# 绪论

1. 数据挖掘任务
2. 数据挖掘主要流程

## 1.数据挖掘任务

- 预测任务

  预测任务主要包括回归和分类

- 描述任务

  描述任务主要包括关联分析、聚类分析、异常检测等

- 四种主要的挖掘任务

1. 聚类分析(cluster analysis)
    1. Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that
2. 预测建模(predictive modeling)
    1. 分类(classification): 用于预测离散的目标变量
    2. 回归(regression): 用于预测连续的目标变量
3. 关联分析(association analysis):
    1. 用来发现描述数据中强关联特征的模式
4. 异常检测(anomaly detection)
    1. 识别其特征不同于其他数据的观测值

## 2.数据挖掘的主要流程

1. 分析问题，定义挖掘目标
2. 数据采集
3. 数据预处理
4. 构建挖掘模型
5. 模型评价优化
6. 模型部署应用

# 第二章 数据

1. 不同的属性类型
2. 数据质量问题
3. 数据预处理的主要方法
4. 连续属性离散化
5. 相似性/相关性度量：**距离、余弦、SMC、Jaccard系数**、皮尔森相关系数

## 1. 不同的属性类型

1. 属性

   属性(attribute): 对象的性质或特性

   属性也称:变量、特性、字段 or 维

2. 测量标度

测量标度(measurement scale): 将数值和符号值与对象的属性相关联的规则(函数)

3. 属性的类型(测量标度的类型)

   取决于下列4种性质

   - Distinctness(相异性): = ≠
   - Order(序):<>
   - Addition: + -
   - Multiplication(乘法): * /

   结合四种属性,可定义四种属性类型

   1. 分类的(定性的)
      - 标称
      - 序数
   2. 数值的(定量的)
      - 区间
      - 比率

| 属性类型 | | 描　述 | 例　子 | 操　作 |
|---|---|---|---|---|
| 分类的(定性的) | 标称 | 标称属性的值仅仅只是不同的名字，即标称值只提供足够的信息以区分对象（=，≠） | 邮政编码、雇员ID号、眼球颜色、性别 | 众数、熵、列联相关、$\chi^2$检验 |
| | 序数 | 序数属性的值提供足够的信息确定对象的序（<，>） | 矿石硬度、{好，较好，最好}、成绩、街道号码 | 中值、百分位、秩相关、游程检验、符号检验 |
| 数值的(定量的) | 区间 | 对于区间属性，值之间的差是有意义的，即存在测量单位（+，−） | 日历日期、摄氏或华氏温度 | 均值、标准差、皮尔逊相关、$t$和$F$检验 |
| | 比率 | 对于比率变量，差和比率都是有意义的（*，/） | 绝对温度、货币量、计数、年龄、质量、长度、电流 | 几何平均、调和平均、百分比变差 |

# 2. 数据质量问题

## 2.1.2 数据集的类型

1. 数据集的一般特性
   - 维度(Dimensionality)
     - 维度是数据集中的对象具有的属性数目
     - 维灾难(curse of dimensionality)
     - 维归约(dimensionality reduction)
   - 稀疏性(sparsity)
     - 一个对象大部分属性上的值为0
     - 只存储和处理非零值
   - 分辨率(resolution)
     - 数据的模式依赖于分辨率——度量尺度 (scale)
2. 数据集类型
   - 记录数据(record)
     - 数据矩阵(Data Matrix)
     - 文本数据(Document Data)：每篇文档可以表示成一个文档-词矩阵
     - 事务数据(Transaction Dat)
   - 基于图形的数据(graph)

        - World Wide Web
        - 分子结构（Molecular Structures）
    - 有序数据(ordered)
        - 空间数据(Spatial Data)
        - 时间数据(Temporal Data)
        - 序列数据(Sequential Data )

# 2.2 数据质量

## 2.2.1 测量和数据收集问题

- 测量误差和数据收集错误
- 噪声和伪像
- 精度、偏倚、准确率
- 离群点
- 遗漏值
- 不一致的值
- 重复的值

# 2.3 数据预处理

数据预处理的主要方法

- 聚集 (Aggregation):Combining two or more attributes (or objects) into a single attribute (or object)
- 抽样 (Sampling):是一种选择数据对象子集进行分析的常用方法
- 维归约 (Dimensionality Reduction)
- 特征子集选择 (Feature subset selection)
    - emmbedded approach
    - wrapper approach
    - filter approach
- 特征创建 (Feature creation)
    - 特征提取(Feature Extraction)
    - 映射数据到新的空间(Mapping Data to New Space)
    - 特征构造(Feature Construction)
- 离散化与二元化 (Discretization and Binarization)
    - 离散属性二元化
    - 连续属性离散化
- 属性变换 (Attribute Transformation)
    - 简单变换
    - 标准化(standardization)或规范化(normalization)

连续属性二元化:

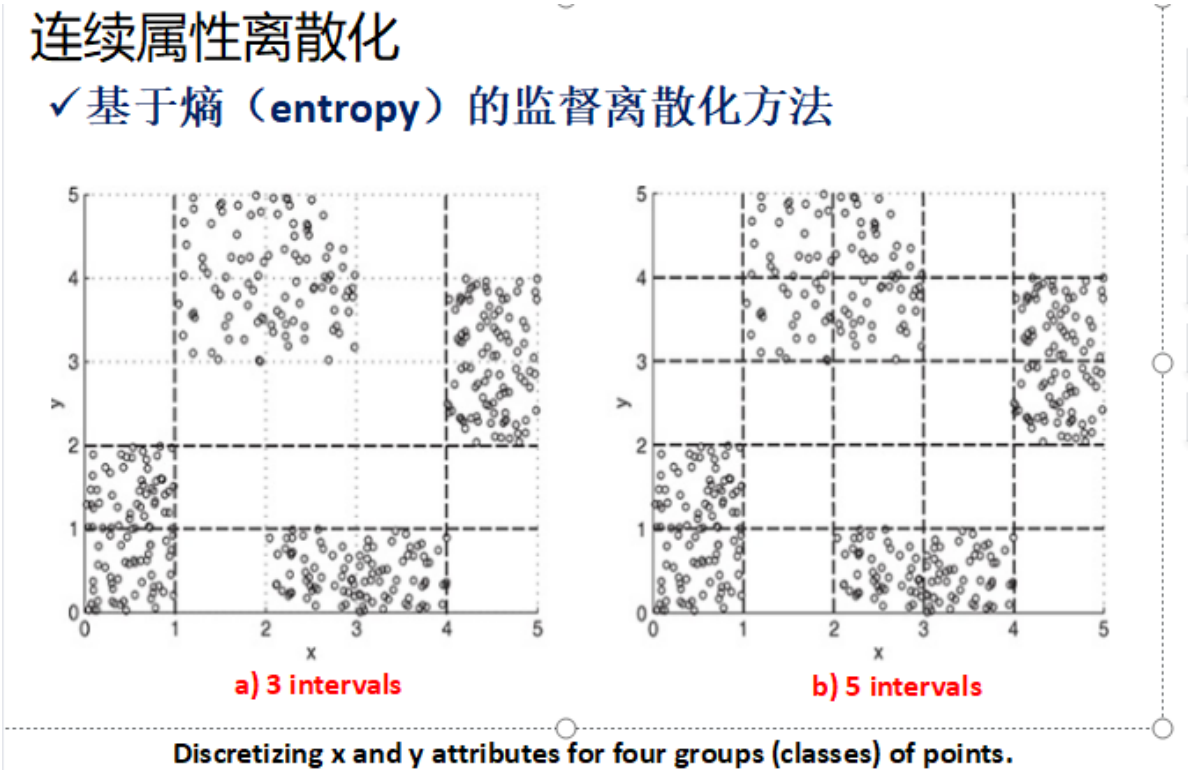
explain:

类信息:

In set theory and its applications throughout mathematics, a class is a collection of sets (or sometimes other mathematical objects) that can be unambiguously defined by a property that all its members share.

使用类信息(supervised)还是不使用类信息(unsupervised)

- 非监督离散化
- 监督离散化




Discretizing x and y attributes for four groups (classes) of points.

# 2.4 相似性和相异度的度量

## 2.4.3 数据对象之间的相异度

距离: 具有特定性质的相异度

Euclidean distance(欧式距离)

d(x,y) = $\sqrt  {\sum_{k=1}^{n} {(x_k-y_k)^2}}$

欧几里得距离可以用Minkowski distance来推广

$d(x,y) = \left(\sum_{k=1}^{n} {|x_k-y_k|^r}\right)^{\frac{1}{r}}$

- r = 1, Manhantann distance
- r = 2, Euclidean distance
- r = ∞, Supremun distance

$d(x,y) =\lim{x \to \infty} \left(\sum{k=1}^{n} {|x_k-y_k|^r}\right)^{\frac{1}{r}} = max{|x_k-y_k|,k=1,2, \dots, n}$

距离的性质

1. 非负性
2. 对称性
3. 三角不等式

满足以上三个性质称为度量(metric)

有些相异度无法满足度量性质

## 2.4.5 邻近性度量的例子

x,y为两个对象, 都由n个二元属性组成

- $f_{00}$: x取0 y取0
- $f_{01}$: x取0 y取1
- $f_{10}$: x取1 y取0
- $f_{11}$: x取1 y取1

1. 简单匹配系数(Simple Matching Coefficient)

