



# 数据科学基础

## 第四章：数据统计

顾慎凯



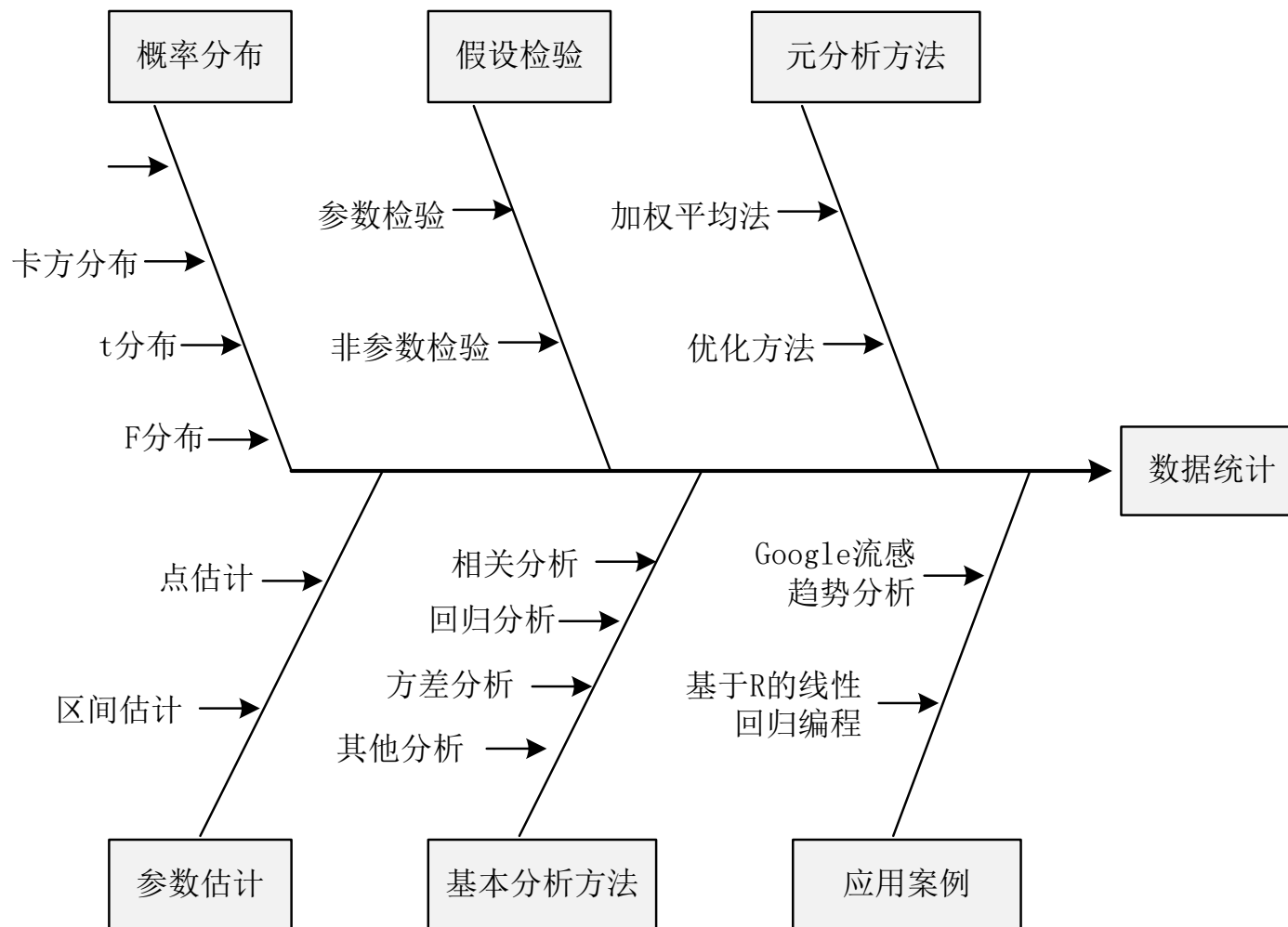
# 内容简介：数据科学中的数据预处理



南京工业大学  
NANJING TECH  
UNIVERSITY



© 2021 Dr Shenkai Gu



- 不要求精通统计知识，但需要了解统计学的基本知识

## 理解

- 概率分布的基础知识

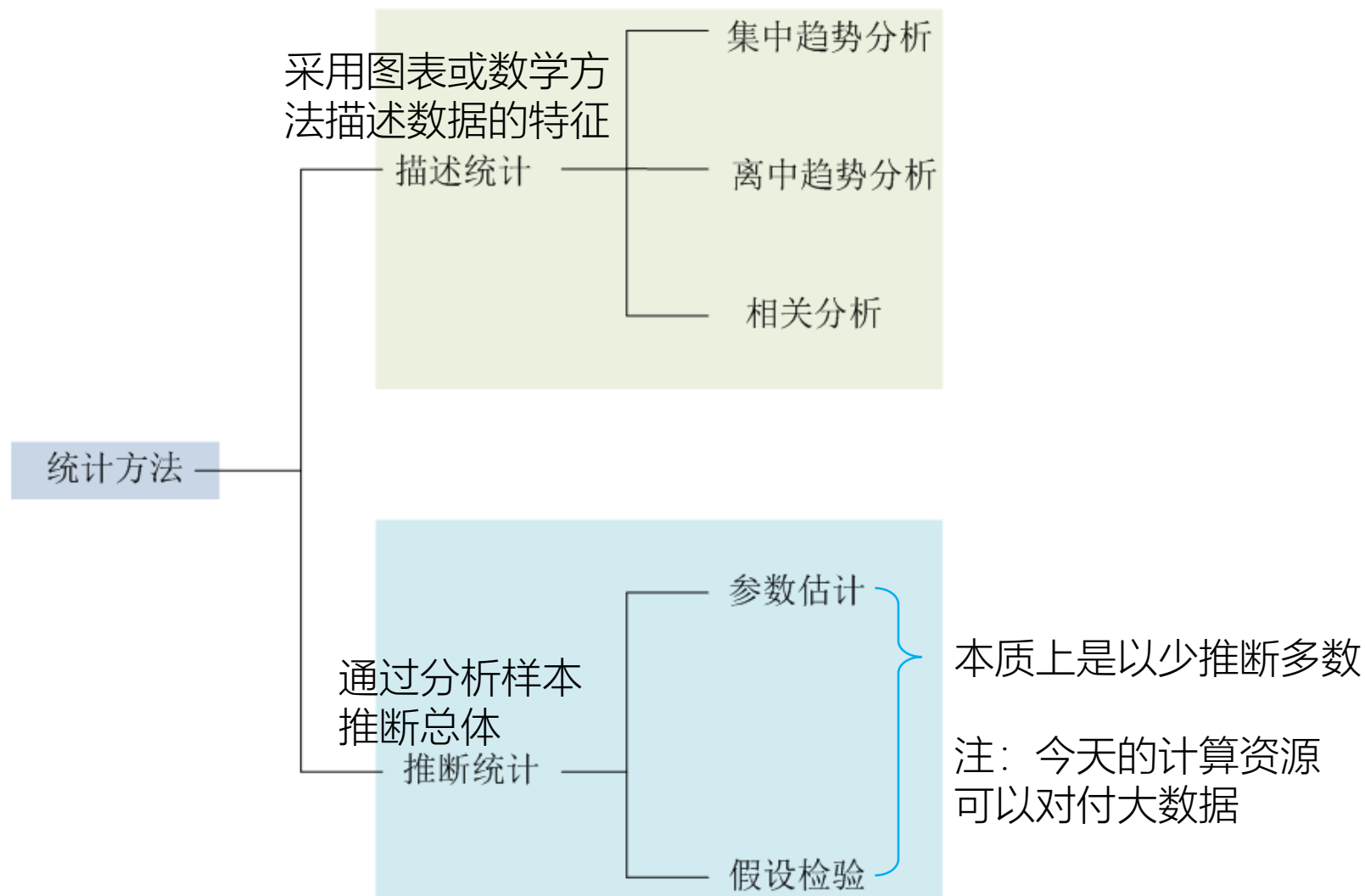
## 掌握

- 参数估计和假设检验、基本分析方法与元分析法

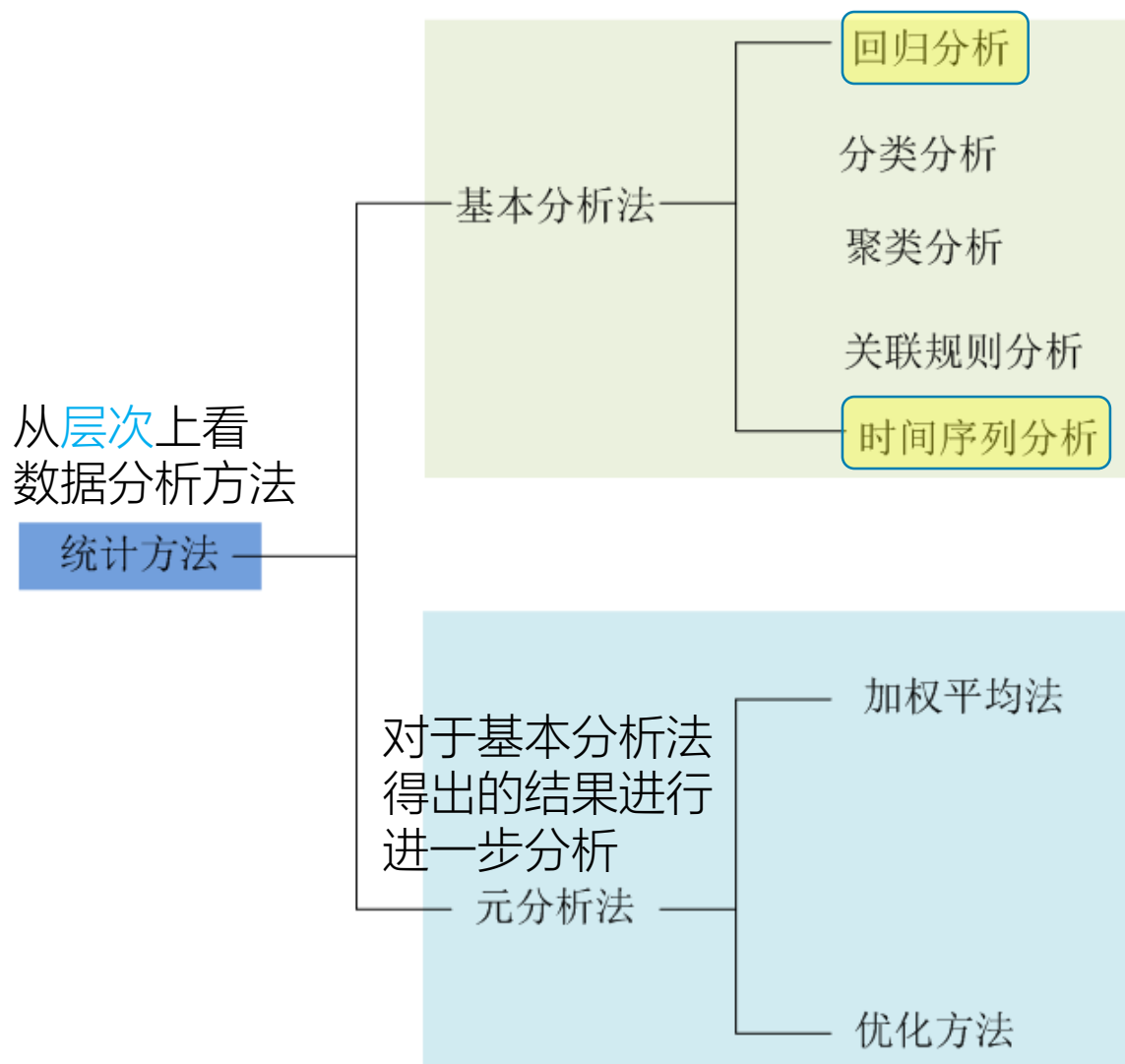
## 熟练掌握

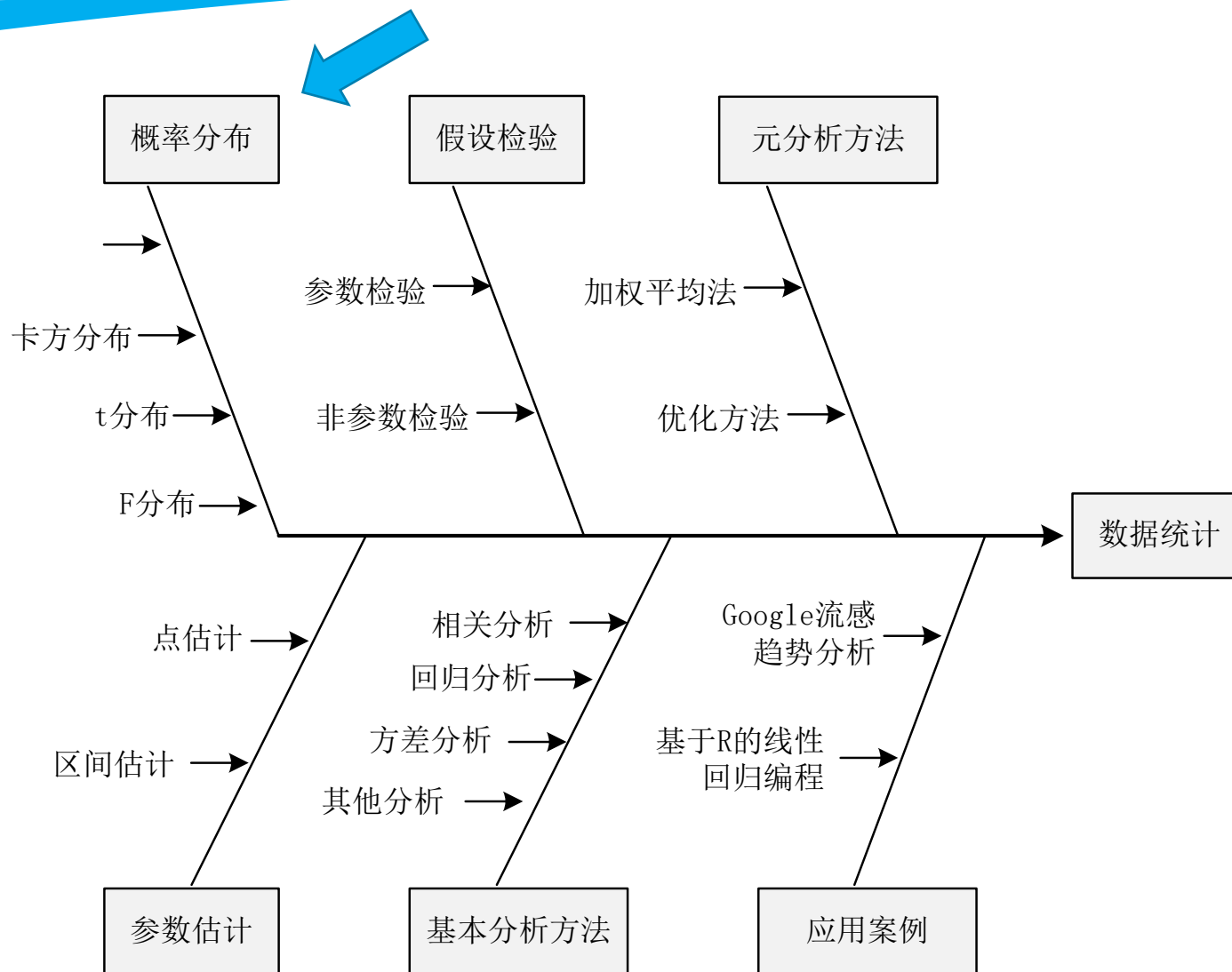
- 读者自己所在专业领域中常用的数据统计分析方法、技术与工具

# 数据统计分析的类型——目的与思路视角



# 数据统计分析的类型——方法论视角





- 用以描述随机变量取值的概率规律
  - 也就是随机变量所有可能的取值以及取每个值所对应的概率
- 离散型随机变量和连续型随机变量的概率分布的描述方法不同

	概率分布的描述方法	典型概率分布
离散型随机变量	概率函数 $P(X)$	二项分布和泊松分布
连续型随机变量	概率密度函数 $f(x)$ 或 概率分布函数 $F(x)$	正态分布、 $\chi^2$ 分布、 t分布和F分布

- 离散型随机变量的概率分布
  - 设  $x_i$  ( $i = 1, 2, \dots$ ) 是离散型随机变量  $X$  所取的一切可能值, 称  $P(X = x_i) = p_i, i = 1, 2, \dots, n$  为离散型随机变量  $X$  的概率分布
  - 因此, 离散型随机变量  $X$  的概率分布可以用下表的方式表示

$x_i$	$x_1$	$x_2$	$x_3$	$\dots$	$x_n$
$p_i$	$p_1$	$p_2$	$p_3$	$\dots$	$p_n$



二项分布实际上就是  
重复  $n$  次的伯努利试验

单次成功/失败试验  
称为伯努利试验

## 二项分布

## 泊松分布

用途	用来描述类似伯努利试验 (Bernoulli Experiment) 的只有两种对立结果的随机事件所服从的概率分布	用来描述在一定时空范围内某一事件的出现次数的分布
举例	掷硬币; 农民是否脱贫; 客户是否喜欢产品	在某企业每个月发生的事故次数; 在某服务台, 单位时间内需要接待的顾客人数
公式	设 $X$ 为 $n$ 次重复试验中事件 $A$ 出现的次数, $X$ 取 $x$ 的概率为 $P\{X = x\} = C_n^x p^x q^{n-x}$ $(x = 0, 1, 2, \dots, n)$ 式中: $C_n^x = \frac{n!}{x!(n-x)!}$	$\{X = x\} = \frac{\lambda e^{-\lambda}}{x!}, (x = 0, 1, 2, \dots, n)$ 式中: $\lambda$ 为给定的时间间隔、长度、面积、体积内“成功”的平均数; $x$ 为给定的时间间隔、长度、面积、体积内“成功”的次数
期望值	$E(X) = np$	$E(X) = \lambda$
方差	$D(X) = npq$	$D(X) = \lambda$

泊松分布的期望  
和方差是一样的

- 泊松分布是以 18~19 世纪的法国数学家西莫恩·德尼·泊松 (Siméon Denis Poisson) 命名的
  - 泊松分布适合于描述单位时间内随机事件发生的次数
- 泊松分布与二项分布的关系
  - 当二项分布的  $n$  很大而  $p$  很小时, 泊松分布可作为二项分布的近似, 其中  $\lambda$  为  $np$ 
    - 通常当  $n \geq 10$ ,  $p \leq 0.1$  时, 就可以用泊松公式近似得计算
  - 事实上, 泊松分布正是由二项分布推导而来的
- 应用场景
  - 当一个随机事件, 例如某电话交换台收到的呼叫、来到某公共汽车站的乘客、某放射性物质发射出的粒子、显微镜下某区域中的白血球等等以固定的平均瞬时速率  $\lambda$  (或称密度) 随机且独立地出现时, 那么这个事件在单位时间 (面积或体积) 内出现的次数或个数就近似地服从泊松分布



- 概率密度函数 $f(x)$

- 设 $X$ 为一连续型随机变量， $x$ 为任意实数， $X$ 的概率密度函数记为 $f(x)$ ，也就是说概率密度函数 $f(x)$ 表示 $X$ 的所有取值 $x$ 及其频数 $f(x)$

$$f(x) \geq 0 \quad \int_{-\infty}^{\infty} f(x)dx = 1$$

- 连续型随机变量的概率密度函数是一个描述这个随机变量的输出值，在某个确定的取值点附近的可能性的函数（若可导，实际就是导数）

- 概率分布函数 $F(x)$

- 建立在概率密度函数的基础上，定义如下：

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(t)dt \quad (-\infty < x < \infty)$$

- 可见， $f(x)$ 与 $F(x)$ 之间的关系为 $f(x) = F'(x)$
- 期望值： $E(X) = \int_{-\infty}^{\infty} xf(x)dx = \mu$
- 方差： $D(X) = \int_{-\infty}^{\infty} [x - E(X)]^2 f(x)dx = \sigma^2$

