实现过程：
  - 使损失函数最小，获得回归系数（梯度下降法）





1.3 逻辑回归

- 逻辑回归实现过程：
  - 使损失函数最小，获得回归系数（梯度下降法）

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update θ_j for $j = 0, \ldots, n$)

}

代表第i个训练记录的第j个特征

举例

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

1.3 逻辑回归

举一个非常简单的例子，如求函数 $f(x)=x^2$ 的最小值。

利用梯度下降的方法解题步骤如下：

1、求梯度， $\nabla = 2x$

2、向梯度相反的方向移动 x ，如下

$x \leftarrow x-\gamma \cdot \nabla$ ，其中， γ 为步长。如果步长足够小，则可以保证每一次迭代都在减小，但可能导致收敛太慢，如果步长太大，则不能保证每一次迭代都减少，也不能保证收敛。

3、循环迭代步骤2，直到 x 的值变化到使得 $f(x)$ 在两次迭代之间的差值足够小，比如0.00000001，也就是说，直到两次迭代计算出来的 $f(x)$ 基本没有变化，则说明此时 $f(x)$ 已经达到局部最小值了。

4、此时，输出 x ，这个 x 就是使得函数 $f(x)$ 最小时的 x 的取值 。

1.3 逻辑回归 – 关键问题导读

P的取值可能在0-1之间，不一定恰好等于0或者1，那怎么划分类？决策边界如何进行设置？

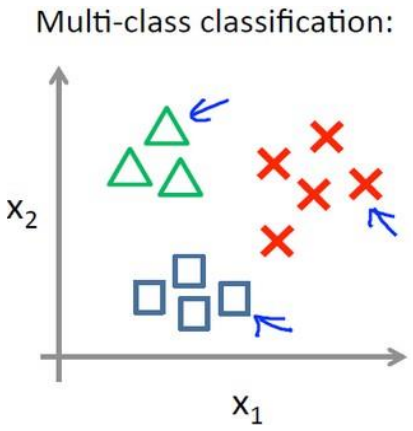
解决方案：

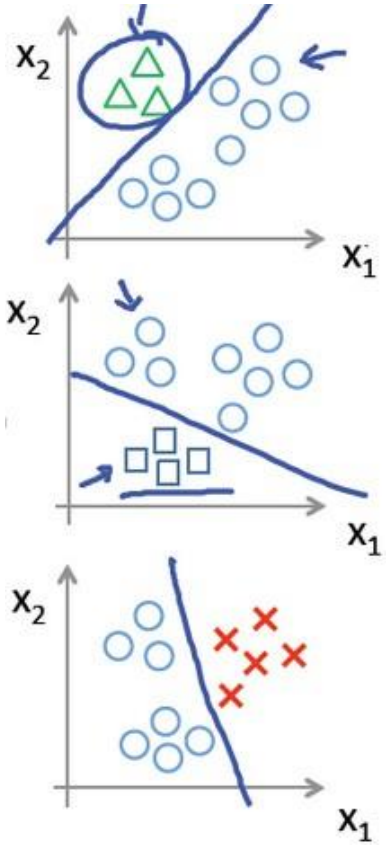
如果分类器用的是回归模型，并且已经训练好了一个模型，可以设置一个阈值：

- 如果$h\theta(x)\geq 0.5$，则预测y=1,既y属于正例；
- 如果$h\theta(x)<0.5$，则预测y=0,既y属于负例；

1.3 逻辑回归 -- 思考

逻辑回归实现多分类：





$$\max_i h_\theta^{(i)}(x)$$

1.4 softmax回归

- softmax回归的特点：
  - 该模型是逻辑回归模型在多分类问题上的推广，在多分类问题中，类标签y可以取两个以上的值，在逻辑回归中，样本数据的y值为{0，1}，而在softmax回归中，样本的y值为{1，k}。

- softmax回归适用的问题
  - 多分类问题

1.4 softmax回归

- softmax回归实现过程：
  - 构造预测函数/回归模型

注：

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1|x^{(i)};\theta) \\ p(y^{(i)}=2|x^{(i)};\theta) \\ \vdots \\ p(y^{(i)}=k|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

$\theta_1, \theta_2, \ldots, \theta_k \in \Re^{n+1}$ 是模型的参数

$\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}}$ 这一项对概率分布进行归一化，使得所有概率之和为 1

1.4 softmax回归

- softmax回归实现过程：
  - 构造损失函数

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\left\{y^{(i)} = j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right]$$

- 注：

$1\{\cdot\}$ 是示性函数

$1\{\text{值为真的表达式}\} = 1$

$1\{\text{值为假的表达式}\} = 0.$

1.4 softmax回归

- softmax回归实现过程：
  - 使损失函数最小，获得回归系数（梯度下降法）

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}\left(1\{y^{(i)}=j\} - p(y^{(i)}=j|x^{(i)};\theta)\right)\right]$$

$$\theta_j := \theta_j - \alpha\nabla_{\theta_j} J(\theta)(j=1,\ldots,k)$$

Softmax 回归 vs k 个二元分类器

当做一个k分类的应用时，选用Softmax分类还是k个独立的二元分类器?

解决方案：

取决于类别之间是否互斥

例如：对人声音乐、舞曲、影视原声和流行歌曲分类，这些类别之间并不是互斥的。一首歌曲可以来源于影视原声，同时也包含人声。这种情况下，使用4个二分类的logistic 回归分类器更为合适

目录

CONTENTS

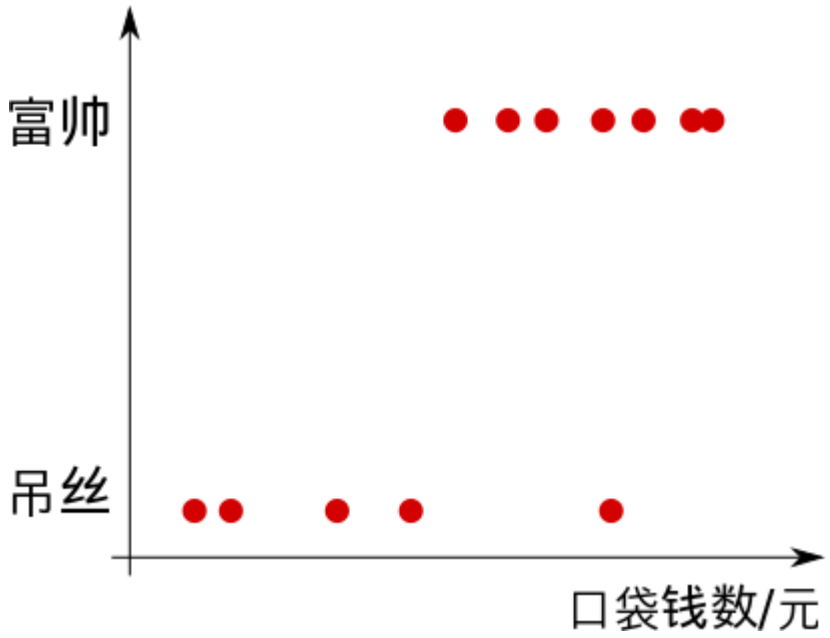
2.1 分类

分类：监督学习，将一些新的数据项映射到给定类别中的某个类别中

动物种类	体型	翅膀数量	脚的只数	是否产蛋	是否有毛	类别
狗	中	0	4	否	是	哺乳动物
猪	大	0	4	否	是	哺乳动物
牛	大	0	4	否	是	哺乳动物
麻雀	小	2	2	是	是	鸟类
天鹅	中	2	2	是	是	鸟类
大雁	中	2	2	是	是	鸟类
动物A	大	0	2	是	无	?
动物B	中	2	2	否	是	?

2.1 分类

- 分类问题也是一类很常见的问题。 比如说，怎么判定一个人是高富帅还是吊丝?



2.1 分类

- 实现分类步骤
  - 将样本转化为等维的数据特征（特征转化）
  - 选择与类别相关的特征（特征选择/提取）
  - 建立分类模型或分类器进行分类（分类）

$$f\left(x_{i1}, x_{i2}, x_{i3}, \ldots\ldots, x_{in}\right) \rightarrow y_i$$

特征转化

39, State-gov, 77516, Bachelors, 13, Never-married, Adm-clerical, Not-in-family, White, Male, 2174, 0, 40, United-States, <=50K
50, Self-emp-not-inc, 83311, Bachelors, 13, Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 13, United-States, <=50K
38, Private, 215646, HS-grad, 9, Divorced, Handlers-cleaners, Not-in-family, White, Male, 0, 0, 40, United-States, <=50K
53, Private, 234721, 11th, 7, Married-civ-spouse, Handlers-cleaners, Husband, Black, Male, 0, 0, 40, United-States, <=50K
28, Private, 338409, Bachelors, 13, Married-civ-spouse, Prof-specialty, Wife, Black, Female, 0, 0, 40, Cuba, <=50K
37, Private, 284582, Masters, 14, Married-civ-spouse, Exec-managerial, Wife, White, Female, 0, 0, 40, United-States, <=50K
49, Private, 160187, 9th, 5, Married-spouse-absent, Other-service, Not-in-family, Black, Female, 0, 0, 16, Jamaica, <=50K
52, Self-emp-not-inc, 209642, HS-grad, 9, Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 45, United-States, >50K
31, Private, 45781, Masters, 14, Never-married, Prof-specialty, Not-in-family, White, Female, 14084, 0, 50, United-States, >50K
42, Private, 159449, Bachelors, 13, Married-civ-spouse, Exec-managerial, Husband, White, Male, 5178, 0, 40, United-States, >50K
37, Private, 280464, Some-college, 10, Married-civ-spouse, Exec-managerial, Husband, Black, Male, 0, 0, 80, United-States, >50K
30, State-gov, 141297, Bachelors, 13, Married-civ-spouse, Prof-specialty, Husband, Asian-Pac-Islander, Male, 0, 0, 40, India, >50K
23, Private, 122272, Bachelors, 13, Never-married, Adm-clerical, Own-child, White, Female, 0, 0, 30, United-States, <=50K
32, Private, 205019, Assoc-acdm, 12, Never-married, Sales, Not-in-family, Black, Male, 0, 0, 50, United-States, <=50K
40, Private, 121772, Assoc-voc, 11, Married-civ-spouse, Craft-repair, Husband, Asian-Pac-Islander, Male, 0, 0, 40, ?, >50K
34, Private, 245487, 7th-8th, 4, Married-civ-spouse, Transport-moving, Husband, Amer-Indian-Eskimo, Male, 0, 0, 45, Mexico, <=50K
25, Self-emp-not-inc, 176756, HS-grad, 9, Never-married, Farming-fishing, Own-child, White, Male, 0, 0, 35, United-States, <=50K
32, Private, 186824, HS-grad, 9, Never-married, Machine-op-inspct, Unmarried, White, Male, 0, 0, 40, United-States, <=50K
38, Private, 28887, 11th, 7, Married-civ-spouse, Sales, Husband, White, Male, 0, 0, 50, United-States, <=50K
43, Self-emp-not-inc, 292175, Masters, 14, Divorced, Exec-managerial, Unmarried, White, Female, 0, 0, 45, United-States, >50K
40, Private, 193524, Doctorate, 16, Married-civ-spouse, Prof-specialty, Husband, White, Male, 0, 0, 60, United-States, >50K