   该度量对出现和不出现都进行计数

   SMC = $\frac{f*{11}+f*{00}}{f*{01}+f*{10}+f*{11}+f*{00}}$

   SMC可用于是非题就按测回答问题相似学生

2. Jaccard系数(Jaccard Coefficient)

   Jaccard 假定x和y是两个数据对象，代表一个事物矩阵的两行,忽略0-0匹配

   J = $\frac{f*{11}}{f*{01}+f*{10}+f*{11}}$

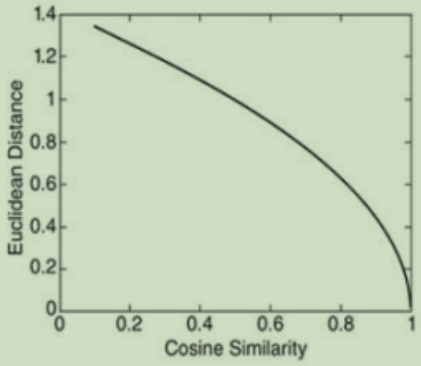
3. 余弦相似度

   忽略0-0匹配 && 处理非二元向量

   $\cos(x,y) = \frac{x \cdot y}{||x|| \ ||y||} = x'y'$

   - $||x||$为向量x的长度
   - $x'=\frac{x}{||x||}$: 长度为1的向量

   6 【教材第2章，习题20】关于余弦度量与欧氏距离的进一步考察。
   （a）余弦度量的取值范围是怎样的？
   （b）若两个对象（向量）的余弦为1，它们一定相等吗？
   （c）假设向量x和y的L2长度为1，即||x||=||y||=1，试推导x和y的欧氏距离||x-y||与余弦cos(x,y)之间的关系。
   在此基础上，阐述你从下图中获得的观测结论。

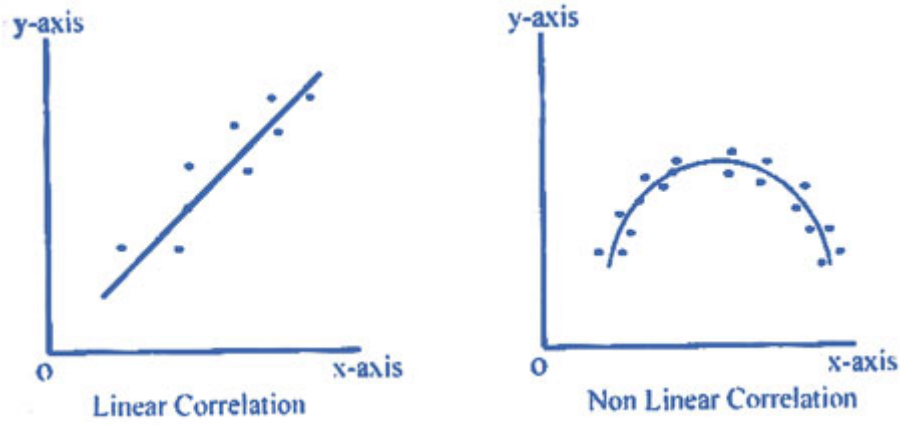
   欧氏距离与余弦之间的关系

   正确答案：
   （a）[-1, 1]
   （b）不一定；举例，x=(1,1), y=(2,2)，余弦为1，但不相等。
   （c）
   注意到，$\|X\| = \|Y\| = 1$
   则有：$\|X-Y\|^2 = (X-Y)^T(X-Y) = X^TX - 2X^TY + Y^TY = \|X\|^2 - 2X^TY + \|Y\|^2 = 1 - 2X^TY + 1 = 2(1 - X^TY)$
   且有：$X^TY = \|X\| \|Y\| cos(X,Y) = cos(X,Y)$
   所以，有：$\|X-Y\|^2 = 2(1 - cos(X,Y))$
   由此可知，X与Y的余弦相似度越大，则它们之间的欧氏距离就越小；反之亦然。

4. 皮尔森相关系数

   explaination:

线性: Correlation is said to be linear if the ratio of change is constant




对象之间的相关性是对象属性之间**线性**联系的度量

$corr(x,y) = \frac{covariance(x,y)}{standard_deviation(x)× standard_deviation(y)} = \frac{s_{xy}}{s_xs_y}$

$$\text{covariance}(\mathbf{x},\mathbf{y}) = s_{xy} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k-\bar{x})(y_k-\bar{y}) \qquad (2\text{-}11)$$

$$\text{standard\_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k-\bar{x})^2}$$

$$\text{standard\_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k-\bar{y})^2}$$

$$\bar{x} = \frac{1}{n}\sum_{k=1}^{n}x_k \text{ 是 } \mathbf{x} \text{ 的均值}$$

$$\bar{y} = \frac{1}{n}\sum_{k=1}^{n}y_k \text{ 是 } \mathbf{y} \text{ 的均值}$$

---

to be studied
stardard_deviation and covariance

---

4 【教材第2章，习题23】给定一个在区间[0, 1]取值的相似性度量，描述两种将该相似度变换成区间[0, ∞]中的相异度的方法。

正确答案：
$d = \frac{1-s}{s}, d = -log(s)$
$s \in [0, 1]$

5 【教材第2章，习题15】给定m个对象的集合，这些对象划分成K组，其中第i组的大小为m_i。如果目标是得到容量为n<m的样本，下面两种抽样方案有什么区别？（假定使用有放回抽样。）
(a) 从每组随机地选择n*m_i/m个元素。
(b) 从数据集中随机地选择n个元素，而不管对象属于哪个组。

正确答案：
方式（a）可以保证按比例从每一组中抽取到固定数目的样本，而方式（b）从每一组抽到的样本数目都是随机变化的，更进一步地说，若反复按照方式（b）进行多次抽样，则第i组抽到的样本数目的平均值近似于n*m_i/m。

# 第三章 探索数据

1. 汇总统计：众数、百分位数、中位数、极差
2. 可视化：盒状图

# 3.2 汇总统计

## 3.2.1 频率和众数

$frequency(v_i) = \frac{The\ object\ number\ has\ property\ vi}{m}$

众数：最高频率的值

## 3.2.2 百分位数 percentile

percentile:] is a score below which a given percentage k of scores in its frequency distribution falls (exclusive definition) or a score at or below which a given percentage falls (inclusive definition).

## 3.2.3 均值和中位数

$mean(x) = \overline{x} = \frac{1}{m}\sum_{i-1}^{m}x_i$

$median(x) = x_{r+1}: m = 2r+1$

$median(x) = \frac{1}{2}(x_r+x_{r+1}): m = 2r$

## 3.2.4 散布度量：极差和方差

连续数据的另一组常用的汇总统计是值集的**divergence(弥散)度量**

此度量表面属性是否散布很宽，或者是否相对集中在单个点(如均值)附近

极差(range): $range*{(x)}* = *max*{(x)} - min*{(x)}* = *x*{(m)} - x_{(1)}$

$$\text{variance}(x) = s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x})^2$$

$$\text{AAD}(x) = \frac{1}{m}\sum_{i=1}^{m}|x_i - \overline{x}|$$

$$\text{MAD}(x) = median\left(\{|x_1 - \overline{x}|, \ldots, |x_m - \overline{x}|\}\right)$$

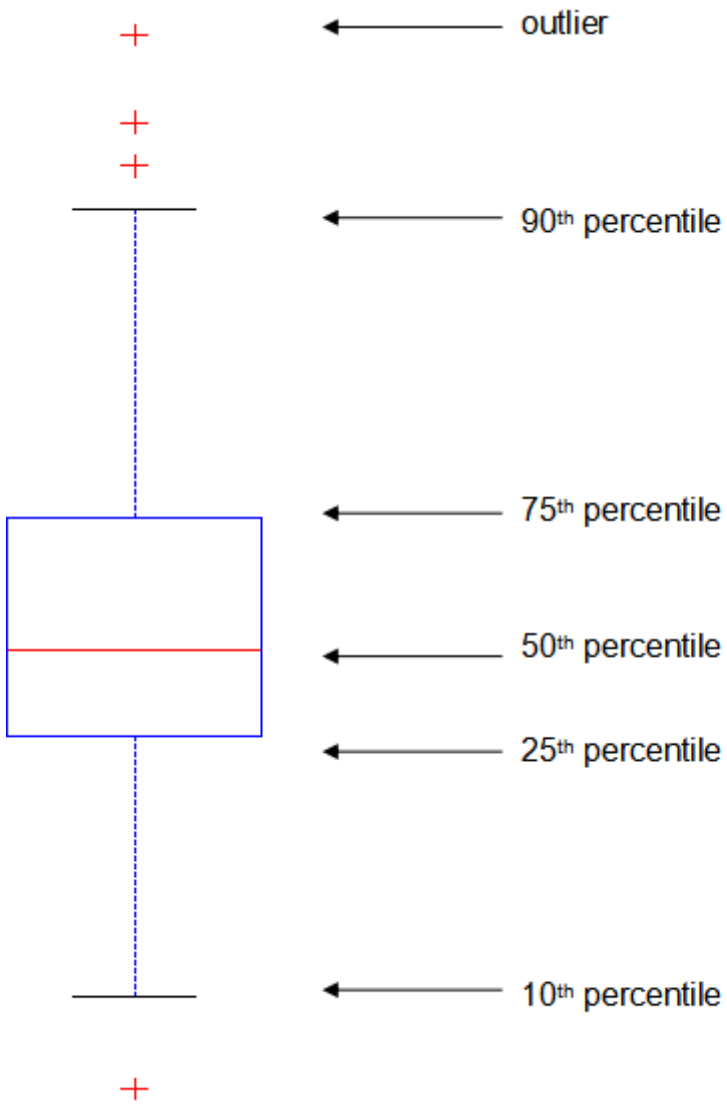
$$\text{interquartile range}(x) = x_{75\%} - x_{25\%}$$