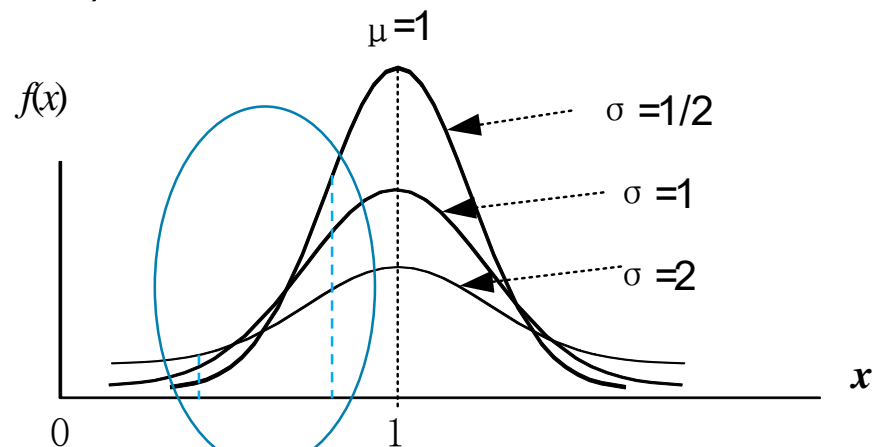
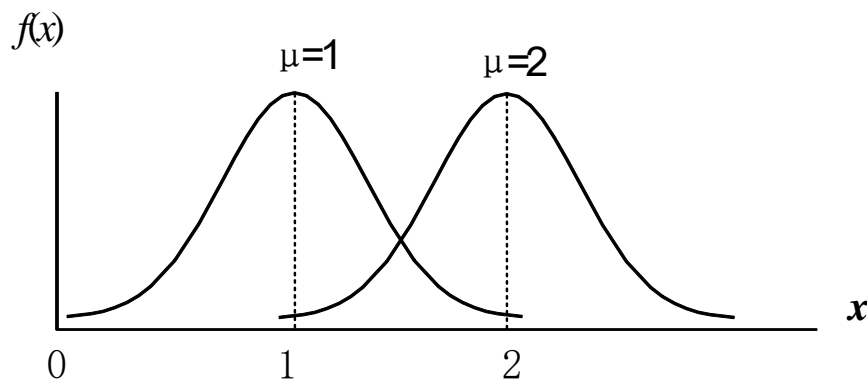
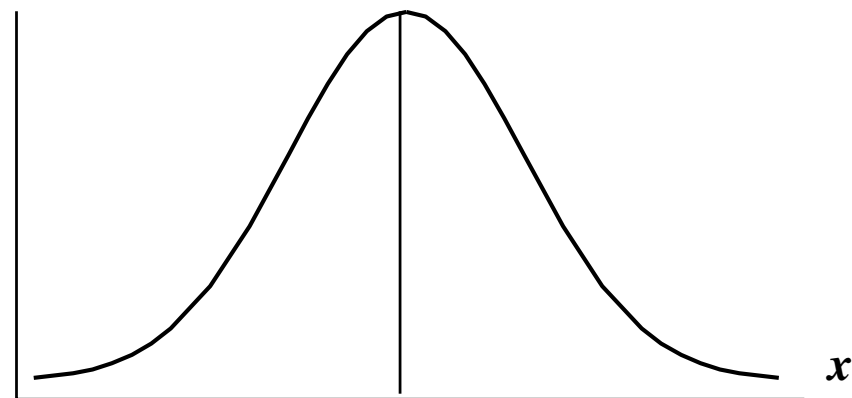
- Normal Distribution, 也称为高斯分布
  - 是描述连续型随机变量的最重要的分布, 它是经典统计推断的基础
- 定义:
  - 如果随机变量 $X$ 的概率密度函数为 $f(x)$
  - $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \quad (-\infty < x < \infty)$
  - $f(x)$  为随机变量  $X$  的频数
  - $\sigma^2$  为总体方差
  - $x$ 为随机变量的取值
  - $\mu$ 为总体均值
  - 那么, 称 $X$ 服从正态分布, 并记作 $X \sim N(\mu, \sigma^2)$

- 正态分布概念是由德国的数学家和天文学家棣·莫弗（De Moivre）于 1733 年首次提出的
- 德国数学家 Gauss 率先将其应用于天文学家研究，故正态分布又叫高斯分布
- 德国 10 马克的印有高斯头像的钞票，其上还印有正态分布的密度曲线
- 正态分布是自然科学与行为科学中的定量现象的一个方便模型



- 概率密度曲线图为钟形曲线
- $X \sim N(\mu, \sigma^2)$
- $\mu$  决定曲线的位置
- $\sigma^2$  决定曲线的平缓程度 (即宽度)

$f(x)$

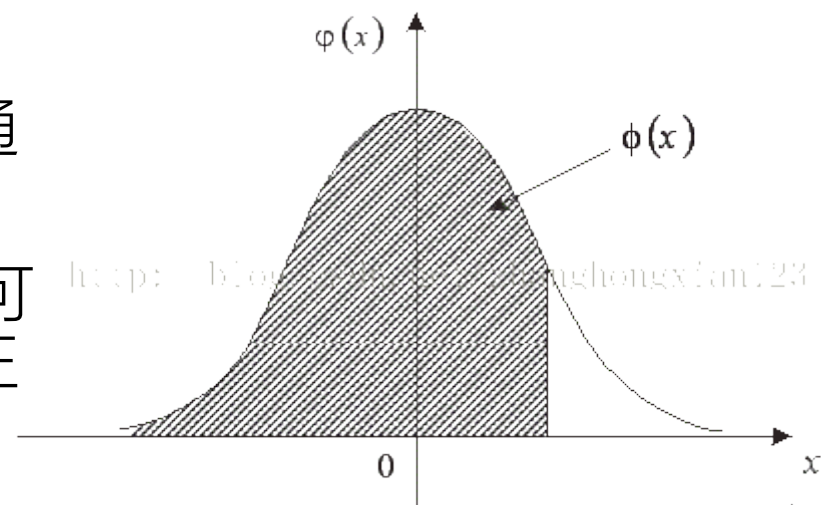


方差越小, 概率取值变化越大

© 2021 Dr Shenkai Gu

- 概率密度曲线图为钟形曲线
- 参数方差 $\sigma^2$ 和均值 $\mu$ 是决定一个正态分布的两个重要因素
- 服从正态分布的随机变量的概率规律
  - 取与均值 $\mu$ 临近的值的概率最高
  - 取离均值 $\mu$ 越远的值的概率越小
- 曲线 $f(x)$ 相对于均值 $\mu$ 对称，尾端向两个方向无限延伸，且理论上永远不会与横轴相交
- 正态曲线下的总面积等于 1
- 不同正态分布的“正态概率分布表”不同。
  - 需要把其它类型的正态分布转化为“标准正态分布”，进而达到“只需要查一张概率分布表”的目的

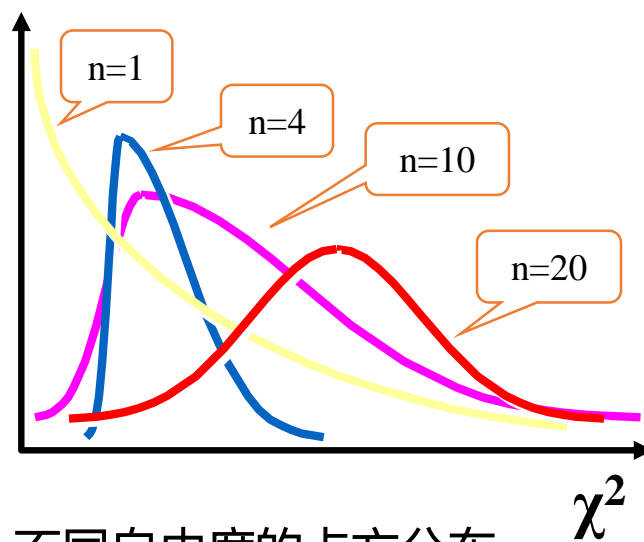
- 标准正态分布是指方差 $\sigma^2 = 1$ 均值 $\mu = 0$ 的一种特殊正态分布
- 标准正态分布概率密度函数 $f(x)$ 可被化简为
  - $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (-\infty < x < \infty)$
- 标准正态分布概率分布函数 $\Phi(x)$ :  $P(X \leq x)$ 的概率
  - $\Phi(x) = \int_{-\infty}^x \phi(x) dt = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$
- 任何一个一般的正态分布都可以通过线性变换之后转化为标准分布
- 对于一般正态分布 $X \sim N(\mu, \sigma^2)$ , 可通过下面的线性变换转化为标准正态分布
  - $Z = \frac{X - \mu}{\sigma} \sim N(0, 1)$





- 主要刻画的是一个总体 ( $n$  个随机变量) 为标准正态分布时, 所对应的样本方差 (随机变量的平方和) 的分布情况
  - $n$  个独立的标准正态分布变量的平方和服从自由度为  $n$  的卡方分布
- 定义: 设总体服从正态分布  $X \sim N(\mu, \sigma^2)$ ,  $X_1, X_2, \dots, X_n$  为来自正态总体的样本, 则样本方差  $s^2$  的分布为
  - $\frac{(n-1)s^2}{\sigma^2} \sim \chi^2(n-1)$
- 式中, 将  $\chi^2(n-1)$  称为自由度为  $(n-1)$  的卡方分布
- 期望值:  $E(\chi^2(n)) = n$
- 方差:  $D(\chi^2(n)) = 2n$
- 注: 样本方差为构成样本的随机变量对离散中心  $x$  之离差的平方和除以  $(n-1)$ , 用来表示一系列数的变异程度

# 卡方 ( $\chi^2$ ) 分布 (续)



不同自由度的卡方分布

- 卡方分布是由正态分布构造而成的一个新的分布
- 当自由度  $n$  很大时,  $\chi^2$  分布接近于正态分布

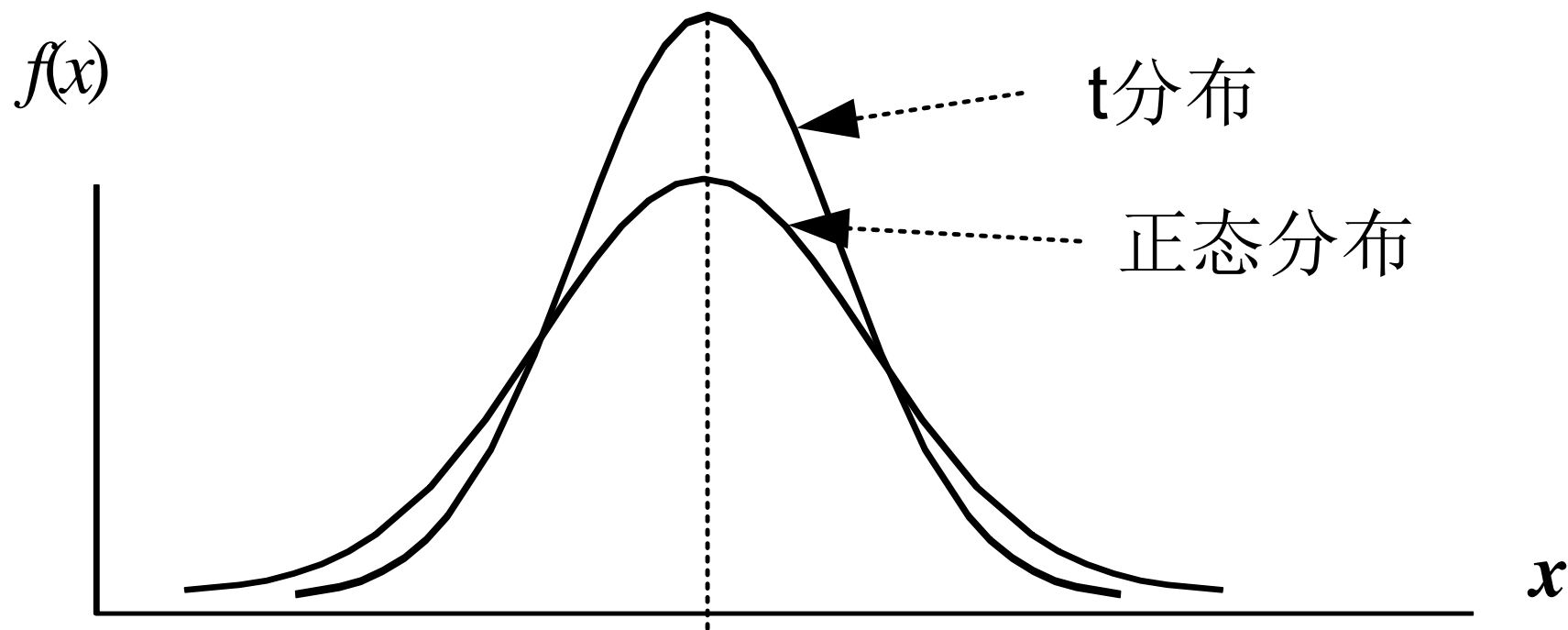
- 在数据分析中，当总体标准差 $\sigma$ 为未知数时，我们可以采用t分布，用样本标准差 $S$ 代替总体标准差 $\sigma$ ，由样本平均数推断总体平均数及两个小样本之间的显著性检验
- 【定义】设 $X_1, X_2, \dots, X_{n-1}$ 是来自正态总体 $N(\mu, \sigma^2)$ 的一个样本，则
  - $\frac{\sqrt{n}(\bar{X} - \mu)}{S} \sim t(n - 1)$
- 称为服从自由度为 $(n - 1)$ 的 t 分布，式中
  - $\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i$
  - $s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2$



"Student" in 1908

威廉·戈塞，英国现代统计方法发展的先驱，以笔名"Student"署名

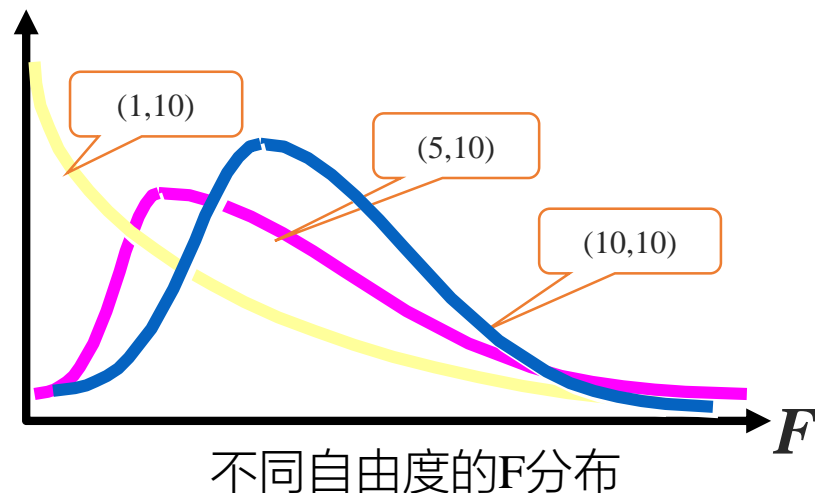
# t分布与正态分布的比较示意图



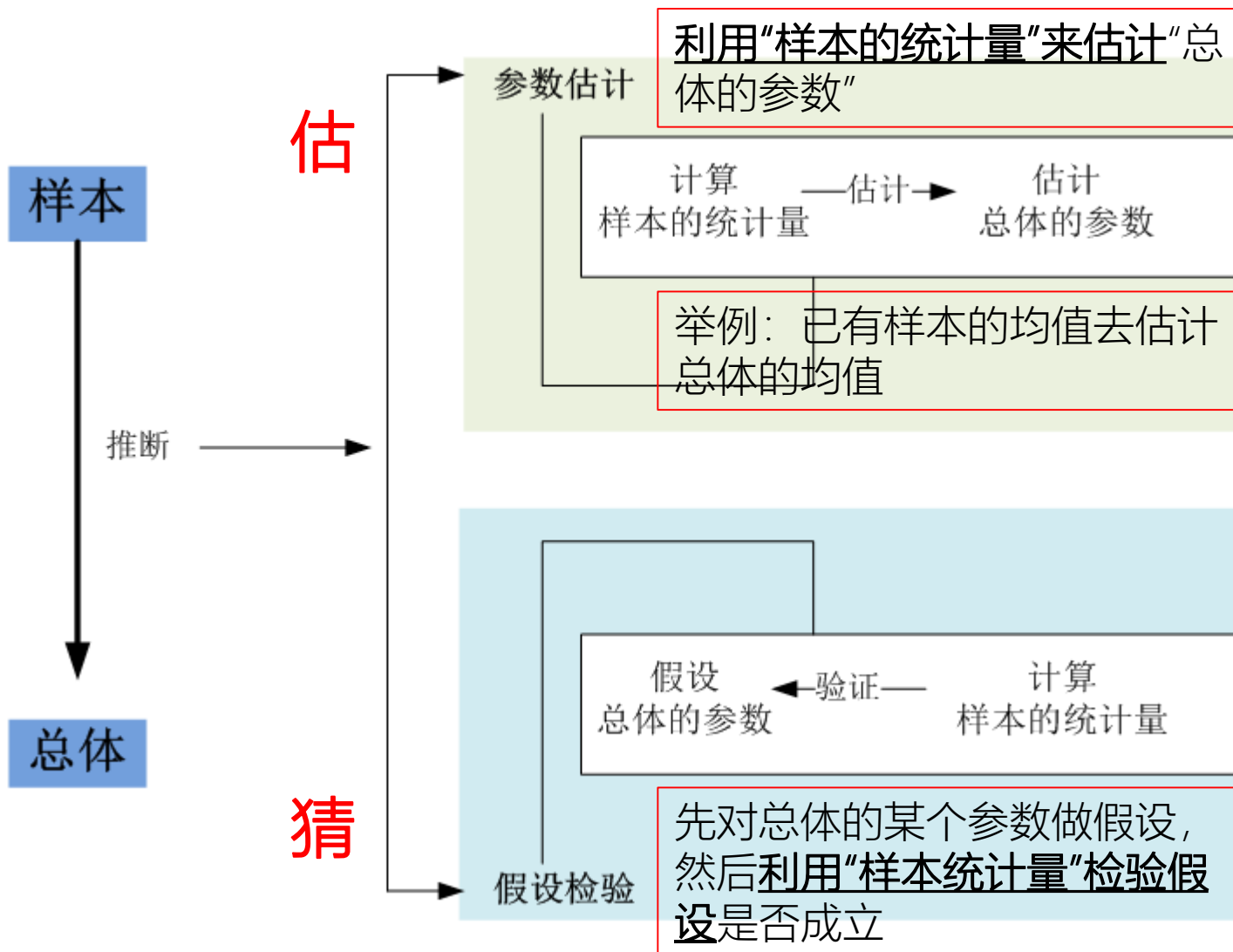
- 当  $n \geq 2$  时, t分布的数学期望  $E(t)=0$
- 当  $n \geq 3$  时, t分布的方差  $D(t) = \frac{n}{n-2}$
- 与标准正态分布的相似, 均为单峰偶函数, 区别在于t分布的密度函数在两侧的尾部比标准正态分布宽一些, t分布的方差比标准正态分布大一些
- 在实际应用中, 当  $n \geq 0$  时, t分布与标准正态分布非常接近
  - 自由度越小, t分布曲线越低平
  - 自由度越大, t分布曲线越接近标准正态分布 (u分布) 曲线

- 1924 年英国统计学家 R. A. Fisher 提出
- F分布刻画的是当两个总体均为正态分布时，分别这两个总体的两个样本差 ( $s_1$ 和 $s_2$ ) 之间的比例的分布情况，主要用于方差分析和回归方程的显著性检验之中
- 定义：设 $X_1, X_2, \dots, X_{n_1}$ 是来自正态总体 $N \sim (\mu_1, \sigma_1^2)$ 的一个样本， $Y_1, Y_2, \dots, Y_{n_2}$ 是来自另一个正态总体 $N \sim (\mu_2, \sigma_2^2)$ 的一个样本，且 $X_i (i = 1, 2, \dots, n_1)$ ， $Y_i (i = 1, 2, \dots, n_2)$ 相互独立，则
  - $\frac{s_x^2/s_y^2}{\sigma_1^2/\sigma_2^2} = \frac{s_x^2/\sigma_2^2}{s_y^2/\sigma_1^2} \sim F(n_1 - 1, n_2 - 1)$
- 我们将 $F(n_1 - 1, n_2 - 1)$ 称为第一自由度为 $(n_1 - 1)$ ，第二自由度为 $(n_2 - 1)$ 的F分布

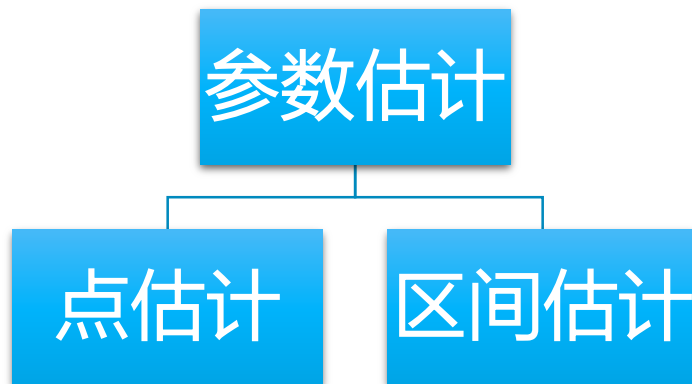
- 一个F分布的随机变量是两个卡方分布变量的比率
- F分布定义为
  - 设  $X$ 、 $Y$  为两个独立的随机变量,  $X$  服从自由度为  $m$  的卡方分布,  $Y$  服从自由度为  $n$  的卡方分布, 这 2 个独立的卡方分布被各自的自由度除以后的比率这一统计量的分布即  $F = x/m / y/n$  服从自由度为  $(m,n)$  的F分布, 上式F服从第一自由度为  $m$ , 第二自由度为  $n$  的F分布
- 期望值:  $E(F) = \frac{n_2}{n_2 - 2}, n_2 > 2$
- 若  $F \sim F(n_1, n_2)$ , 则  $1/F \sim F(n_2, n_1)$
- F分布于t分布的关系
  - 若  $T \sim t(n)$ , 则  $T^2 \sim F(1, n)$



