数据科学理论基础：统计学

北京邮电大学

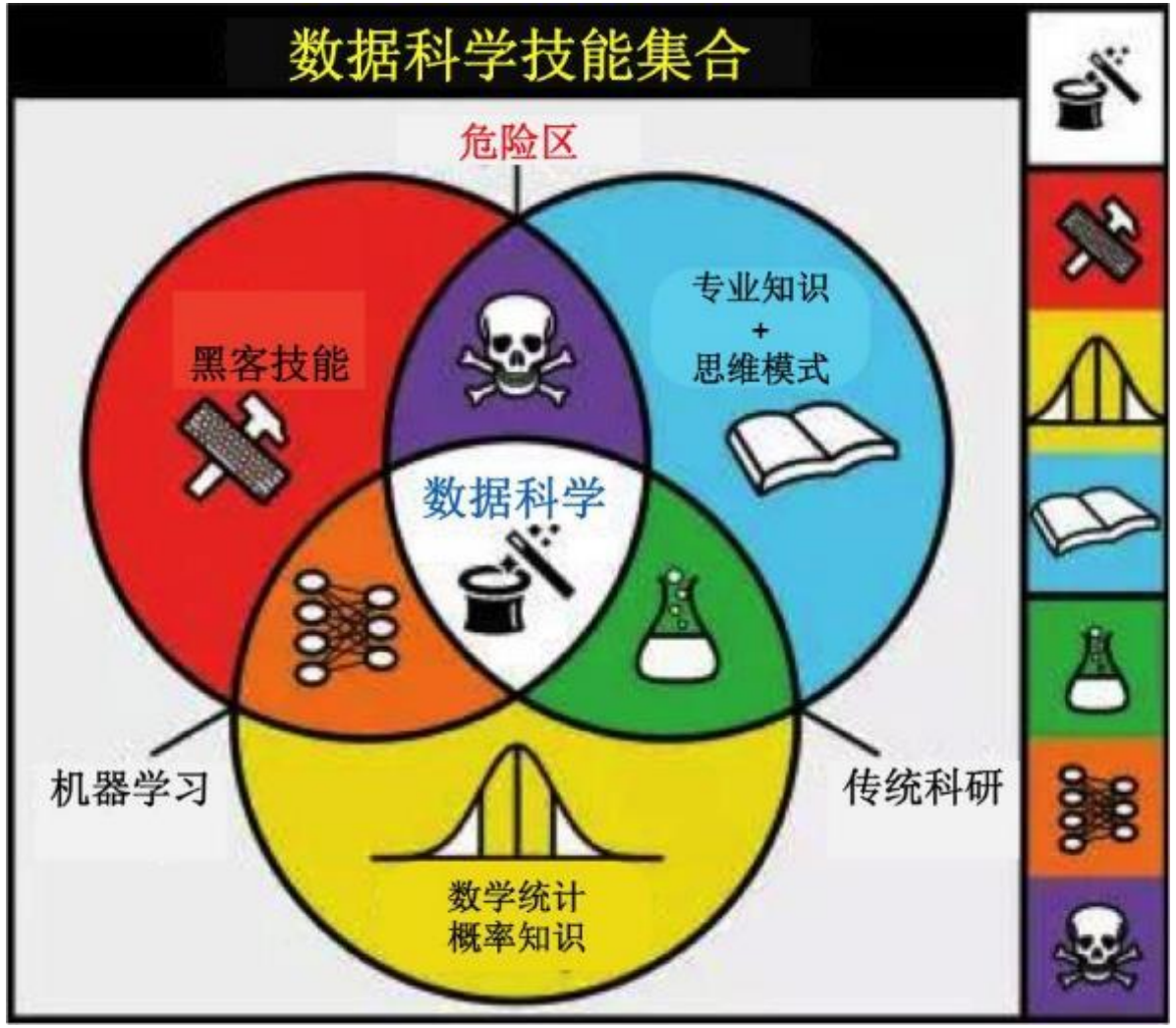
智能感知与计算教研中心

数据科学中心

统计学与数据科学

统计学是数据科学的理论基础之一

- 数据科学的文氏图（韦恩图）



统计学原理

统计学原理

- 统计学基本概念
  - ✓ “统计学是收集、分析、表述和解释数据的科学”
  - ✓ “统计学是一组方法，用来设计实验，获得数据，然后在这些数据的基础上组织、概括、演示、分析、解释和得出结论”
  - ✓ ……

统计学是关于数据资料的收集、整理、分析和推断的一门科学。

统计学的分类

- 描述统计学(descriptive statistics)

  数据收集、处理、汇总、图表描述、概括与分析等统计方法。

- 推断统计学(inferential statistics)