# 3.3 可视化

## 3.3.3 技术

盒状图:

盒的下端和上端分别指示第25和第75个百分位数，而盒中的线指示第50个百分位数的值，底部和顶部的尾线分别指示第10和第90个百分位数，离群值用“+”显示




# 第四章 分类

1. 建立分类模型的一般方法
2. 决策树算法
    - 1）算法框架
    - 2）划分评估：基于熵的方法、基于Gini系数的方法、基于分类误差的方法
3. 分类模型的评估方法 （部分内容见课本的第5章

- 1）混淆矩阵|p303-304,y2|P91-92
- 2）各种性能指标
    - p303-311,y9
    - 准确率|P91
    - 错误率|P91
    - Precision|P181-182
    - Recall|P181-182
    - P-R曲线|p309
    - ROC曲线等高|P182-184
- 3）评估方法，包括保留法，K-折交叉验证法、自助法等。|p312-317,y6|P114-115

分类任务:

通过学习得到一个目标函数(target function), 把每个属性集x映射到一个预先定义的类标号y

# 4.2 解决分类问题的一般方法

建立分类模型的一般方法

- 基本方法
  - Logistic Regression（逻辑回归）
  - Decision Tree based Methods（决策树）
  - Rule-based Methods（基于规则的方法）
  - Nearest-neighbor（近邻方法）
  - Neural Networks（神经网络）
  - Deep Learning（深度学习）
  - Naïve Bayes and Bayesian Belief Networks（贝叶斯方法）
  - Support Vector Machines（支持向量机）
- 集成方法
  - Boosting（提升）
  - Bagging（装袋）
  - Random Forests（随机森林

# 4.2Next 解决分类问题的一般方法

1. 混淆矩阵(confusion matrix)

   定义:

   a confusion matrix, also known as an error matrix,  is a specific table layout that allows visualization of the performance of an algorithm,

| | | 预测的类标签 | |
|---|---|---|---|
| | | 正：P (1) | 负：N (0) |
| 真实的类标签 | 正：P (1) | TP ($f_{11}$) | FN ($f_{10}$) |
| | 负：N (0) | FP ($f_{01}$) | TN ($f_{00}$) |

   - TPR = $\frac{TP}{TP+FN}$
   - TNR = $\frac{TN}{FP+TN}$
   - FPR = $\frac{FP}{FP+TN}$
   - FNR = $\frac{FN}{TP+FN}$

   accuracy = $\frac{f*{11}+f*{00}}{f*{11}+f*{10}+f*{01}+f*{00}} = \frac{TP+TN}{N}$

   error_rate = $\frac{FP+FN}{N}$

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

   Recall(TPR) = $\frac{TP}{TP+FN}$

2. ROC曲线

   接受者操作特征(receive operating characteristic, ROC)曲线是显示分类器真正率(TPR)和假正率(FPR)之间折中的一种图形化方法

   关键点

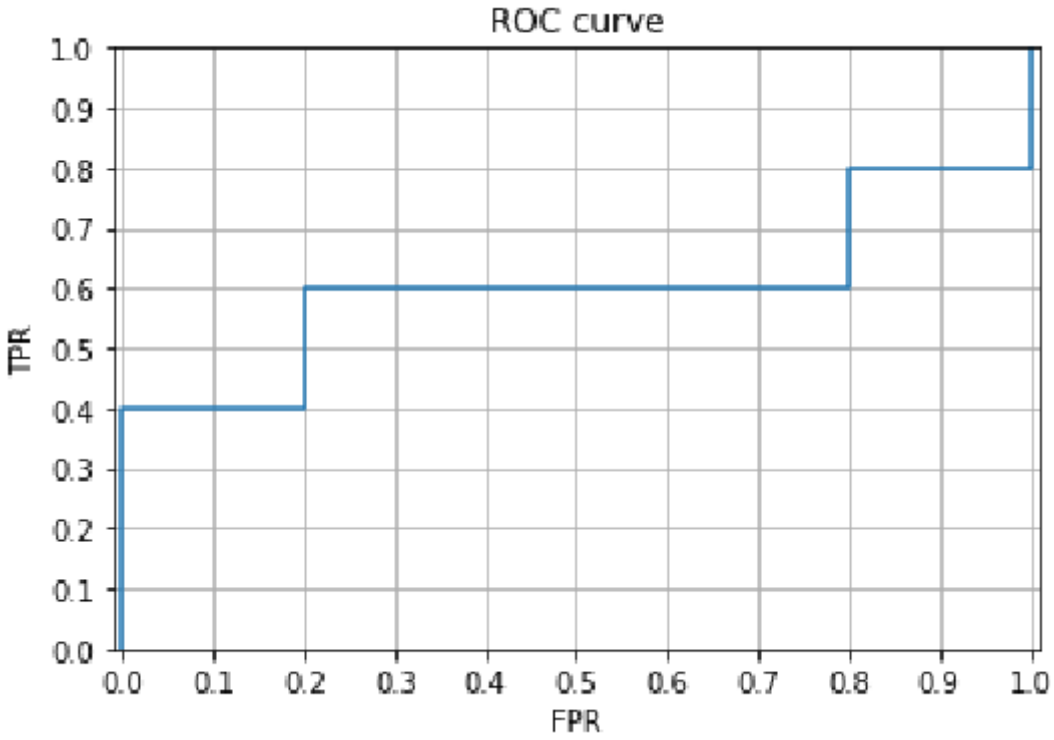
   - (TPR = 0.FPR =0):把每个实例都预测为负类的模型。
   - (TPR = 1, FPR= 1):把每个实例都预测为正类的模型。
   - (TPR = 1, FPR = 0):理想模型。

   等高: 有相同的TPR,但FPR逐渐变大

---

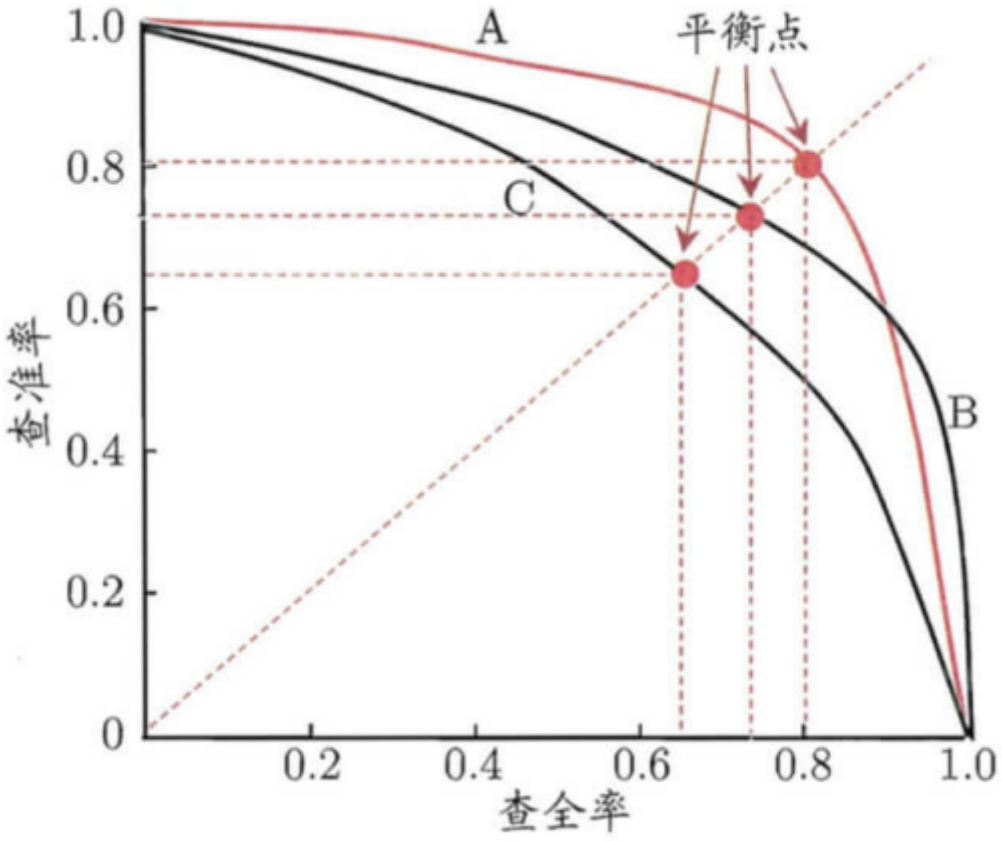
   question:

图5-42显示了一个如何计算 ROC 曲线的例子。检验集中有5个正实例和5个负实例。检验记录的类标号显示在表的第一行。第二行对应于每个记录排序后的输出值，例如，它们可能对应于朴素贝叶斯分类器产生的后验概率 $P(+|\mathbf{x})$。接下来的六行包括 *TP* 计数、*FP* 计数、*TN* 计数和 *FN* 计数，以及它们对应的 *TPR* 和 *FPR*。于是从左到右填表。开始时，所有的记录都被预测为正类，因此 $TP = FP = 5$，$TPR = FPR = 1$。然后，指派有最低输出值的检验实例为负类。因为选择的记录实际上是正类，因此 *TP* 计数从5减到4，而 *FP* 计数不变，并相应地更新 *TPR* 和 *FPR* 值。重复这个过程直至到达列表的末尾，这时 $TPR=0$，$FPR=0$。这个例子的 ROC 曲线如图5-43所示。




3. P-R曲线

Typically, Precision and Recall are inversely related, ie. as Precision increases, recall falls and vice-versa. A balance between these two needs to be achieved by the IR system, and to achieve this and to compare performance, the precision-recall curves come in handy.