# 参数估计：“样本”推断“总体”

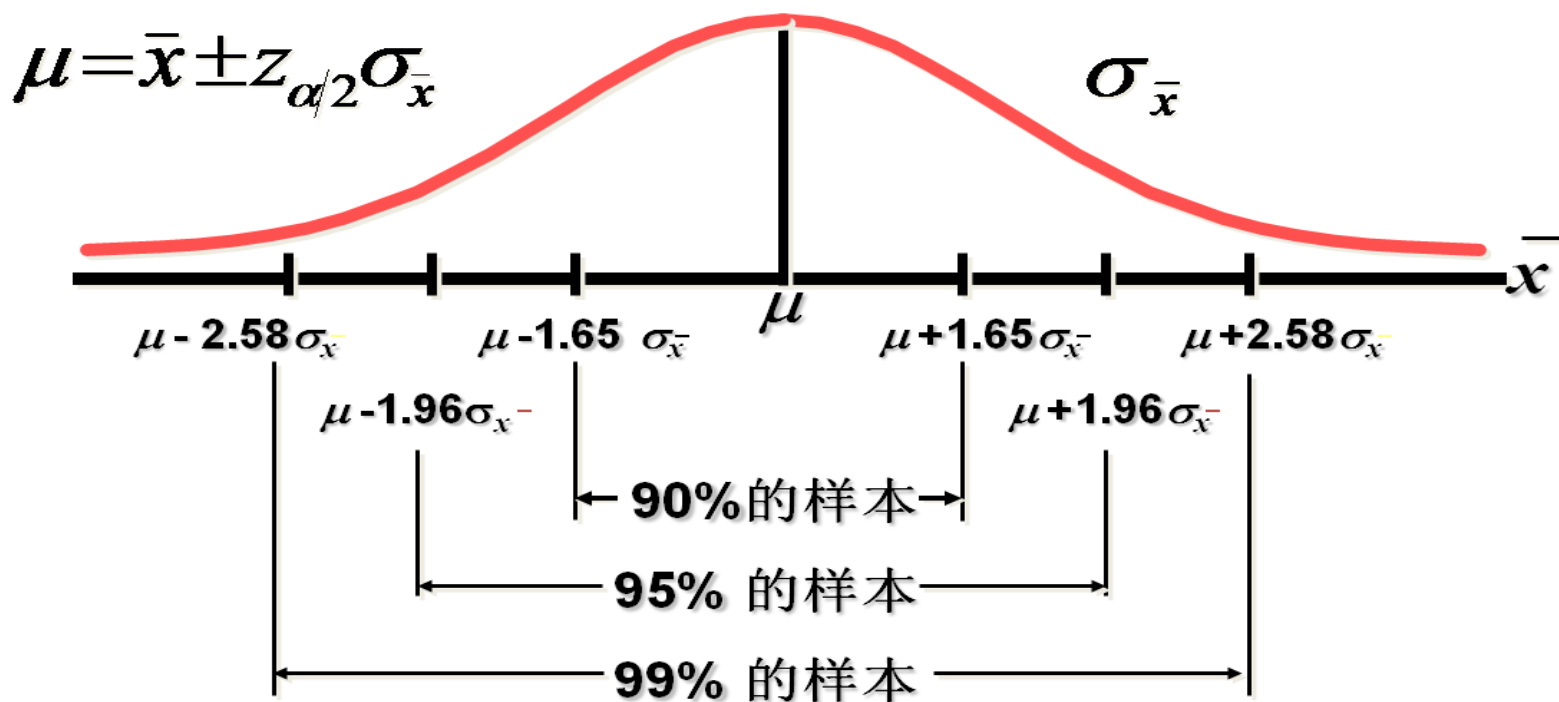






- 基本思路
  - 从总体中抽取一个样本（不是一个值，一个样本包含很多值），根据该样本的统计量对总体的未知参数做出一个数值点的估计
- 例如
  - 用样本均值作为总体未知均值 $\mu$ 的估计值
- 注意
  - 点估计没有给出估计值接近总体未知参数程度的信息
- 方法
  - 矩估计法、顺序统计量法、最大似然法、最小二乘法等

- 在点估计的基础上，给出总体参数落在某一区间的概率
- 区间就是根据一个样本的观察值给出的总体参数的估计范围，可通过样本统计量加减抽样误差的方法计算
- 例如，总体均值落在 50~70 之间的概率为 0.95

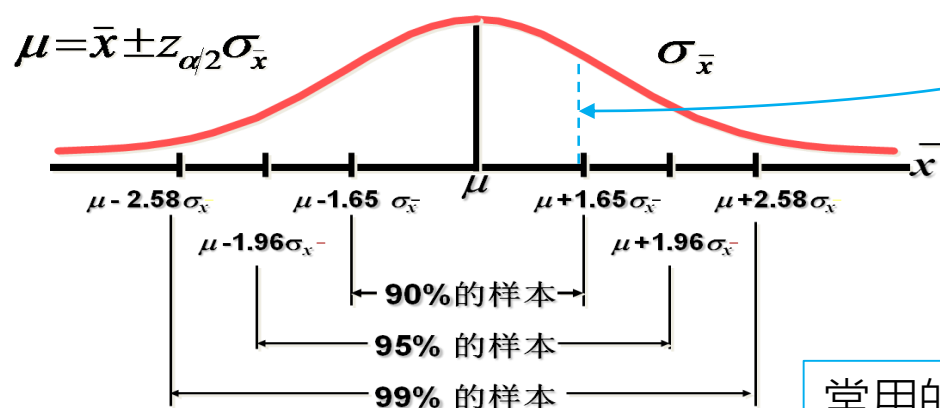


- 置信区间 Confidence interval

- 指由样本统计量所构造的总体参数的估计区间，其中区间的最小值和最大值分别称为置信下限和置信上限

- 置信水平 Confidence level

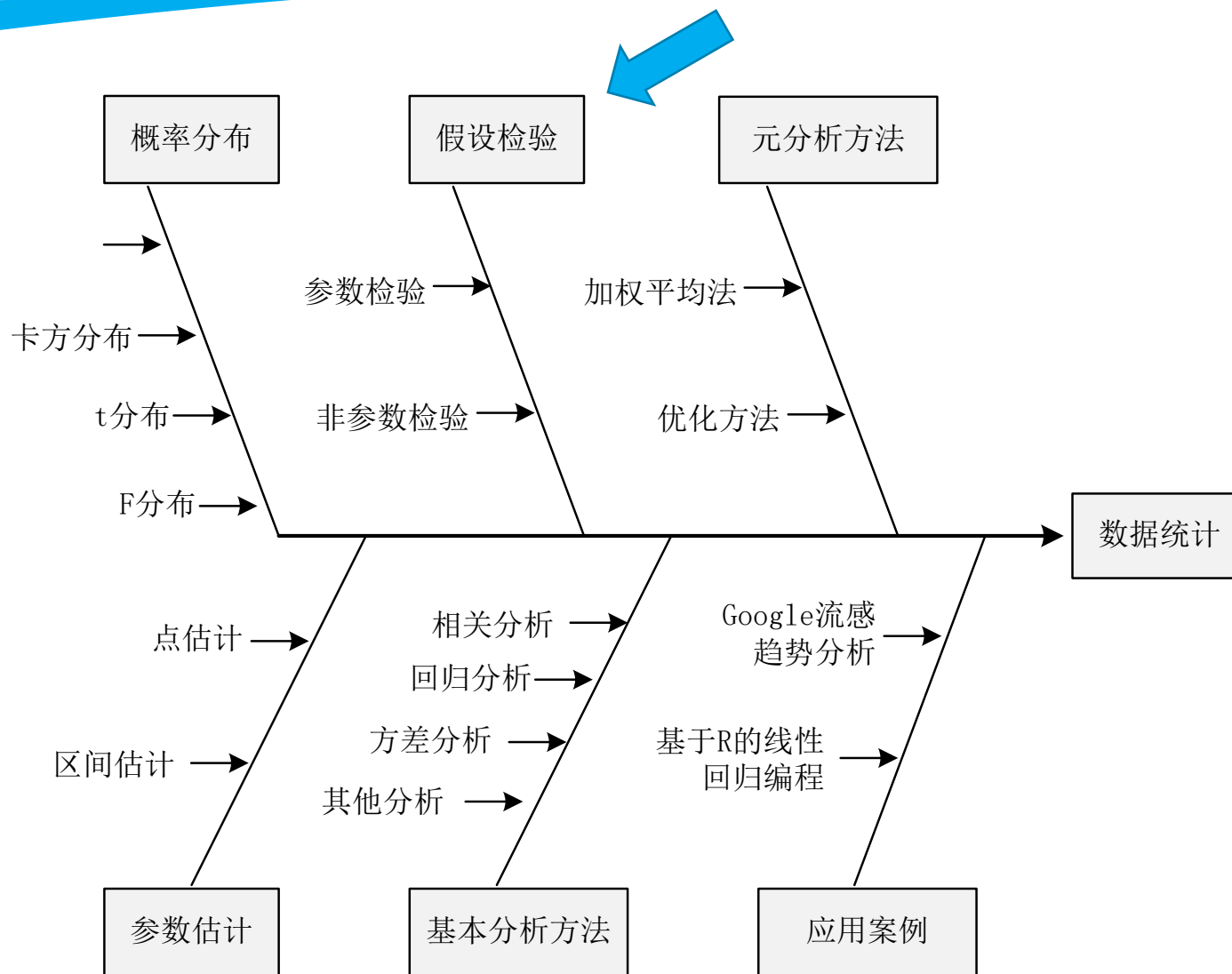
- 指总体未知参数落在区间内的概率，表示为  $(1-\alpha)$  (置信度)， $\alpha$  为显著性水平（犯错误的概率），即总体参数未落在区间内的概率
- 在构造置信区间时，可以任意设置目标置信水平
- 常用的置信水平及正态分布曲线下右侧面积为  $\alpha/2$  时的  $z$  值 ( $z_{\alpha/2}$ )



置信水平	$\alpha$	$\alpha/2$	$z_{\alpha/2}$
90	0.1	0.05	1.645
95	0.05	0.025	1.96
99	0.01	0.005	2.58

常用的置信水平及正态分布曲线下右侧面积为  $\alpha/2$  时的  $z$  值 ( $z_{\alpha/2}$ )

- 参数估计中，用于估计总体某一参数的随机变量称为**估计量 estimator**
  - 如**样本均值** $\bar{x}$ 就是总体均值 $\mu$ 的一个估计量
- 判断估计量的优良性准则：
  - 无偏性 Unbiasedness（期望相等）
    - 估计量的**数学期望**等于被估计的总体参数
  - 有效性 Efficiency（方差要小）
    - 一个**方差较小**的无偏估计量称为一个**更有效**的估计量
    - 如，与其他估计量相比，样本均值是一个更有效的估计量
  - 一致性 Consistency（极限接近）
    - 随着样本容量的增大，估计量**越来越接近**被估计的**总体参数**



- 假设检验

- 根据已掌握的资料对一个总体参数是否等于某一个数值，某一随机变量是否服从某种概率分布的假设
- 然后根据所取得的样本资料，利用一定的统计方法计算出有关检验的统计量，依据一定的概率原则，以较小的风险来判断估计数值与总体数值（或估计分布与实际分布）是否存在显著差异，是否应当接受原假设的一种检验方法
- 假设检验是进行科学决策的有力工具

- 参数假设检验

- 是指在总体的分布形式已知的条件下，对总体参数的某一假设进行的检验

- 非参数假设检验

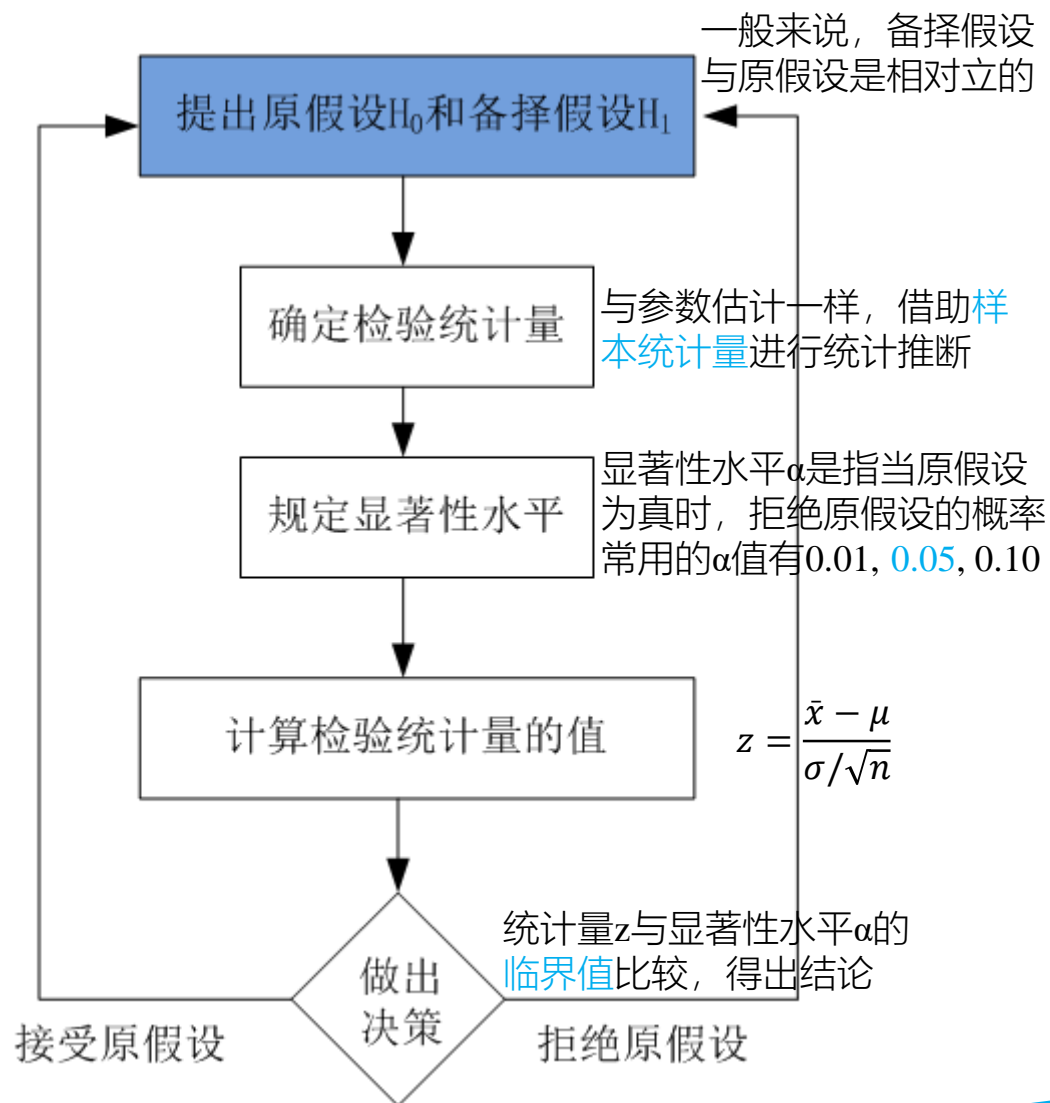
- 是指在总体分布未知时，对其分布的具体形式、特征值等信息进行的假设检验

- 首先对所研究的命题提出一种无显著差异的假设，并假定这一假设成立，然后由此导出其必然的结果
- 如果能证明这种结果出现的可能性很小，那么我们就有理由认为原假设是错误的，从而拒绝接受这个假设
- 否则，我们就没有理由拒绝原假设，而认为原假设是可接受的
- 假设检验是根据小概率事件的实际不可能性原理来推断的
- 假设检验中的小概率标准称为显著性水平，用 $\alpha$ 表示
- 因此，假设检验也称为显著性检验

- 依据显著性水平的大小将检验统计量的所有可能值组成的样本空间分为两个区域
- 否定域或拒绝域
  - 在原假设成立的情况下，如果检验统计量的值落在这个区域里，则否定原假设
- 接受域
  - 在原假设成立的情况下，如果检验统计量的值落在这个区域里，则接受原假设



- 主要以小概率原理为基础
- 小概率：一次试验中，一个几乎不可能发生的事件的概率
- 标准是概率 $\leq 0.05$
- 小概率原理：如果一次试验中出现小概率事件，则有理由拒绝假设

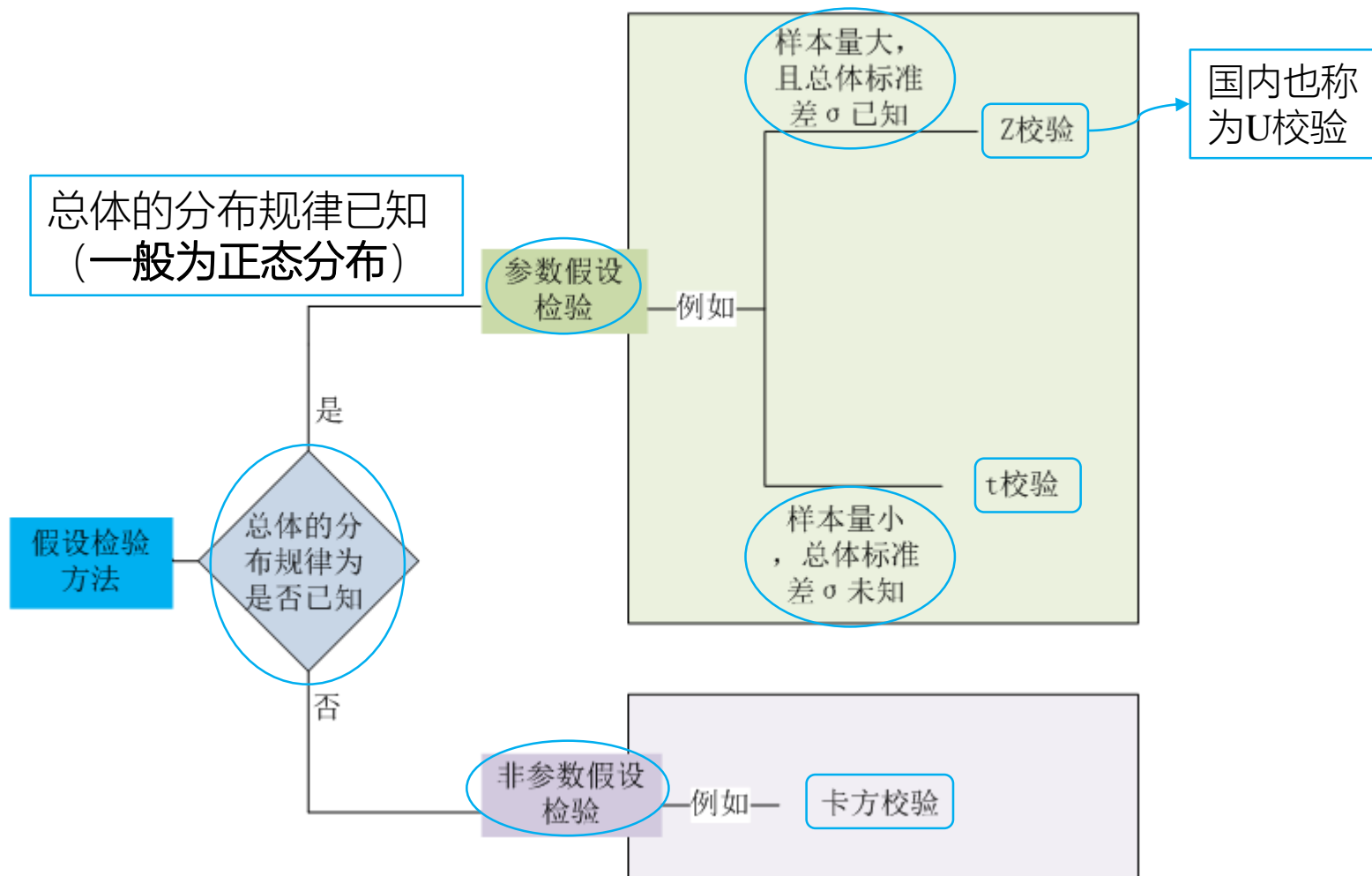


- 弃真错误： $\alpha$ 错误（真的当假的）
  - 显著性水平： $\alpha$ 错误的出现概率
  - 常用的 $\alpha$ 值有0.01, 0.05, 0.10
- 取伪错误： $\beta$ 错误（假的当真的）

项目	未拒绝原假设 $H_0$	拒绝原假设 $H_0$
$H_0$ 为真	$1-\alpha$ （正确）	$\alpha$ （弃真错误）
$H_0$ 为假	$\beta$ （取伪错误）	$1-\beta$ （正确）

- 注意：假设检验中这两种错误存在矛盾关系
  - 如果减少 $\alpha$ 错误，就会增加犯 $\beta$ 错误的可能
  - 如果减少 $\beta$ 错误，就会增加犯 $\alpha$ 错误的可能
- 在样本量不变的情况下，假设检验中无法同时减少这两类错误发生的概率，所以实际中要有所侧重
- 一般来说，统计领域经常遵循“先控制 $\alpha$ 错误”的原则
  - 就是不能把真的丢了

# 假设检验：两种类型



- 应用前提

- 总体的分布规律为已知（一般为正态分布）
- 一般用于
  - 检验一个样本平均数与一个已知的总体平均数的差异
  - 检验来自两个的两组样本平均数的差异性，从而判断它们各自代表的总体的差异是否显著
- 在参数检验中，一般选择正态分布检验和t检验等方法

- 制定假设 $H_0$ ：假设指标（如两个均值）没有差异，**概率 $P$** 与 $H_0$ 的关系
- 如果**样本量大**（大于 30，称为大样本）且**总体标准差 $\sigma$** 已知，那么，采用 **$z$ 统计量**（ $Z$ 校验）

$$z = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$$

样本标准差已知途径，多种途径，如从他人研究成果的值

表  $P$  与  $H_0$  的关系

P值	$H_0$ 成立概率大小	差异显著程度
$P \leq 0.01$	$H_0$ 成立概率极小	差异非常显著
$P \leq 0.05$	$H_0$ 成立概率较小	差异显著
$P \geq 0.05$	$H_0$ 成立概率较大	差异不显著

- 然后，比较计算所得 $Z$ 值与理论 $Z$ 值，推断发生的概率，依据 $Z$ 值与差异显著性关系表作出判断
- 根据计算出来的 $Z$ 值和 $P$ 值，确定差异程度
- 注意：如果样本量大，且总体标准差 $\sigma$ 未知，可用样本标准差来代替

$Z$ 值与 $P$ 值关系

$ Z $	P值	差异程度
$\geq 2.58$	$\leq 0.01$	非常显著
$\geq 1.96$	$\leq 0.05$	显著
$< 1.96$	$> 0.05$	不显著

- 如果样本量小，或总体标准差 $\sigma$ 未知，那么，一般只能采用样本标准差 $s$ 进行计算，即采用 $t$ 统计量
  - $t = \frac{\bar{x} - \mu_0}{s / \sqrt{n}}$
- 然后，比较计算所得 $Z$ 值与理论 $Z$ 值，推断发生的概率，依据 $Z$ 值与差异显著性关系表作出判断

# 参数假设检验例子 (source: 百度百科)



- 某项教育技术实验，对实验组和控制组的前测和后测的数据分别如下表所示，比较两组前测和后测是否存在差异
- 由于 $n > 30$ ，属于大样本，所以采用Z检验
- 由于这是检验来自两个不同总体的两个样本平均数，看它们各自代表的总体的差异是否显著，所以采用双总体的Z检验方法

实验组和控制组的前测和后测数据表				
前测	实验组	$n_1 = 50$	$\bar{X}_{1a} = 76$	$S_{1a} = 14$
	控制组	$n_2 = 50$	$\bar{X}_{2a} = 78$	$S_{2a} = 16$
后测	实验组	$n_1 = 50$	$\bar{X}_{1b} = 85$	$S_{1b} = 8$
	控制组	$n_2 = 50$	$\bar{X}_{2b} = 80$	$S_{2b} = 14$