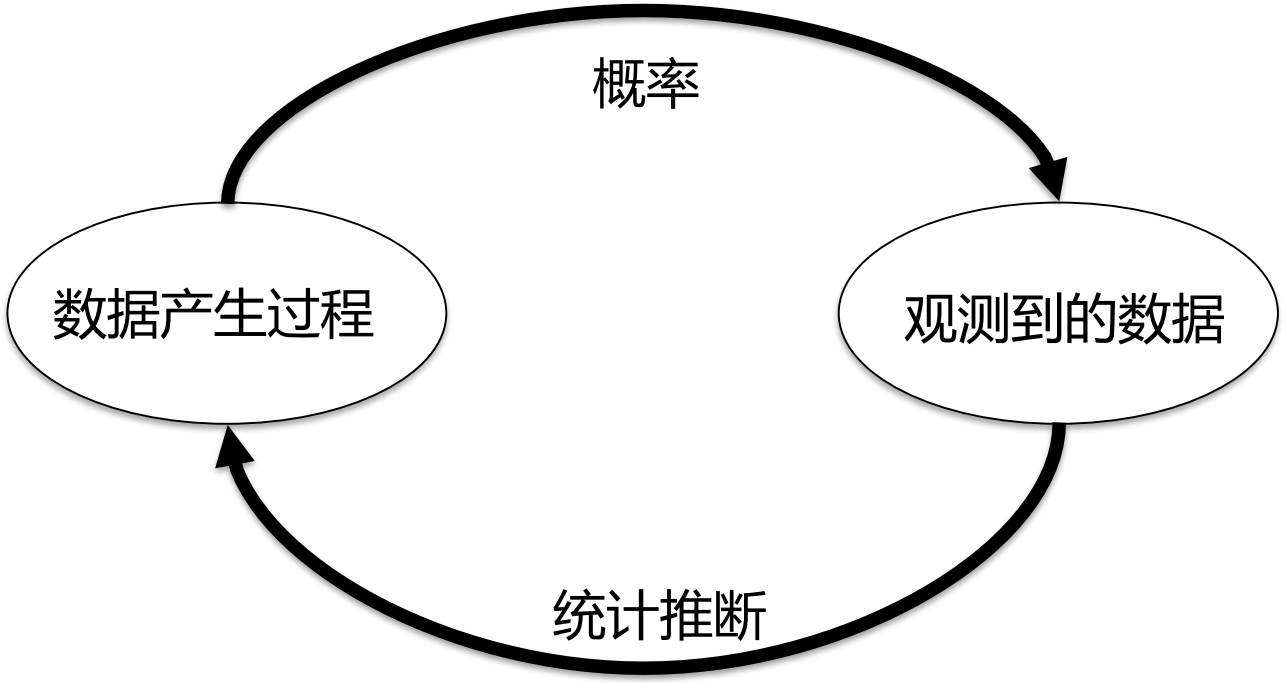
  研究如何利用样本数据来推断总体特征的统计方法。

  需要抽取部分个体即样本进行测量,然后根据获得的样本数据对所研究的总体特征进行推断。

  与数据挖掘和机器学习是近亲。

统计学同概率之间的关系

- 推断统计学与概率密不可分，是我们要研究的主要内容



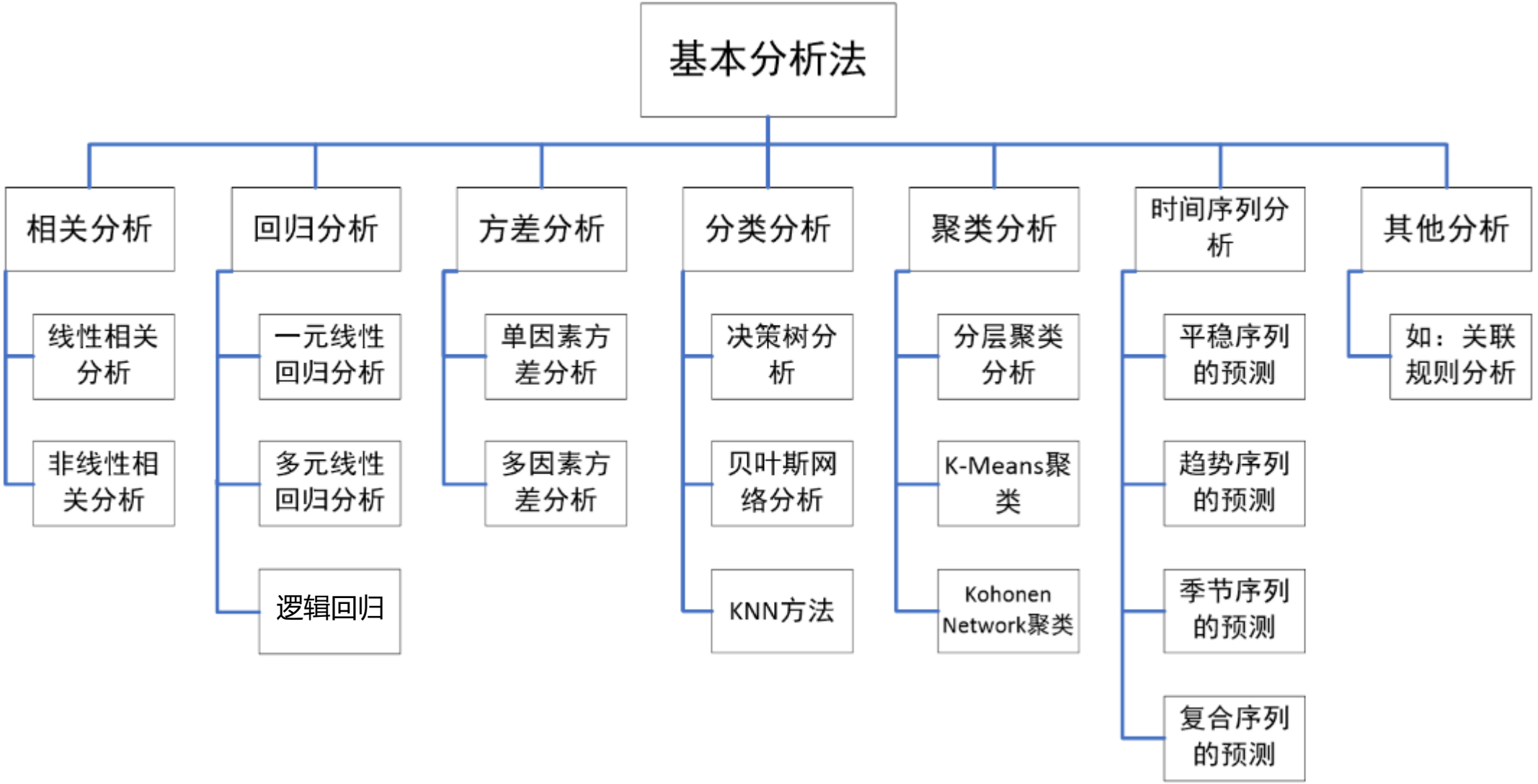
概率： 一个事件或事件集合出现的可能性

当事件不可精确计数

统计：对一个事件或事件集合进行调查整理出结果

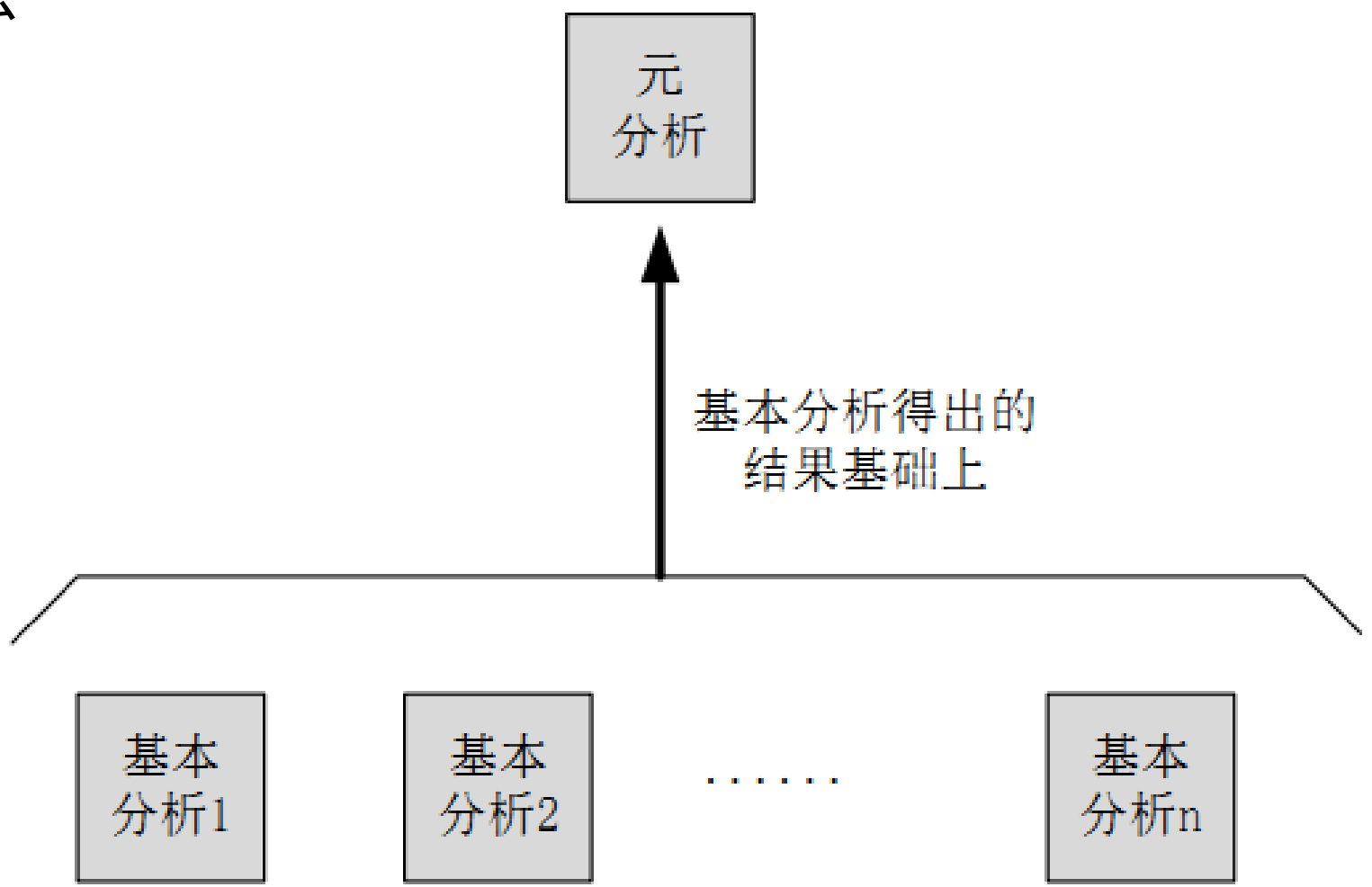
数据科学中常用的统计学方法

- 基本分析法

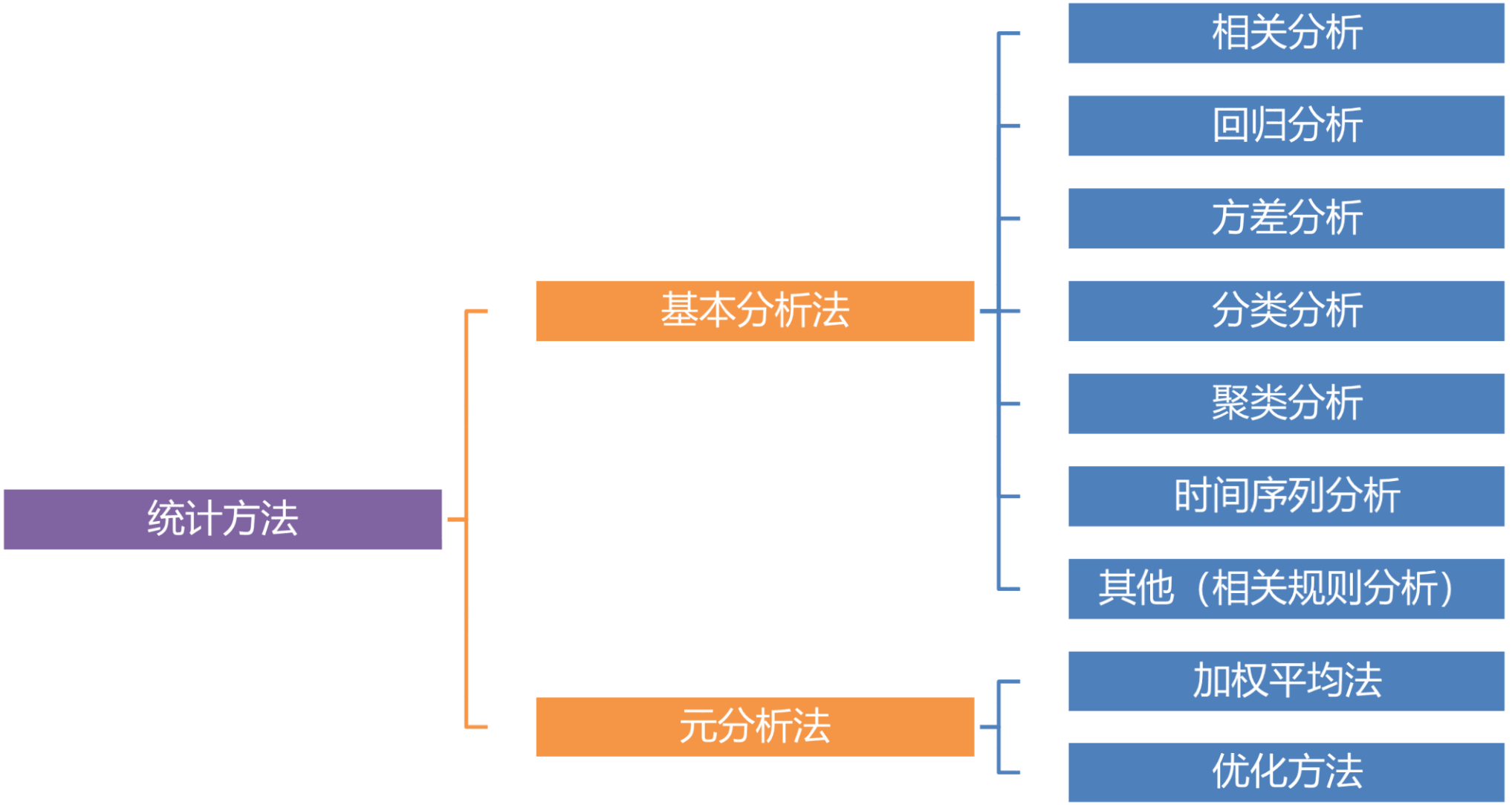


数据科学中常用的统计学方法

- 元分析法



方法的两个层次小结

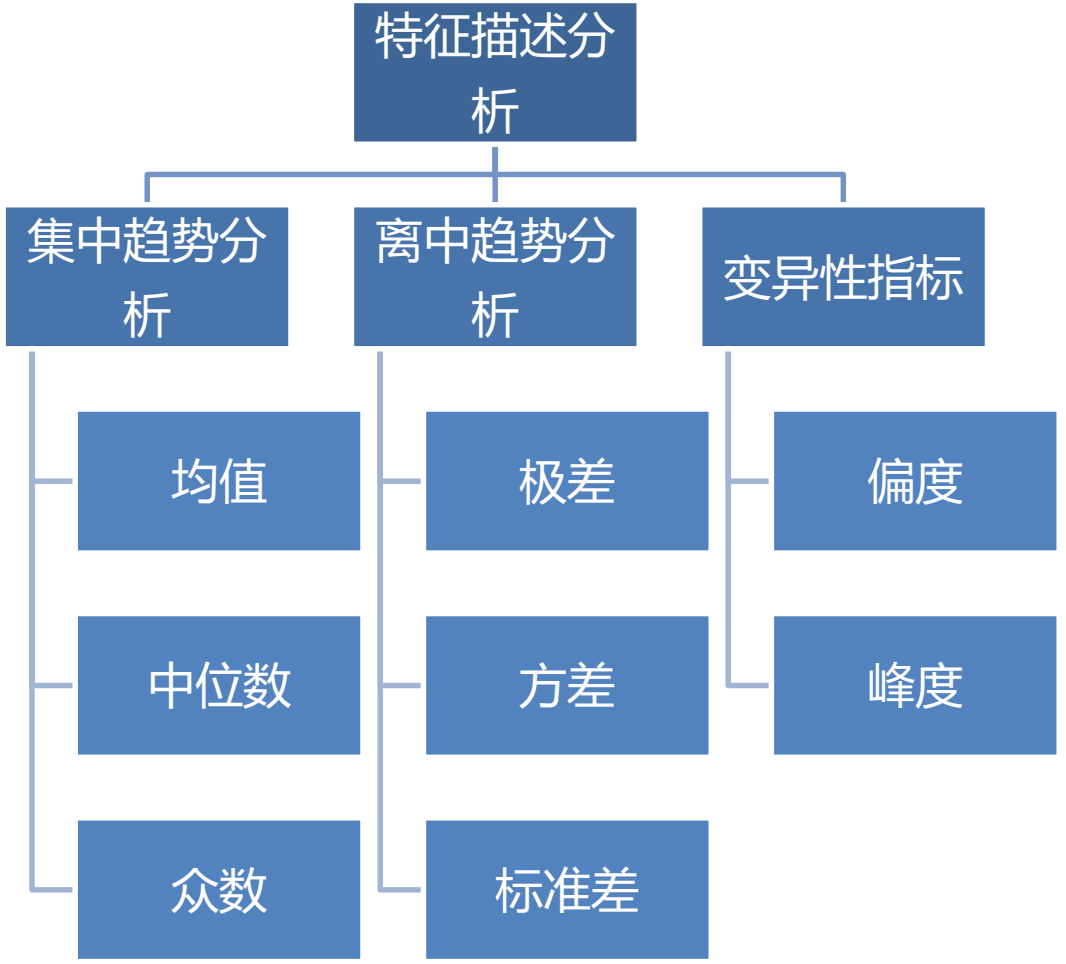

统计方法
基本分析法
元分析法
相关分析
回归分析
方差分析
分类分析
聚类分析
时间序列分析
其他（相关规则分析）
加权平均法
优化方法

分析方法

- 特征描述分析
- 相关和回归分析
  - 两个变量之间的密切程度
- 聚类分析
- 关联分析

特征描述分析

- 特征描述分析
  - 集中趋势分析
    - ✓ 均值、中位数、众数
  - 离中趋势分析
    - ✓ 极差、方差、标准差
  - 变异性指标
    - ✓ 偏度、峰度



分析方法

- 特征描述分析
- 相关和回归分析
  - 相关分析
  - 回归分析
- 聚类分析
- 关联分析

函数关系和相关关系

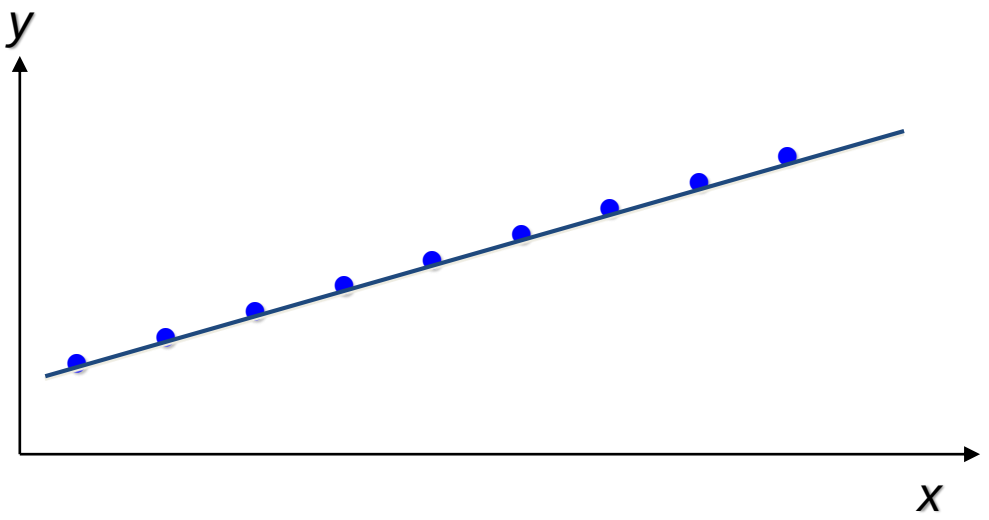
- 变量联系存在着两种不同的类型
  - ✓ 函数关系
  - ✓ 相关关系

- 函数关系

  当一个(或一组)变量每取一个值时，相应的另一个变量必然有一个确定值与之对应 。