如何转化为机器识别的数据？

特征转化

转化为机器识别的数据

1.Categorical → Integer

- 编码（二进制，十进制）
- 概率密度

2.数据转化为无纲量数据

- 数据归一化:x{0,1}

归一化方法	计算公式
Minimum Maximum 归一化（本文简称 Maxmin）	$f_1(x_i)=\frac{x_i-\min(x)}{\max(x)-\min(x)}$ $\min(x)$ 和 $\max(x)$ 分别为 x 可取值中的最小值和最大值
Statistical 归一化	$f_2(x_i)=\frac{x_i-\mu}{\sigma}$ μ 的取值为所有 x 取值的平均值，σ 为 x 的标准差
Decimal 归一化	$f_3(x_i)=\frac{x_i}{10^e}$ e 的取值为能使 x 中的最大值经过处理后处于 $[0,1]$ 的最小值

特征选择/提取

- 选择与分类相关的特征，提升分类效果，提高分类效率：
  - 初步观察法
  - 计算相关系数
  - 计算互信息
  - 降维

2.2 KNN

- KNN（k近邻分类）建模思想
  - 已知样本集中每一数据与所属分类的对应关系，输入没有标签的新数据后，将新数据的每个特征与样本集中的数据对应的特征进行比较，提取样本集中特征最相似的数据（最近邻）的分类标签。一般来说，我们只选择样本集中前k个最相似的数据，这就是k-近邻算法中k的出处，通常k是不大于20的整数，最后，选择k个最相似的数据中出现次数最多的分类，作为新数据的分类。
- 建模关键
  - 训练集、距离或相似性的衡量、k的大小

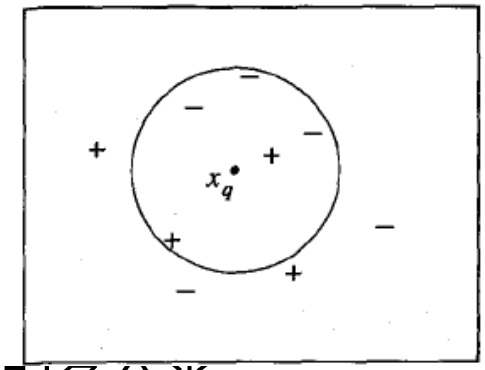
2.2 KNN

分类步骤：给定一个要分类的查询实例 x_q

- 算距离：给定测试对象，计算它与训练集中的每个对象的距离：

$$D(X_i, X_j) = \sqrt{\sum_{k=1}^{d} (X_{ik} - X_{jk})^2}$$

- 找邻居：圈定距离最近的k个训练对象
  作为测试对象的近邻
- 做分类：
  根据这k个近邻归属的主要类别，来对测试对象分类



2.2 KNN

- 适用领域
  - 客户流失预测
  - 欺诈侦测等
  - 更适合于稀有事件的分类问题

$$\hat{f}(x_q) \leftarrow \arg\max_{y} \sum_{i=1}^{k} \delta(y, f(x_i)) \quad \text{其中，} \quad \delta(a,b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases}$$

2.3 Bayes

- Bayes（贝叶斯）建模思想：
  - 贝叶斯分类器的分类原理是通过某对象的先验概率，利用贝叶斯定理计算出其后验概率，即该对象属于某一类的概率，选择具有最大后验概率的类作为该对象所属的类。
- 贝叶斯定理：
  - 由于P(X)对于所有类为常数，只需要P(X|H)*P(H)最大即可

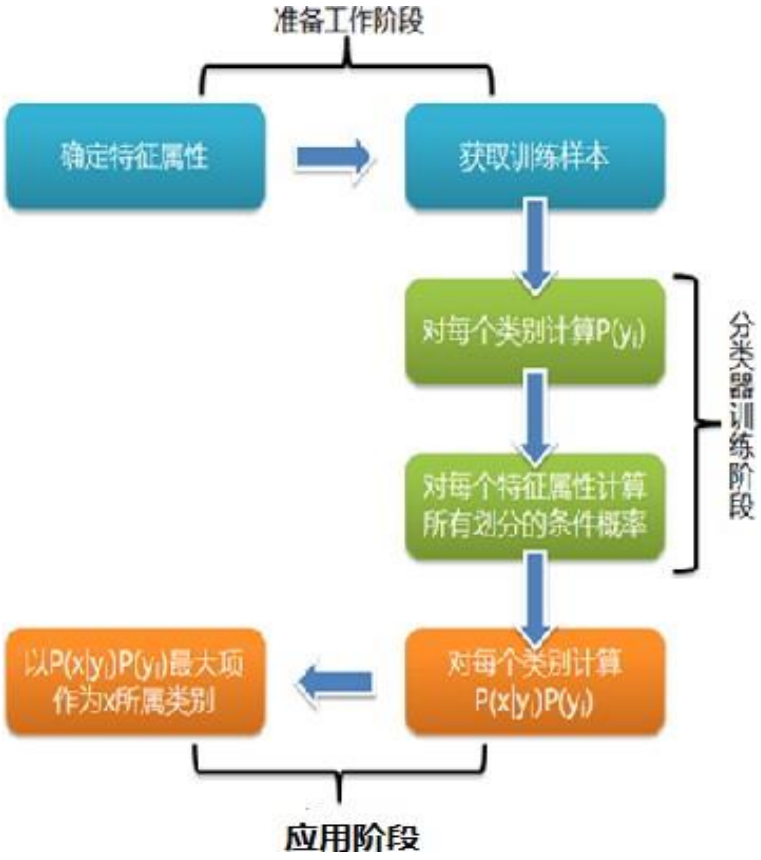
$$P(H \mid X) = \frac{P(XH)}{P(X)} = \frac{P(X \mid H)P(H)}{P(X)}$$

2.3 Bayes

- 朴素贝叶斯分类
  - 是一种简单的贝叶斯分类方法；
  - 对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。

2.3 Bayes

朴素贝叶斯分类步骤：



朴素贝叶斯分类实例--检测SNS社区中不真实账号

对于SNS社区来说，不真实账号（使用虚假身份或用户的小号）是一个普遍存在的问题，作为SNS社区的运营商，希望可以检测出这些不真实账号，从而在一些运营分析报告中避免这些账号的干扰，亦可以加强对SNS社区的了解与监管。

是真是假？

朴素贝叶斯分类实例--检测SNS社区中不真实账号

类别标记：H = 0表示真实账号，H = 1表示不真实账号

1、确定特征属性及划分

- 三个特征属性：a1：日志数量/注册天数
  a2：好友数量/注册天数
  a3：是否使用真实头像
- 在SNS社区中这三项都是可以直接从数据库里得到或计算出来的，下面给出划分：

  a1：{a1<=0.05, 0.05<a1<0.2, a1>=0.2}

  a2：{a<2=0.1, 0.1<a2<0.8, a2>=0.8}

  a3：{a3=0（不是）,a3=1（是）}

朴素贝叶斯分类实例--检测SNS社区中不真实账号

2、获取训练样本

- 使用运维人员曾经人工检测过的1万个账号作为训练样本，8900条为真实账号，1100条为不真实账号。

3、计算训练样本中每个类别的频率

- P(H = 0) = 8900/10000 = 0.89
- P(H = 1) = 1100/10000 = 0.11

朴素贝叶斯分类实例--检测SNS社区中不真实账号

4、计算每个类别条件下各个特征属性划分的频率（P(x|H)）

- P(a1<=0.05| H = 0) = 0.3
- P(0.05<a1<0.2|H = 0) = 0.5
- P(a1>0.2| H = 0) = 0.2
- P(a2<=0.1| H = 0) = 0.1
- P(0.1<a2<0.8 | H=0) = 0.7
- P(a2>0.8| H = 0) = 0.2
- P(a3 = 0|H = 0) = 0.2
- P(a3 = 0|H = 1) = 0.9