- 计算前测Z的值：
$$Z = \frac{76 - 78}{\sqrt{\frac{14^2}{50} + \frac{16^2}{50}}} = -0.658$$
 (公式) 
$$Z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

- $|Z| = 0.658 < 1.96 \Rightarrow$  前测两组差异不显著

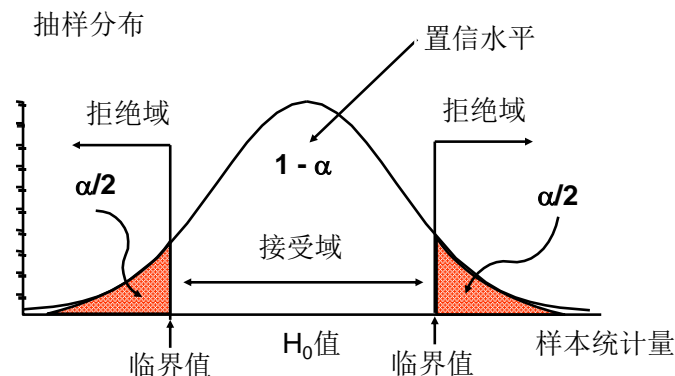
- 计算后测Z的值：
$$Z = \frac{85 - 80}{\sqrt{\frac{8^2}{50} + \frac{14^2}{50}}} = 2.16$$

- $|Z| = 2.16 > 1.96 \Rightarrow$  后测两组差异显著

Z值与P值关系

Z	P值	差异程度
$\geq 2.58$	$\leq 0.01$	非常显著
$\geq 1.96$	$\leq 0.05$	显著
$< 1.96$	$> 0.05$	不显著

检验	研究的问题		
	双侧检验	左单侧检验	右单侧检验
$H_0$	$\mu = \mu_0$	$\mu \geq \mu_0$	$\mu \leq \mu_0$
$H_1$	$\mu \neq \mu_0$	$\mu < \mu_0$	$\mu > \mu_0$



- 检验统计量的检验结论按下列方法作出
  - 若采用**双侧**检验，则检验的**双侧临界值**为 $-t_{\alpha/2}$ 和 $t_{\alpha/2}$ 
    - 当 $|t| > t_{\alpha/2}$ 时，拒绝原假设；反之，接收原假设
  - 若采用**左侧**检验，则检验的**左侧临界值**为 $-t_{\alpha}$ 
    - 当 $t < -t_{\alpha}$ 时，拒绝原假设；反之，接收原假设（怕小不怕大）
  - 若采用**右侧**检验，则检验的**右侧临界值**为 $t_{\alpha}$ 
    - 当 $t > t_{\alpha}$ 时，拒绝原假设；反之，接收原假设（怕大不怕小）



- 设 $\mu_0$ 表示在零假设和备择假设中考虑的某一特定数值
- 一般来说，对总体均值的假设检验采取下面的三种形式之一

$$\left. \begin{array}{l} H_0 : \mu \geq \mu_0 \\ H_1 : \mu < \mu_0 \end{array} \right\} \rightarrow \text{左侧检验——怕小不怕大}$$

如，质量检查抽样要求罐头重要不低于50g

$$\left. \begin{array}{l} H_0 : \mu \leq \mu_0 \\ H_1 : \mu > \mu_0 \end{array} \right\} \rightarrow \text{右侧检验——怕大不怕小}$$

$$\left. \begin{array}{l} H_0 : \mu = \mu_0 \\ H_1 : \mu \neq \mu_0 \end{array} \right\} \rightarrow \text{双侧检验——过大过小均拒绝}$$

- 某厂购买一台新的机器，生产零件长度规定为10cm。为检验机器性能是否良好，质检员随机抽取了25件产品，测得平均长度为9.8cm，标准差为0.4cm。假设生产的零件长度服从正态分布，问在显著性水平 $\alpha=0.05$ 时，该机器的性能是否良好？
- 设 $X$ 表示该机器生产零件的长度，则有 $X \sim N(\mu, \sigma^2)$ ，样本容量 $n=25$ ，样本均值 $\bar{x}=9.8\text{cm}$ ，样本标准差 $s=0.4\text{cm}$ 。根据问题提出的假设为： $H_0: \mu = \mu_0 = 10\text{cm}$ ； $H_1: \mu \neq \mu_0 = 10\text{cm}$
- 这是一个双侧检验问题，因为总体服从正态分布但总体方差未知，用检验的小样本数据检验，故当 $H_0$ 成立时，检验统计量为：
$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{9.8 - 10}{0.4/\sqrt{25}} = -2.5$$
- 规定显著性水平为 $\alpha=0.05$ ，查t检验临界值置信水平表得到临界值 $t_{\alpha/2} = 2.064$ ，所以原假设的否定域为： $|t| > 2.064$
- 因为 $|-2.5| = 2.5 > 2.064$ ，落在否定域，所以否定 $H_0$ ，接受 $H_1$ 。表明在显著性水平 $\alpha=0.05$ 时，不能说该机器的性能良好

- 非参数检验的优点

- 应用范围的广泛性，可用于非正态的、方差不等的、分布规律未知的数据分析工作中

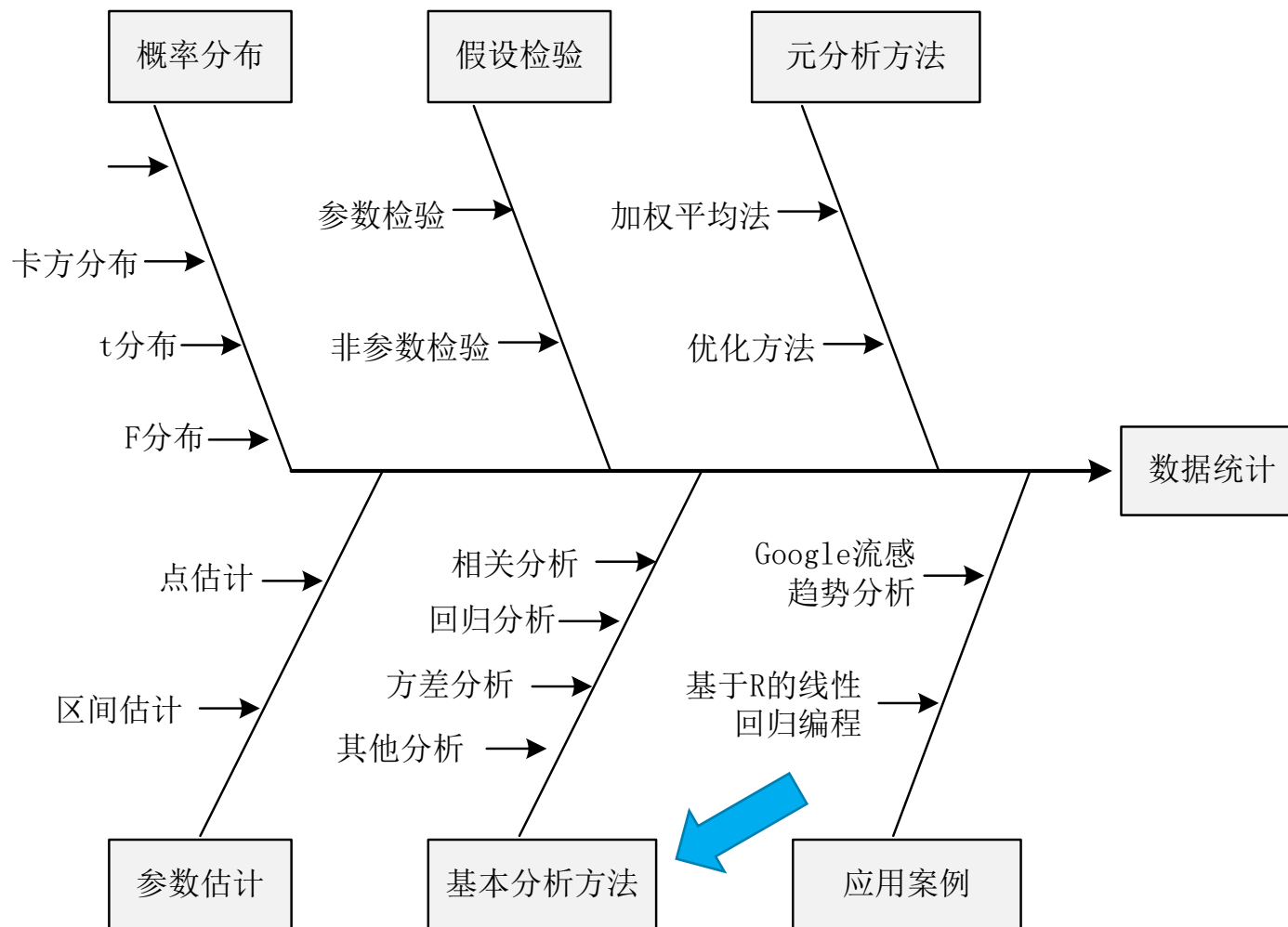
- 非参数检验方法

- 卡方检验：统计样本的实际观测与理论推断值之间的偏离程度
  - 卡方值越大，实际观测值与理论推断值之间的偏离程度越大
  - 卡方值越小，区域符合
  - 卡方值为0时，量值完全不符合

- 卡方值的计算方法

$$\chi^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i} \sim \chi^2(k - 1)$$

- 式中， $O_i$ 为观测值， $E_i$ 为期望值



- 常用统计分析方法：

- 相关分析
  - 线性相关分析
  - 非线性相关分析
- 回归分析
  - 一元线性回归
  - 多元线性回归
- 方差分析
  - 单因素方差分析
  - 多因素方差分析
- 时间序列分析
  - 平稳序列的预测
  - 趋势序列的预测
  - 季节序列的预测
  - 符合序列的预测

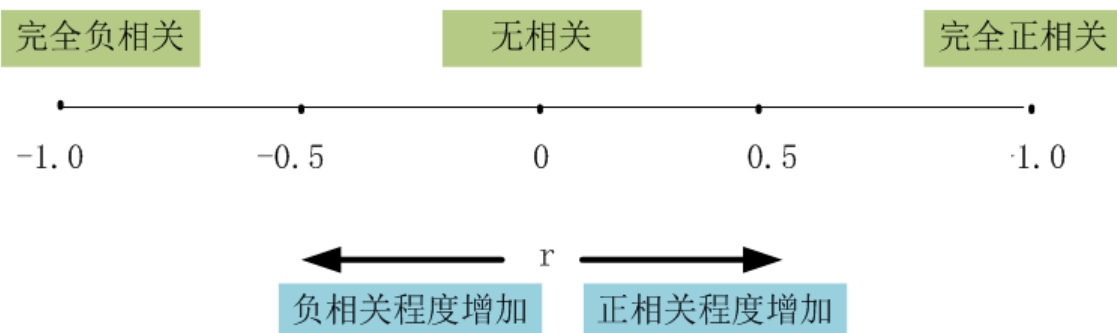
数据挖掘  
的范畴

- 分类分析
  - 决策树分析
  - 贝叶斯网络分析
  - kNN法
- 聚类分析
  - 分层聚类分析
  - *K*-means聚类
  - Kohonen Network聚类
- 其它学科方法
  - 如关联规则分析等

- 对变量之间关系密切程度的度量
  - 相关系数的分析方法
  - 不代表具体关系（如线性关系等）
- 相关系数有两种
  - 总体相关系数 ( $\rho$ )
  - 样本相关系数 ( $r$ )
    - 相对于总体相关系数的提法
    - 若相关系数是根据样本数据计算的，称为样本相关系数
- 通常，由于总体相关系数是未知数，我们一般采用样本相关系数进行相关性分析
- 样本相关计算公式

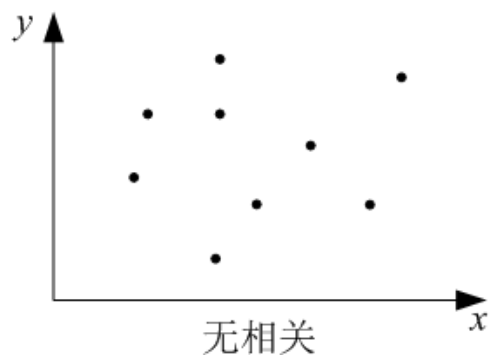
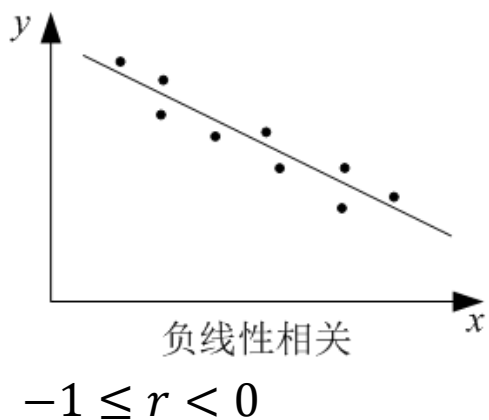
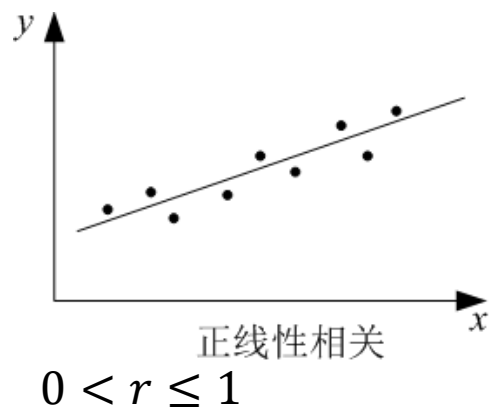
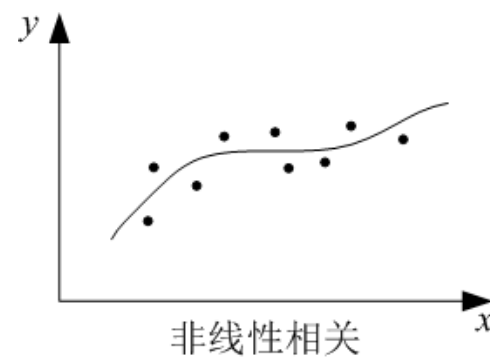
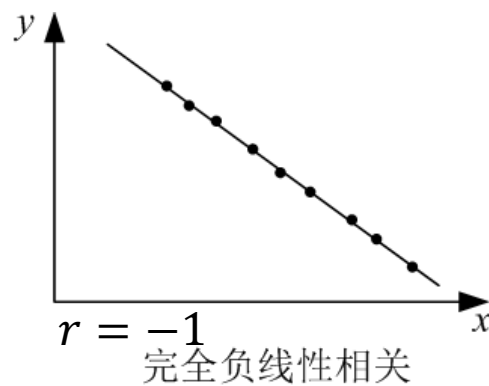
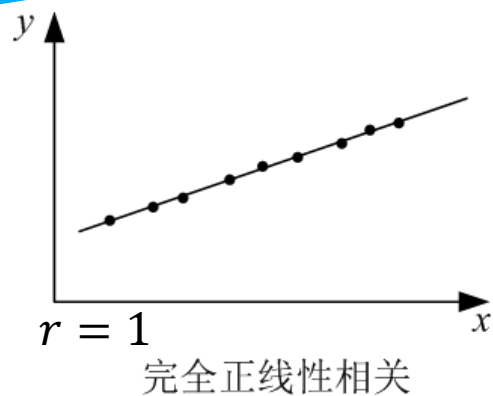
$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \cdot \sum (y - \bar{y})^2}} = \frac{n \sum xy - \sum x \sum y}{\sqrt{n \sum x^2 - (\sum x)^2} \cdot \sqrt{n \sum y^2 - (\sum y)^2}}$$

- 样本相关系数 $r$ 的取值范围是 $[-1, 1]$ 
  - $|r|$ 越趋于1表示关系越密切
  - $|r|$ 越趋于0表示关系越不密切



- 相关关系
  - $|r| = 1$ : 完全相关
  - $r = 1$ : 完全正相关
  - $r = -1$ : 完全负相关
  - $r = 0$ : 不存在线性相关关系
  - $-1 \leq r < 0$ : 负相关
  - $0 < r \leq 1$ : 正相关

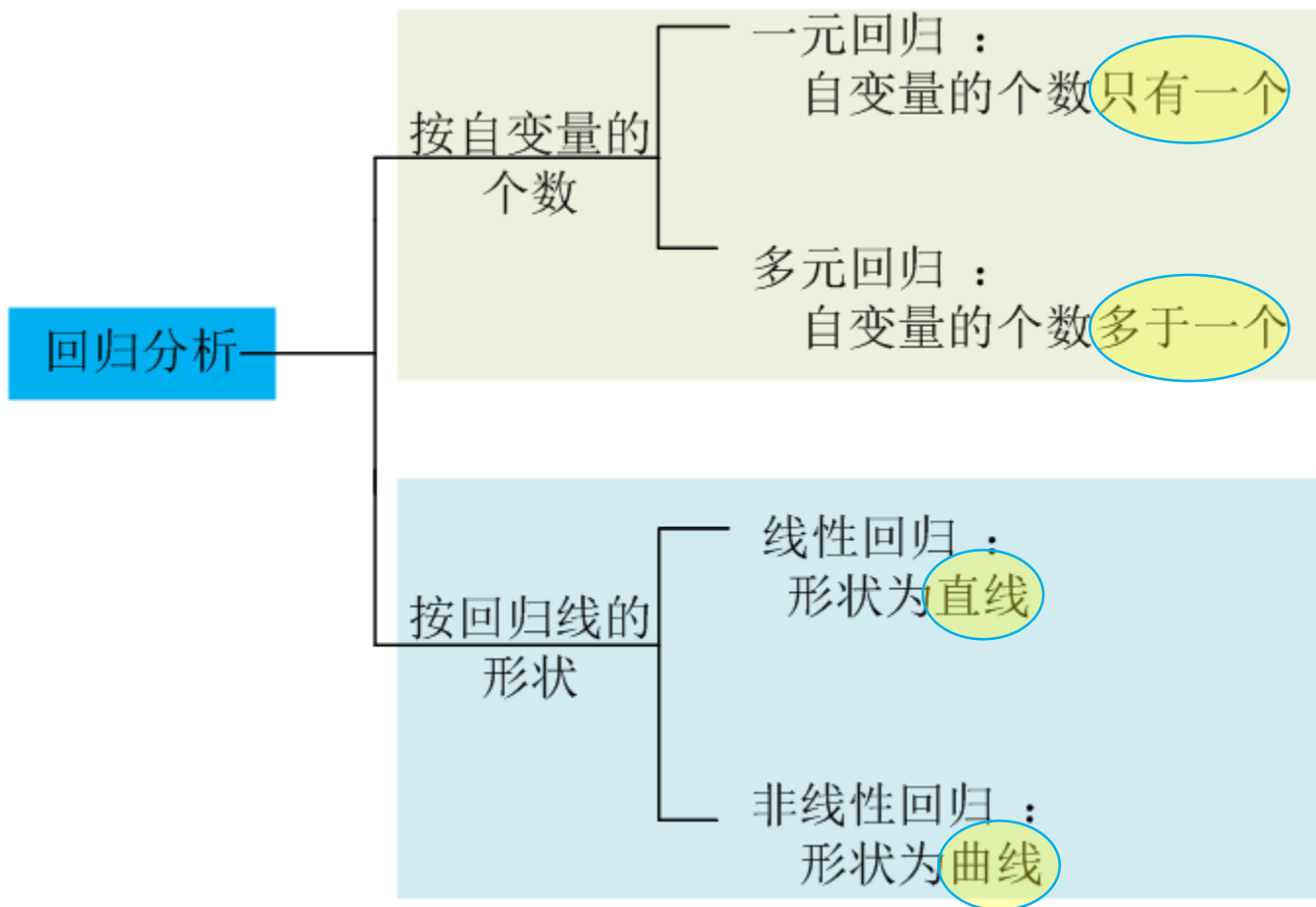
# 相关关系示意图





- 考虑到 $r$ 的抽样分布不一定是正态分布，通常对其不进行正态检验，而采用 $t$ 检验（样本量小，标准差未知）
- $r$ 的显著性检验步骤如下：
  - 第1步：提出原假设（ $H_0$ ）和备择假设（ $H_1$ ）
    - $H_0: \rho = 0$ （表示不相关）
    - $H_1: \rho \neq 0$
  - 第2步：计算检验的统计量 $t$ 
$$t = |r| \frac{\sqrt{n-2}}{\sqrt{1-r^2}} \sim t(n-2)$$
  - 第3步：进行决策
    - 显著性水平 $\alpha$ 和自由度 $n-2$ 查阅 $t$ 分布表，得出 $t_{\alpha/2}$ 的临界值
    - 若 $|t| > t_{\alpha/2}$ ，拒绝 $H_0$ ；若 $|t| < t_{\alpha/2}$ ，接受 $H_0$

- 试图找出变量之间的函数关系的一种统计分析方法
- 函数关系和相关关系是两个不同概念
  - 在函数关系 $y = f(x)$ 中，自变量（解释变量） $x$ 和因变量（被解释变量） $y$ 之间必须存在以下关系
    - 如果 $x$ 确定一个值， $y$ 就有唯一的一个值与 $x$ 对应
  - 如果一个 $x$ 值对应多个 $y$ 值，不能称之为函数关系，只能认为 $x$ 和 $y$ 之间存在相关关系
- 回归分析和相关分析的关系
  - 回归分析就是对具有相关关系的两个或两个以上变量之间数量变化的一般关系进行测定，确立一个相应的数学表达式，以便从一个已知量来推测另一个未知量，为估算预测提供一个重要的方法
  - 回归分析和相关分析是互相补充、密切联系的，相关分析需要回归分析来表明现象数量关系的具体形式，而回归分析则应该在相关分析的基础上进行
  - 即：先做相关分析看有没有关系，再做回归分析确定具体关系