函数关系

- ——对应的确定关系
- 设有两个变量 x 和 y，变量 y 随变量 x 一起变化，并完全依赖于 x，当变量 x 取某个数值时，y 依确定的关系取相应的值，则称 y 是 x 的函数，记为 $y = f(x)$，其中 x 称为自变量，y 称为因变量
- 各观测点落在一条线上



相关关系（correlation analysis）

- 相关关系

变量之间存在有依存关系，但这种关系是不完全确定的随机关系

- 即当一个(或一组)变量每取一个值时，相应的另一个变量可能有多个不同值与之对应 。

相关关系的分类

- 按相关的程度分

  - 完全相关
  - 不完全相关
  - 不相关(或零相关)

例：

完全相关:在价格P不变的情况下,销售收入Y与销售量X的关系。

不相关:股票价格的高低与气温的高低是不相关的。

相关关系的分类

- 按相关的方向分
  - 正相关
  - 负相关

正相关：两个变量之间的变化方向一致，都是增长趋势或下降趋势。

例： 收入与消费的关系;
工人的工资随劳动生产率的提高而提高。

负相关：两个变量变化趋势相反，一个下降而另一个上升，或一个上升而另一个下降。

例： 物价与消费的关系。
商品流转的规模愈大,流通费用水平则越低。

相关关系的分类

- 按相关的形式分
  - 线性相关
  - 非线性相关

线性相关（直线相关）：当一个变量每变动一个单位时，另一个变量按一个大致固定的增(减)量变动。

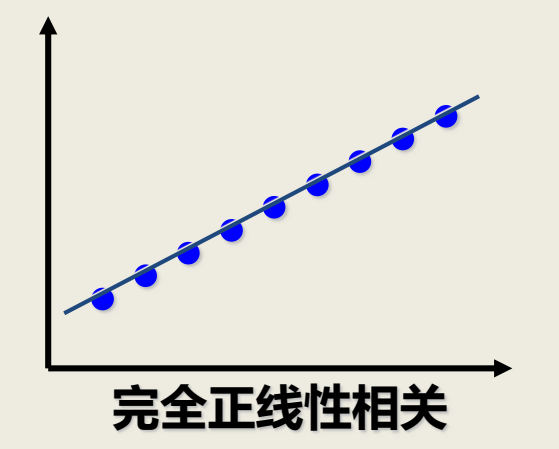
例:人均消费水平与人均收入水平

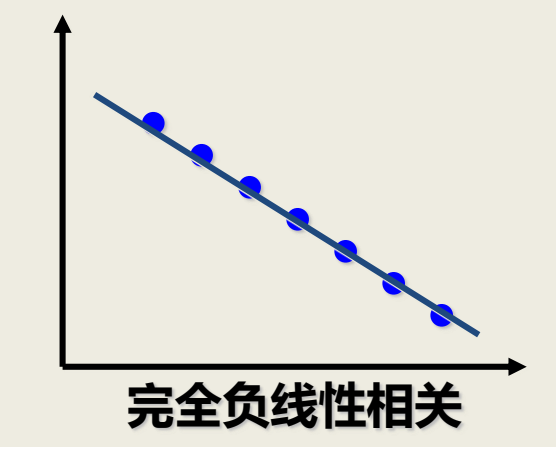
非线性相关（曲线相关）：当一个变量变动时，另一个变量也相应发生变动，但这种变动是不均等的。

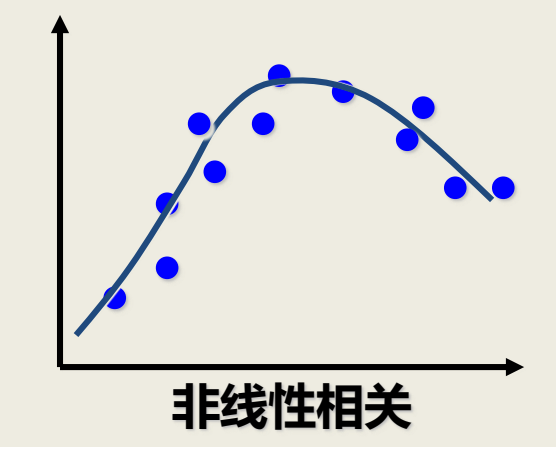
例:

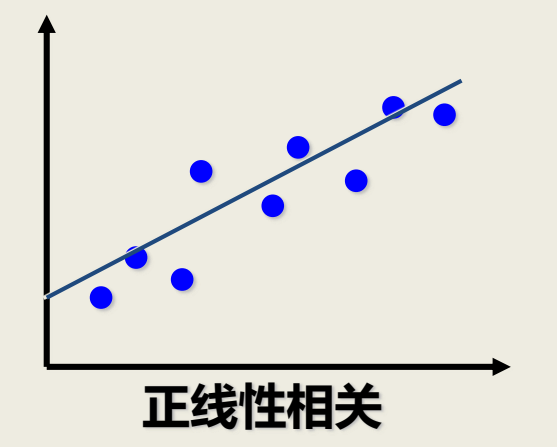
产品的平均成本与总产量;
农产量与施肥量.