P(a1<=0.05| H = 1) = 0.8

P(0.05<a1<0.2| H = 1) = 0.1

P(a1>0.2| H = 1) = 0.1

P(a2<=0.1| H = 1) = 0.7

P(0.1<a2<0.8 | H=1) = 0.2

P(a2>0.8| H = 0) = 0.1

P(a3 = 1|H = 0) = 0.8

P(a3 = 1|H = 1) = 0.1

朴素贝叶斯分类实例--检测SNS社区中不真实账号

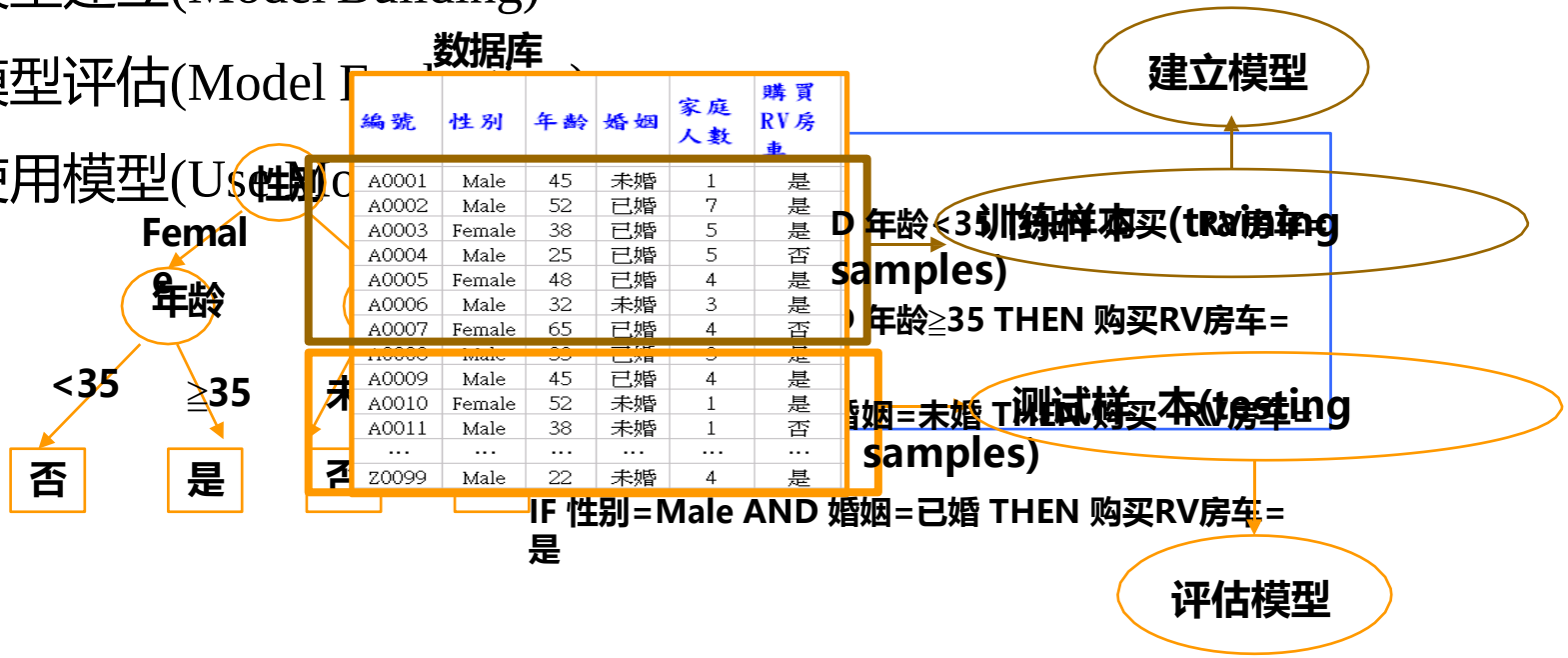
5、使用分类器进行鉴别

- 待鉴别账号属性如下 a1：日志数量与注册天数的比率为0.1
  a2：好友数与注册天数的比率为 0.2
  a3：不使用真实头像 (a = 0)

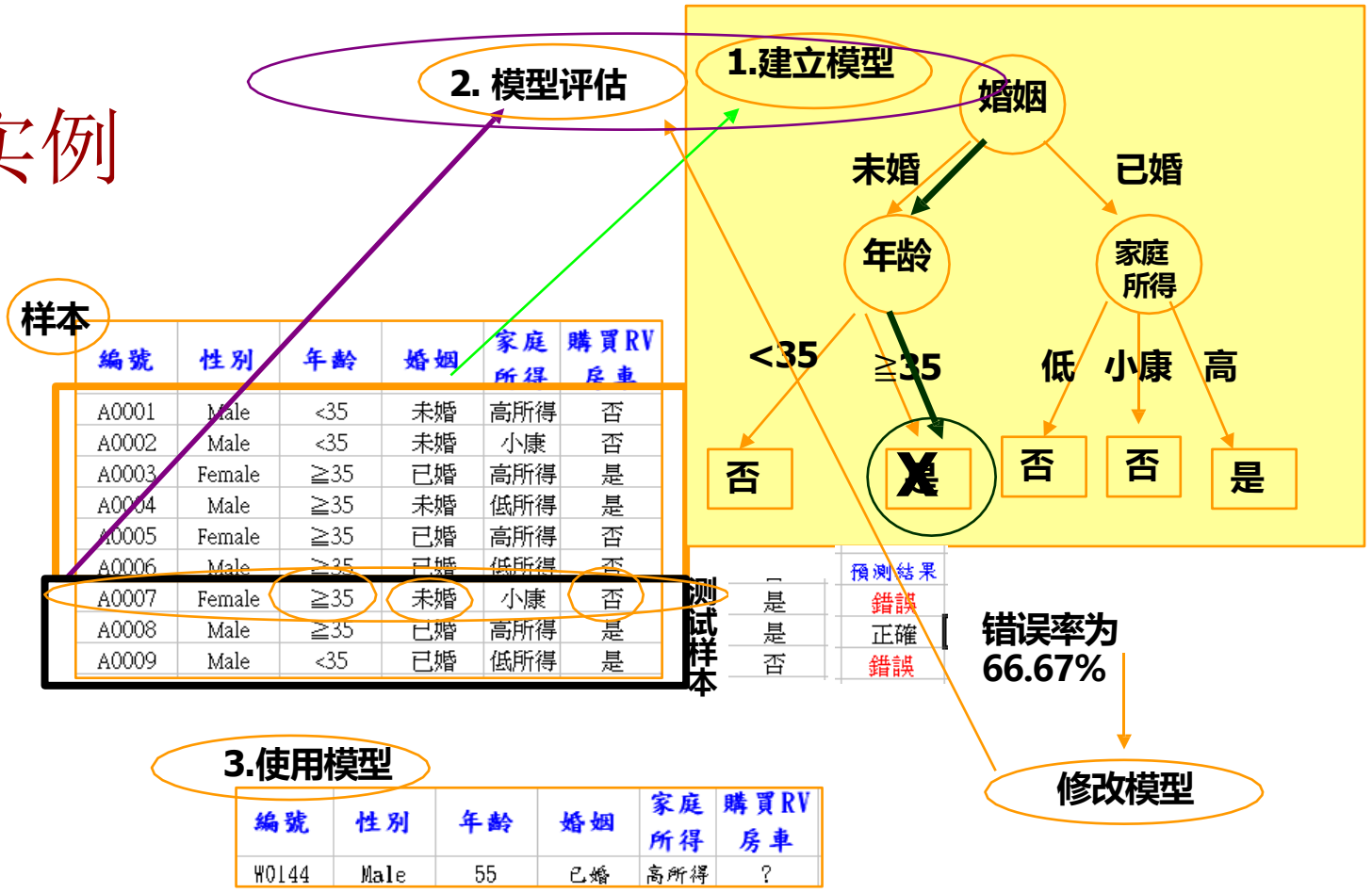
- P(H = 0)P(x|H = 0)
  = P(H = 0) P(0.05<a1<0.2|H = 0)P(0.1<a2<0.8|H = 0)P(a3=0|H = 0)
  = 0.89*0.5*0.7*0.2
  = 0.0623
- P(H = 1)P(x|H = 1)
  = P(H = 1) P(0.05<a1<0.2|H = 1)P(0.1<a2<0.8|H = 1)P(a3=0|H = 1)
  = 0.11*0.1*0.2*0.9
  = 0.00198

2.4 Decision Tree

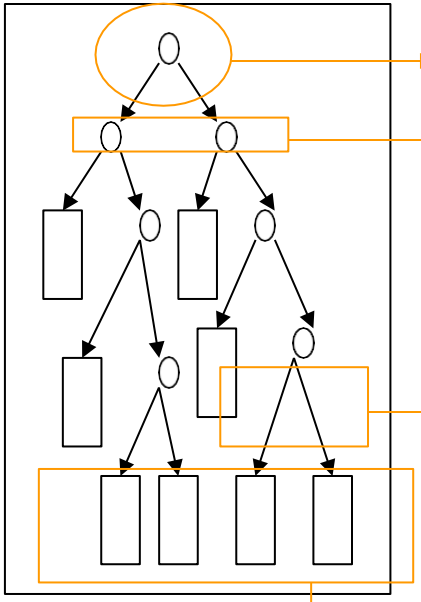
- 分类过程：
  - 模型建立(Model Building)
  - 模型评估(Model E...)
  - 使用模型(Use M...)



实例


2. 模型评估
1.建立模型
婚姻
未婚
已婚
年龄
家庭所得
<35
≧35
低
小康
高
否
否
否
是
样本
編號 性別 年齡 婚姻 家庭所得 購買RV房車
A0001 Male <35 未婚 高所得 否
A0002 Male <35 未婚 小康 否
A0003 Female ≧35 已婚 高所得 是
A0004 Male ≧35 未婚 低所得 是
A0005 Female ≧35 已婚 高所得 否
A0006 Male ≧35 已婚 低所得 否
A0007 Female ≧35 未婚 小康 否
A0008 Male ≧35 已婚 高所得 是
A0009 Male <35 已婚 低所得 是
測試樣本
是
是
否
預測結果
錯誤
正確
錯誤
错误率为66.67%
修改模型
3.使用模型
編號 性別 年齡 婚姻 家庭所得 購買RV房車
W0144 Male 55 已婚 高所得 ?