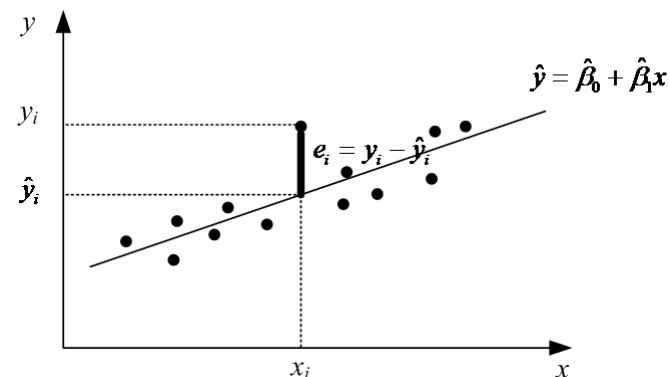


- $y = \beta_0 + \beta_1 x + \varepsilon$
- 模型含义：  $y$  是  $x$  的线性函数（部分） + 误差项  $\varepsilon$
- 线性部分反映了由于  $x$  的变化而引起的  $y$  的变化
- 误差项  $\varepsilon$  是随机变量
  - 反映了除  $x$  和  $y$  之间的线性关系之外的随机因素对  $y$  的影响
  - 误差项的期望值为 0，即  $E(\varepsilon) = 0$ 
    - 也就是说对于一个给定的  $x$  值，  $y$  的期望值为  $E(y) = \beta_0 + \beta_1 x$
- $\beta_0$  是回归直线在  $y$  轴上的截距
  - 表示当  $x = 0$  时  $y$  的期望值
- $\beta_1$  是直线的斜率，有时称为回归系数
  - 表示当  $x$  每变动一个单位时，  $y$  的平均变动值

# 参数 $\beta_0$ 和 $\beta_1$ 的估计方法



- 可以采用最小二乘法进行估计
- 通常，总体回归参数 $\beta_0$ 和 $\beta_1$ 是未知的
  - 我们需利用样本参数（ $\hat{\beta}_0$ ,  $\hat{\beta}_1$ ）估计
- 因此，一元线性回归方程可以改写为
  - $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$



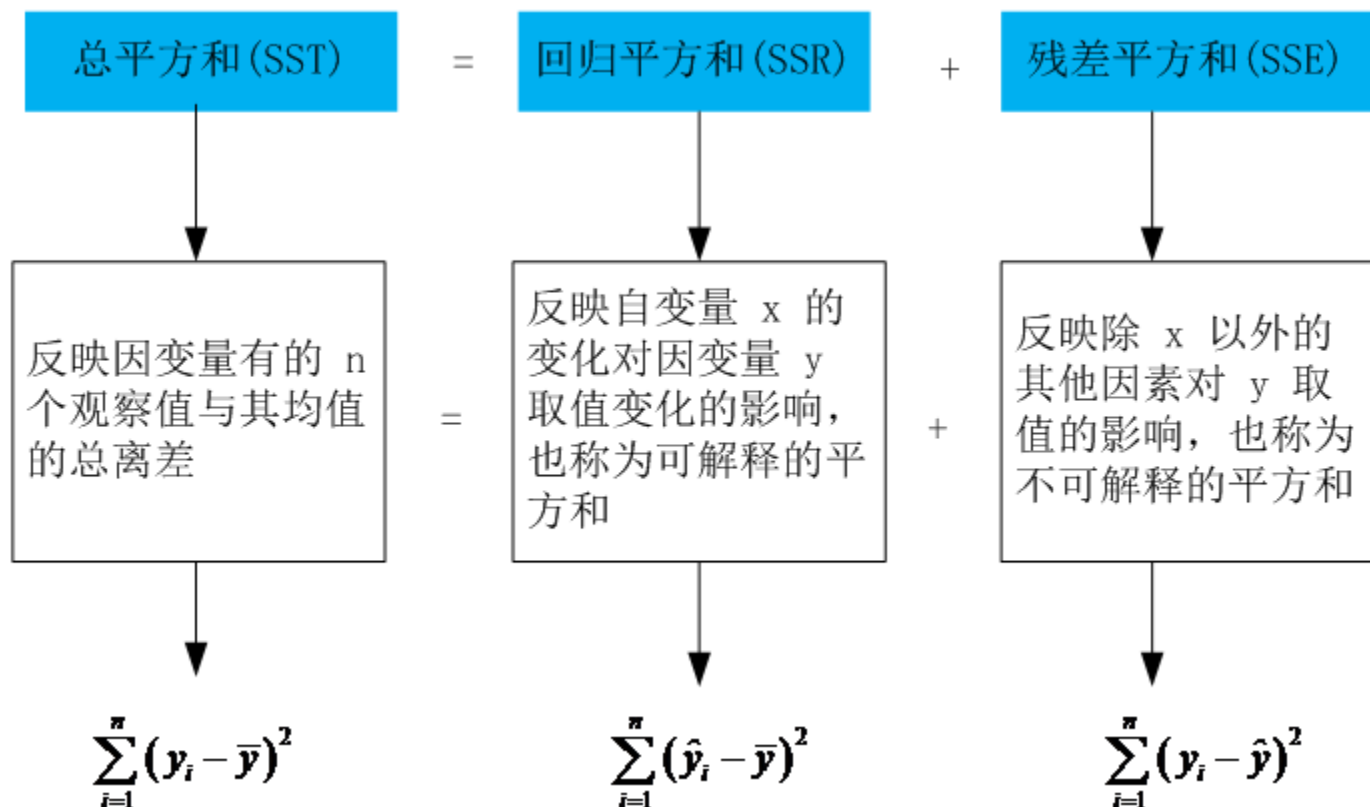
## • 式中

- $\hat{\beta}_1 = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$
- $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$ （只与平均值有关）
- 最小二乘法就是使因变量的观察值与估计值之间的离差平方和达到最小来求得 $\hat{\beta}_0$ 和 $\hat{\beta}_1$ 的方法，即
  - $\min Q(\hat{\beta}_0, \hat{\beta}_1) = \min \sum_{i=1}^n (y_i - \hat{y})^2$
  - 高斯证明了最小二乘方法的一个最优性质：在所有无偏的线性估计类中，最小二乘方法是其中方差最小的！



- 1801年，意大利天文学家朱赛普·皮亚齐发现了第一颗小行星谷神星
- 经过40天的跟踪观测后，由于谷神星运行至太阳背后，使得皮亚齐失去了谷神星的位置
- 随后全世界的科学家利用皮亚齐的观测数据开始寻找谷神星，但是根据大多数人计算的结果来寻找谷神星都没有结果
- 时年24岁的高斯也计算了谷神星的轨道（最小二乘法）
- 奥地利天文学家海因里希·奥尔伯斯根据高斯计算出来的轨道重新发现了谷神星

- 线性回归分析结果的评价涉及3个重要概念



- 证明: [https://blog.csdn.net/weixin\\_43145361/article/details/103546382](https://blog.csdn.net/weixin_43145361/article/details/103546382)

- 判定系数 $R^2$

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad \begin{array}{l} \text{回归平方和} \\ \text{总平方和} \end{array}$$

- $R^2$ 的取值范围在 $[0,1]$ 之间
  - $R^2$ 越接近1 (说明残差越小), 说明回归方程拟合的越好
  - $R^2$ 越接近0 (说明残差越大), 说明回归方程拟合的越差
- 线性回归方程的显著性检验的基本步骤
  - 第1步: 提出假设:  $H_0: \beta_1 = 0$ , 即线性关系不显著
  - 第2步: 计算检验统计量 $F$ 
    - $$F = \frac{SSR/1}{SST/(n-2)} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2 / 1}{\sum_{i=1}^n (y_i - \bar{y})^2 / (n-2)} \sim F(1, n-2)$$
  - 第3步: 确定显著性水平
    - 并根据分子自由度1和分母自由度 $n-2$ 找出临界值 $F_\alpha$
  - 第4步: 做出决策
    - 若 $F > F_\alpha$ , 拒绝 $H_0$ , 若 $F < F_\alpha$ , 接受 $H_0$



- 一种将复杂问题简单化之后，再进行分析 and 处理
  - 即将大量数据若干个类别之后，分析类别的统计特征，通过类别的特征概括具体数据。
- 注意：从**统计分析**角度看，**分类**和**预测**是两个相互关联和转化的概念
  - 当目标值的类型为“分类型”时，称之为分类（就是离散类）
  - 当目标值的类型为“连续型”时，称之为预测
- 分类分析的算法
  - 决策树（Decision tree）
  - 决策表（Decision table）
  - 贝叶斯网络（Bayesian network）
  - 神经网络（Neural network）
  - 支持向量机（Support vector machine）
  - KNN算法（K-nearest neighbor）

注：不同算法对分类和预测的支持不同  
例如贝叶斯网络和决策表只能处理分类型目标值的分类分析

- 人们对动物的树形分类方法类似，分类过程是通过递归方式进行，每次分类都是基于最显著属性进行划分
- 决策树的最顶层为树的根节点
- 每个非叶结点表示一个显著性属性上的测试
- 而其后的分支代表基于这个显著性属性的划分结果
- 决策树的实现方法有很多种，如
  - ID3
  - C4.5
  - C5.0
  - 分类和回归树 C&R Tree
    - Classification and Regression Tree
  - 卡方自动交互检验法 CHAID
    - Chi-squared Automatic Interaction Detector
  - ...

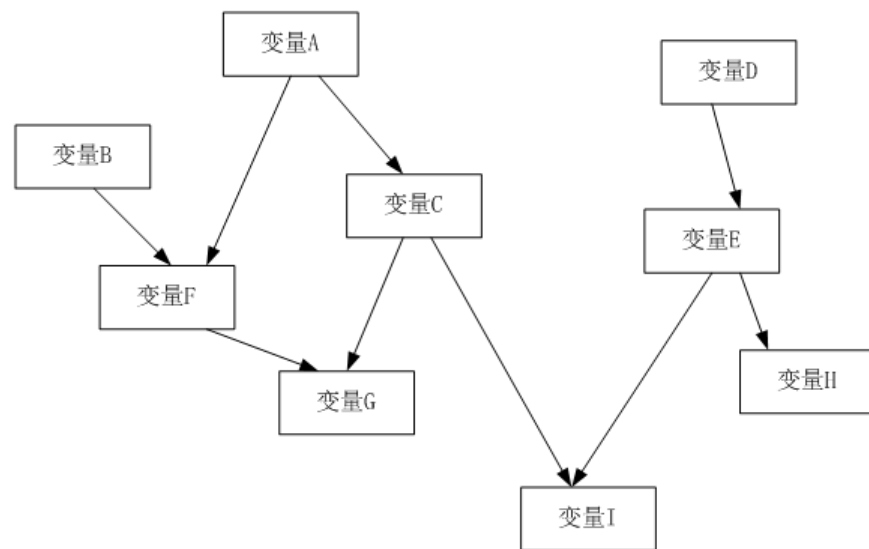
例：C5.0算法

1. 在每次决策树节点的计算上，将计算所有没有被用过的字段的信息熵
2. 取这个信息熵最小的字段
3. 将步骤2中确定的字段作为节点上保留的字段，并据此继续分支

- 基于概率推理的数学模型
  - 概率推理是指通过一些变量的信息来获取其他的概率信息的过程，主要用于解决不定性和不完整性问题
- 可以从网络结构和条件概率表两个视角理解贝叶斯网络
- 基于条件概率论提出的，所涉及的条件概率是贝叶斯公式进行计算，计算方法如下

$$P(A_i|B) = \frac{P(A_iB)}{P(B)} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^n P(A_j)P(B|A_j)}$$

$A_i$ 条件下 $B$ 发生的概率  
 $B$ 在各种 $A_i$ 条件下发生的概率



贝叶斯网络的网络结构示意图

- 基本思路为寻找 $k$ 个最近的邻居
- 当一个样本在特征空间中的 $k$ 个最相邻的样本中的大多数属于某一个类别时，该样本也属于这个类别，并具有这个类别上样本的特性
- 对于类域的交叉或重叠较多的待分样本集来说，KNN方法较其他方法更为适合

- 与分类分析不同的是，聚类分析所要求划分的类别是未知
- 聚类分析中的“聚类要求”有两条：
  - 每个分组内部的数据具有比较大的相似性
  - 组间的数据具有较大不同
- 聚类分析方法
  - 分层聚类 Hierarchical
  - K-means 聚类
  - Kohonen Network 聚类等
    - 一种神经网络，属于无监督学习网络

- 通过尝试对给定数据集进行分层以达到聚类
- 根据分层分解采用的分解策略，分层聚类法又可以分为
  - 凝聚的分层聚类
  - 分裂的分层聚类
- 分层聚类的关键在于选择合并点或者分裂点
  - 如果合并或者分裂的决定不合理，就可能得出低质量的聚类结果
- 分层聚类算法的另一个缺点是，计算量大
  - 在决定合并或者分裂之前需要检查和估算大量的数据和类
- 因此，基本的分层分类方法很少直接应用于数据分析之中

- 一种典型的基于距离的聚类算法
  - 两个对象的距离越近，其相似度就越大
- 目的是寻找固定数目的簇
  - 这些簇是由距离靠近的对象组成
- 算法步骤
  - 第1步：从n个数据对象任意选择 k 个对象作为初始聚类中心
  - 第2步：计算在聚类中心之外的每个剩余对象与中心对象的距离，并根据最小距离重新对相应对象进行划分
  - 第3步：重新计算每个有变化的聚类的均值，确定新的聚类中心
  - 第4步：迭代步骤2-3直至每个聚类不再发生变化或小于指定阈值
- 相对于分层聚类，K-means 聚类的效率更高，在数据分析领域，更为广泛使用

1、随机选取k个聚类质心点 ( cluster centroids ) 为  $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$

2、重复下面过程直到收敛 {

对于每一个样例i，计算其应该属于的类

$$c^{(i)} := \arg \min_j \|x^{(i)} - \mu_j\|^2.$$

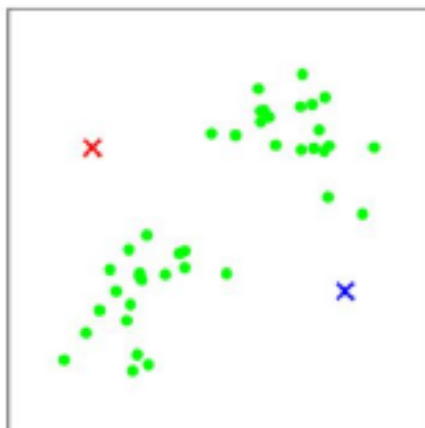
对于每一个类j，重新计算该类的质心

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

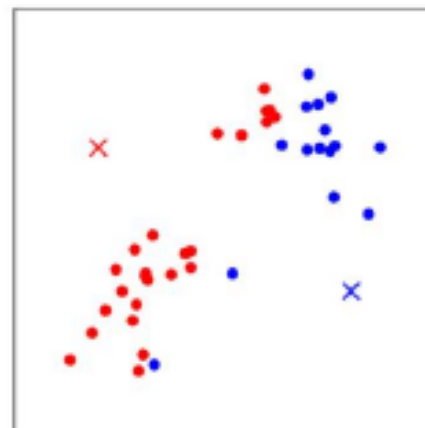
# 聚类分析——K-means聚类例子



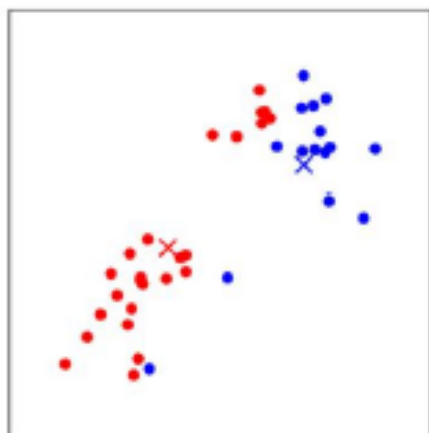
(a)



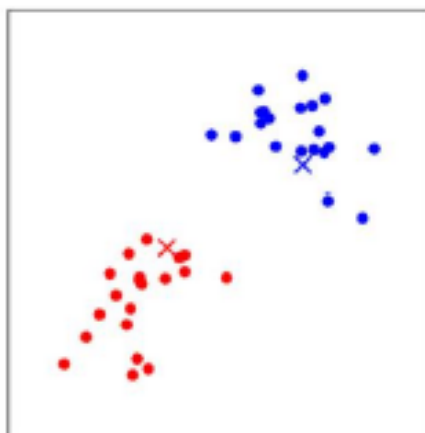
(b)



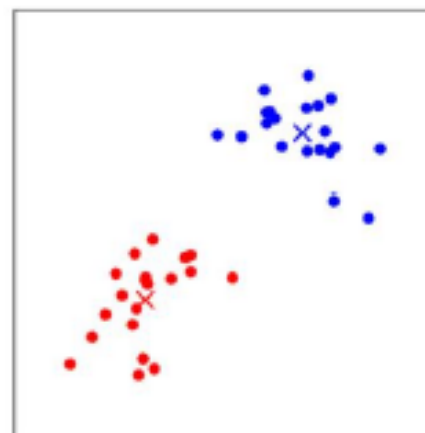
(c)



(d)



(e)



(f)



- 选择批次距离尽可能远的k个点
  - 首先随机选择一个点作为第一个初始类簇中心点
  - 然后选择距离该点最远的那个点作为第二个初始类簇中心点
  - 再选择距离前两个点的最近距离最大的点作为第三个初始类簇的中心点
  - 以此类推，直至选出k个初始类簇中心点
- 选用层次聚类或者Canopy算法进行初始聚类，然后利用这些类簇的中心点作为K-Means算法初始类簇中心点
  - Canopy聚类最大的特点是不需要事先指定k值（即类簇的个数）
    - 因此具有很大的实际应用价值
  - Canopy聚类虽然精度较低，但其在速度上有很大优势
  - 可以使用Canopy聚类先对数据进行“粗”聚类，得到k值后再使用K-means进行进一步“细”聚类
- 聚类扩展：密度聚类、层次聚类

- “There is no single clustering algorithm that has been shown to dominate other algorithms across all application domains” A.K. Jain, 2009, PRL, 2009
- 没有一个聚类算法能适应所有的应用

- 时间序列是按时间顺序的一组数字序列
- 时间序列分析就是利用这组数列，应用数理统计方法加以处理，以预测未来事物的发展
- 时间序列分析的基本假设有以下两条
  - 承认事物发展的延续性
    - 用过去数据，就能推测事物的未来发展趋势
  - 考虑事物发展的随机性
    - 任何事物发展可能会受到偶然因素影响
- 时间序列预测一般反映3种实际变化规律
  - 趋势变化
  - 周期性变化
  - 随机性变化

- 通常，一个时间序列由4种要素组成
  - 趋势：时间序列长时间内呈现出来的持续向上或持续向下的变动
  - 季节变动：时间序列在一年内重复出现的周期性波动
    - 例如气候变化、周期性变化和随机性变化
    - 如销售数据的具有季节性变化规律。每年9~10月份出口增长
  - 循环波动：时间序列呈现出来的非固定长度的周期性变动
  - 不规则波动：时间序列中去除以上趋势、季节变动和周期波动之后的随机波动（就是把有规律可循的全部去掉后剩下的）
- 时间序列建模包括以下3个主要阶段
  - 第1阶段：用观测、调查、统计、抽样等方法取得被观测系统时间序列动态数据
  - 第2阶段：根据动态数据作相关图，进行相关分析，求自相关函数
    - 相关图能显示出变化的趋势和周期，并发现动态数据中的跳点与拐点
  - 第3阶段：辨识合适的随机模型，进行曲线拟合
    - 即用通用随机模型去拟合时间序列的观测数据

- 购物篮分析：引发性例子
- Q：哪组商品顾客可能会在一次购物时同时购买？
- 关联分析 Solutions
  - 经常同时购买的商品可以摆近一点，以便进一步刺激这些商品一起销售
  - 规划哪些附属商品可以降价销售，以便刺激主体商品的捆绑销售

- 案例——沃尔玛“尿布与啤酒”
- “关联”是指形如 $X \rightarrow Y$ 的蕴涵式
  - $X$ : 关联规则的**先导** (Antecedent 或 Left-Hand-Side, LHS)
  - $Y$ : 关联规则的**后继** (Consequent 或 Right-Hand-Side, RHS)
- 两个指标
  - 支持度: 数据集中包含 $X \cup Y$ 的样本**占全部样本的百分比**
    - $\text{Support}(X \rightarrow Y) = P(X \cup Y)$
    - 集合 $X$ 与集合 $Y$ 中的项在一条记录中同时出现的次数/总数据记录数
  - 信任度: 包含 $X \cup Y$ 的样本数与**包含 $X$ 的样本数的比值**
    - $\text{Confidence}(X \rightarrow Y) = P(X|Y)$
    - 也就是事件 $X$ 发生基础上 $Y$ 发生的概率
- 例: 比如说在规则 $\text{Computer} \rightarrow \text{antivirus\_software}$ 
  - 其中  $\text{support}=2\%$ ,  $\text{confidence}=60\%$  中, 就表示的意思是
    - 所有的商品交易中有2%的顾客同时买了电脑和杀毒软件
    - 购买电脑的顾客中有60%也购买了杀毒软件

- k项集
  - 如果事件A中包含k个元素，那么称这个事件A为k项集
  - 事件A满足最小支持度阈值的事件称为频繁k项集
- 同时满足最小支持度阈值和最小置信度阈值的规则称为强规则
- 两个阶段
  - 1) 必须先从资料集合中找出所有的高频项目组 (Frequent Item Sets)
    - 也就是，根据定义，这些项集出现的频繁性至少和预定义的最小支持计数一样
  - 2) 再由这些高频项目组中产生关联规则 (Association Rules)
    - 也就是，根据定义，这些规则必须满足最小支持度和最小置信度
- 关联规则分析的算法
  - Apriori 算法
  - FP-growth (Frequent Pattern-growth) 算法