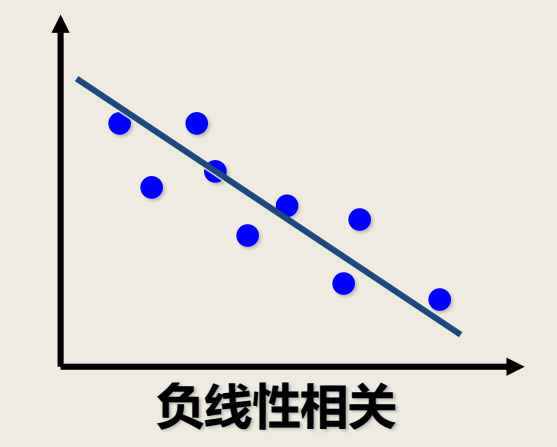
图示

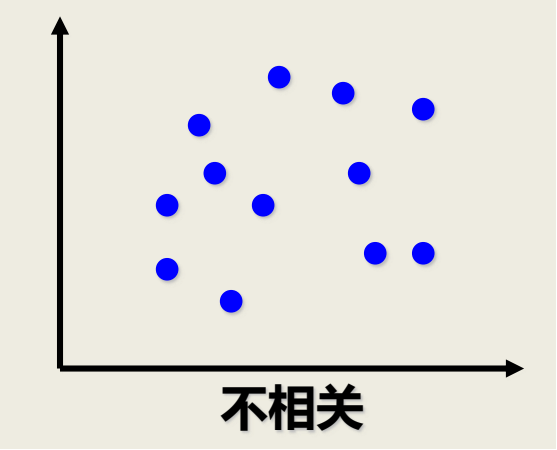












相关系数

皮尔逊(1890年,英国)相关系数

– 两个变量之间的协方差和标准差的商。

– 度量**两个变量**之间**线性**相关**密切程度**和相关方向的统计指标。

若相关系数是根据总体全部数据计算的，称为总体相关系数,记为 .

$$\rho_{X,Y}=\frac{cov(X,Y)}{\sigma_X\sigma_Y}=\frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X\sigma_Y}$$

估算样本的协方差和标准差，可得到样本皮尔逊系数，常用英文小写字母 r 代表

$$r=\frac{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)\left(Y_i-\overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2}\sqrt{\sum_{i=1}^{n}\left(Y_i-\overline{Y}\right)^2}}$$

相关系数

与上式等价的表示为：

$$r_{xy}=\frac{\sum x_i y_i - n\bar{x}\bar{y}}{(n-1)s_x s_y}=\frac{n\sum x_i y_i-\sum x_i \sum y_i}{\sqrt{n\sum x_i^2-\left(\sum x_i\right)^2}\sqrt{n\sum y_i^2-\left(\sum y_i\right)^2}}$$

1. r 的取值范围是 [-1,1], $|r|=1$，为完全相关
   $r=1$，为完全正相关，$r=-1$，为完全负相关
2. $r=0$，不存在**线性相关**关系
3. $-1\le r<0$，为负相关
4. $0<r\le 1$，为正相关
5. $|r|$越趋于1表示关系越密切；$|r|$越趋于0表示关系越不密切

分析方法

- 特征描述分析
- 相关和回归分析
  - 相关分析
  - 回归分析
- 聚类分析
- 关联分析

回归分析

- 回归分析（regression analysis）：确定两种或两种以上变量间相互依赖的定量关系的一种分析方法。
  - 研究的是因变量（目标）和自变量（预测器）之间的关系
    - ✓ 源于英国统计学家Galton研究父母平均身高与子女身高的关系

$$y = ax + b$$

  - 按变量多少分类
    - ✓ 分为一元回归
    - ✓ 多元回归分析
  - 按自变量和因变量之间的关系类型分类
    - ✓ 线性回归分析
    - ✓ 非线性回归分析

相关分析与回归分析的区别

- 相关分析
  - ✓变量 x 变量 y 处于平等的地位
  - ✓变量 x 和 y 都是随机变量
  - ✓用于描述两个变量之间线性关系的密切程度

- 回归分析
  - ✓变量 y 称为因变量，处在被解释的地位，x 称为自变量，用于预测因变量的变化
  - ✓y 是随机变量， x 则作为研究时给定的非随机变量
  - ✓不仅可以揭示变量 x 对变量 y 的影响大小，还可以由回归方程进行预测和控制

相关分析与回归分析的联系

- 相关分析是回归分析的基础和前提
  - 只有当变量之间存在着高度相关时，进行回归分析寻求其相关的具体形式才有意义。
- 回归分析是相关分析的深入和继续
  - 相关分析需要依靠回归分析来表明现象数量相关的具体形式。

线性回归

- 线性回归：因变量和自变量之间的关系是线性的。
  – 能找到线性关系，是预测结果尽可能接近真实值

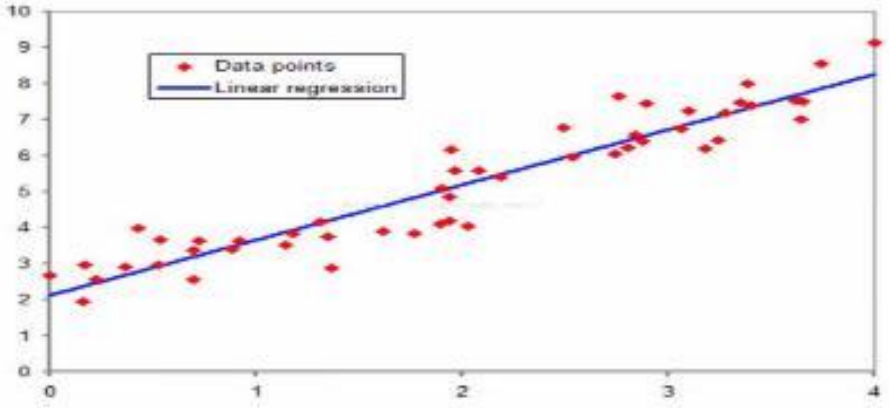
$y = b_0 + b_1x_1 + \cdots + b_nx_n + \varepsilon$

y：结果变量；

x_j：输入变量，$j = 1,2,\ldots, n$；

b_i：未知参数，$i=0,1,2,\ldots,n$；

ε随机误差，假设$\varepsilon \sim N(0,\sigma^2)$，以及ε相互独立。



建立回归方程，要估计未知参数b_i

进行m次独立观测，得到m组样本数据 $(x_{i1}, \ldots, x_{in}; y_i)$ $(i=1, \ldots, m)$

线性回归

$$\begin{cases} y_1 = b_0 + b_1 x_{11} + \cdots + b_n x_{1n} + \varepsilon_1 \\ \cdots \\ y_m = b_0 + b_1 x_{m1} + \cdots + b_n x_{mn} + \varepsilon_m \end{cases}$$

可表示为

$Y = XB + \epsilon$

$Y = (y_1, y_2, \ldots, y_m)^T$

$B = (b_0, b_1, \ldots, b_n)^T$

$\epsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m)^T \sim N_m(0, \sigma^2 I_m)$，　I_m为m阶单位矩阵。

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1n} \\ & & \cdots & \\ 1 & x_{m1} & \cdots & x_{mn} \end{bmatrix}$$

X为$m*（n+1）$阶矩阵。

线性回归

- 参数B的估计：最小二乘法

  – 选择B使残差平方和$Q（B）$最小

$$Q(B)= (Y-XB)^T(Y-XB)$$

$$= \sum_{i=1}^{m}(y_i - b_0 - b_1x_{i1} - \cdots - b_nx_{in})^2$$

$Q(B)$是B的非负二次函数，存在最小值。

对于每个b_i对$Q(B)$求偏导，并等于0，利用联合方程可求出b_i。

逻辑回归

- 逻辑回归
  - 广义线性回归模型，与线性回归模型都具有 $b_0 + b_1x_1 + \cdots + b_nx_n$，其中$b_i$是待求参数
  - 线性回归直接将$b_0 + b_1x_1 + \cdots + b_nx_n$作为因变量，即

$$y = b_0 + b_1x_1 + \cdots + b_nx_n$$

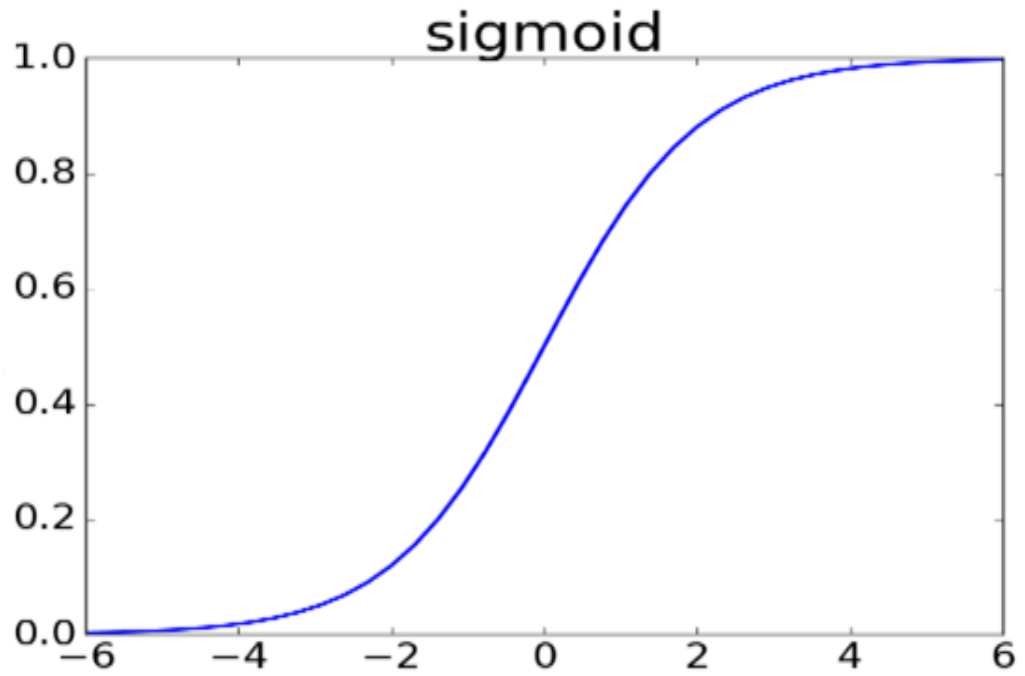
  - 逻辑回归将$b_0 + b_1x_1 + \cdots + b_nx_n$映射到[0,1]
    - ✓ 逻辑回归：属于某一类的概率多大?
  - $p(x)$则可以用来表示概率$p(y=1/x)$，即当一个x发生时，y被分到1那一组的概率。