2.4 Decision Tree



根部节点(root node)

中间节点(non-leaf node)

(代表测试的条件)

分支(branches)

(代表测试的结果)

叶节点(leaf node)

(代表分类后所获得的分类标记)

2.4 Decision Tree

- 决策树结构：
  - 节点→特征属性
  - 分支→属性值
  - 根结点→信息量最大的属性
  - 中间结点→该结点为根的子树所包含的样本子集中信息量最大的属性
  - 叶结点→样本的类别标签

2.4 Decision Tree

- 决策树建树方法—ID3
  - 对当前例子集合，计算各特征的互信息；
  - 选择互信息最大的特征Ak作为根节点；
  - 把在Ak处取值相同的例子归于同一子集，Ak取几个值就得几个子集；
  - 对既含正例又含反例的子集，递归调用建树算法；
  - 若子集仅含正例或反例，对应分枝标上类别。

NO.	属性				类别
	天气	气温	湿度	风	
1	晴	热	高	无风	**N**
2	晴	热	高	有风	**N**
3	多云	热	高	无风	**P**
4	雨	适中	高	无风	**P**
5	雨	冷	正常	无风	**P**
6	雨	冷	正常	有风	**N**
7	多云	冷	正常	有风	**P**
8	晴	适中	高	无风	**N**
9	晴	冷	正常	无风	**P**
10	雨	适中	正常	无风	**P**
11	晴	适中	正常	有风	**P**
12	多云	适中	高	有风	**P**
13	多云	热	正常	无风	**P**
14	雨	适中	高	有风	**N**

天气可取值：晴，多云，雨

气温可取值：冷 ， 适中，热

湿度可取值：高 ， 正常

风　可取值：有风， 无风

类别可取值：N，P

Decision Tree

(1) 信息熵: $H(U) = -\sum_i P(u_i)\log P(u_i)$

- 类别出现概率: $P(u_i) = \frac{|u_i|}{|S|}$
- |S|表示例子集S的总数，$|u_i|$表示类别u_i的例子数，对9个正例和5个反例有:
- P（u_1）=9/14
- P（u_2）=5/14
- H（U）=（9/14）log（14/9）+（5/14）log（14/5）=0.94bit

Decision Tree案例

（2）条件熵：$H(U/V)=-\sum_j P(v_j)\sum_i P(u_i/v_j)\log P(u_i/v_j)$

属性A_1取值v_j时，类别u_i的条件概率：$P(u_i/v_j)=\frac{|u_i|}{|v_j|}$

A_1=天气 取值 v_1=晴，v_2=多云，v_3=雨

在A_1处取值晴的例子5个，多云的例子4个，雨的例子5 个，则：

P（v_1）=5/14 P（v_2）=4/14 P（v_3）=5/14

取值为晴的5个例子中有2个正例、3个反例，则：

P（u_1/v_1）=2/5，P（u_2/v_1）=3/5

同理有：P（u_1/v_2）=4/4，P（u_2/v_2）=0，P（u_1/v_3）=2/5， P（u_2/v_3）=3/5

H(U/V)=(5/14)((2/5)log(5/2)+(3/5)log(5/3))+(4/14)((4/4)log(4/4+0)+(5/14)((2/5)log(5/2)+(3/5)log(5/3)) = 0.694bit

Decision Tree案例

3　　互信息：信息熵-条件熵

对 A1=天气 处有：

I（天气）=H（U）- H（U|V）= 0.94 - 0.694 = 0.246 bit

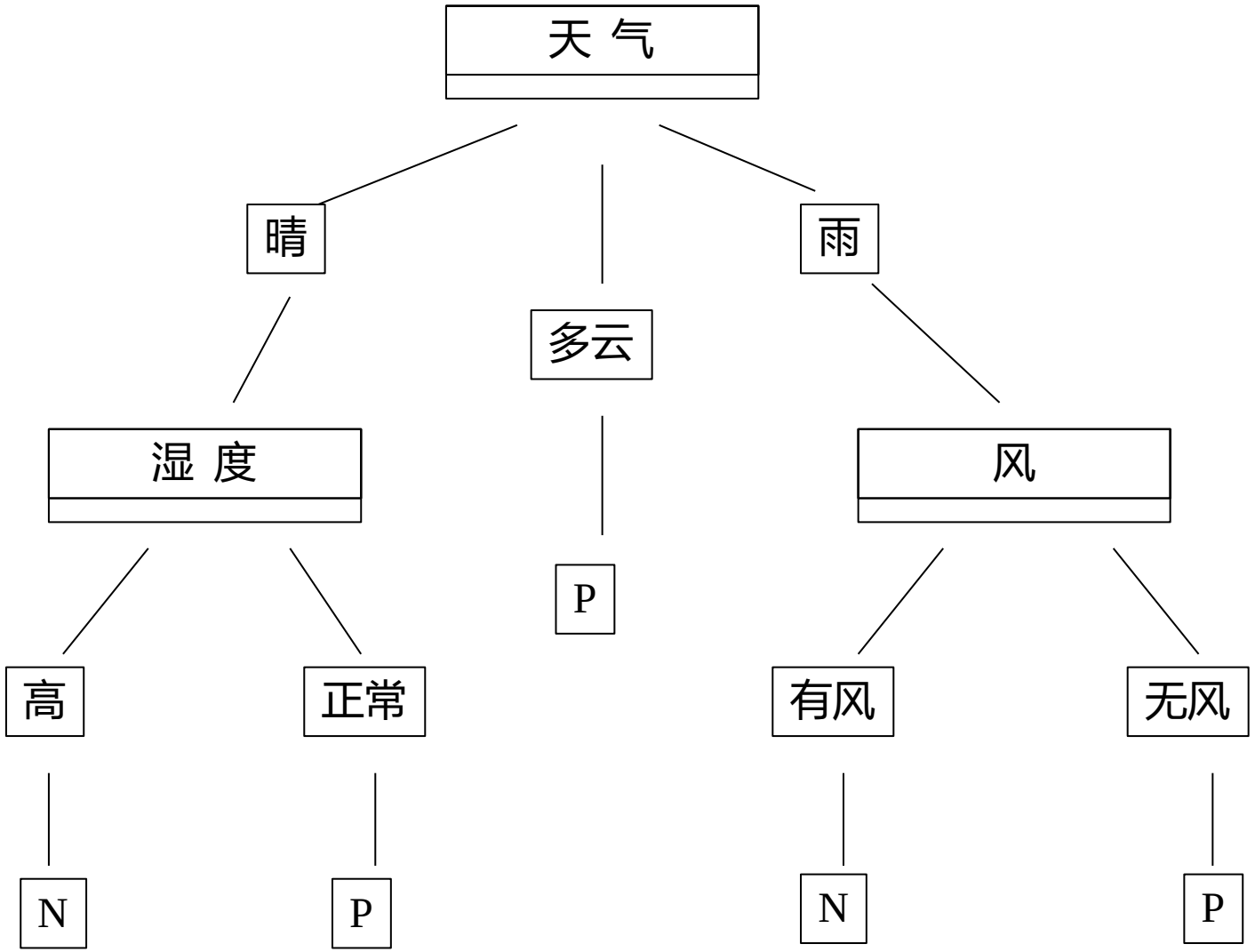
类似可得：

I（气温）=0.029 bit　　I（湿度）=0.151 bit　　I（风）=0.048 bit

4　　建决策树的树根和分枝

ID3算法将选择互信息最大的特征天气作为树根，在14个例子中对天气的3个取值进行分枝，3 个分枝对应3 个子集，分别是：F1={1，2，8，9，11}，F2={3，7，12，13}，F3={4，5，6，10，14}

其中F2中的例子全属于P类，因此对应分枝标记为P，其余两个子集既含有正例又含有反例，将递归调用建树算法。

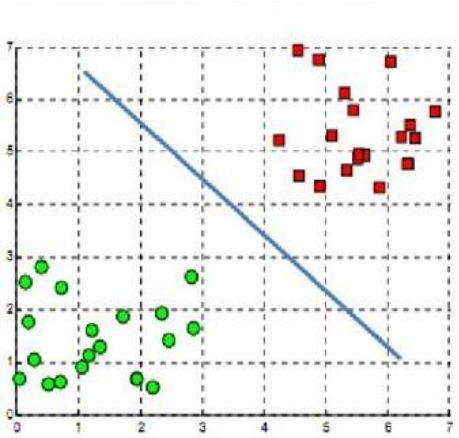

天 气
晴
多云
雨
湿 度
P
风
高
正常
有风
无风
N
P
N
P

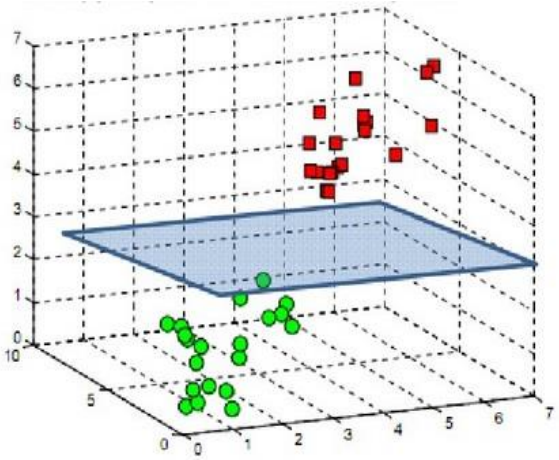
2.4 Decision Tree

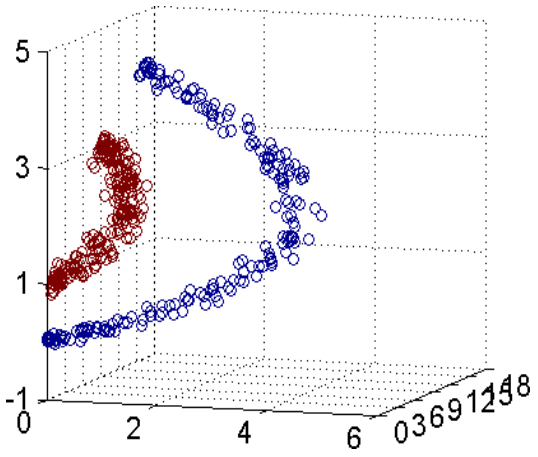
4 决策树模型特点：

- 优点：
  - 不需要任何领域知识或参数假设。
  - 适合高维数据。
  - 短时间内处理大量数据，得到可行且效果较好的结果。
- 缺点：
  - 对于各类别样本数量不一致数据，信息增益偏向于那些具有更多数值的特征。
  - 忽略属性之间的相关性。
  - 不支持在线学习。