书本未找到

# 4.3 决策树归纳

## 4.3.1 决策树的工作原理

决策数是一种由结点和有向边组成的层次结构,包括

- root node
- internal node
- leaf node or terminal node

## 4.3.3 表示属性测试条件的方法

- 二元属性
- 标称属性
- 序数属性
- 连续属性

## 4.3.4 选择最佳划分的度量

选择最佳划分的度量通常是根据划分后子女结点不纯(impurity)性的程度

impurity的程度越低, 类分布就越倾斜

选择增益最大的部分

增益: $\Delta = I(parent) - \sum_{j=1}^{k}\frac{N(v_j)}{N}I(v_j)$

- $N(v_j)$是与子女结点$v_j$相关联的记录个数

Gini:

$$Gini(t) = 1 - \sum_j [p(j \mid t)]^2$$

Entropy:

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

计算熵时, $Olog_20 = 0$

Classification error:

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

增益率Gain ratio:

$$GainRatio = \frac{Gain}{SplitINFO}$$

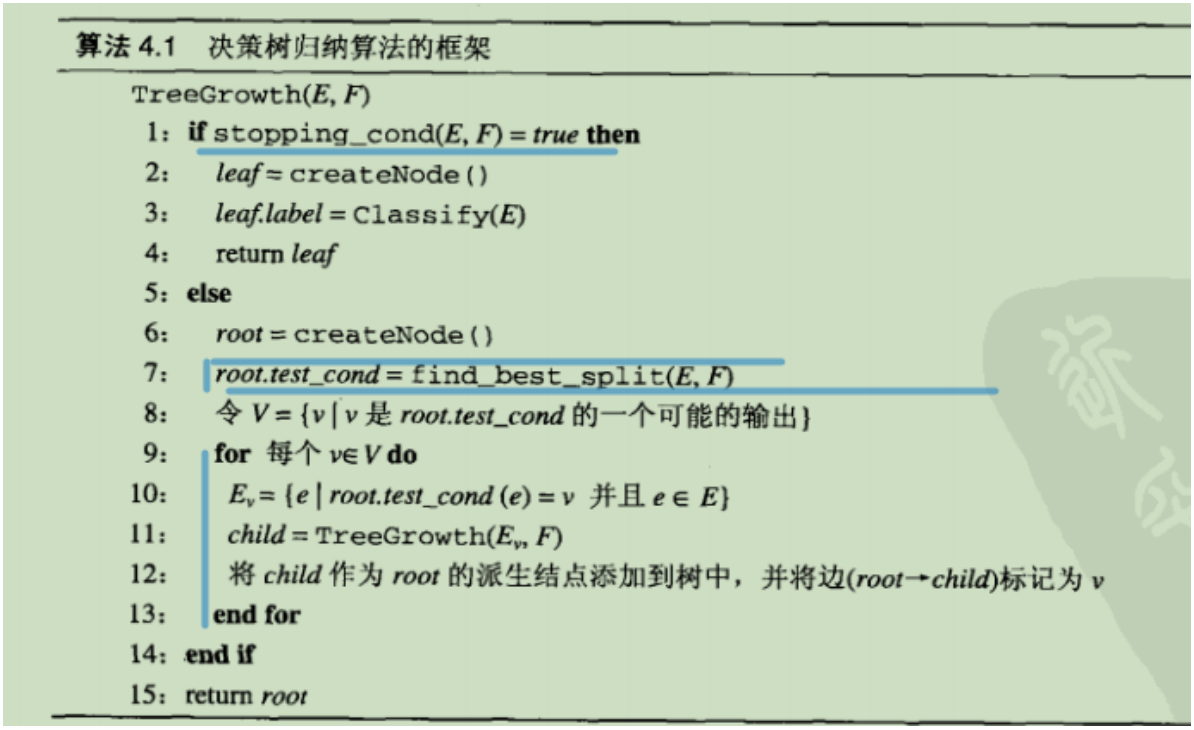
SplitINFO

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

---

question:

决策树算法框架



算法 4.1 决策树归纳算法的框架

```
TreeGrowth(E, F)
1: if stopping_cond(E, F) = true then
2:   leaf = createNode()
3:   leaf.label = Classify(E)
4:   return leaf
5: else
6:   root = createNode()
7:   root.test_cond = find_best_split(E, F)
8:   令 V = {v | v 是 root.test_cond 的一个可能的输出}
9:   for 每个 v∈V do
10:    E_v = {e | root.test_cond(e) = v 并且 e ∈ E}
11:    child = TreeGrowth(E_v, F)
12:    将 child 作为 root 的派生结点添加到树中，并将边(root→child)标记为 v
13:  end for
14: end if
15: return root
```

## 4.5 评估分类器的性能

hold out与k-fold cross validation方法的不足:

由于保留一部分样本用于检验，实际的训练集比D小，因此会引入一些因训练样本规模不同而导致的估计偏差。
leave-one-out方法受训练规模变化的影响较小，但是计算复杂度又太高 。

1. 保持法(hold out)

   将数据集D划分为两个互斥的集合，其中一个作为训练集S，另一个作为检验集T

2. 随机二次抽样

   多次重复保持方法来改进对分类器性能的估计

3. 交叉验证(cross-validation)

   - 二折交叉验证

   选择一个子集作训练集, 而另一个作检验集, 然后交换两个集合的角色

   - k折交叉验证

   把数据分为大小相同的k份, 在每次运行, 选择其中一份作检验集, 该过程重复k次, 使得每份数据都用于检验恰好一次.

   - Leave-one-out（留一法）

   k=n时的k-fold cross validation

4. 自助法(bootstrapping)

   训练记录采用有放回抽样, 即已经选作训练的记录将放回原来的记录集中, 使得它等几率地被重新抽取.

   对于n个样本组成的数据集D，有放回抽样n次，则一个样本始终未被抽中的概率是$(1-\frac{1}{n})^n$，当n足够大时，$(1-\frac{1}{n})^n \to \frac{1}{e} ≈ 0.368$

# 第五章 贝叶斯分类器

1）贝叶斯公式|p350,y2|P139-140

2）贝叶斯分类的一般方法-?

3）朴素贝叶斯分类器|p319-329,330-342,y24|P141-145

## 5.3 贝叶斯分类器

原理：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。

流程：根据具体情况确定特征属性，并对每个特征属性进行适当划分，然后由人工对一部分待分类项进行分类，形成训练样本集合。这一阶段的输入是所有待分类数据，输出是特征属性和训练样本。这一阶段是整个朴素贝叶斯分类中唯一需要人工完成的阶段，其质量对整个过程将有重要影响，分类器的质量很大程度上由特征属性、特征属性划分及训练样本质量决定。然后计算每个类别在训练样本中的出现频率及每个特征属性划分对每个类别的条件概率估计，并将结果记录。其输入是特征属性和训练样本，输出是分类器。最后使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。

Bayes theorem: 一种把类的先验知识和从数据中收集的新证据相结合的统计原理

条件概率:一随机变量在另一随机变量取值已知的情况下取某一特定值的概率

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

P(Y=y|X=x): 变量X取值x的情况下, 变量Y取值y的概率

X和Y的联合概率和条件概率满足:

$\displaystyle P(X,Y) = P(Y|X) × P(X) = P(X|Y) × P(Y)$

ie.

$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$

## 5.3.2 贝叶斯定理在分类中的应用

X表示属性集, Y表示类表量

P(Y|X): Y的后验概率 posterior probability

P(Y): Y的先验概率 prior probability

$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$

P(X)为常数

P(Y)可以通过训练记录占比估计

分类器任务:

给定一个记录对应的属性向量X，判断其对应的类标签y

$$\hat{y} = \arg\max_{c_j} P(y = c_j \mid \mathbf{X})$$

### 5.3.3 朴素贝叶斯分类器

在计算条件概率时, 假设属性之间是条件独立的,进而,将联合概率的计算简化成边缘概率的计算(朴素)

1. 条件独立性

$$P(\mathbf{X} \mid y = c_j) = P(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d \mid y = c_j) = \prod_{i=1}^{d} P(\mathbf{x}_i \mid y = c_j)$$

   类条件概率$\prod_{i}P(X_i|Y)$

2. 离散属性计算
   - 首先，将数据集中属于第j个类别的记录筛选出来，相应的记录个数为Nj；
   - 其次，统计其中第i个属性的取值为xi的记录个数mi;

   $P(xi | y=cj) = \frac{mi}{Nj}$

3. 条件概率的m估计(m-estimate)

   如果有一个属性的类条件概率等于0, 则整个类的后验概率就等于0

   解决该问题的途径是使用m估计

   $\displaystyle P(x_i) = \frac{n_c+mp}{n+m}$

# 第五章 人工神经网络ANN artificial neural networks

1）人工神经网络的要素：网络结构、激活函数、优化模型（损失函数）、优化算法

2）感知器的网络结构以及学习算法

3）三层前馈神经网络：网络结构、激活函数、基于均方误差损失函数的优化模型

决定人工神经网络性能的三大要素

1. 神经元的特性(**激活函数**)
2. 神经元之间相互连接的形式(**拓扑结构**)
3. 为适应环境而改善性能的**学习规则**

## 5.4.1 感知器

- 神经元：sign()
- 网络结构：两层
- 训练规则：给定训练数据集D，如何确定最优的w和b

感知器:

- 几个输入结点, 用来表示输入属性
- 一个输出结点, 用来提供模型输出
  - 输出结点是一个数学装置, 计算输入的加权和, 减去偏置项, 然后根据结果的符号产生输出
  - 数学方式表示:

---

ques:<=0?