逻辑回归

- sigmoid函数

$$g(y)=\frac{1}{1+e^{-y}}$$

  - 把（$-\infty$，$+\infty$）映射到[0,1]
  - 可微分



逻辑回归

$p = g(y) = \frac{1}{1+e^{-y}}$可转化为

$$ln\frac{p}{1-p} = y$$

成为Logistic变换。

$$ln\frac{p}{1-p} = b_0 + b_1x_1 + \cdots + b_nx_n$$

用给定的样本估计参数b_0，b_1，…b_n

逻辑回归

- 参数估计
  - 最小二乘法
    - ✓ 非凸函数，会导致陷入局部最优解
  - 极大似然估计
    - ✓ 某个参数能够使样本出现的概率最大，就作为估计的真实值。

设 $p(x,\theta)$ 概率分布，θ 为未知参数

设$x_1, \ldots, x_n$是一个样本值，该事件发生的概率为

$$L(\theta) = L(x_1, \ldots, x_n; \theta) = \prod_{i=1}^{n} p(x_i; \theta)$$

$L(\theta)$称为样本的似然函数。

极大似然估计：已知样本值$x_1, \ldots, x_n$，选取参数θ，使概率$L(\theta)$达到最大值，此时的θ为最大估计值。

逻辑回归

序号	年龄	收入	结婚	购买汽车
1	25	3000	0	1
2	39	6000	1	1
3	40	4000	1	1
4	55	2500	1	0
5	45	6000	0	0
6	28	1500	1	0
7	56	6500	1	1
8	35	3500	1	1
9	45	5000	1	1
10	30	3500	0	0

逻辑回归

y是二项分类，1：买汽车，0：不买汽车

p买汽车的概率，1-p：不买汽车的概率

$$ln\frac{p}{1-p}=b_0+b_1x_1+\cdots+b_3x_3$$

$$p=\frac{e^{b_0+b_1x_1+\cdots+b_3x_3}}{1+e^{b_0+b_1x_1+\cdots+b_3x_3}}$$

分类中有6个买汽车，4个不买汽车，建立联合概率

$$L=P_1*P_2*P_3*(1-P_4)*(1-P_5)*(1-P_6)*P_7*P_8*P_9*(1-P_{10})$$

$$=\frac{e^{b_0+b_1x_{11}+\cdots+b_3x_{13}}}{1+e^{b_0+b_1x_{11}+\cdots+b_3x_{13}}}*\ldots*\left(1-\frac{e^{b_0+b_1x_{101}+\cdots+b_3x_{103}}}{1+e^{b_0+b_1x_{101}+\cdots+b_3x_{103}}}\right)$$

求使得L最大的b_i，也是使$ln(L)$最大的b_i。对b_i求偏导，令偏导为0。得到：

b_0=137.95，b_1=-17.34，b_2=0.10，b_3=173.11

分析方法

- 特征描述分析
- 相关和回归分析
- 聚类分析
- 关联分析

聚类分析

- 聚类分析：将一批样本(或变量)按照在性质上的“亲疏”程度，在没有先验知识的情况下，自动进行分类的方法。
  - 对类的要求：类内个体具有较高的相似性，类间的差异性较大。

聚类算法中常用的距离度量

加权l_p度量：

$$d_p(x,y)=\left(\sum_{i=1}^{l} w_i|x_i-y_i|^p\right)^{1/p}$$

x_i, y_i 是x和y的第i个值，$i=1,2,\cdots,l$，$w_i \geq 0$是第i个权重参数

当 $p=2$ 时，上式变为**欧几里德距离：**

$$d_2(x,y)=\left(\sum_{i=1}^{l} w_i|x_i-y_i|^2\right)^{1/2}$$

最常用的距离度量

当 $p=1$ 时，l_p度量变为**曼哈顿函数：**

$$d_1(x,y)=\sum_{i=1}^{l} w_i|x_i-y_i|$$

聚类算法中常用的距离度量

Sup 距离（Sup distance）	$D_{ij}=\max\left\|x_{il}-x_{jl}\right\|$，（1≤l≤d）
投影朴素欧氏距离（Projected Pure Euclidean Distance）	$D_{ij}=(\sum_{l=1,x_{il}\neq 0}^{d}(x_{il}-x_{jl})^2)^{1/2}$
马氏距离（Mahalanobis distance）	$D_{ij}=(\mathbf{x_i}-\mathbf{x_j})^T\mathbf{S}^{-1}(\mathbf{x_i}-\mathbf{x_j})$（S 为样本协方差矩阵）
余弦相似度（Cosine similarity）	$S_{ij}=\cos\alpha=\frac{\mathbf{x_i}^T\mathbf{x_j}}{\|\mathbf{x_i}\|\|\mathbf{x_j}\|}$
模糊距离（Fuzzy distance）	$D_{ij}=1-\frac{\sum_{l=1}^{d}\min(x_{il},x_{jl})}{\sum_{l=1}^{d}\max(x_{il},x_{jl})}$

聚类算法

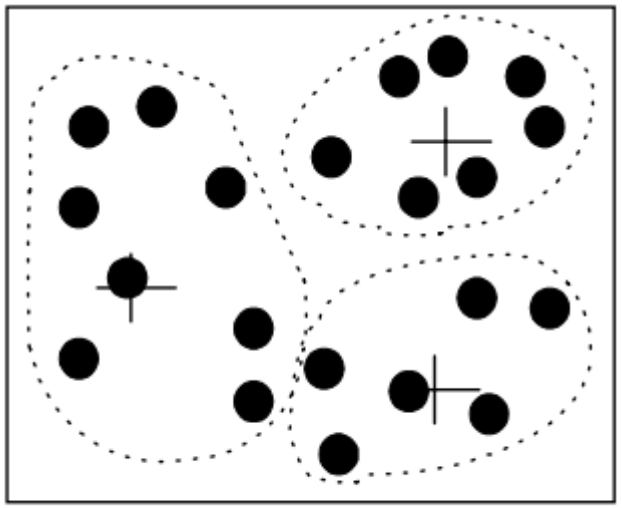
- k-means算法
- 层次聚类算法
- 模糊聚类算法
- 竞争学习算法

k-means算法

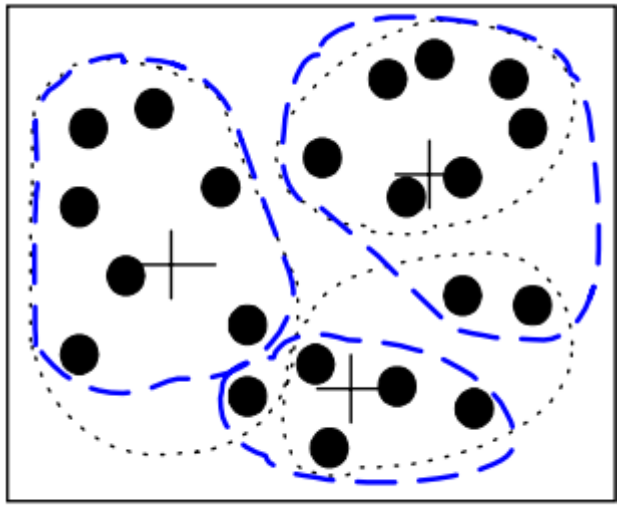
最为经典的聚类方法

- **基本思想：**
  - ✓ 以空间中k个点为中心进行聚类，对最靠近他们的对象归类。
  - ✓ 通过迭代的方法，逐次更新各聚类中心的值，直至得到最好的聚类结果。

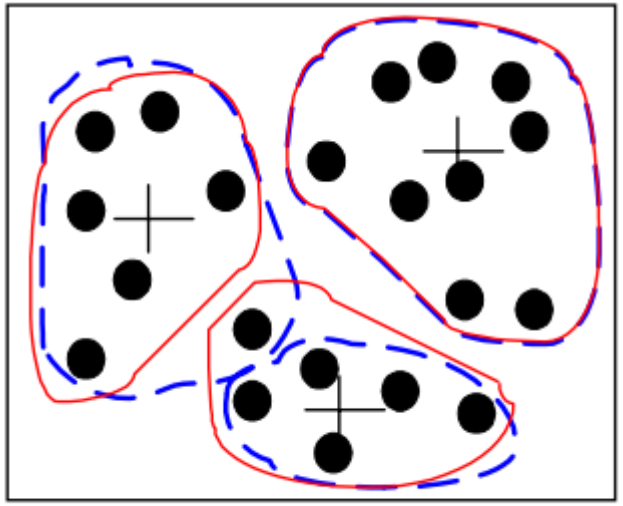
k-means算法



(a)



(b)



(c)

k-means算法

1. 指定最终聚类数k。
2. 用户指定k个样本作为初始类中心或系统自动确定k个样本作为初始类中心。
3. 系统按照距k个中心距离最近的原则，把距中心最近的样本分派到各中心所在的类中，形成一个新的k类。
4. 重新计算k个类的类中心(以各类均值点作为类中心)。
5. 重复3步和4步，直到达到下列条件之一：
   - 指定的迭代次数；
   - 达到终止迭代的条件；
   - 误差平均值局部最小。（常见）

K-means算法的优点

K-means的的时间复杂度

O(Nmq)

- q是收敛所需的迭代次数，m是聚类数目，N是聚类元素数。
- m和q远小于N所以时间复杂度接近O(N)，因此k均值适合处理大数据集。

K-means算法中存在的问题

- 算法对初始种子十分敏感。不同的初始划分使k均值产生不同的聚类结果。

  解决：

  – 根据经验选择初始点

  – 使用其他聚类算法的结果的中心点作为k-means初始点的输入

- 聚类数k需指定。错误估计k使算法不能揭示真正的聚类结构。

  解决：多次实验或者使用其他聚类算法的最终聚类数作为k-means的输入。

- k均值对异常点和噪声是敏感的。

  解决：

  – 直接剔除那些比其他任何数据都远离中心的异常值。

  – 为防止误删，需多次监测这些异常点。

K-means算法

- 尽管k-means算法有不足之处，但它简洁、高效。
- 其他聚类算法在某些特殊数据的表现优于k-means，但是没有任何一种算法在整体上是优于k-means的，各有所长。
- k-means仍然是实践中采用最为广泛的算法。