- 1994年提出，一种最有影响的挖掘关联规则频繁项集的算法
  - 使用候选项集找频繁项集（频集）
- 使用一种称作逐层搜索的迭代方法
  - “K-1项集”用于搜索“K项集”
  - 首先，找出频繁“1项集”的集合，该集合记作 $L_1$
  - $L_1$ 用于找频繁“2项集”的集合 $L_2$ ，而 $L_2$ 用于找 $L_3$
  - 如此下去，直到不能找到“K项集”
  - 找每个 $L_k$ 都需要一次数据库扫描
  - 简单的讲，发现频集的过程为
    - 1) 扫描
    - 2) 计数
    - 3) 比较
    - 4) 产生频繁项集
    - 5) 连接、剪枝，产生候选项集
    - 重复1~5，直到不能发现更大的频集



- 定律1：如果一个集合是频繁项集，则它的**所有子集都是频繁项集**
  - 举例：假设一个集合 $\{A, B\}$ 是频繁项集，即A、B同时出现在一条记录的次数大于等于最小支持度 $\text{min\_support}$ ，则它的子集 $\{A\}, \{B\}$ 出现次数必定大于等于 $\text{min\_support}$ ，即它的子集都是频繁项集
- 定律2：如果一个集合不是频繁项集，则它的**所有超集都不是频繁项集**
  - 举例：假设集合 $\{A\}$ 不是频繁项集，即A出现的次数小于 $\text{min\_support}$ ，则它的任何超集如 $\{A, B\}$ 出现的次数必定小于 $\text{min\_support}$ ，因此其超集必定也不是频繁项集

# Apriori算法例子1



最小支持度为3

消除不满足条件的候选集

扫描数据库, 统计一级  
候选项集的出现次数

Itemset	Count
{牛奶}	4
{面包}	4
{尿布}	4
{啤酒}	3
{鸡蛋}	1
{可乐}	2

Itemset	Count
{牛奶}	4
{面包}	4
{尿布}	4
{啤酒}	3

在一级频繁项集上  
生成二级候选集

扫描数据库, 统计二级  
候选项集的出现次数

Itemset	Count
{牛奶, 面包}	3
{牛奶, 尿布}	3
{牛奶, 啤酒}	2
{面包, 尿布}	3
{面包, 啤酒}	2
{尿布, 啤酒}	3

消除不满足条件的候  
选集

Itemset	Count
{牛奶, 面包}	3
{牛奶, 尿布}	3
{面包, 尿布}	3
{尿布, 啤酒}	3

在二级频繁项集上  
生成三级候选集

Itemset
{牛奶, 面包, 尿布}

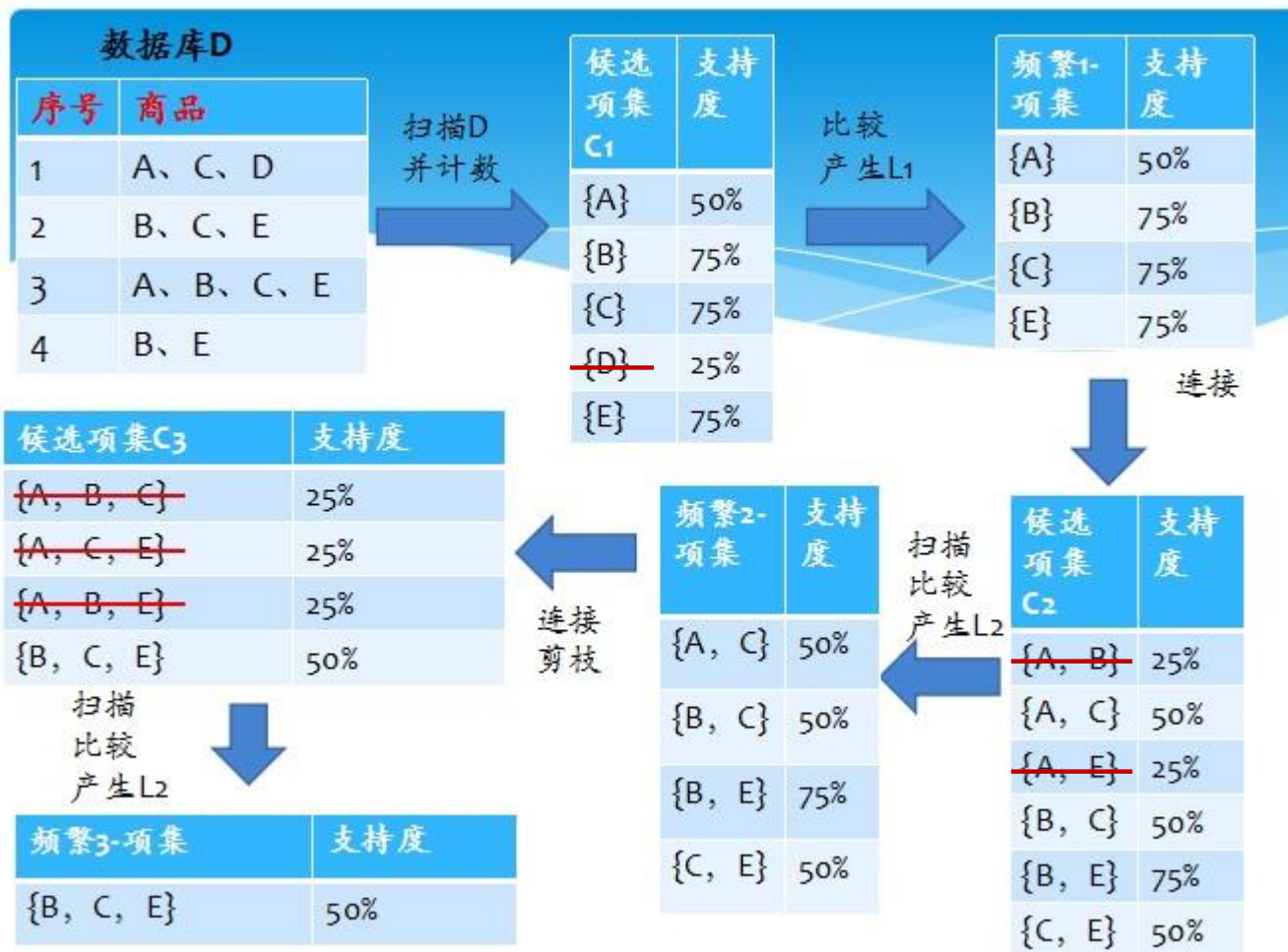
扫描数据库, 统计三级候  
选项集的出现次数

Itemset	Count
{牛奶, 面包, 尿布}	3

没有{牛奶,  
面包, 啤酒}  
为什么?

注意看由二级频繁项集生成三级候选项集时, 没有{牛奶, 面包, 啤酒}, 那是因为{面包, 啤酒}不是二级频繁项集, 这里利用了Apriori定理

最小支持度设为50%



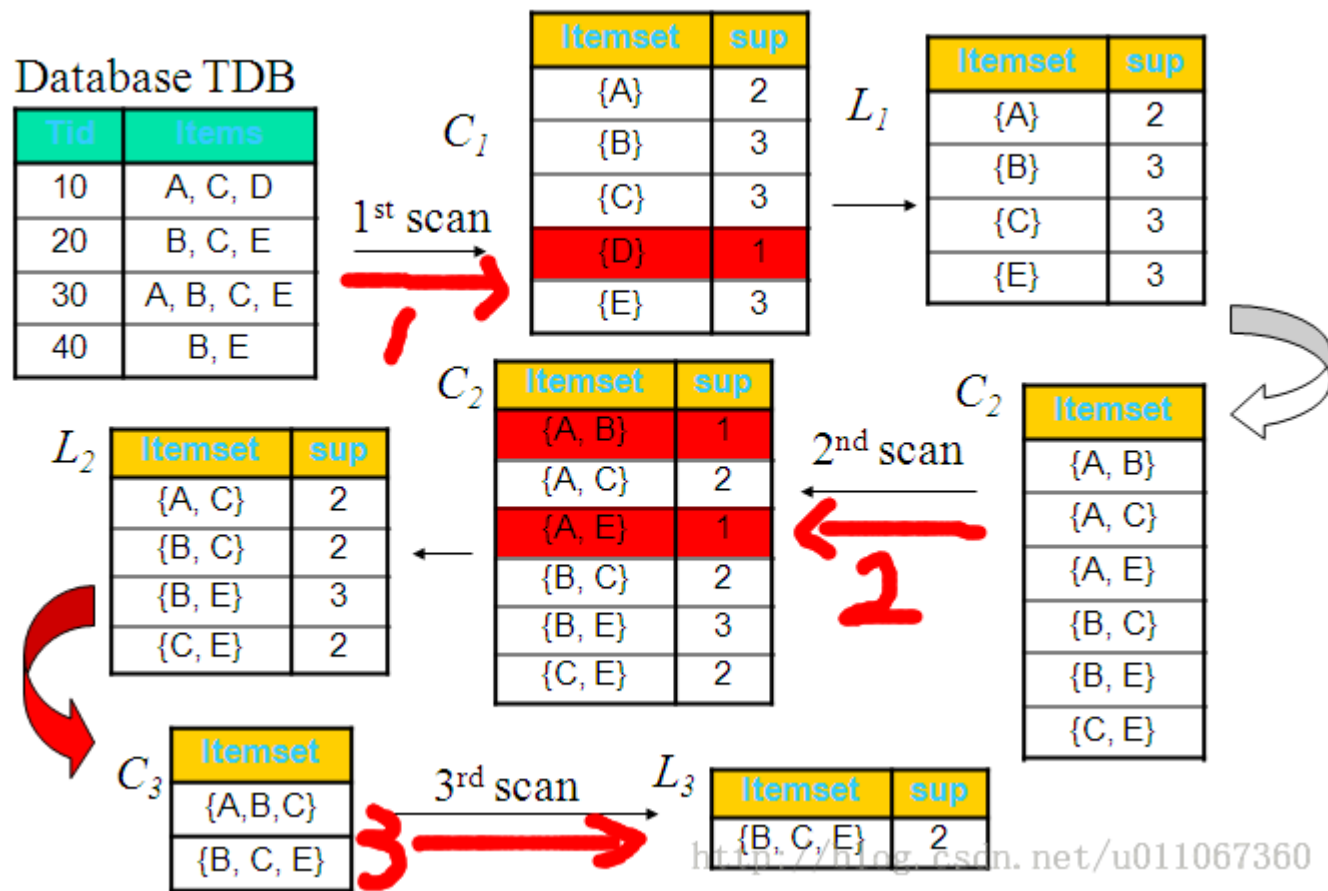
实际上,  
{A,B,C}  
{A,C,E}和  
{A,B,E}可以  
直接排除掉  
为什么?

- 参考:
- <https://github.com/taizilongxu/datamining>

# Apriori算法例子2 (另一种形式)



- 要求：
  - 支持度为2



- $C_3$ 少了部分候选项集, 如{A,C,E}, {A,B,E}, 为什么?
- {A,B,C}可否直接去掉的 (即不用计算支持度), 为什么?

- Apriori 算法采用了逐层搜索的迭代的方法
  - 算法简单明了，没有复杂的理论推导，也易于实现
- 但是，Apriori 算法存在一些难以克服的缺陷
- 对数据库的扫描次数过多
  - 每次计算项集的支持度时，都对数据库中的全部记录进行了一遍扫描比较
  - 如果是一个大型的话，这种扫描比较会大大增加计算机系统的 I/O 开销
  - 而这种代价是随着数据库的记录的增加呈现出几何级数的增加
- 会产生大量的中间项集
- 采用唯一支持度
- 算法的适应面窄

算法不仅仅是要结果好，还要其实现过程开销小！

## • 划分方法

大数据  
分解为  
小规模

- 挖掘频繁项集只需要两次数据扫描，数据库中的任何频繁项集必须作为局部频繁项集至少出现在一个部分中
- 第一次扫描：将数据划分为多个部分并找到局部频繁项集
- 第二次扫描：评估每个候选项集的实际支持度，以确定全局频繁项集
- （这样就可以采用 MapReduce 方法）
- 采用（在给定数据的一个子集挖掘）方法

用精度  
换速度

- 基本思想：选择原始数据的一个样本，在这个样本上用 Apriori 算法挖掘频繁模式
- 通过牺牲精确度来减少算法开销，为了提高效率，样本大小应该以可以放在内存中为宜，可以适当降低最小支持度来减少遗漏的频繁模式
- 可以通过一次全局扫描来验证从样本中发现的模式
- 可以通过第二次全局扫描来找到遗漏的模式

- FP-growth 方法是在不生成候选项的情况下，完成 Apriori 算法的功能
- 实用性比较强的算法，利用了巧妙的树形数据结构（类似于 Prefix Tree），大大降低了 Apriori 挖掘算法的代价，不需要不断得生成候选项目队列和不断得扫描整个数据库进行比对
- 采用分而治之的策略，在经过第一遍扫描之后，把数据库中的频集压缩进一棵频繁模式树（FP-tree）
- 同时依然保留其中的关联信息
- 随后再将 FP-tree 分化成一些条件库，每个库和一个长度为 1 的频集相关
- 然后再对这些条件库分别进行挖掘
- 当原始数据量很大的时候，也可以结合划分（MapReduce）的方法，使得一个 FP-tree 可以放入主存中



- 第一遍扫描数据
  - 找出频繁 1 项集 L, 按降序排序
- 第二遍扫描数据:
  - 对每个 transaction, 过滤不频繁集合, 剩下的频繁项集按 L 顺序排序
  - 把每个 transaction 的频繁 1 项集插入到 FP-tree 中, 相同前缀的路径可以共用
  - 同时增加一个 header table, 把 FP-tree 中相同项连接起来, 也是降序排序
- 参考论文
  - Mining Frequent Patterns without Candidate Generation (SIGMOD2000)



- 第一步：第一次扫描整个数据库产生频繁项集  $L$

按支持度降序排列，  
最小支持度为 3

<i>TID</i>	<i>Items bought</i>
100	{ <i>f</i> , <i>a</i> , <i>c</i> , <i>d</i> , <i>g</i> , <i>i</i> , <i>m</i> , <i>p</i> }
200	{ <i>a</i> , <i>b</i> , <i>c</i> , <i>f</i> , <i>l</i> , <i>m</i> , <i>o</i> }
300	{ <i>b</i> , <i>f</i> , <i>h</i> , <i>j</i> , <i>o</i> }
400	{ <i>b</i> , <i>c</i> , <i>k</i> , <i>s</i> , <i>p</i> }
500	{ <i>a</i> , <i>f</i> , <i>c</i> , <i>e</i> , <i>l</i> , <i>p</i> , <i>m</i> , <i>n</i> }



$L$

<i>Item</i>	<i>frequency</i>
<i>f</i>	4
<i>c</i>	4
<i>a</i>	3
<i>b</i>	3
<i>m</i>	3
<i>p</i>	3

没有 *g*, *e* .....

By-Product of First  
Scan of Database

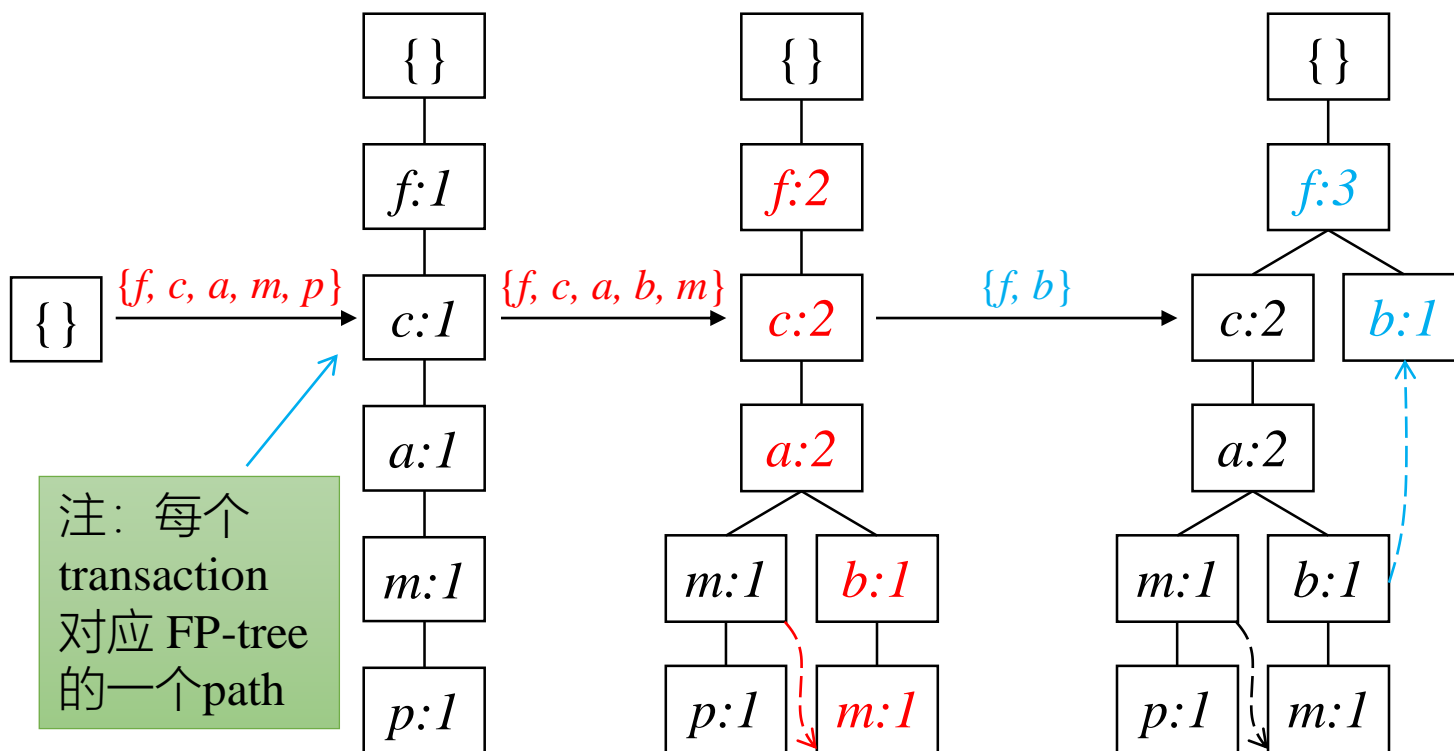
- 第二步：第二次扫描数据库，得到每个 transaction 中的频繁数据项集

按支持度降序排列

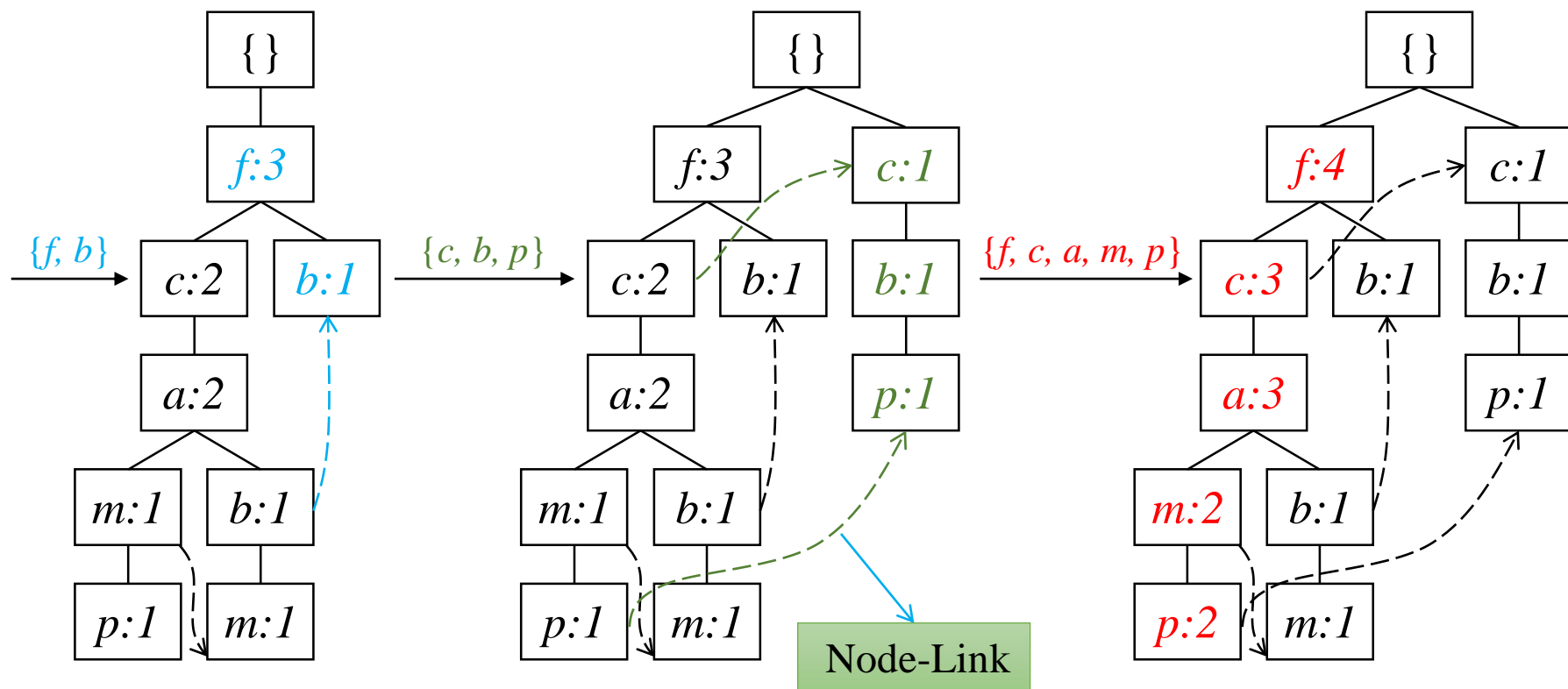
<i>TID</i>	<i>Items bought</i>	<i>(ordered) frequent items</i>
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

<i>Item</i>	<i>frequency</i>
<i>f</i>	4
<i>c</i>	4
<i>a</i>	3
<i>b</i>	3
<i>m</i>	3
<i>p</i>	3