因为<=0说明判断的类和预测的类相反

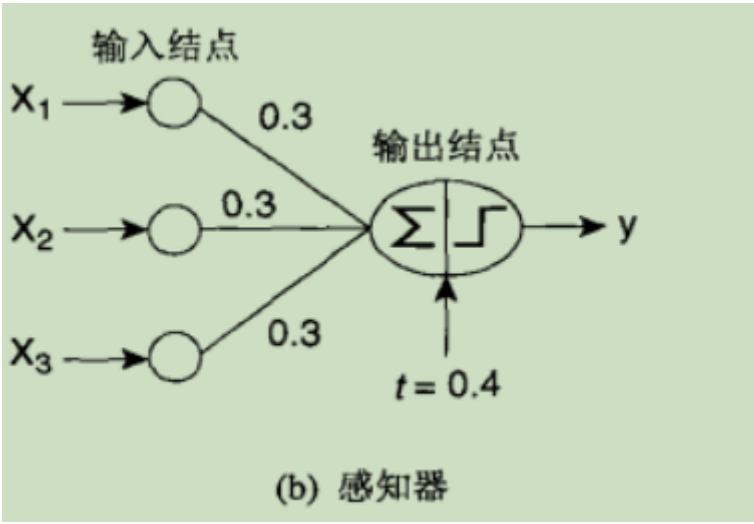
---

感知器学习算法

输入：训练数据集 $D=\{(X_i,y_i)\mid i=1,2,\ldots,N\}$,学习率 $\eta$

输出：$W,b$　　// 感知器模型 $f(X)=sign(W^TX+b)$

1. 初始化：$W\leftarrow W_0, b\leftarrow b_0$
2. **WHIEL** (存在某个数据 $(X,y)\in D$ 使得 $y(W^TX+b)\le 0$)
3. $W\leftarrow W+\eta yX,\quad b\leftarrow b+\eta y$
4. **RETURN** $W,b$



2 给定如下的训练样本集，现考虑采用感知器学习算法进行训练，以构建相应的分类判别函数$f(X)=sign(w_1\times x_1+w_2\times x_2+b)$。
设定学习率η=0.5，初始的参数w1=1，w2=-1, b=-0.4。
要求给出详细的感知器学习算法迭代过程以及最终获得的分类判别函数f(X)。

| 样本序号 | x1 | x2 | 类别y |
|---|---|---|---|
| 1 | 3 | 3 | 1 |
| 2 | 4 | 3 | 1 |
| 3 | 1 | 1 | -1 |

正确答案：

正类样本：$X_1=(3,3)^T$, $X_2=(4,3)^T$

负类样本：$X_3=(1,1)^T$

分类判别函数f(X)=sign($w_1$*$x_1$+$w_2$*$x_2$+b)

初始的参数($w_1$, $w_2$, b)=(1, -1, -0.4)
学习率设为η=0.5

| 样本序号 | x1 | x2 | 类别y |
|---|---|---|---|
| 1 | 3 | 3 | 1 |
| 2 | 4 | 3 | 1 |
| 3 | 1 | 1 | -1 |

| 迭代序号 | W | | | f(X) | | | 误分样本 | $\Delta W=\eta y_i X_i$ | |
|---|---|---|---|---|---|---|---|---|---|
| | w1 | w2 | b | f(X1) | f(X2) | f(X3) | | Δw1 | Δw2 |
| 1 | 1 | -1.00 | -0.40 | -1.00 | 1.00 | -1.00 | X1 | 1.50 | 1.50 |
| 2 | 2.50 | 0.50 | 0.10 | 1.00 | 1.00 | 1.00 | X3 | -0.50 | -0.50 |
| 3 | 2.00 | 0.00 | -0.40 | 1.00 | 1.00 | 1.00 | X3 | -0.50 | -0.50 |
| 4 | 1.50 | -0.50 | -0.90 | 1.00 | 1.00 | 1.00 | X3 | -0.50 | -0.50 |
| 5 | 1.00 | -1.00 | -1.40 | -1.00 | -1.00 | -1.00 | X1 | 1.50 | 1.50 |
| 6 | 2.50 | 0.50 | -0.90 | 1.00 | 1.00 | 1.00 | X3 | -0.50 | -0.50 |
| 7 | 2.00 | 0.00 | -1.40 | 1.00 | 1.00 | 1.00 | X3 | -0.50 | -0.50 |
| 8 | 1.50 | -0.50 | -1.90 | 1.00 | 1.00 | -1.00 | NULL | | |

最终构建的分类判别函数为f(X)=sign(1.5x1-0.5x2-1.9)

## 5.4.2 多层人工神经网络

前馈神经网络: 每一层的结点仅和下一层的结点相连

1. 网络结构
   - 输入层Input Layer
   - 隐藏层Hidden Layer
     - 中间的结点称为隐藏结点(hidden node)
   - 输出层Output Layer
2. 激活函数

   神经元中使用的激活函数(activation function), 除了符号函数之外, 还可以使用其它激活函数, those activation function 允许隐藏结点和输出结点的输出值与输入参数呈非线性关系

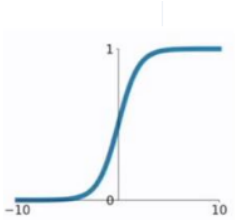
$$sign(z) = \begin{cases} 1 & if\ z \geq 0 \\ -1 & if\ z < 0 \end{cases}$$

$$sigmoid(z) = \frac{1}{1+e^{-z}}$$

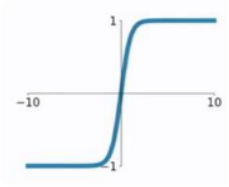
$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

$$ReLu(z) = \max\{0, z\}$$

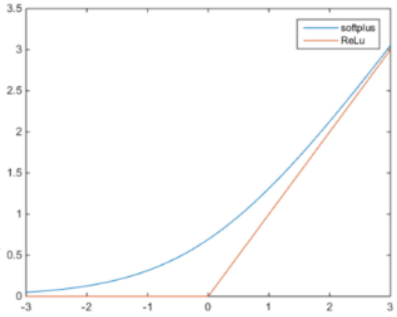
$$softplus(z) = \log(1 + e^{z})$$



sigmoid （logistic）函数



双曲正切函数




3. 损失函数

当结构给定的情况，神经网络对应的函数由参数（连接权重）W决定，即参数化的函数fW()

神经网络的训练,本质上是要找到合适参数W

1. 首先，需要有一个度量来衡量参数W的好坏，这个度量通常体现为损失函数loss(W)。
2. 其次，在损失函数的基础上，将参数W的确定归结为一个优化问题。

$$D = \{(X_i, Y_i) \mid i = 1, 2, \ldots, N\}$$

$$\min_{W} \ loss(W \mid D) = \sum_{i=1}^{N} loss(W \mid (X_i, Y_i))$$

均方误差损失函数:

$$均方误差损失：\ (y - f_W(X))^2$$

$$训练数据集 D = \{(X_i, Y_i) \mid i = 1, 2, \ldots, N\}$$

$$X_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})^T, Y_i = (y_i^{(1)}, y_i^{(2)}, \ldots, x_i^{(m)})^T$$

采用均方误差损失函数，构建优化模型：

$$\min_{W,\theta,V,\gamma} E = \sum_{k=1}^{N} E_k = \sum_{k=1}^{N} \frac{1}{2} \| \hat{Y}_k - Y_k \|^2 = \sum_{k=1}^{N} \left( \frac{1}{2} \sum_{j=1}^{m} (\hat{y}_k^{(j)} - y_k^{(j)})^2 \right)$$

$\hat{Y}_k = (\hat{y}_k^{(1)}, \hat{y}_k^{(2)}, \ldots, \hat{y}_k^{(m)})^T$ 为 $X_k$ 对应的网络输出