聚类算法

- k-means算法
- 层次聚类算法
- 模糊聚类算法
- 竞争学习算法

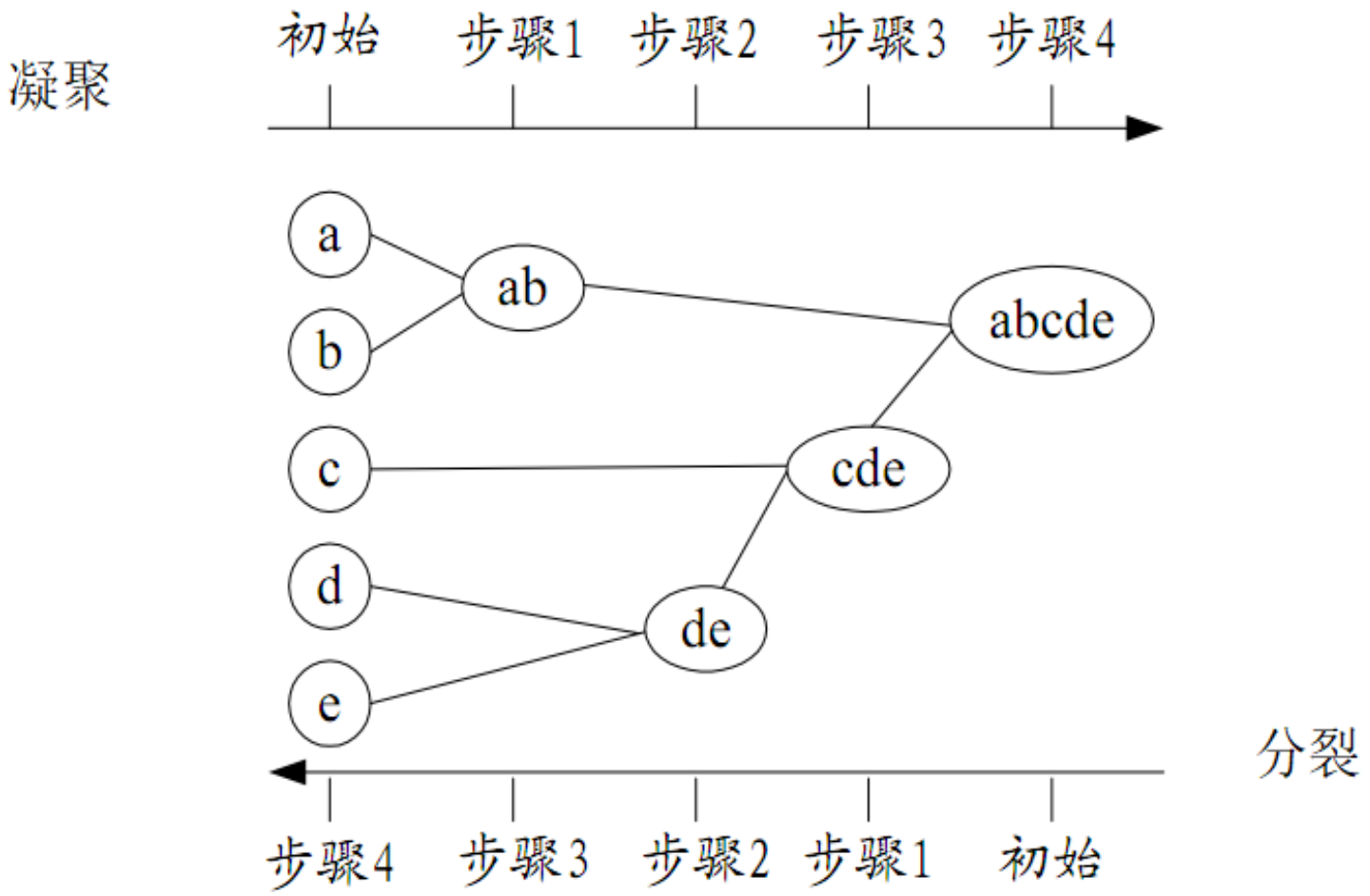
层次聚类算法

对给定的数据集进行层次的分解，直到某种条件满足为止。

- 凝聚的层次聚类
  - 自底向上的策略；
  - 将每个对象作为一个簇，合并这些原子簇为越来越大的簇，直到所有的对象都在一个簇中，或者某个终结条件被满足。

- 分裂的层次聚类
  - 与凝聚的层次聚类相反，采用自顶向下的策略；
  - 首先将所有对象置于同一个簇中，然后逐渐细分为越来越小的簇，直到每个对象自成一簇，或者达到了某个终止条件 。

层次聚类算法



层次聚类-经典凝聚算法

- 初始化

  -选择R_0 = {C_i = {x_i}，i = 1,...,N}作为初始聚类

  -令t = 0

- 重复执行下列步骤

  -t = t + 1

  -在R_{t-1} 的所有可能聚类对(C_r, C_s)中找到一组(C_i, C_j)，满足

  $d(C_i, C_j) = \min_{1\le r\le N, 1\le s\le N, r\neq s} d(C_r, C_s)$

  -定义$C_q = C_i \cup C_j$，并产生新聚类$R_t = (R_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$

- 直所有向量全被加入到单一聚类中或满足终止条件。

层次聚类算法存在的问题

- 某次合并/分裂没有选择好，可能导致低质量的聚类结果。
  - 下一步的处理是在新生成的簇进行。
  - 已做的处理不能撤销，簇之间也不能交换对象。
- 时间复杂度较高
  - 每一次合并/分裂前，需计算所有两两簇之间的距离，计算量大。
  - 若聚类对象数目为n，则层次聚类算法的时间复杂度为$O(n^2)$。

聚类算法

- k-means算法
- 层次聚类算法
- 模糊聚类算法
- 竞争学习算法

模糊聚类算法

- 传统的聚类是一种硬划分
  - 把每个样本对象严格的划分到某一类中，具有非此即彼的性质。
- 实际大多数对象并没有严格的属性
  - 它们在类别属性方面存在亦此亦彼的性质，适合进行模糊划分。

模糊聚类算法

- θ_j 表示第j个聚类的中心点，$\theta \equiv \left[\theta_1^{\mathrm{T}}, \cdots, \theta_m^{\mathrm{T}}\right]^T$，
- U是一个N*m矩阵，(i, j)元素为$u_j(x_i)$，表示**元素i对聚类j的簇隶属度**，
- $d(x_i, \theta_j)$ 是x_i 和 θ_j 之间的距离，q(>1)是模糊性参数。

模糊聚类算法是找到 θ_j使代价函数式最小：

$$\mathrm{J_q}(\theta, U) = \sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}^{q} d(x_i, \theta_j) \qquad (式1)$$

其中 U满足约束条件：

$$\sum_{j=1}^{m} u_{ij} = 1, \quad i=1, 2,..., N \qquad (式2)$$

$$u_{ij} \in [0,1], \quad i=1, 2,..., N, j=1, 2,...,m$$

模糊聚类算法

$J_q(\theta, U)$ 最小化：

先考虑 U，求 $J_q(\theta, U)$ 关于U的最小值，约束式是 $\sum_{j=1}^{m} U_{ij} = 1$，得出拉格朗日函数：

（原先的函数减去 λ 乘以约束函数）

$$J(\theta, U) = \sum_{i=1}^{N}\sum_{j=1}^{m} U_{ij}^{q} d(x_i, \theta_j) - \sum_{i=1}^{N} \lambda_i \left(\sum_{j=1}^{m} U_{ij} - 1 \right)$$

对 $J_q(\theta, U)$ 求 u_{rs}偏导：

$$\frac{\partial J(\theta, U)}{\partial u_{rs}} = q u_{rs}^{q-1} d(x_r, \theta_s) - \lambda_r \qquad \text{其中s=1,2,...,m}$$

解出 $u_{rs} = \left(\frac{\lambda_r}{q d(x_r, \theta_s)} \right)^{\frac{1}{q-1}}$ 其中s=1,2,...,m （式3）

模糊聚类算法

将上述方程带入约束条件 $\sum_{j=1}^{m} U_{ij} = 1$，得到

$$\lambda_r = \frac{q}{\left(\sum_{j=1}^{m} \left(\frac{1}{d(x_r, \theta_j)} \right)^{\frac{1}{q-1}} \right)^{q-1}} \quad \text{（式4）}$$

合并式3和式4，经变化得到

$$u_{rs} = \frac{1}{\sum_{j=1}^{m} \left(\frac{d(x_r, \theta_s)}{d(x_r, \theta_j)} \right)^{\frac{1}{q-1}}}$$

其中 *r=1,2,...,N, s=1,2,...,m* （式5）

对 $J_q(\theta, U)$ 求 θ_j 偏导数，并令偏导数为0，得到

$$\frac{\partial J(\theta, U)}{\partial \theta_j} = \sum_{i=1}^{N} u_{ij}^{q} \frac{\partial d(x_i, \theta_j)}{\partial \theta_j} = 0 \qquad j=1,2,...,m \qquad \text{(式6)}$$

将式5带入，即可求出θ_j

模糊聚类算法

随机选择 $\theta_j(0)$ 作为聚类中心 θ_j 的初始值，其中，j=1,2,...,m

t=0

重复执行下列步骤：

-For i=1 to N

For j=1 to m

$$u_{\vec{ij}}(t) = -\frac{1}{\sum_{k=1}^{m}\left(\frac{d\left(x_i,\theta_j(t)\right)}{d\left(x_i,\theta_k(t)\right)}\right)^{\frac{1}{q-1}}}$$