2.5 SVM







2.5 SVM

(1)支持向量机（SVM）基本思想：

- 是二值分类算法：系统随机产生一个超平面并移动它，直到训练集中属于不同类别的样本点正好位于该超平面的两侧。显然，这种机理能够解决线性分类问题，但不能够保证产生的超平面是最优的。
- 支持向量机建立的最优分类超平面能够在保证分类精度的同时，使超平面两侧的空白区域最大化，从而实现对线性可分问题的最优分类。

2.5 SVM

2　支持向量机（SVM）关键问题：

- SVM（支持向量机）主要针对小样本数据进行学习、分类和预测的一种方法。
- “支持向量”：则是指训练集中的某些训练点，这些点最靠近分类决策面，是最难分类的数据点

最优分类线/面？

2.5 SVM—线性分类

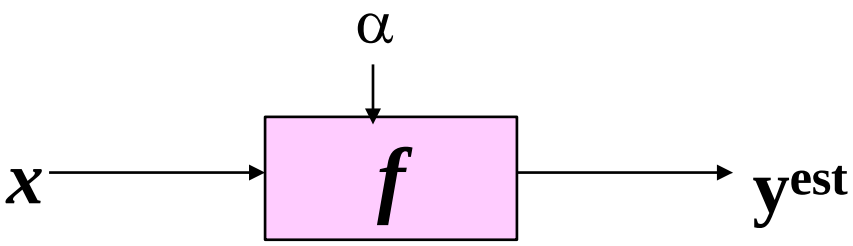
- 分类面：把一个空间按照类别切分两部分的平面，在二维空间中，分类面相当于一条直线，三维空间中相当于一个平面，高维空间为超平面。
- 线性分类面函数形式为：$f(\mathbf{x}) = w^T \mathbf{x} + b$

w^T,b是分类面函数参数，x是输入的样本, w^T权向量，b是偏移量

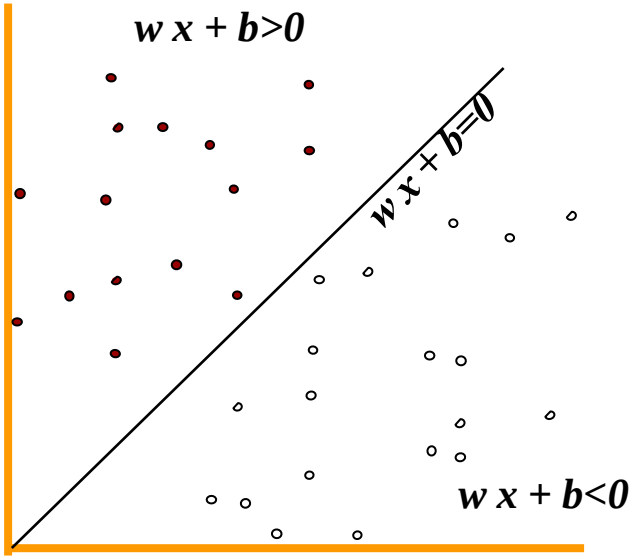
$$f(x) = w^T \mathbf{x}_i + b = 0$$

$$y = \operatorname{sgn}\left(f(x)\right) = \begin{cases} f(x) = w^T \mathbf{x}_i + b > 0 & \text{for } y = +1 \\ f(x) = w^T \mathbf{x}_i + b < 0 & \text{for } y = -1 \end{cases}$$

2.5 SVM—线性分类



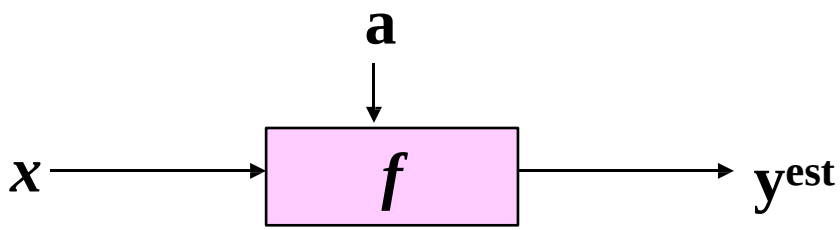
- 表示 +1
- 表示 -1

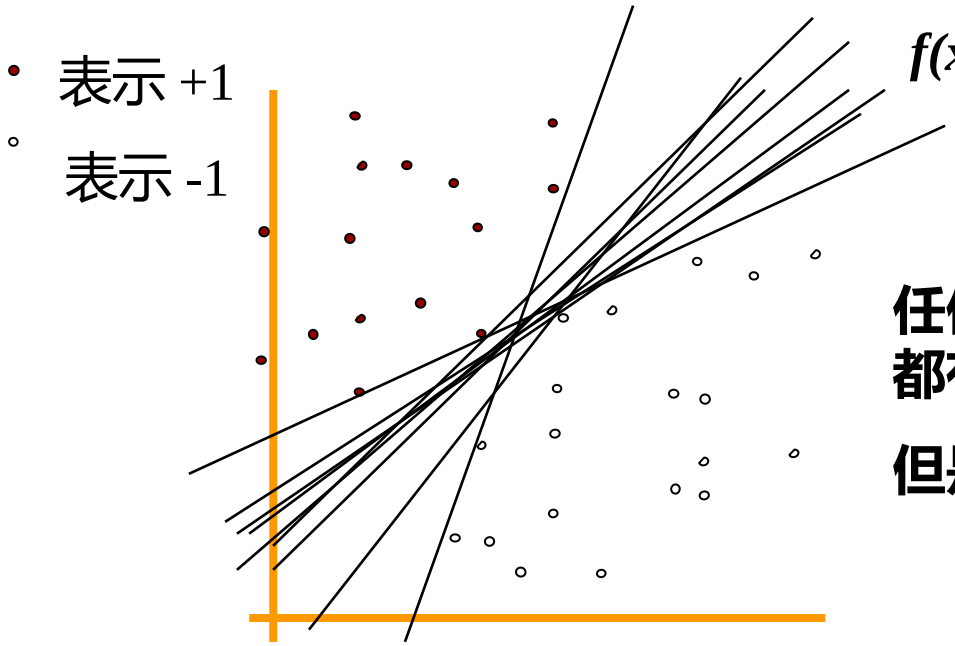


$f(x,w,b) = sign(w\ x + b)$

如何分类这些数据?

2.5 SVM—线性分类



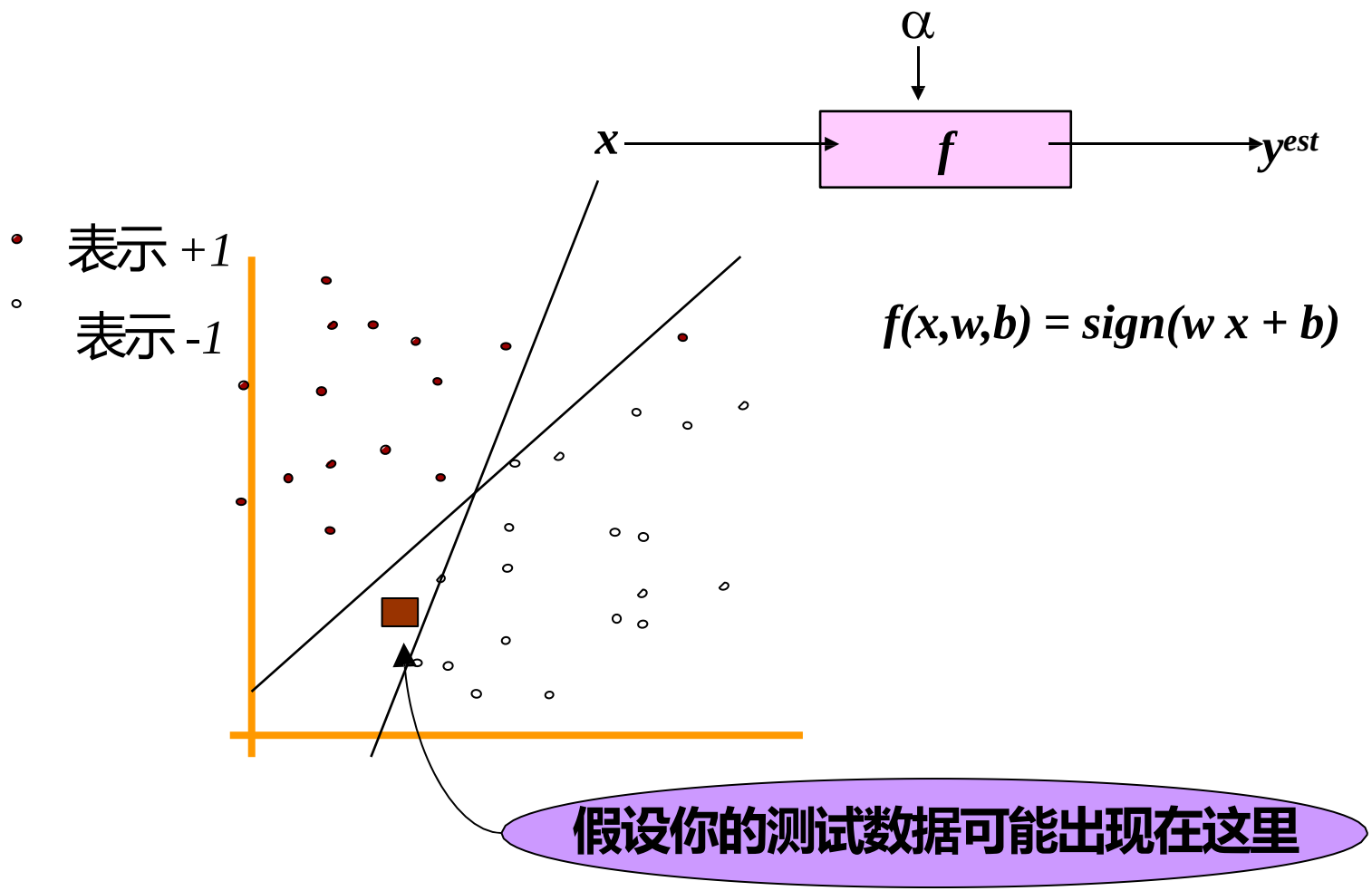


$f(x,w,b) = sign(w\,x + b)$

任何一个分类器（一条线）都有效

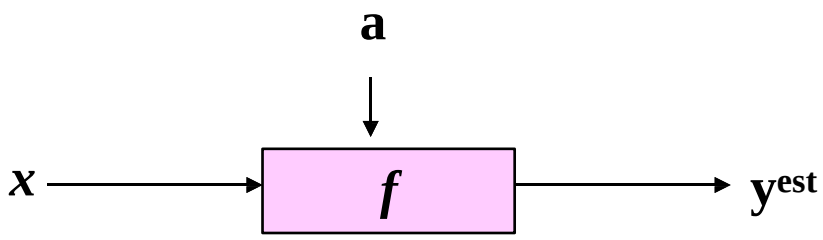
但是哪一个是最好的?