$E_k = \frac{1}{2} \sum_{j=1}^{m} (\hat{y}_k^{(j)} - y_k^{(j)})^2$ 则为相应的误差

网络参数：

$V$：$n \times q$个权值$v_{ih}$

$W$：$q \times m$个权值$w_{hj}$

$\theta$：$m$个阈值$\theta_j$

$\gamma$：$q$个阈值$\gamma_h$

# 第五章 集成学习

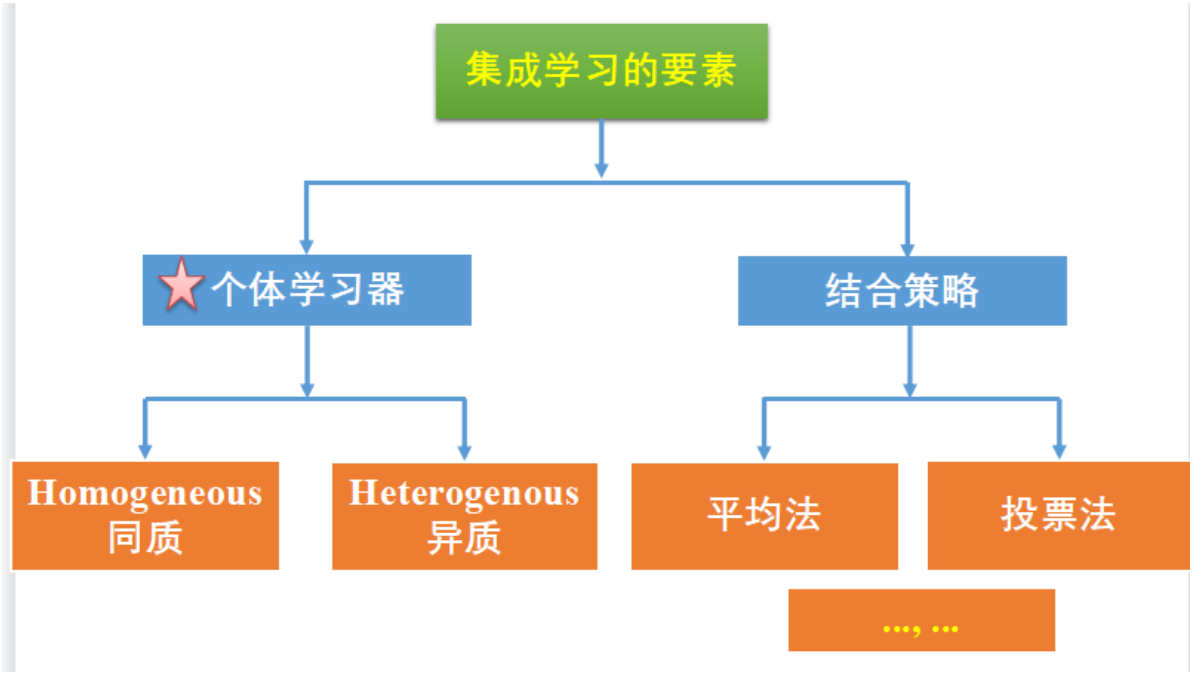
1）集成学习的一般框架及要素|p435-436,y2

2）Bagging算法（算法流程）|p439-446,y8|P173-175

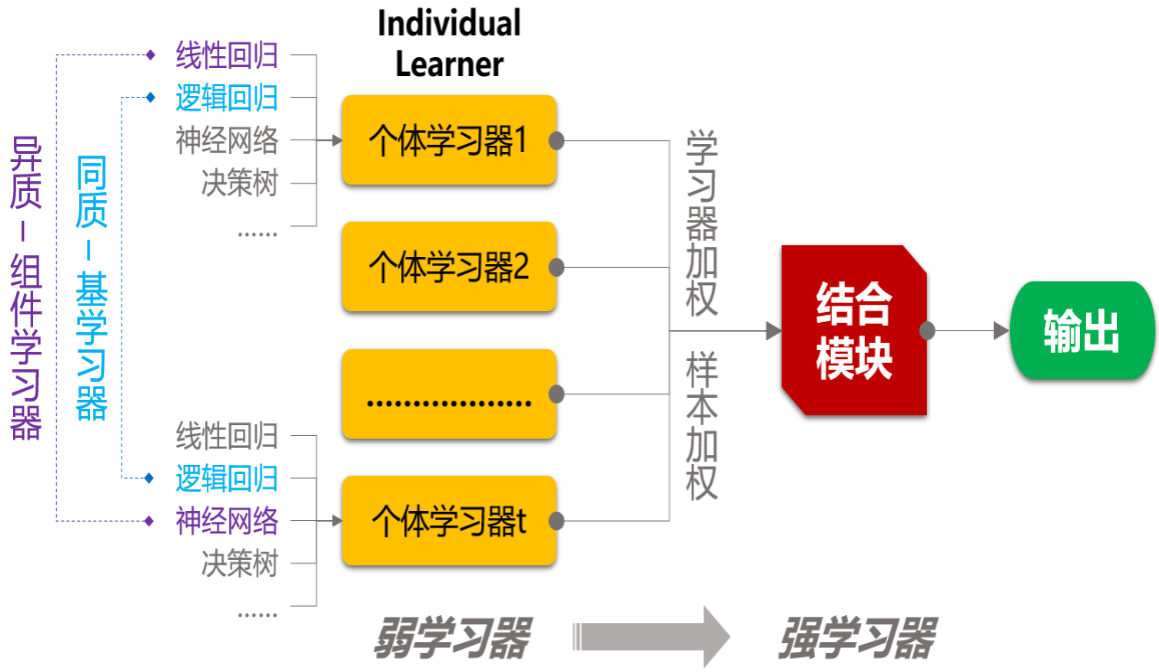
3）AdaBoost算法（基本思想）|p447-456,y10|P175-178

## 集成学习的要素

要素:

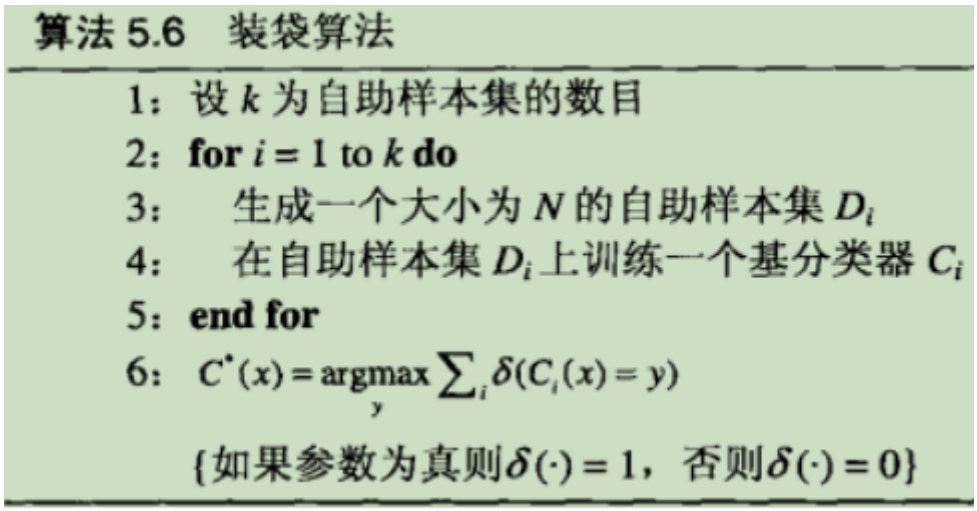



一般框架:




## Bagging算法

Bagging 又称为 boot strap aggregating, 是一种根据均匀概率分布从数据集中重复抽样(有放回)的技术



**算法 5.6　装袋算法**

1：设 $k$ 为自助样本集的数目
2：**for** $i = 1$ to $k$ **do**
3：　生成一个大小为 $N$ 的自助样本集 $D_i$
4：　在自助样本集 $D_i$ 上训练一个基分类器 $C_i$
5：**end for**
6：$C^*(x) = \underset{y}{\operatorname{argmax}} \sum_i \delta(C_i(x) = y)$

{如果参数为真则 $\delta(\cdot) = 1$，否则 $\delta(\cdot) = 0$}

## AdaBoost算法

AdaBoost将每一个分类器$C_j$的预测值根据$a_j$进行加权，而不采用多数表决的方案，这种机制有助于AdaBoost惩罚那些准确率很差的模型，如那些在较早提升轮产生的模型。另外，如果任何中间伦产生高于50%的误差，则权值将被恢复为开始的一致值$w_i = \frac{1}{n}$

并在每轮训练中增加被错误分类样本的权值，并减少已经被正确分类的样本的权值

最终预测结果通过取每个基分类器的加权平均得到

---

question: why 0.9 not boost???

why the back not be effected???

---

# 第六七章 关联分析

1. 基本概念：

   项集、K-项集、支持度计数、支持度、频繁项集、关联规则、关联规则的支持度和置信度

2. 先验原理
3. 产生频繁项集的Apriori算法
   - 1）算法框架
   - 2）产生候选频繁项集的Fk-1Fk-1方法
4. 序列模式发现
   - 1）序列的基本概念
   - 2）子序列与序列包含
   - 3）序列模式发现的类Apriori算法

---

关联分析(association analysis):

用于发现隐藏在大型数据集中的有意义的联系,所发现的联系可以用关联规则(association rule)或频繁项集的形式表示

例如: {尿布}->{啤酒}

# 6.1 问题定义

- 项集(Itemset)

  a set include zero or multiple items

- k-项集

  including k items

- 项集支持度计数(Support count)

  $σ(X) = |{t_i|X\subseteq t_i,t_i\in T }|$

- 项集支持度(Support)

  $s = σ(X)/amount$

关联规则:形如 X->Y 的蕴含表达式

1. 支持度(support):给定数据集的频繁程度

   $\displaystyle s(X->Y) = \frac{σ(X\cup Y)}{N}$

2. 置信度(confidence):确定Y在包含X的事务中出现的频繁程度

   $\displaystyle c(X->Y) = \frac{σ(X\cup Y)}{σ(X)}$

关联规则挖掘问题:

给定事务的集合T,找出支持度≥ minsup并且置信度≥minconf的所有规则

大多数关联规则挖掘算法通常采用的策略是, 将关联规则挖掘任务分解为如下两个主要的子任务

1. 频繁项集产生(Frequent Itemset Generation)

   发现满足最小**支持度**阈值的所有项集, 这些项集称作频繁项集(frequent itemset)

2. 规则的产生(Rule Generation)

   从上一步发现的频繁项集中提取所有高**置信度**的规则, 这些规则称作强规则(strong rule)

## 6.2 频繁项集的产生

一个包含k个项的数据集可能产生$2^k-1$个频繁项集, 不包括空集

有几种方法可以降低产生频繁项集的计算复杂度

1. 减少候选项集的数目(M):如先验原理
2. 减少比较次数

## 6.2.1 先验原理

定义: If a simple pattern is not supported, then a more complicated one with that simple pattern in it can not be supported (e.g. if AC isn't supported, there is no way that ABC is supported)