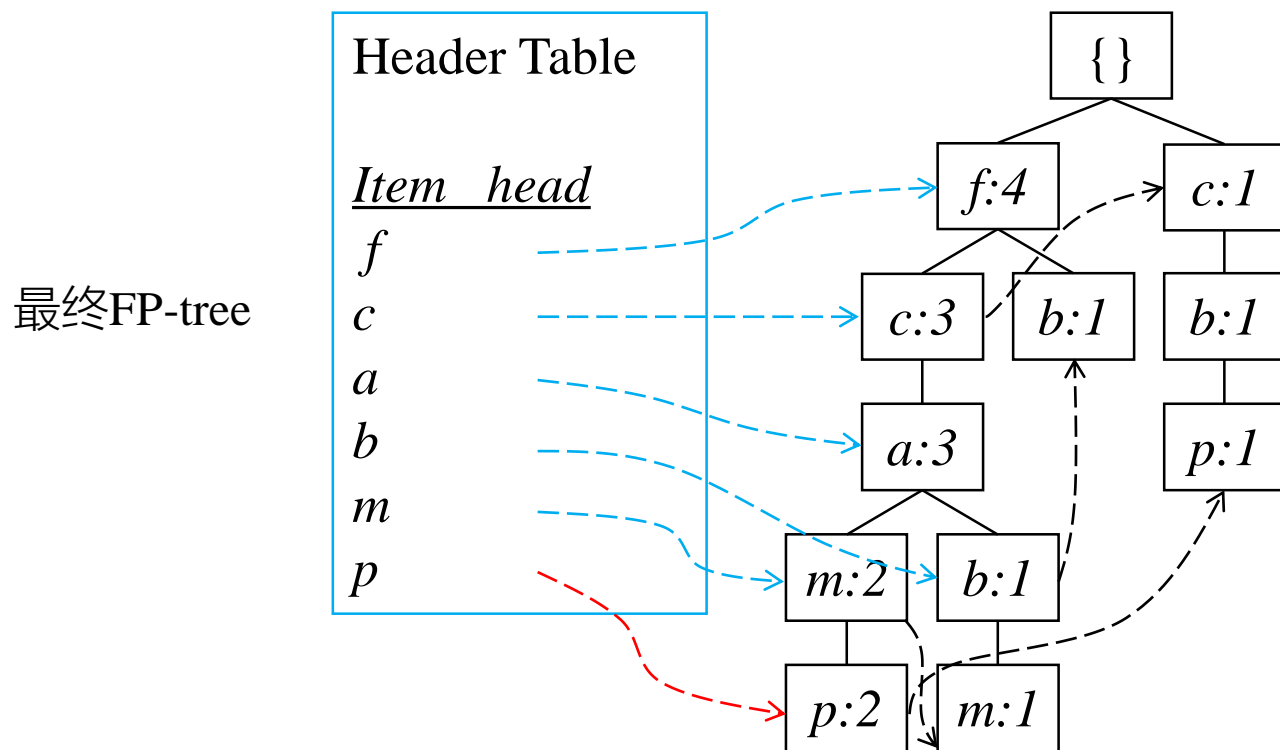
- 第三步：构造FP-tree



- 第三步：构造FP-tree



- FP Tree 频繁特征树 (注意：要包含Header)

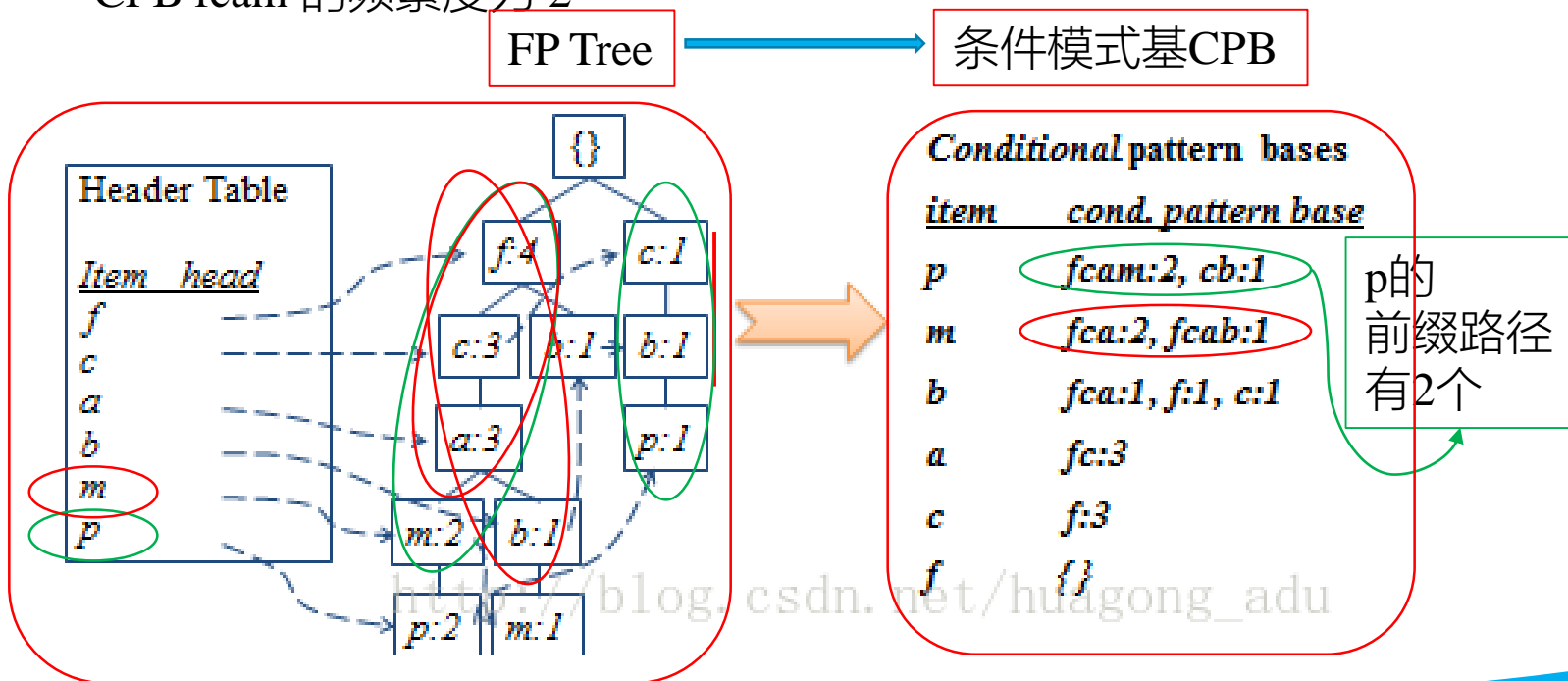


- 算法优势
  - 只需扫描 2 次数据库
  - 完整性 Completeness
    - FP-tree 包含挖掘频繁模式相关的所有信息（最小支持阈值）
  - 紧凑性 Compactness
    - FP-tree 的大小只和出现的频繁项相关
    - FP-tree 的高度只和一个 transaction 内的项目数量有关
- FP-tree 挖掘频繁项集方法：分而治之 Divide and conquer
  - 对每一个频繁项，构造每个项的条件模式基 Conditional Pattern Base, CPB)
  - 重复该过程，直到FP-Tree为空

# 步骤1：构造每个项的条件模式基



- 从 header table 的最下面的项开始，构造每个项的 CPB
  - 顺着 header table 中项的链表，找出所有包含该项的**前缀路径**，这些前缀路径就是该项的 CPB
  - 所有这些 CPB 的频繁度（计数）为该路径上项的频繁度（计数）
    - 如包含 p 的其中一条路径是 fcamp，该路径中 p 的频繁度为 2，则该 CPB fcamp 的频繁度为 2

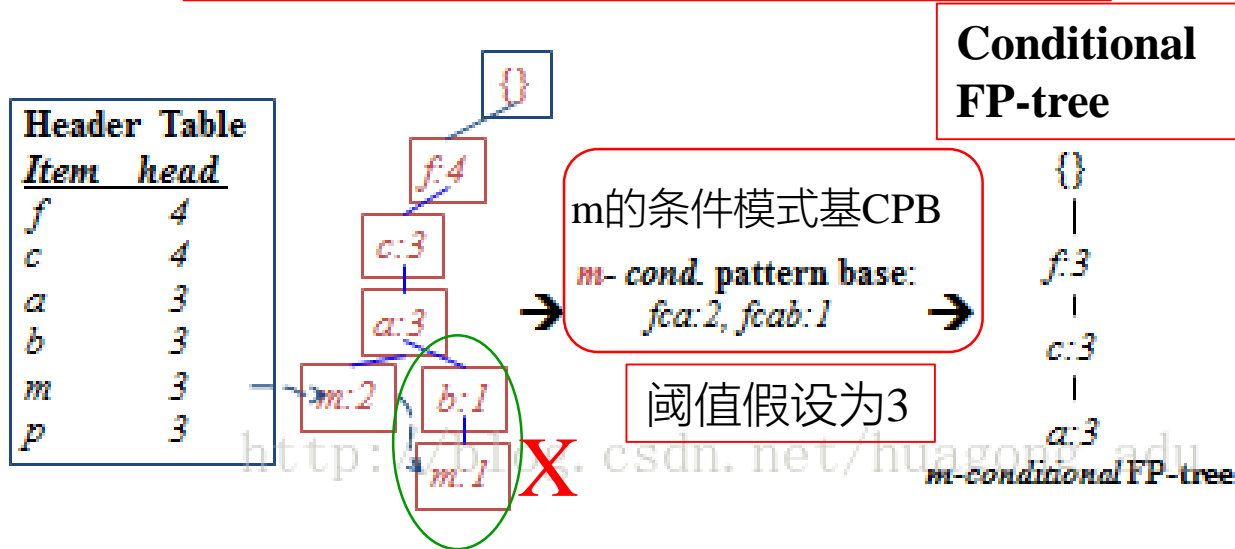


## 步骤2：构造条件FP-tree (Conditional FP-tree)



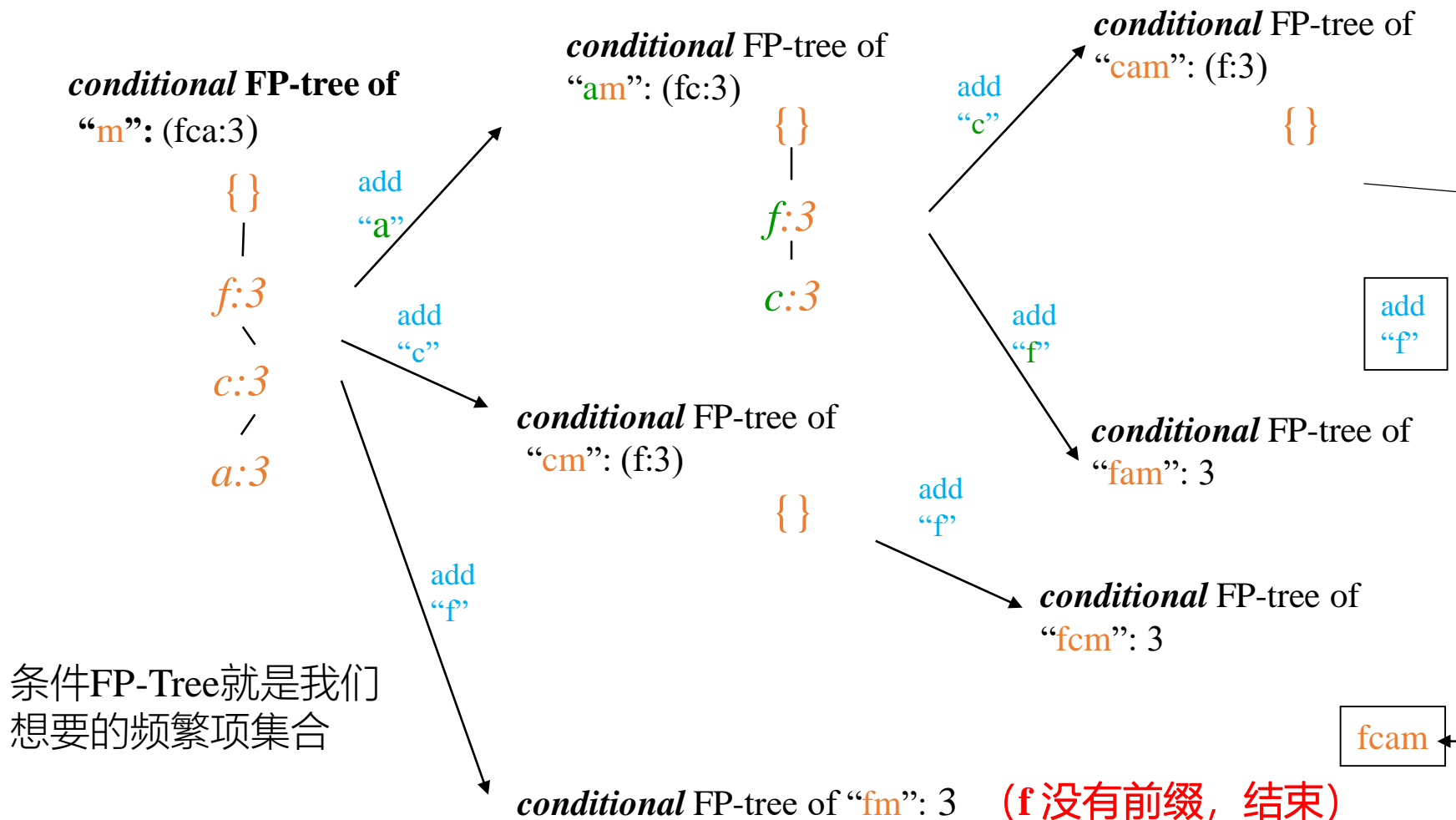
- 根据每个 CPB 上的项的**频繁度** (计数), 过滤低于阈值的项, 构建 FP-tree
- 如 m 的 CPB {<fca:2>, <fcab:1>}, f:3, c:3, a:3, b:1, 阈值假设为 3, 过滤掉 b

由条件模式基 CPB 构造 条件 FP Tree





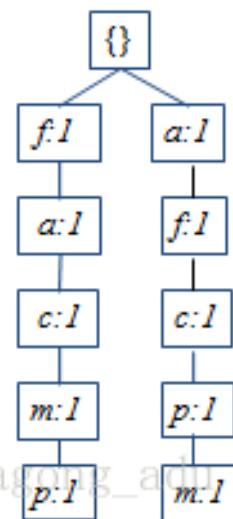
# 步骤3: 递归挖掘条件FP-tree



- 共用前缀
  - 不排序会造成不能共用前缀
- 更多的共用前缀
  - 频繁的项会在树的上层，可以被更多的共享
  - 升序排序会造成那些频繁出现的项出现在树的分支中，不能更多的共用前缀（i.e.不用降序的原因）

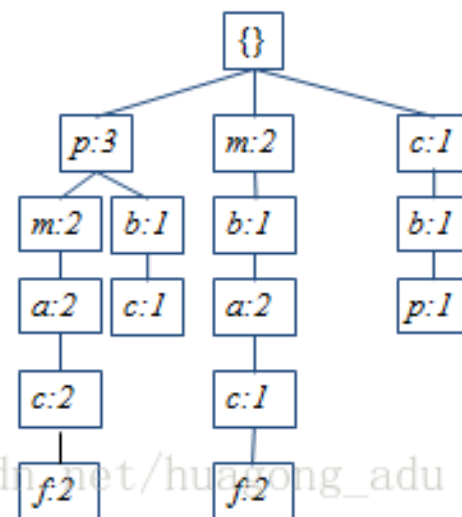
• Example 1:

<u>TID</u>	<u>(unordered) frequent items</u>
100	{f, a, c, m, p}
500	{a, f, c, p, m}

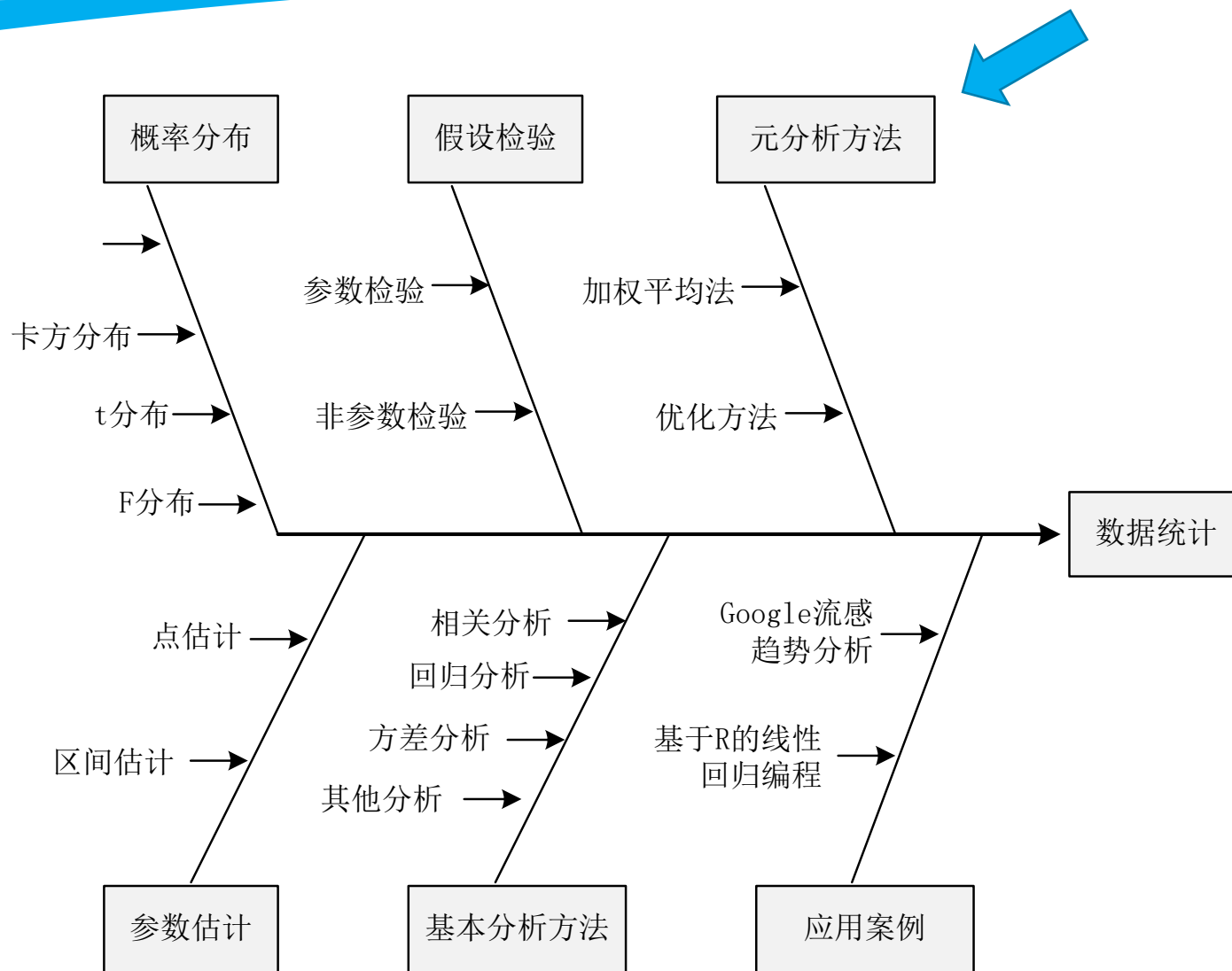


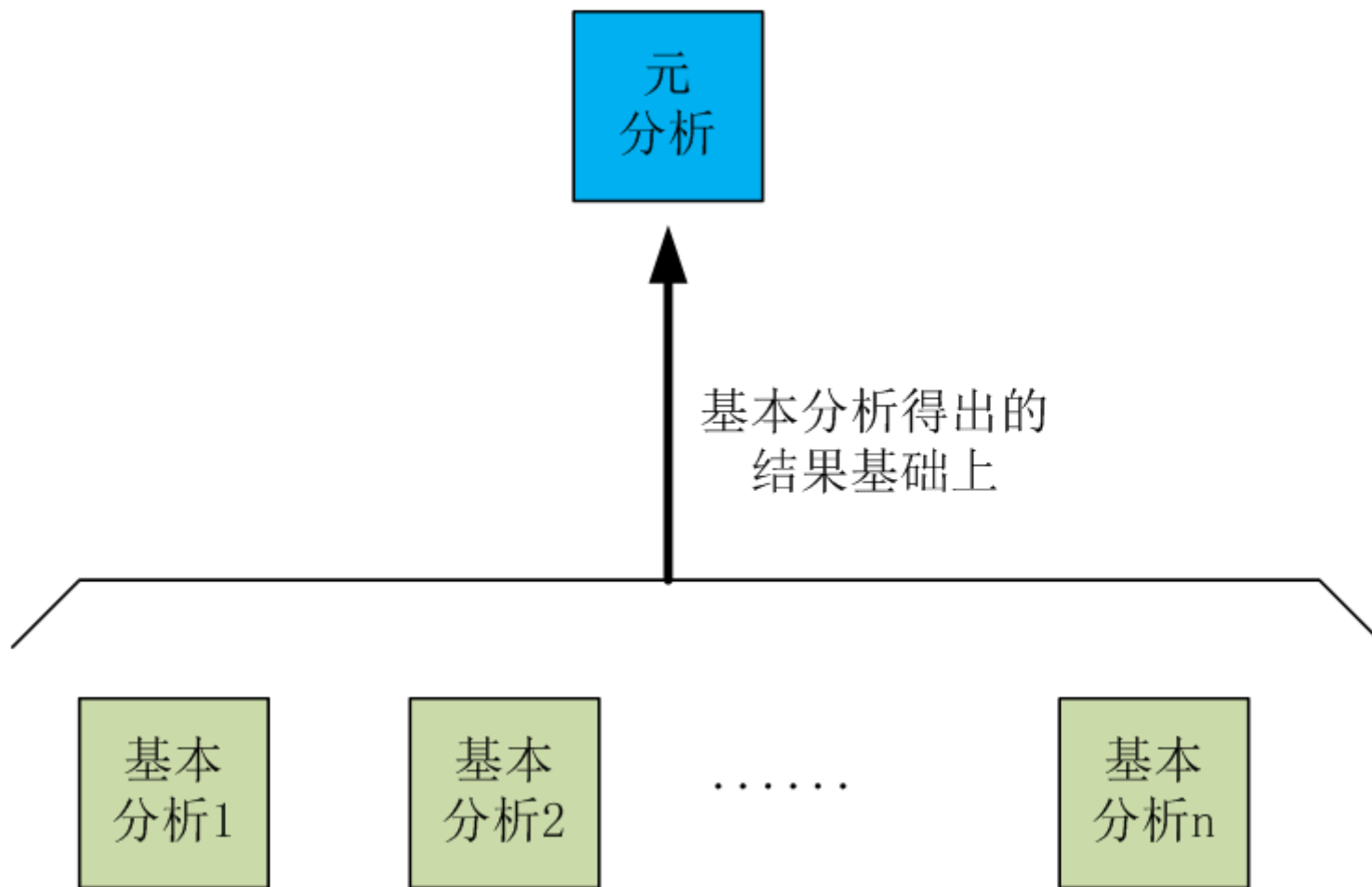
Example 2:

<u>TID</u>	<u>(ascended) frequent items</u>
100	{p, m, a, c, f}
200	{m, b, a, c, f}
300	{b, f}
400	{p, b, c}
500	{p, m, a, c, f}



This tree is larger than FP-tree, because in FP-tree, more frequent items have a higher position, which makes branches less





- 加权平均法中，**权重计算**是关键
  - 通常根据数据来源的**精确性**和**可靠性**来确定权重高低
    - 较高精度（较可靠）的数据赋予较高的权重
    - 较低精度（不可靠）的数据赋予较低的权重

- 计算方法

- 样本大小加权方法
    - 一般采用**样本大小**为依据进行加权

$$w_i = \frac{x_i}{\sum_{j=1}^k x_j}$$

- 式中， $w_i$ 代表的是第 $i$ 个变量 $x_i$ 的权重， $k$ 为变量个数

- **逆方差**（inverse-variance）加权方法

$$w_i = \frac{\sum_i \frac{y_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2}}$$

- 式中， $y_i$ 为第 $i$ 个分析数据集，其对应的方差为 $\sigma_i^2$

- 从多个备选方案（可行性解空间）中挑选（搜索）最优方案的一种方法，其主要理论基础来自于运筹学
- 主要包括
  - 线性规划
  - 整数规划
  - 多目标规划
  - 动态规划

- 主要研究线性约束条件下线性目标函数的极值问题的数学理论和方法，主要步骤包括
  - 列出约束条件及目标函数
  - 画出约束条件所表示的可行域
  - 在可行域内求解目标函数的最优解及最优值
- 例
  - 假设一个农夫有一块 $A$ 平方千米的农地，打算种植小麦或大麦，或是两者依某一比例混合种植
    - 该农夫只可以使用有限数量的肥料 $F$ 和农药 $P$
    - 单位面积的小麦和大麦都需要不同数量的肥料和农药，小麦以 $(F_1, P_1)$ 表示，大麦以 $(F_2, P_2)$ 表示
    - 设小麦和大麦的售出价格分别为 $S_1$ 和 $S_2$ ，则小麦与大麦的种植面积 $x_1$ 和 $x_2$ 问题可表示为右侧线性规划问题

$$\begin{aligned} \max P &= S_1x_1 + S_2x_2 \\ \text{s.t.} \end{aligned}$$

$$\begin{aligned} x_1 + x_2 &\leq A \\ F_1x_1 + F_2x_2 &\leq F \\ P_1x_1 + P_2x_2 &\leq P \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

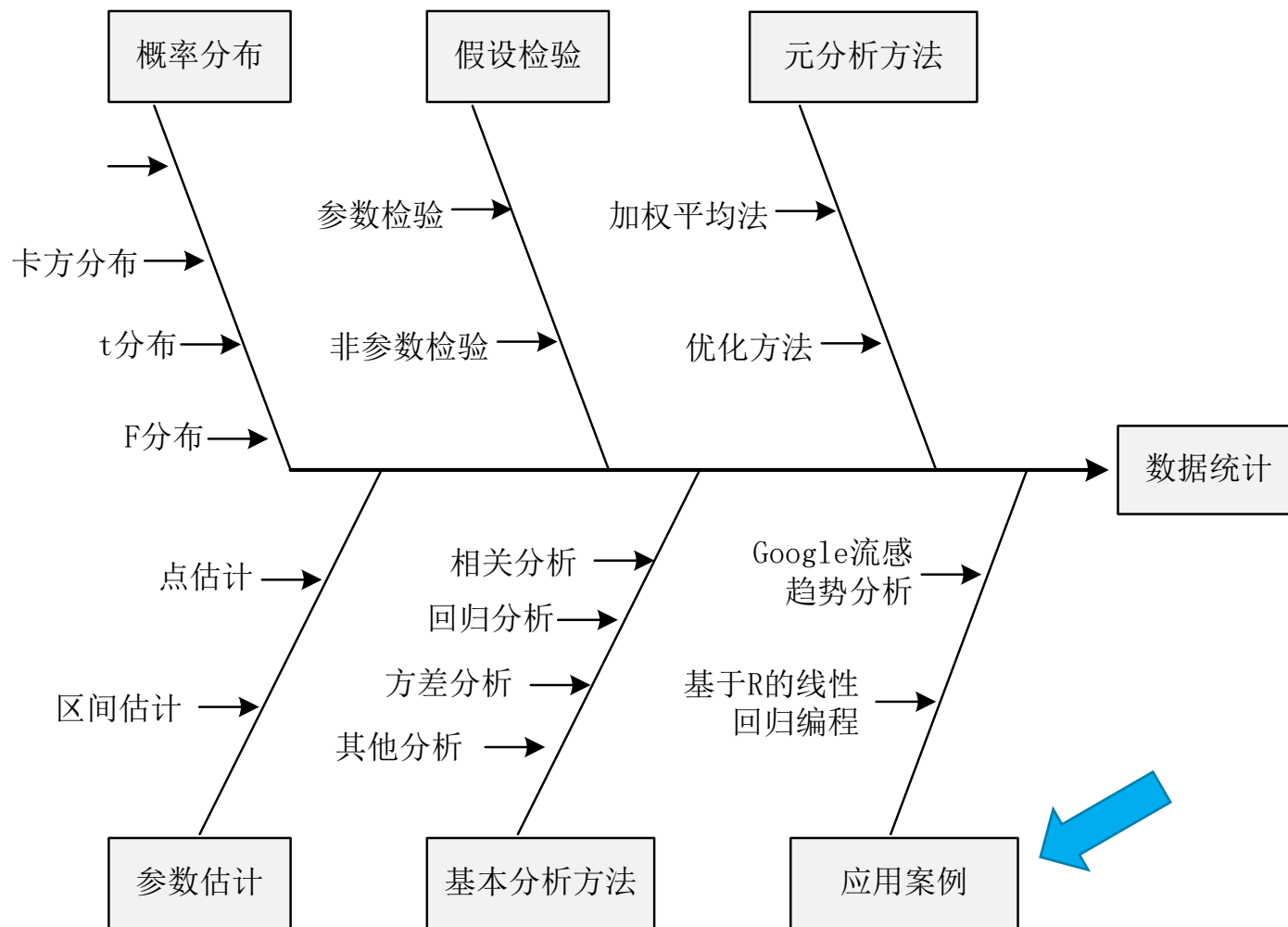
注：线性规划有最优解

- 所涉及变量（全部或部分）限制为整数的规划称为整数规划
- 在整数规划中
  - 如果所有变量都限制为整数，则称为纯整数规划
  - 如果仅一部分变量限制为整数，则称为混合整数规划
- 整数规划的一种特殊情形是 01 规划，它的变数仅限于 0 或 1
- 不同于线性规划问题，整数规划和 01 规划问题至今尚未找到一般的多项式解法



- 主要研究多于一个的目标函数在给定区域上的最优化
- 求解多目标规划的方法主要有
  - 化多为少法
  - 分层序列法
  - 层次分析法

- 求解决策过程最优化的数学方法
- 基本类型
  - 线性，区域，树形和背包动态规划
- 基本步骤
  - 确定问题的决策对象
  - 对决策过程划分阶段
  - 对各阶段确定状态变量
  - 根据状态变量确定费用函数和目标函数
  - 建立各阶段状态变量的转移过程，确定状态转移方程



- 2009 年，谷歌的工程师 Ginsberg 等人在 [Nature](#) 上发表的文章 Detecting influenza epidemics using search engine query data

nature.com > nature > letters > article


MENU ▾

**nature**  
International journal of science

Altmetric: 453 Citations: 1502 [More detail >>](#)

Letter

## Detecting influenza epidemics using search engine query data

Jeremy Ginsberg, Matthew H. Mohebbi , Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski & Larry Brilliant

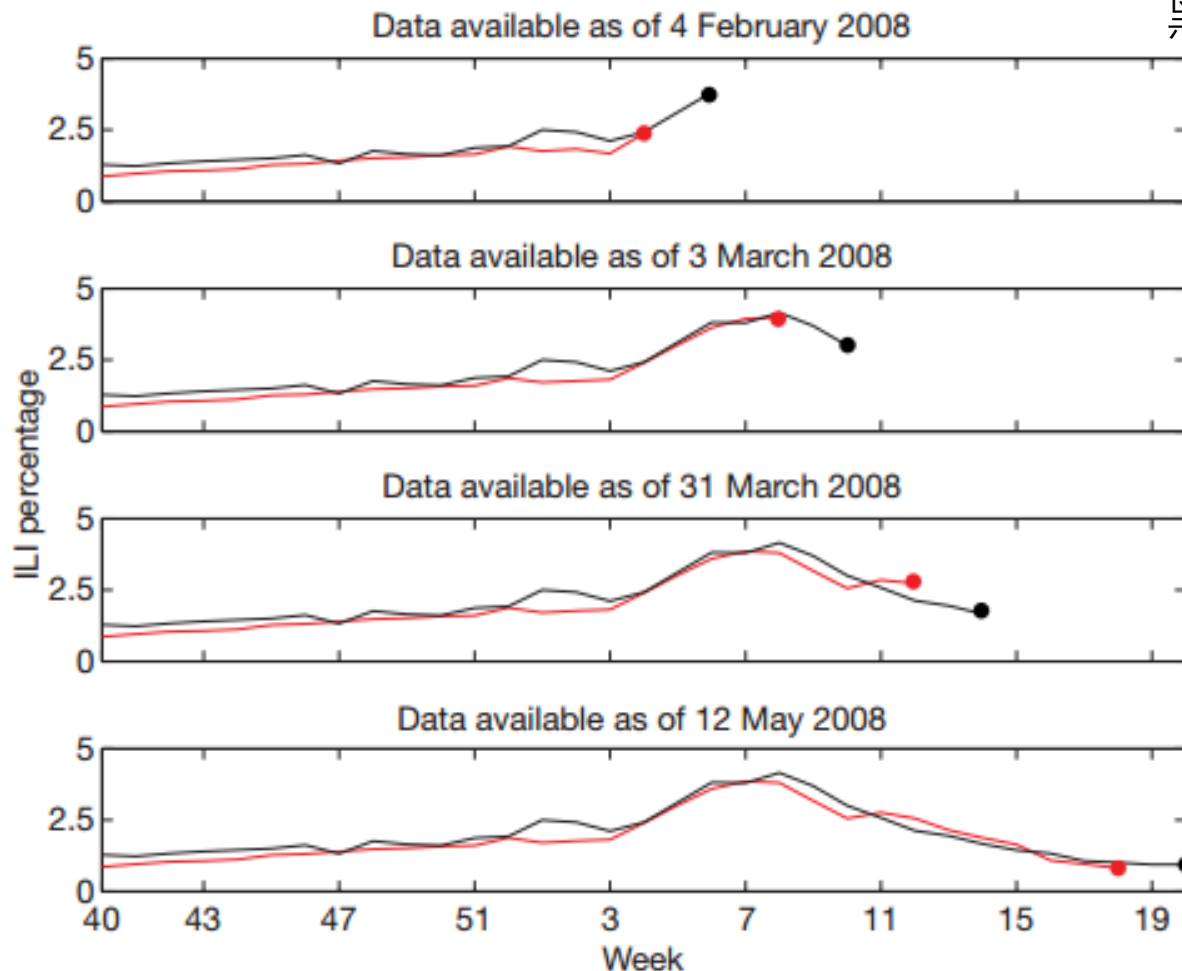
*Nature* **457**, 1012–1014 (19 February 2009)  
doi:10.1038/nature07634  
[Download Citation](#)

Received: 14 August 2008  
Revised: 19 February 2009  
Accepted: 13 November 2008  
Published: 19 November 2008  
Corrected online 19 February 2009

- 应用搜索引擎数据来预测流感疫情的方法和工具 (Google Flu Trends, GFT)
- 当时美国国家疾控中心只能做到在流感爆发一两周之后才能发布预测
- GFT 实时预测了当年的 H1N1 在全美范围的传播
- GFT 的实时性和准确性震撼了当时的学术界和政界
- GFT 的成功极大引发了人们对大数据思维的讨论，是大数据理念的一个里程碑

# 谷歌流感趋势分析 (续)

红色线是CDC提供的真实数据  
黑色线是谷歌模型预测的数据



注意：  
黑色线比红色线提前

Figure 3: ILI percentages estimated by our model (black) and provided by CDC (red) in the Mid-Atlantic region, showing data available at four points in the 2007-2008 influenza season. During week 5, we detected a sharply

- 核心模型: influenza-like illness (ILI) model

预测变量

输入变量

$$\text{logit}(P) = \beta_0 + \beta_1 \times \text{logit}(Q) + \varepsilon$$

类似于线性拟合

where  $P$  is the percentage of ILI physician visits,  $Q$  is the ILI-related query fraction,  $\beta_0$  is the intercept,

$\beta_1$  is the multiplicative coefficient, and  $\varepsilon$  is the error term.  
 $\text{logit}(P)$  is the natural log of  $P/(1-P)$ .



## Detecting influenza epidemics using search engine query data

Jeremy Ginsberg<sup>1</sup>, Matthew H. Mohebbi<sup>1</sup>, Rajan S. Patel<sup>1</sup>, Lynnette Brammer<sup>2</sup>, Mark S. Smolinski<sup>1</sup> & Larry Brilliant<sup>1</sup>

<sup>1</sup>Google Inc. <sup>2</sup>Centers for Disease Control and Prevention

- Target variable to be predicted 预测变量  $I(t)$ 
  - 待求的变量
  - For each week, for each region 对于每周每个区域
    - $I(t)$  = percentage physician visits that are ILI (as compiled by CDC)
    - 医生外出访问的百分比
- Input variable 输入变量  $Q(t)$ 
  - 从搜索记录里过滤而来 (数据预处理)
    - $Q(t)$  = sum of top n highest correlated queries 最相关的n个搜索请求 / total number of queries that week 总的搜索请求
- “模式学习”

$$\log\left(\frac{I(t)}{[1-I(t)]}\right) = \alpha \log\left(\frac{Q(t)}{[1-Q(t)]}\right) + \text{noise}$$

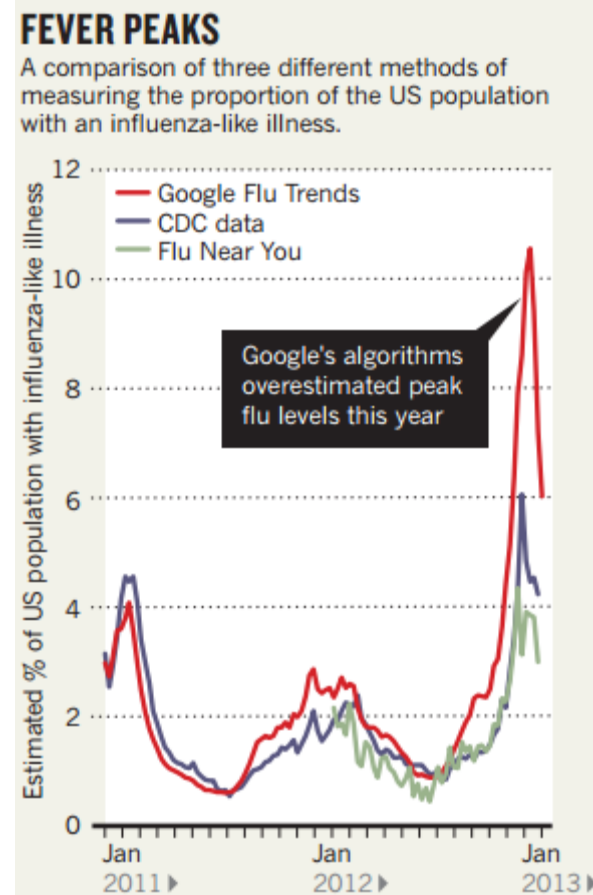
预测变量

输入变量

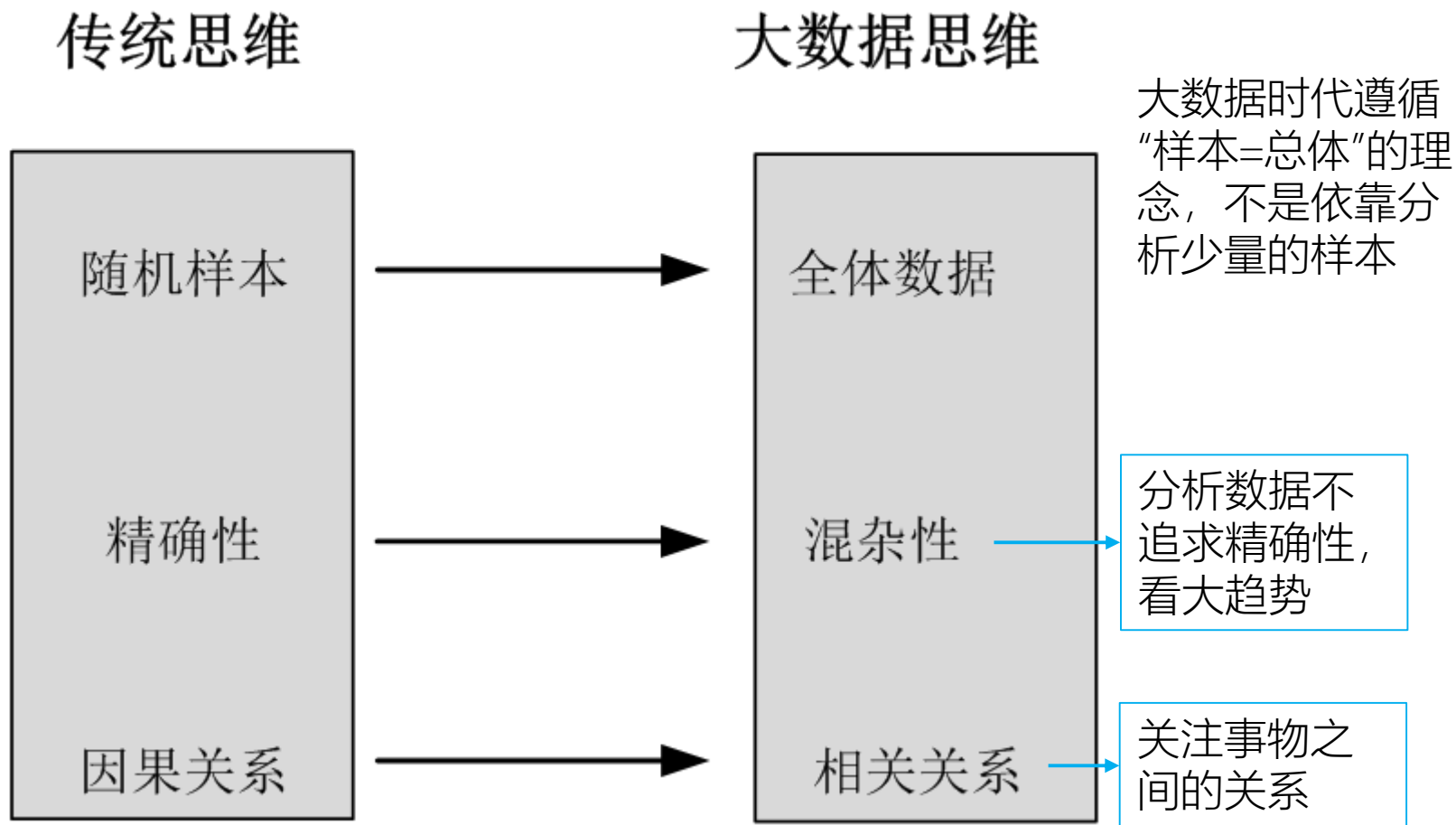
$\beta_0 + \varepsilon$



- 2011-2013 谷歌流感趋势分析不准了
  - 比实际高了 2 倍
  - 但趋势却是一致的
- 2014 年 3 月, Lazer 等人在 [Science](#) 上发表了一篇文章提出 GFT 出现预测不准确的原因
- Parable of Google Flu: Traps in Big Data Analysis  
谷歌流感的寓言: 大数据分析的陷阱
- 大数据浮夸 Bigdata Hubris
  - 指在[没有拥有真正的大数据或掌握大数据管理和分析能力](#)的情况下, 人们对大数据赋予[盲目期望](#)的现象
- 算法动态性 Algorithm Dynamic 和用户行为习惯的进化
  - 自 2009 年以来, 谷歌为改善其搜索引擎服务而改变了算法, 从而使得[用户习惯](#)也发生了变化, 导致 GFT 的高估



# 总结：大数据时代的思维模式的转变





顾慎凯 博士 | [s.gu@njtech.edu.cn](mailto:s.gu@njtech.edu.cn)

本讲义内容仅限课程听众个人学习、研究。使用者应遵守著作法及其他相关法律规定，不得擅自以营利或非营利性目的发布、传播本讲义。  
将本讲义内容或幻灯片模板用于其他用途时，须征得作者书面许可。

# 感谢聆听