2.5 SVM—线性分类

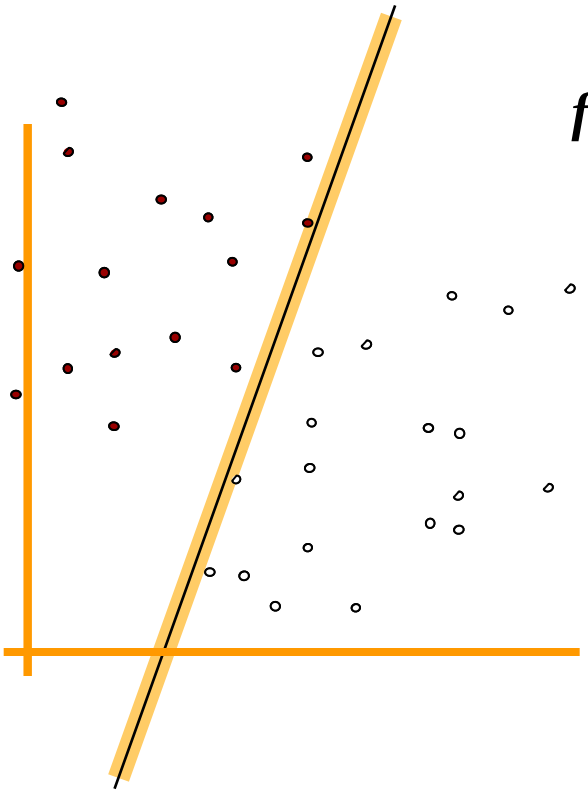


2.5 SVM—线性分类

Max-margin



- 表示 +1
- 表示 -1



$$f(x,w,b) = sign(w\,x + b)$$

定义分类器的边界

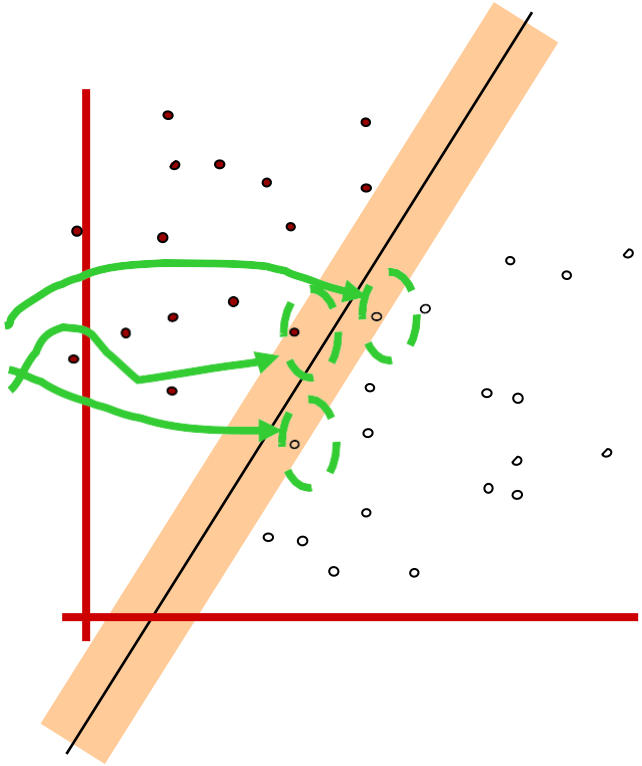
以改善分类性能.

2.5 SVM—线性分类

Max-margin

- 表示 +1
- 表示 -1

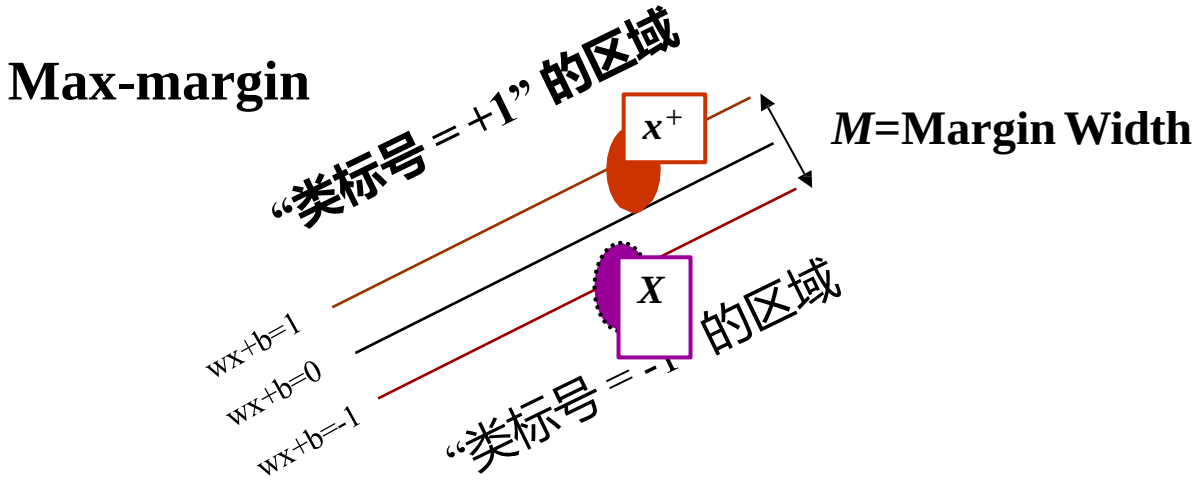
Support Vectors 是边界上的一些样本点



1.这种理论说明只有 Margin上的样本点是 重要的，其他样本都 不重要

2.实践证明这种假设 效果非常好.

2.5 SVM—线性分类



- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{w} \cdot (\mathbf{x}^+ - \mathbf{x}^-) = 2$

$$\Rightarrow \quad \text{Margin} = \frac{w \cdot (x^+ - x^-)}{|w|} = \frac{2}{|w|}$$

2.5 SVM—线性分类

- 假定训练数据 $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_n), \mathbf{x} \in R^d, y \in \{+1, -1\}$
- 线性分类面函数 $(w^T . \mathbf{x}) + b = 0, w \in R^d, b \in R$
- Max-margin转化成优化问题

$$\max\left(\frac{2}{\|w\|}\right) \Leftrightarrow \min\left(\|w\|^2\right)$$

2.5 SVM—线性分类

最优分类面求解问题表示成约束优化问题

最小化目标函数 $Q(w)=\frac{1}{2}\|w\|^2=\frac{1}{2}(w^T\cdot w)$

约束条件 $y_i((w^T\cdot\mathbf{x}_i)+b)\geq 1, i=1,...,n$

拉格朗日函数 $L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i(y_i\cdot((w^T\cdot\mathbf{x}_i)+b)-1)$

2.5 SVM—线性分类

- Lagrange函数 $L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i(y_i\cdot((w^T\cdot\mathbf{x}_i)+b)-1)$

$$\frac{\partial}{\partial b}L(w,b,\alpha)=0;\ \frac{\partial}{\partial w}L(w,b,\alpha)=0$$

$$\sum_{i=1}^{n}a_i y_i=0\ ;\quad w=\sum_{i=1}^{n}\alpha_i y_i x_i$$

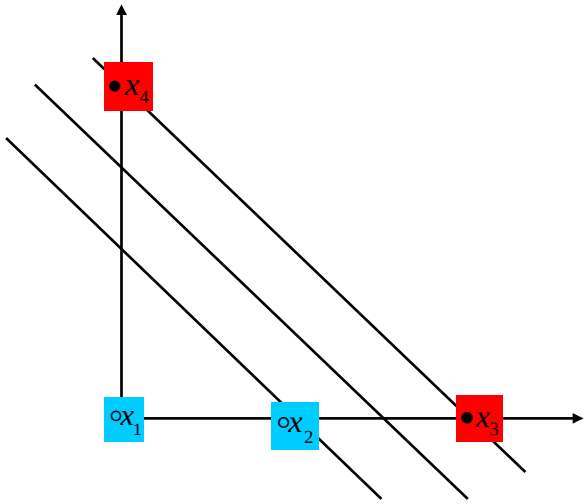
- KKT条件

$$W(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)$$

$$\alpha_i\geq 0,\quad i=1,\ldots,n,\quad and\quad \sum_{i=1}^{n}\alpha_i y_i=0$$

$$\alpha_i\,(y_i\cdot((w^T\cdot\mathbf{x}_i)+b)-1)=0$$

线性SVM求解实例



$\mathbf{x}_1 = (0, 0)^T,\ y_1 = +1$

$\mathbf{x}_2 = (1, 0)^T,\ y_2 = +1$

$\mathbf{x}_3 = (2, 0)^T,\ y_3 = -1$

$\mathbf{x}_4 = (0, 2)^T,\ y_4 = -1$

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

代入x,y值 . . .

$$W(\alpha) = (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2}(\alpha_2^2 - 4\alpha_2\alpha_3 + 4\alpha_3^2 + 4\alpha_4^2)$$