This process of removing patterns that can't be supported because their subsets (or shorter combination) aren't supported is called pruning. (support-based pruning)

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

## 6.2.2 Apriori算法的频繁项集产生

```
k = 1
{发现所有频繁1-项集}
repeat
    k = k + 1
    {产生候选项集}
    for 每个事务 t ∈ T do
        {识别属于t的所有候选}
        for 每个候选项集 c ∈ Ct do
           {支持度技术增值}
        end for
    end for
   {提取频繁k-项集}
until Fk = ∅
Result = ∪ Fk
```

算法描述:

- $F_k$: frequent k-itemsets (频繁k-项集的集合)
- $C_k$: candidate k-itemsets (候选k-项集的集合)

---

Let k = 1

Generate F1 = {frequent 1-itemsets}

Repeat until $F_k$ is empty

- k = k + 1
- Candidate Generation: Generate $C_k$ from $F_k-1$
- Candidate Pruning: Prune candicate itemsets in $C_k$ containing subsets of length k-1 that are infrequent
- Support Counting: Count the support of each candidate in $C_k+1$ by scanning the DB
- Candicate Elimination: Eliminate candidates in $C_k$ that are infrequent, leaving only those that are frequent=>Fk

Output frequent itemsets $\cup F_k$

---

候选集产生方法-要求

1. 避免产生太多不必要的候选
2. 确保候选项集的集合是完全的
3. 不产生重复候选项集

$F*{k-1}* × *F*{k-1}$

Merge two frequent (k-1)-itemsets if the last (k-2) items of the first one is identical to the first (k-2) items of the second.

在两个频繁的Fk-1中,一个截去头部一个截去尾部,再将头尾补回Fk-2中,即形成Fk

question:

这个例子表明了候选项产生过程的完全性和使用字典序避免重复的候选的优点。然而，由于每个候选都由一对频繁$(k-1)$-项集合并而成，因此需要附加的候选剪枝步骤来确保该候选的其余$k-2$个子集是频繁的。

UNDERSTAND

(c) What is the pruning ratio of the *Apriori* algorithm on this data set? (Pruning ratio is defined as the percentage of itemsets not considered to be a candidate because (1) they are not generated during candidate generation or (2) they are pruned during the candidate pruning step.)

Answer:

## 7.4 序列模式

### 7.4.1 问题描述

序列: 将与对象A有关的所有时间按时间戳增序排序, 就得到对象A的一个序列(sequence)

子序列: 如果t中每个有序元素都是s中一个有序的子集,序列t是另一个序列s的子序列(subsequence).

ie...:存在整数$1 \le j_1 \le j_2 \dots \le j_m \le n$, 使得$t_1 \subseteq s*{j_1}, t_2 *subseteq* s{j_2},\dots,t_m \subseteq s_{j_m}$

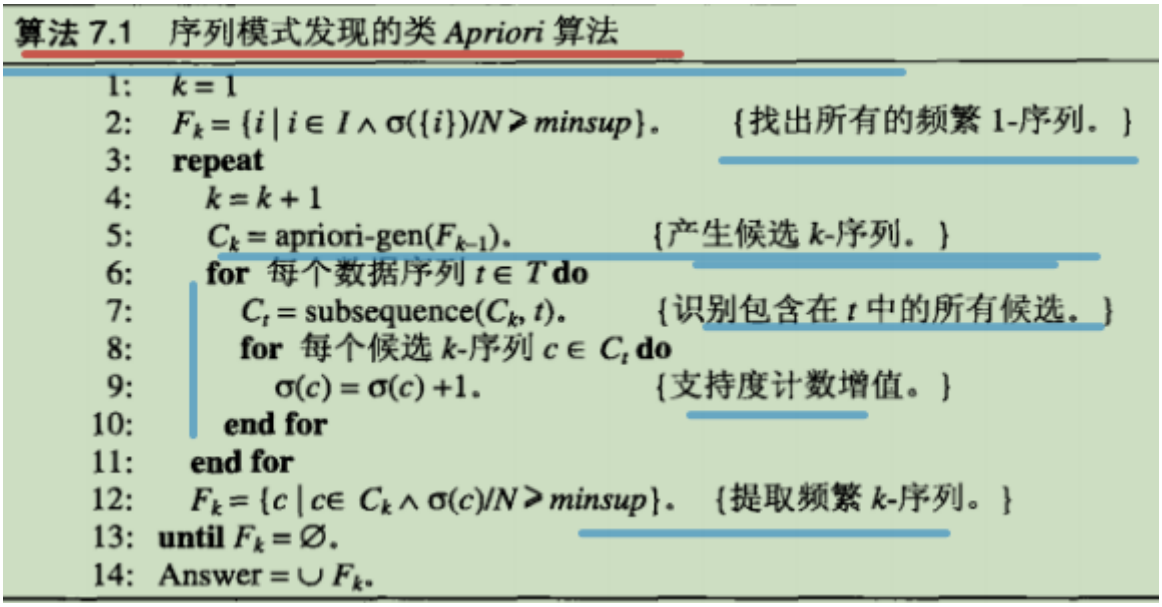
### 7.4.2 序列模式发现

数据序列: 是指与单个数据对象相关联的事件的有序列表

序列s的支持度: 包含s的所有数据序列所占的比例

序列模式发现:给定序列数据库D和用户指定的最小支持度阈值minsup，序列模式发现的任务是找出支持度大于或等于minsup的所有序列 。

序列模式发现的类Apriori算法



算法 7.1　序列模式发现的类 *Apriori* 算法

```
1:  k = 1
2:  F_k = {i | i ∈ I ∧ σ({i})/N ≥ minsup}.      {找出所有的频繁 1-序列。}
3:  repeat
4:    k = k + 1
5:    C_k = apriori-gen(F_{k-1}).               {产生候选 k-序列。}
6:    for 每个数据序列 t ∈ T do
7:      C_t = subsequence(C_k, t).              {识别包含在 t 中的所有候选。}
8:      for 每个候选 k-序列 c ∈ C_t do
9:        σ(c) = σ(c) +1.                       {支持度计数增值。}
10:     end for
11:   end for
12:   F_k = {c | c∈ C_k ∧ σ(c)/N ≥ minsup}.     {提取频繁 k-序列。}
13: until F_k = ∅.
14: Answer = ∪ F_k.
```

# 第八章 聚类分析

1. 聚类分析的任务描述
2. 聚类任务（无监督学习）与分类任务（有监督学习）的不同
3. K-均值聚类算法
4. 凝聚层次聚类算法
5. DBSCAN算法的基本原理

## 聚类分析

将数据对象分组，使得同一组内的对象彼此相似（或相关），而不同组中的对象是不同的（或不相关）；组内的相似性（同质性）越大，组间差别越大，则聚类（分组）越好。

## 聚类任务（无监督学习）与分类任务（有监督学习）的不同

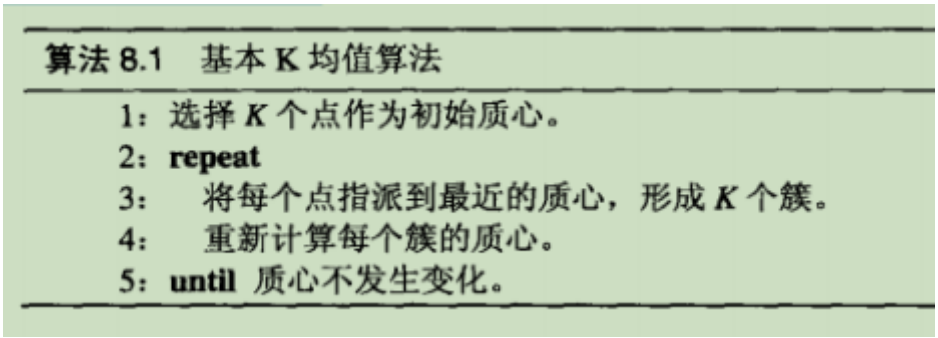
- 聚类是非监督分类(unsupervised classification), 它用类(簇)标号创建对象的标记. 然而, 只能从数据导出这些标号.
- 相比之下, 分类任务是监督分类(supervised classification), 即使用由类标号已知的对象开发的模型.

### 8.1.3 不同的簇类型

- K均值: 基于原型的、划分的聚类技术。试图发现用户指定个数(K)的簇(由质心代表 簇的原型通常是质心，即簇中所有点的平均值)
- 凝聚的层次聚类：开始，每个点作为一个单点簇；然后，重复地合并两个最靠近的簇，直到产生单个的、包含所有点的簇。其中某些技术可以用于基于图的聚类接受，而另一些可以用基于原型的方法解释
- DBSCAN：一种产生划分聚类的基于密度的聚类算法，簇的个数由算法自动地确定。低密度区域中的点被视为噪声而忽略，因此DBSCAN不产生完全聚类

### K-均值聚类算法

首先，选择K个初始质心，其中K为用户指定的参数，即所期望的簇的个数。每个点指派到最近的质心，而指派到一个质心的点集为一个簇。然后根据指派到簇的点，更新每个簇的质心，直到簇不发生变化。



**算法 8.1　基本 K 均值算法**

1：选择 $K$ 个点作为初始质心。
2：**repeat**
3：　将每个点指派到最近的质心，形成 $K$ 个簇。
4：　重新计算每个簇的质心。
5：**until** 质心不发生变化。

## 凝聚的层次聚类

1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5: **until** Only one cluster remains.