End {For j}

-End {For i}

模糊聚类算法

-t=t+1

-For j=1 to m

*参数更新：求解

$$\sum_{i=1}^{N} u_{ij}^{q}(t-1)\frac{\partial d(x_i, q_j)}{\partial q_j} = 0$$

对于每个θ_j，令$\theta_j(t)$ 为上述方程的解

-End {For j}

到满足终止条件 $\|\theta(t+1) - \theta(t)\| < \varepsilon$

聚类算法

- k-means算法
- 层次聚类算法
- 模糊聚类算法
- 竞争学习算法

竞争学习算法

当集合的一个向量 x 出现时，所有聚类中心w_j 相互竞争。

离x最近（根据某种距离测度）的w_j是竞争的胜利者。

胜利者w_j被更新，使之向x移动；

失败者保持原来的位置不变或者被更新为以较慢的速度向x移动。

设t为当前的迭代次数，t_{max}为允许的最大迭代次数。设m为当前聚类数。

竞争学习算法

t=0，在数据集中随机选择m个向量w_j，j=1，2，….，m；

重复下面步骤：

1. t=t+1

2. 在随机选择x，对于x，求出获胜的w_j，获胜条件是

$$d(x, w_j) = \min_{k=1,\cdots,m} d(x, w_k)$$

3. 更新

$$w_j(t) = \begin{cases} w_j(t)+\eta(x-w_j(t)), \text{如果} w_j(t) \text{胜出} \\ w_j(t)，\ w_j(t) \text{失败} \end{cases}$$

η是学习率，在[0,1]之间取值

直到满足收敛条件$\|W(t) - W(t-1)\| < \varepsilon$，其中 $W = \left[W_1^T, \cdots, W_m^T\right]^T$

或$t>=t_{max}$

分析方法

- 特征描述分析
- 相关和回归分析
- 聚类分析
  - k-means算法
  - 层次聚类算法
  - 模糊聚类算法
  - 竞争学习算法
- 关联分析

聚类评估方法

- 依据：好的聚类结果簇内相似度较大而簇间相似度较小。
- 凝聚度（Cohesion）和分离度（Separation）

$$Cohesion = \sum_{i=1}^{K} \sum_{U \in C_i} dis(U, C_i) \qquad Separation = \sum_{i=2}^{K} \sum_{j=1}^{i-1} dis(C_i, C_j)$$

$dis(U, C_i)$ ：簇C_i内的对象到簇中心的距离

- 凝聚度越小，说明聚类结果簇内越紧凑，聚类效果越好。

$dis(C_i, C_j)$ ：簇C_i与C_j之间的距离

- 分离度越大，说明聚类结果簇间越松散，聚类效果越好。

分析方法

- 特征描述分析
- 相关和回归分析
- 聚类分析
- 关联分析

关联规则

- 啤酒和尿布

– 20世纪90年代的沃尔玛超市发现

✓ 在某些特定的情况下，啤酒与尿布两件看上去毫无关系的商品经常会出现在同一个购物篮中。

– 后续调查发现

✓ 这种现象出现在年轻父亲的身上。在美国有婴儿的家庭中，一般是母亲在家中照看婴儿，父亲前去超市购买尿布。父亲在购买尿布的同时，往往会顺便为自己买啤酒，这样就会出现啤酒与尿布这两件看上去不相干的商品经常会出现在同一个购物篮的现象。

关联规则算法

- 定义
  - 设$T=\{t_1,t_2,\cdots,t_m\}$是m个不同项的集合，t_k称为项，集合T称为项集。
  - 关联规则：$x \longrightarrow y$，如：{牛奶，尿布} $\longrightarrow$ {啤酒}
  - 支持度：确定规则可以用于给定数据集的频繁程度
    - 含z的项数：$\sigma(z)=|t_i|\ z \subseteq t_i, t_i \in T$
      项集{啤酒，牛奶，尿布}的项数为$\sigma(z)$ =2
    - $x \longrightarrow y$ 的支持度
      $S(x \longrightarrow y)=\sigma(x \cup y)$/N = 2/5
  - 置信度：确定Y在包含X的事务中出现的频繁程度
    - $C(x \longrightarrow y)=\sigma(x \cup y)/\sigma(x)$ = 2/3

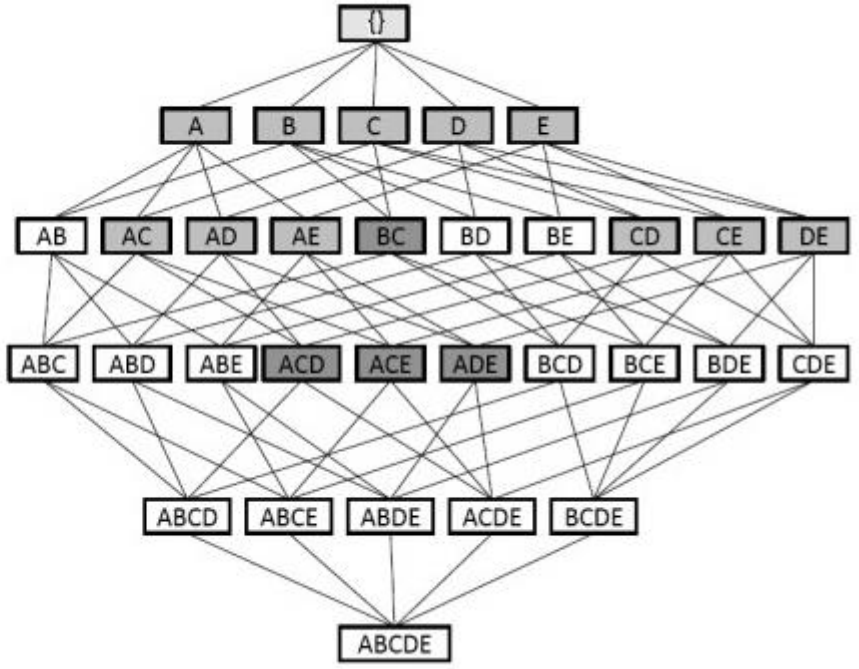
TID	面包	牛奶	尿布	啤酒	鸡蛋	可乐
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

- 目的
  - 支持度用于删去无意义的规则，另外支持度具有一种期望的性质可以用于关联规则的有效发现。
  - 对于给定规则$x \longrightarrow y$，置信度越高表示y在包含x的事务中出现的可能性越大。

关联规则的产生

- 规则的产生
  - 频繁项集的产生：发现满足最小支持度阈值的所有项集，即为频繁项集
  - 规则的产生：从上一步发现的频繁项集中提取所有高置信度的规则，即为强规则
- 频繁项集的产生

TID	Item set
1	$\{a, d, e\}$
2	$\{b, c, d\}$
3	$\{a, c, e\}$
4	$\{a, c, d, e\}$
5	$\{a, e\}$
6	$\{a, c, d\}$
7	$\{b, c\}$
8	$\{a, c, d, e\}$
9	$\{b, c, e\}$
10	$\{a, d, e\}$



lattice structure格结构用于枚举所有可能的项集，确定每个候选项集的支持度计数

问题：开销极大

Apriori算法

- 先验原理

  如果一个项集是频繁的，则它的所有子集一定也是频繁的。

  解释：在上述的项集格中，假定{c,d,e}是频繁项集，则任何包含项集{c,d,e}的事务一定包含它的子集{c,d}, {c,e},{d,e},{c},{d} 和{e}，均为频繁的。

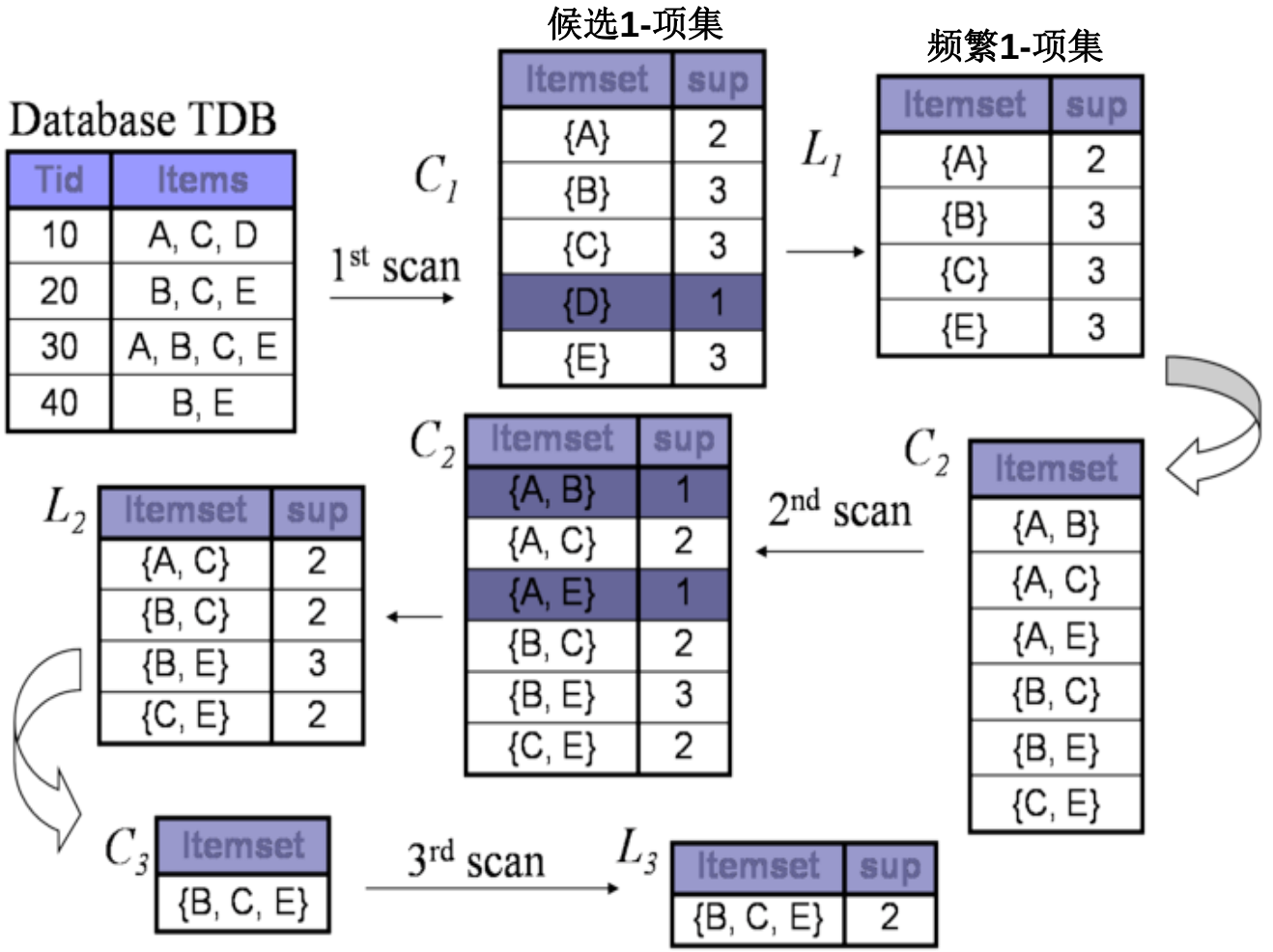
- 思路

  初始时每个项都被看作候选1-项集，筛选出其中的频繁1-项集。

  在进行下一次迭代时仅适用频繁1-项集来产生候选2-项集（根据先验原理，所有非频繁1-项集的超集都是非频繁的）。以此类堆，计算下一层的频繁项集。

Apriori算法

- 算法流程图



Apriori算法总结

- 算法特点
  - 使用先验原理对指数级搜索空间进行剪枝，成功处理频繁项集产生的组合爆炸问题。
  - 但导致不可低估的I/O开销，因为它需要多次扫事务描述数据集。