求得$\alpha_1, \alpha_2, \alpha_3, \alpha_4$的值，进而求得w和b的值。

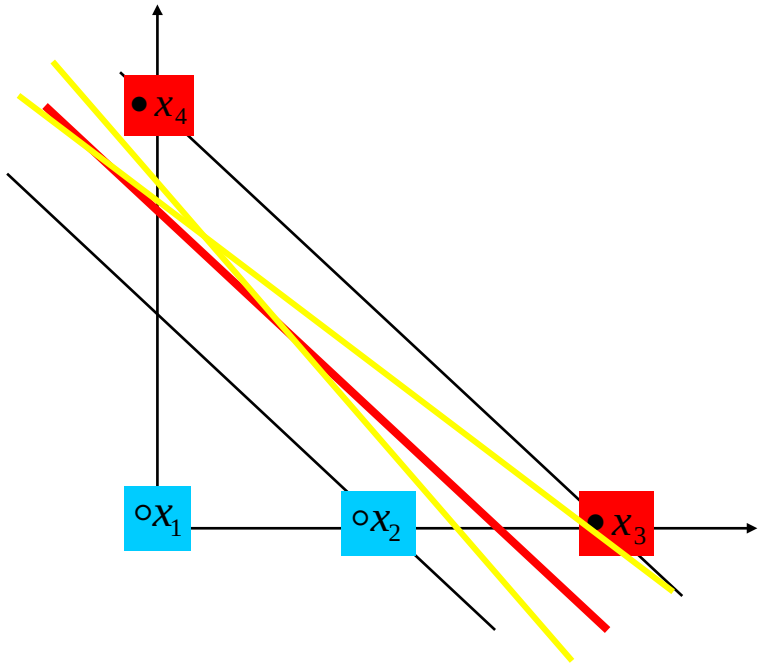
$$\begin{cases} \alpha_1 = 0 \\ \alpha_2 = 1 \\ \alpha_3 = 3/4 \\ \alpha_4 = 1/4 \end{cases}$$

$$w = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \frac{3}{4}\begin{bmatrix} 2 \\ 0 \end{bmatrix} - \frac{1}{4}\begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

$$b = -\frac{1}{2}\left[-\frac{1}{2}, -\frac{1}{2}\right]\begin{bmatrix} 3 \\ 0 \end{bmatrix} = \frac{3}{4}$$

$$f(x) = W^T x + b = \begin{bmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}^T x + \frac{3}{4}$$

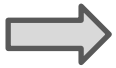
代入(3/2,0),(0,3/2)点可以知道



2.5 SVM—线性分类

$$\min \frac{1}{2}\|w\|^2$$

$$s.t., y_i(w^T x_i + b) \geq 1, i = 1, \cdots, n$$



$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{R} \varepsilon_i$$

$$s.t., y_i(w^T x_i + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0$$

$$Q(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M} \xi_i - \sum_{i=1}^{M} \alpha_i \left(y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{M} \beta_i \xi_i$$

$$\mathbf{w} = \sum_{i=1}^{M} \alpha_i y_i \mathbf{x}_i,$$

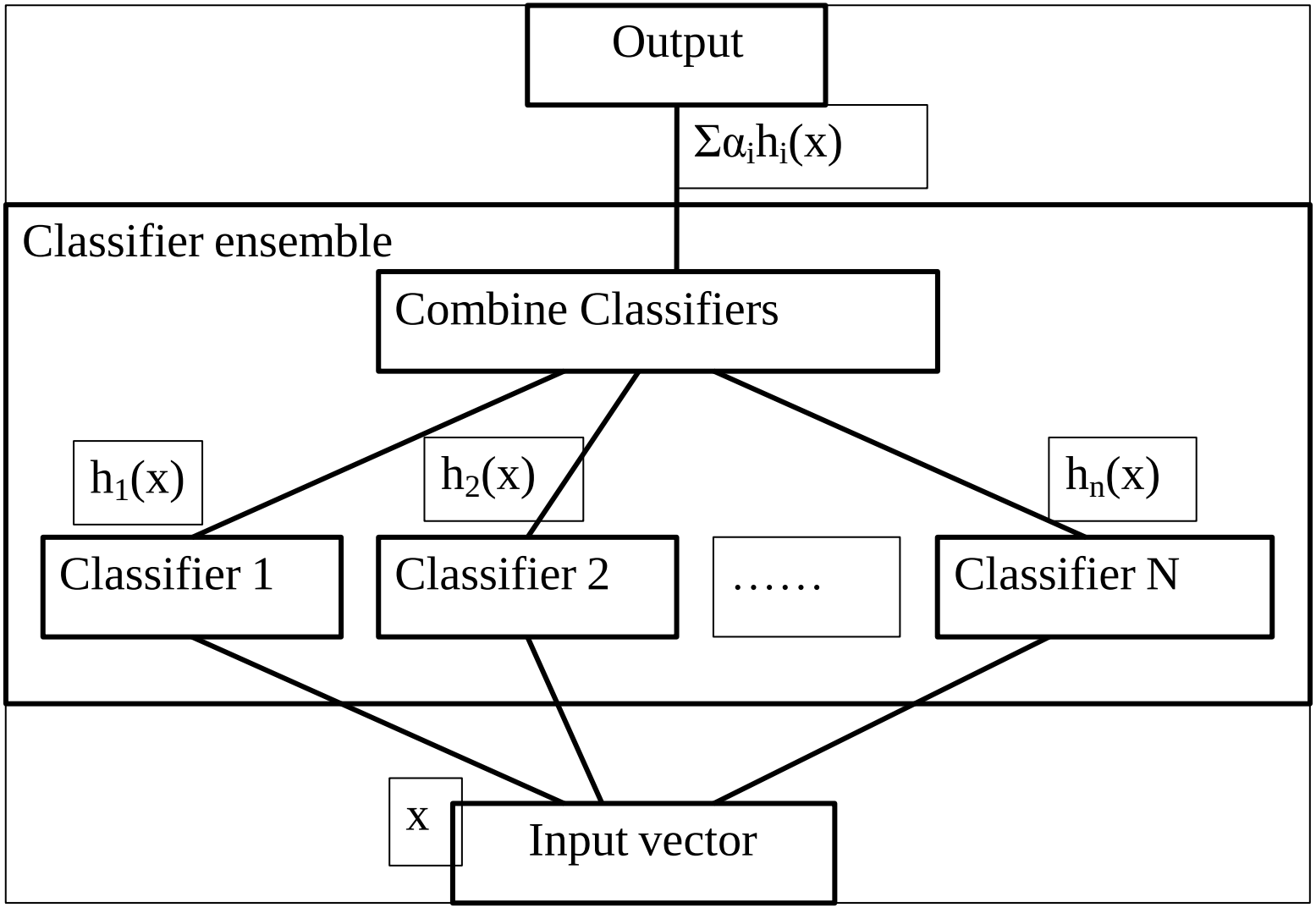
$$\sum_{i=1}^{M} \alpha_i y_i = 0,$$

$$\alpha_i + \beta_i = C \quad \text{for } i = 1, \ldots, M.$$

2.6 Ensemble learning

1 集成学习（Ensemble learning）基本思想

- 在机器学习中，直接建立一个高性能的分类器是很困难的。
- 如果能找到一系列性能相对较差的个体分类器（弱分类器），并把它们集成起来的话，也许就能得到更好的分类器，从而提高整体分类器的泛化能力。
  - 所有个体学习器均为决策树时，称为“决策树集成”
  - 所有个体学习器均为神经网络时，称为“神经网络集成”
  - 所有个体学习器不全是一个种类的，.....


Output
$\Sigma\alpha_i h_i(x)$
Classifier ensemble
Combine Classifiers
$h_1(x)$
$h_2(x)$
$h_n(x)$
Classifier 1
Classifier 2
……
Classifier N
x
Input vector

2.6 Ensemble learning

2　集成学习关键内容

- 如何构建具有差异性的个体分类器？
  - 通过改变训练集来构造不同的个体分类器，且个体学习器越精确、差异越大，集成越好；
  - 构建方法：Bagging；Boosting；AdaBoost；Random Forest；GBDT
- 如何将这些分类器的结果进行整合（集合策略）？
  - 多数投票法；加权平均；学习法（stacking）。

2.6 Ensemble learning

3　　构建个体分类器方法 ---Bagging

- 从大小为n的原始数据集D中独立随机地抽取n’个数据(n’<=n)，形成一个自助数据集；
- 重复上述过程，产生出多个独立的自助数据集；
- 利用每个自助数据集训练出一个“个体分类器”；
- Bagging个体分类器整合策略：
  - 最终的分类结果由这些“个体分类器”各自的判别结果投票决定（投票法）

2.6 Ensemble learning

使用 Bagging 算法的时候，理论上每个基本分类器的训练集中有 63.2%的重复样例

Breiman 指出，要使得 Bagging 有效，基本分类器的学习算法必须是不稳定的，也就是说对训练数据敏感。基本分类器的学习算法对训练数据越敏感，Bagging 的效果越好，因此对于决策树和人工神经网络这样的学习算法 Bagging 是相当有效的。

Bagging 算法的时候应该用多少个基本分类器合适呢？Breiman 指出基本分类器数目应当随着分类种数增多而增加。

2.6 Ensemble learning

4 构建个体分类器方法---Boosting

- Step1: 原始训练集输入
- Step2: 计算训练集中各样本的权重
- Step3: 采用已知算法训练个体分类器，并对每个样本进行判别
- Step4: 计算对此次的个体分类器的权重
- Step5: 转到Step2, 直到循环到达一定次数或者某度量标准符合要求
- Boosting个体分类器集成策略：
  - 将弱学习机按其相应的权重加权组合形成强学习机（加权平均）