- 单链：两个簇的邻近度定义为两个不同簇中任意两点之间最短距离（最大相似度）
- 全链：两个簇的邻近度定义为两个不同簇中任意两点之间最长距离（最小相似度）

## DBSCAN是基于密度的算法

- 密度: 指定半径Eps内的点数
- 核心点: 如果点X的密度超过给定的阈值MinPts(这些是在聚类内部的点
- 边界点: 点X的密度不超过给定的阈值MinPts，但是落在某个核心点的Eps邻域内(这些点在核心点附近
- 噪声点指核心点或边界点之外的其它点。

算法:

- 1）将所有点标记为核心点、边界点或噪声点
- 2）删除所有的噪声点
- 3）对剩余点执行聚类

# 第十章 异常检测

1.异常检测的应用
2.离群点
3.异常检测的主要技术方法

## 1. 异常检测的应用

1. 欺诈检测
2. 天气预测
3. 电子商务
4. 公共安全
5. 入侵检测
6. 医疗

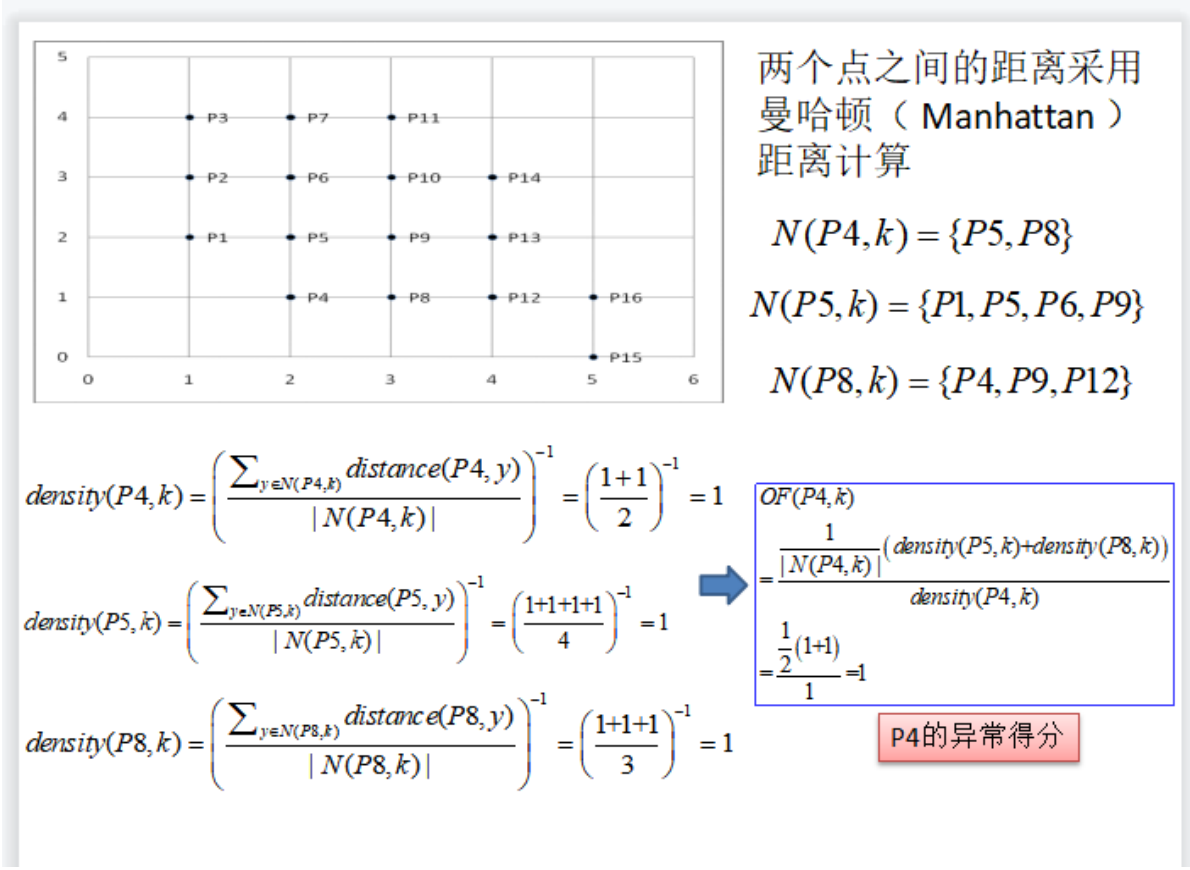
# 2.离群点

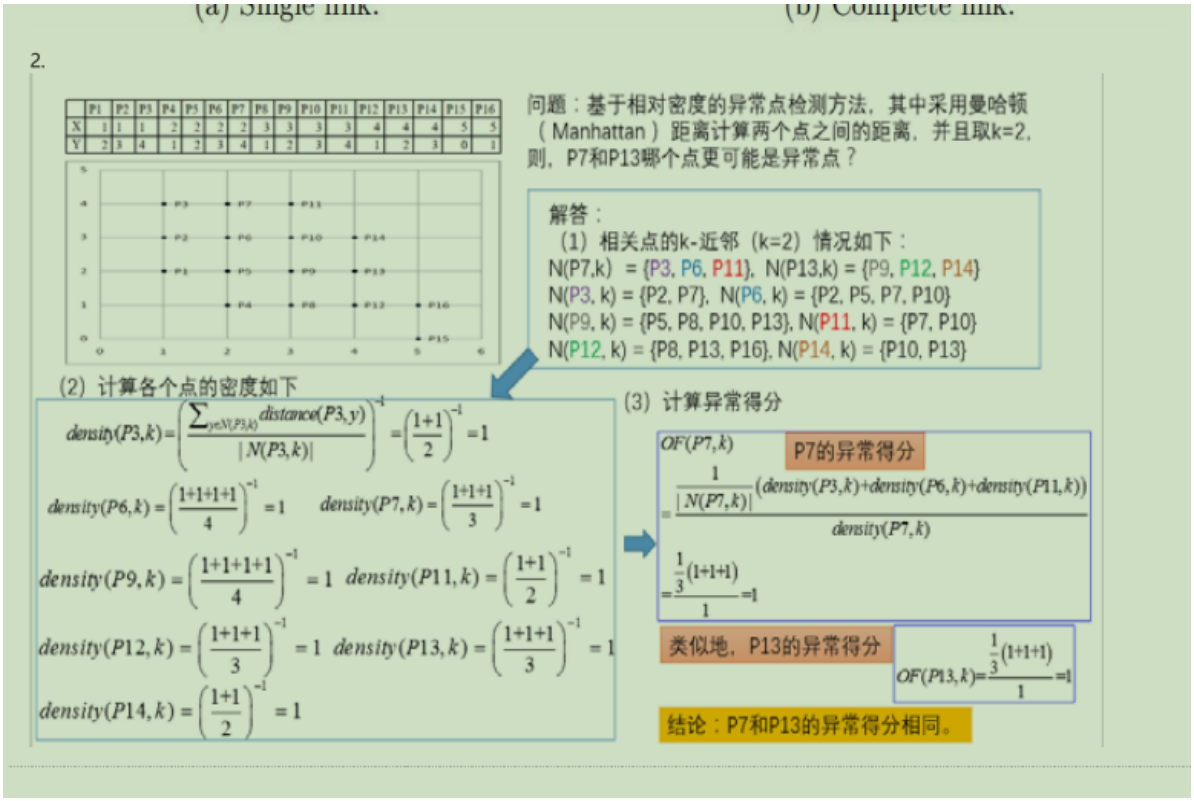
异常对象通常也叫“离群点”（outlier）

- Hawkins：异常是在数据集中偏离大部分数据的数据，使人怀疑这些数据的偏离并非由随机因素产生，而是产生于完全不同的机制。
- Weisberg：异常是与数据集中其余部分不服从相同统计模型的数据。
- Samuels：异常是足够地不同于数据集中其余部分的数据。
- Porkess：异常是远离数据集中其余部分的数据

# 3.异常检测的主要技术方法

1. 按类标号（正常/异常）利用的程度
    - 无监督的异常检测方法：没有提供类标号
    - 有监督的异常检测方法：存在异常点类和正常类的训练集
    - 半监督的异常检测方法：训练数据包含被标记的正常数据，但是没有关于异常对象的信息
2. 按使用的主要技术路线角度
    - 基于统计的方法
    - 基于邻近度的方法
    - 基于密度的方法
    - 基于聚类的方法






# 附录 回归

1. 线性回归
2. Logistic回归

回归定义:回归的目标是找到一个可以最小误差拟合输入数据的目标函数.

回归任务的误差函数(error function)可以用绝对误差/平方误差和表示

- 绝对误差: $\sum{i}|y_i - f(x_i)|$
- 平方误差: $\sum{i} \left( y_i - f(x_i) \right)^2$

## D.2 简单线性回归

假设目标函数是线性函数: $y = f(x) = β0 + β1x$

$$\text{误差（损失）函数：} E(\beta_0, \beta_1) = \sum_{i=1}^{N} (y_i - f(x_i))^2$$

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\min_{\beta_0, \beta_1} E(\beta_0, \beta_1) = \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_i)^2$$

### Logistic回归

In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc.

给定观测数据集$D=\{(\mathbf{x}_i, y_i) \mid i=1,2,\cdots,N\}$

$\mathbf{x}_i \in R^{d\times 1}$：第$i$个观测数据的输入（属性集、自变量）

$y_i \in \{+1,-1\}$：第$i$个观测数据的输出（类标签）

根据数据集$D$，学习一个映射（分类函数）

$$y=f(\mathbf{x}): \mathbf{x} \in R^{d\times 1} \mapsto y \in R$$

使得利用$f$能正确预测未见输入$\mathbf{x}$对应的输出$y$