FP-Growth算法

- FP(Frequent-Pattern)-Growth频繁模式增长
- FP树：输入数据的压缩表示，通过读入事务并把每个事务映射到FP树中的一条路径来构造。
- 特点：不同的事务可能有若干个相同的前缀项，因此它们的路径可能存在部分重叠，路径重叠越多使用FP数结构获得的压缩效果越好。
- 优势：如果FP树足够小能够放入内存中就可以直接从这个内存中的结构提取频繁项集，避免重复地扫描存在硬盘中的数据。

FP-Tree的构建

- Step 1：

  ➢丢弃非频繁项，
  ➢将频繁项按照支持度的递减排序。
  - ✓ a为最频繁的项，接下来依次为b,c,d和e

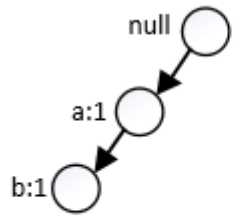
- Step 2：

  ➢二次扫描构建FP树，FP树仅包含一个根节点用null标记，使得树中的每个节点都包括一个项的标记和一个计数，计数显示映射到给定路径的事务的个数。

  - ✓ 读入第一个事务{a,b}后创建标记为a和b的节点，形成null->a->b的路径，该路径上所有节点的频度计数为1。

事务数据类型

TID	项
1	{a, b}
2	{b, c, d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

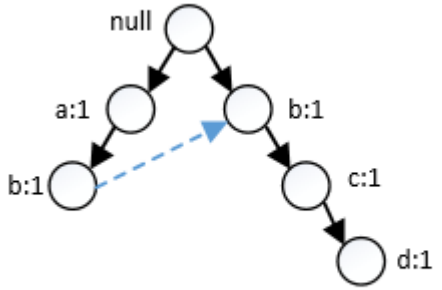


(i) 读入TID=1

FP-Tree的构建

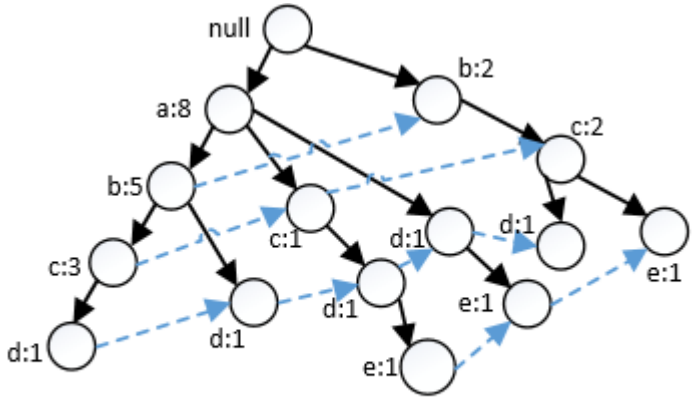
事务数据类型

TID	项
1	{a, b}
2	{b, c, d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

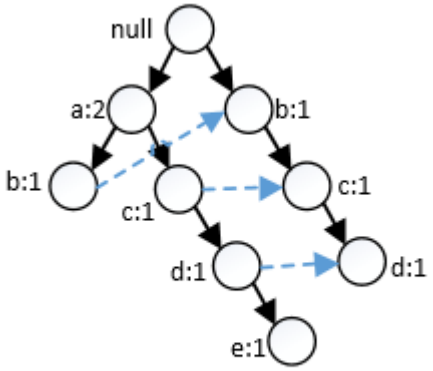


(ii) 读入TID=2

- ✓ 读入第二个事务{b,c,d}，连接null->b->c->d形成路径。



(iv) 读入TID=10



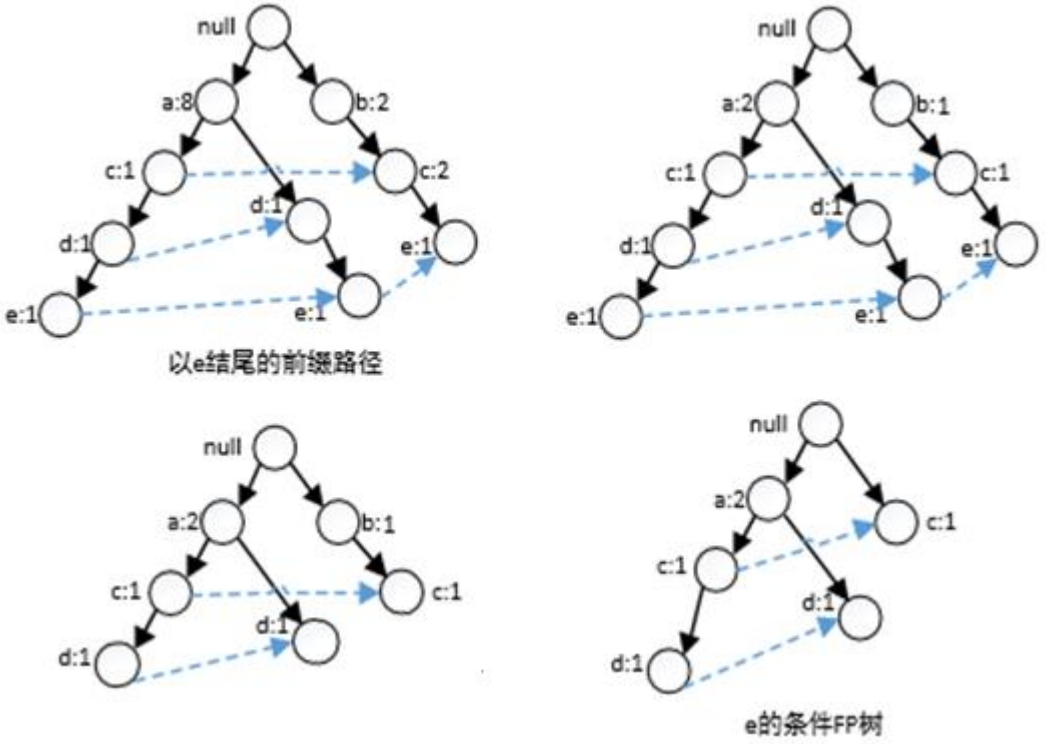
(iii) 读入TID=3

- ✓ 读入第三个事务{a,c,d,e}
- ✓ 与第一个事务共享一个相同的前缀a，a节点的频度计数增到2

- ✓ 依次读取所有的事务形成FP树
- ✓ FP树还包含一个连接具有相同项的指针列表，用虚线表示，可快速地访问树中的项

FP-Growth的频繁项集产生

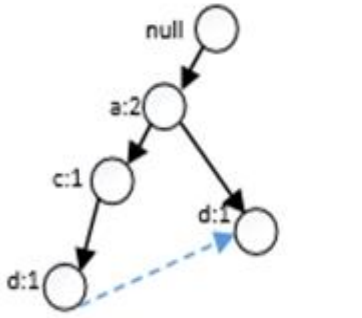
- 频繁项集产生



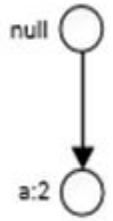
- 步骤
  - 收集包含e结点的所有路径------“前缀路径”
  - 判断{e}为频繁项集
    - ✓ 假定最小支持度为2
    - ✓ e的支持度为3
  - {e}为频繁项集，将前缀路径转化为e的条件FP树
    - ✓ 更新前缀路径上的支持度计数，
    - ✓ 删除e的节点，修建前缀路径
    - ✓ 删除非频繁项，如结点b。

FP-Growth的频繁项集产生

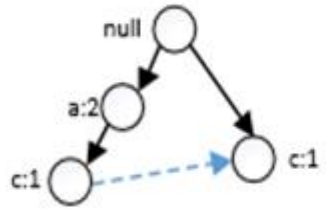
- 频繁项集产生



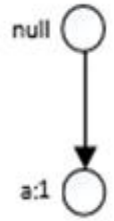
以de结尾的前缀路径



de的条件FP树



以ce结尾的前缀路径



ce的条件FP树

- 步骤
  - 对e的条件FP树重复以上步骤，来发现以de,ce,be和ae结尾的频繁项集
    - ✓ 判断{d,e}为频繁项集：是
    - ✓ 构建de的条件FP树
    - ✓ 更新计数并删除非频繁项
    - ✓ 得出以de结尾的频繁项集为{a,d,e}
    - ✓ 类似分别得出以ce，be和ae结尾的频繁项集

FP-Growth算法测试

- 输入测试数据

牛奶,鸡蛋,面包,薯片

鸡蛋,爆米花,薯片,啤酒

鸡蛋,面包,薯片

牛奶,鸡蛋,面包,爆米花,薯片,啤酒

牛奶,面包,啤酒

鸡蛋,面包,啤酒

牛奶,面包,薯片

牛奶,鸡蛋,面包,黄油,薯片

牛奶,鸡蛋,黄油,薯片

- 频繁项集结果

6	薯片	鸡蛋		
5	薯片	面包		
5	鸡蛋	面包		
4	薯片	鸡蛋	面包	
5	薯片	牛奶		
5	面包	牛奶		
4	鸡蛋	牛奶		
4	薯片	面包	牛奶	
4	薯片	鸡蛋	牛奶	
3	面包	鸡蛋	牛奶	
3	薯片	面包	鸡蛋	牛奶
3	鸡蛋	啤酒		
3	面包	啤酒		


北京郵電大學
Beijing University of Posts and Telecommunications


数据科学中心
Center for Data Science