2.6 Ensemble learning

Boosting方法中各样本的分配权重：提高分错样本的权重

- 没有先验知识的情况下，初始的分布应为等概分布，也就是训练集如果有N个样本，每个样本的分布概率为1/N；
- 每次循环一后提高错误样本的分布概率，分错样本在训练集中所占权重增大， 使得下一次循环的弱学习机能够加强对这些错误样本的训练；
- 反映了strong learner对样本的假设是否正确
- 采用的计算函数：

$$y_i \cdot H_t(\mathrm{X}_i) \qquad y_i \cdot H_t(\mathrm{X}_i) \begin{cases} > 0 & right \\ < 0 & wrong \end{cases}$$

$$\exp\left(- y_i \cdot H_t(\mathrm{X}_i)\right)$$

2.6 Ensemble learning

Boosting方法中个体分类器权重设置：

- 体分类器的权重:准确率越高的弱学习机权重越高
  - $\varepsilon_t = P_{(x,y)\in D_t}[y \neq h_t(x)]$ 为个体分类器的错误概率
  - 采用的计算函数: $w_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
- 循环控制：损失函数达到最小
  - 在强学习机的组合中增加一个加权的弱学习机，使准确率提高，损失函数值减小。

2.6 Ensemble learning

5　　构建个体分类器方法---AdaBoost

- 和boosting算法效率几乎相同，但是不需要任何个体分类器的先验知识，更容易应用到实际应用中

- AdaBoost构造分类器步骤：

输入：(1)训练样本集 $\mathcal{D}=\{(\boldsymbol{x}_i,y_i)\,|\,i=1,\ldots,N\}\ y_i\in\{-1,+1\}$

其中 $\begin{cases} y_i=-1, \text{训练样本}\boldsymbol{x}_i\text{为负样本} \\ y_i=+1, \text{训练样本}\boldsymbol{x}_i\text{为正样本} \end{cases}$

(2)**弱分类器**的学习算法L

(3)弱分类器的数目M

2.6 Ensemble learning

初始化训练样本x_i权重 $\mathcal{D}_1(i)\quad i=1,\ldots,N$

(1)若正负样本数目一致，则 $\mathcal{D}_1(i)=\frac{1}{N}$

(2)若正负样本数目分别为N_+, N_-，则
$$\begin{cases} \text{正样本}\ \mathcal{D}_1(i)=\frac{1}{2N_+} \\ \text{负样本}\ \mathcal{D}_1(i)=\frac{1}{2N_-} \end{cases}$$

for $m=1,\ldots,M$

(1)**训练弱分类器** $f_m(x)=L(\mathcal{D},\mathcal{D}_m)\in\{-1,+1\}$

(2)估计弱分类器$f_m(x)$的分类错误率e_m

如：$e_m=\frac{1}{2}\sum_{i=1}^{N}\mathcal{D}_m(i)\cdot\left|f_m(x_i)-y_i\right|$ [注：$e_m<0.5$]

2.6 Ensemble learning

for $m = 1, \ldots, M$ (续前)

(3) 估计弱分类器 $f_m(x)$ 的权重 $c_m = \log \dfrac{1-e_m}{e_m}$

(4) 基于弱分类器 $f_m(x)$ 调整各样本权重，并归一化

调　整：$\mathcal{D}_{m+1}(i) = \mathcal{D}_m(i) \cdot \exp\left[c_m \cdot 1_{(f_m(x_i) \neq y_i)}\right]$

$$= \begin{cases} \mathcal{D}_m(i) & \text{若} f_m(x_i) = y_i \\ \mathcal{D}_m(i) \cdot \dfrac{1-e_m}{e_m} & \text{若} f_m(x_i) \neq y_i \end{cases}$$

归一化：$\mathcal{D}_{m+1}(i) \leftarrow \dfrac{\mathcal{D}_{m+1}(i)}{\sum_{j=1}^{N} \mathcal{D}_{m+1}(j)} \quad i = 1, \ldots, N$

强分类器

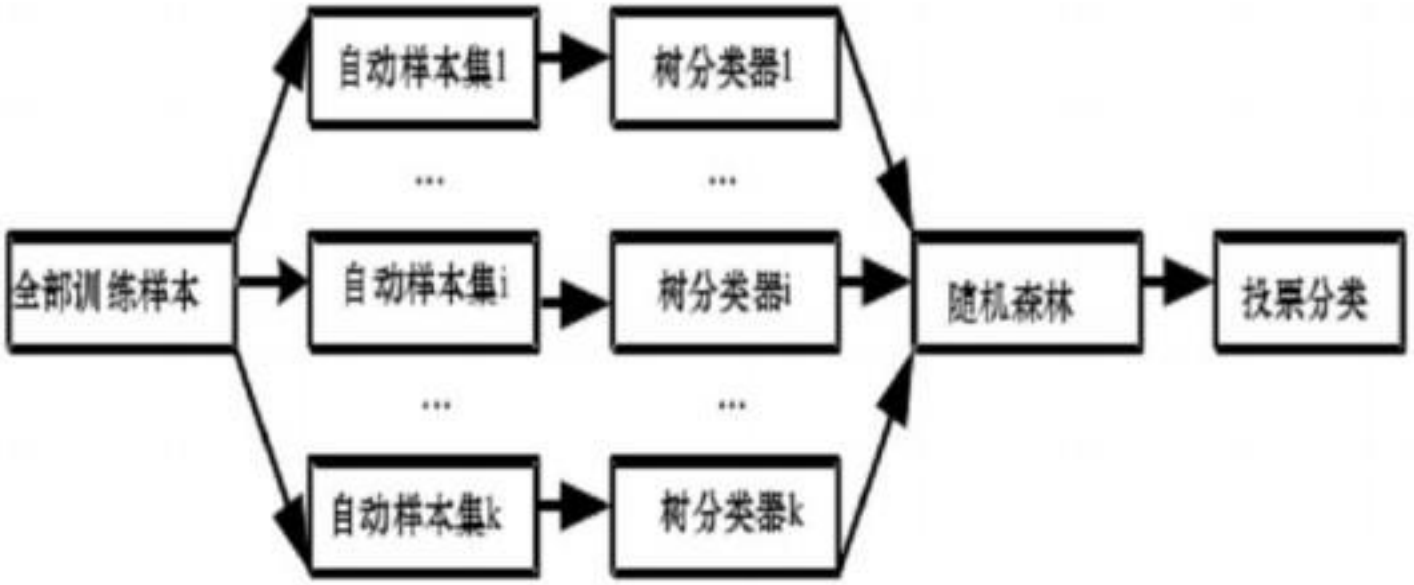
$$H(x) = \operatorname{sgn}\left[\sum_{m=1}^{M} c_m f_m(x)\right]$$

2.6 Ensemble learning

6　构建个体分类器方法---Random Forest

- 一种新型分类和预测模型，它具有需要调整的参数少，不容易过度拟合，分类速度快，能高效处理大样本数据等特点。
- Bagging和AdaBoost等方法只是通过改变样本的权重来获得不同的弱分类器。随机森林（RF）则通过同时改变样本和特征子集来获得不同的弱分类器。
- 采用随机的方式建立一个森林，个体分类器由决策树组成，且之间没有关联。对于新的测试样本，让森林中的每一棵决策树分别进行一下判断，依据多数者投票方法决定样本的类别。

2.6 Ensemble learning



2.6 Ensemble learning

随机决策树的构造

首先，随机森林对输入的数据要进行行、列的采样。对于行采样，采用有放回的方式，也就是在采样得到的样本集合中，可能有重复的样本。假设输入样本为 N 个，那么采样的样本也为 N 个。这样使 得在训练的时候，每 一棵树的输入样本都不是全部的样本，每次 采样后大约有37%左右的数据不被采 到， 从而在一定程度上避免出现过拟合。然后进行列采样，从 M 个特征中，选 择 m 个(m << M)。

之后 ， 就是对采样之后的数据使用完全分裂的方式建立出决策树，这样决策树的某一个叶子节点要么是无法继续分裂的，要么里面 的所有样本的都是指向的同一个分类

2.6 Ensemble learning

随机特征选取

当特征个数M较多时，随机选择m个用于训练决策树。m越小，树的相关性越小，且训练速度越快。

当特征个数M较少时，可以由M个特征进行随机线性组合来产生M′个扩展特征，然后，在（M+M′）上随机选择m个特征，构建决策树。

其中，每一个扩展特征的构造如下： 从现有M特征中随机抽取L个，它们的权重系数是[-1,+1]区间的均匀随机数。然后，由L个已有特征线性组合出扩展特征。

2.6 Ensemble learning – 关键问题导读

7既然多个个体的集成比单个个体更好，那么是不是 个体越多越好?

- 在预测时需要更大的计算开销，因为要计算更多的个体预测
- 更大的存储开销，因为有更多的个体需要保存
- 个体的增加将使得个体间的差异越来越难以获得

算法分析	算法描述
回归分析	回归分析是确定去测属性（数值型）与其他变量间相互依赖的定量关系最常用的统计学方法。包括线性回归、非线性回归、Logistic回归、岭回归、主成分回归、偏最小二乘回归等模型
决策树	决策树采用自顶向下的递归方式，在内部节点进行属性值的比较，并根据不同的属性值从该节点向下分支，最终得到的叶节点是学习划分的类
人工神经网络	人工神经神经网络是一种模仿大脑神经网络和功能而建立的信息处理系统，表示神经网络的输入与输出变量之间关系的模型
贝叶斯网络	贝叶斯网络又称信度网络，是Bayes方法的扩展，是目前不确定知识表达和推理领域最有效的理论模型之一
支持向量机	支持向量机是一种通过魔种非线性映射，把低纬的非线性可分转化为高维的线性可分，在高维空间进行线性分析的算法

回归模型名称	试用条件	算法描述
线性回归	因变量与自变量是线性关系	对一个或多个自变量和因变量之间的线性关系进行建模可用最小二乘法求解模型系数
非线性回归	因变量与自变量之间不都是线性关系	对一个或多个自变量和因变量之间的非线性关系进行建模。如果非线性关系可以通过简单的函数变换转化成线性关系，用线性回归的思想求解；如果不能转化，用非线性最小二乘法方法求解
Logistic	因变量一般有1和0两种取值	是广义线性回归模型的特例，利用Logistic函数将因变量的取值范围控制在0和1之间，表示取值为1的概率
主成分回归	参与建模的自变量之间具有多重共线性	主成分回归是根据主成分分析的思想提出来，是对最小二乘法的一种改进，它是参数估计的一种有偏估计。可以消除自变量之间的多重共线性