《Python统计计算》

（2021年秋季学期）

翟祥
北京林业大学

E-mail: zhaixbh@126.com

统计计算

第12章 统计计算基础

统计计算

- 随着科学技术的飞速发展，科学计算愈来愈显示出其重要性。科学计算的应用之广已遍及各行各业，例如：气象资料的分析图像，飞机、汽车及轮船的外形设计，高科技研究等都离不开科学计算。而计算有时统计学当中构建模型的重要环节，所以，统计计算快速发展，并且极大的促进了深度学习和人工智能的发展。

统计计算

- 统计学的真正广泛应用得益于计算机信息技术的发展。
- 统计计算就是统计方法和实际计算的结合。
- – 统计方法的实现算法，把统计方法变成可靠、高效的算法，并编程实现。属于经典的统计计算（statistical computing）；
- – 借助于现代计算机的强大处理能力，发展新的统计方法。称为计算统计computational statistics，或计算密集统计（computing intensive statistics）

统计计算内容

- 误差分析
- 算法复杂度
- 随机数生成，随机模拟
- 优化计算与方程求根
- 近似计算，包括函数逼近、差值、数值积分、数值微分
- 矩阵运算

统计计算

- 解决科学计算问题时经历的几个过程
  - 实际问题——〉模型——〉计算方法——〉程序设计——〉上机运行求出解
  - 实际问题——〉模型：由实际问题应用科学知识和数学统计学理论建立模型的过程，是统计学的任务。

统计计算

- 计算方法——〉程序设计——〉计算结果：根据模型提出求解的计算方法，直到编出程序上机算出解，是计算的任务。

- 统计计算方法重点研究：求解的数值方法及与此有关的理论
  - 包括：方法的收敛性，稳定性，误差分析，计算时间的最小（也就是计算费用）,占用内存空间少.

统计计算

- 有的方法在理论上虽不够严格，但通过实际计算，对比分析等手段，被证明是行之有效的方法，也可以采用。因此，数值分析既有纯数学高度抽象性与严密科学性的特点，又有应用的广泛性与实验的高度技术性特点，是一门与使用计算机密切结合的实用性很强的统计学课程。

数学问题的数值解法例示

- 例1..1.1试求函数方程x=cosx在区间 $(0,\frac{\pi}{2})$ 内的一个根。

解

令$f(x)=x-\cos x$, 易知$f(x)$在$[0,\frac{\pi}{2}]$上是连续函数, 且

$$f(0)\,f(\frac{\pi}{2})=(-1)*\frac{\pi}{2}<0$$

由零点定理知, 方程$f(x)=0$在$(0,\frac{\pi}{2})$内至少有一个零点.

又由$f'(x)=1+\sin x>0, x\in(0,\frac{\pi}{2})$

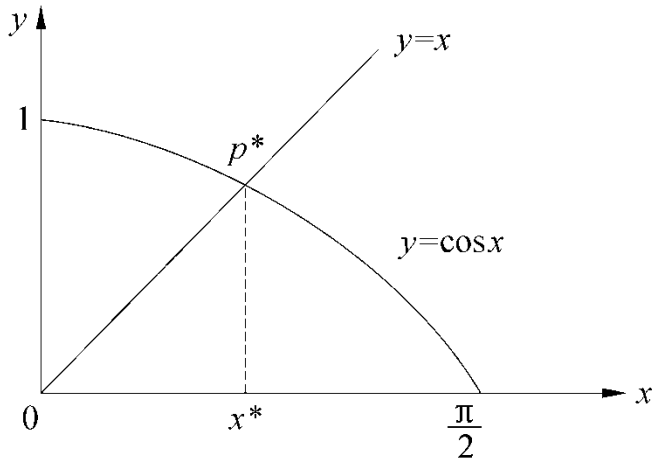
知上述零点唯一.

问题的数值解法例示

本题用解析法求解较为困难.若用图解法,可大致判定此零点位置.作图像

$$\begin{cases} y = x \\ y = \cos x \end{cases}$$

取两曲线交点p^*的横坐标x^*为所求方程的解.,从图中可以看出x^*大致位于$\frac{\pi}{4}$附近.



例2.计算定积分

(1) $\mathrm{I}_1 = \int_0^1 \frac{4}{1+x^2} dx$ (2) $I_2 = \int_0^1 e^{-x^2} dx$

解：（1）由牛顿—莱布尼兹公式

$$I_1 = 4\arctan x\,|_0^1 = 4\arctan 1 - 4\arctan 0 = \pi$$

数值方法有多种，如选择$n=2, h=\frac{1}{2}$,被积函数

$f(x)=\frac{4}{1+x^2}$的复化*Simpson*公式有

$$I_1 \approx \frac{h}{6}[f(0)+4f(\frac{1}{4})+2f(\frac{1}{2})+4f(\frac{3}{4})+f(1)]$$

$$=3.141568627$$

（2）$I_2=\int_0^1 \mathrm{e}^{-x^2}dx$,由于$f(x)=\mathrm{e}^{-x^2}$无原函数，因此，由Newton－Leibniz公式无法求解，仅可用数值方法求解。仍选择$n=2,h=\frac{1}{2}$,的复化*simpson*公式进行数值求解有 $I_2 \approx 0.746855379$。

统计计算

- 随机模拟：在计算机上模拟生成一个统计问题的数据并进行大量的重复，这样相当于获得了此问题的海量的样本。最常用的一种是MCMC。
- 基于随机模拟的方法，如贝叶斯推断，bootstrap 和jacknife, permutation检验，等等
- 机器学习、统计学习、深度学习等方面的算法设计和优化算法

统计计算的要求

- 结果正确，即要求算法的最后结果是我们问题的正确解，最好能够验证结果的正确性。
- 指令可行，即指令含义明确无歧义，指令可以执行并且在现有的计算条件下算法能在允许的时间计算结束。
- 高效，尽可能少地消耗时间和内存、外存资源。

误差概念和有效数

- 在任何计算中其解的精确性总是相对的，而误差则是绝对的.
- 统计计算的算法要得到正确的结果，就需要尽可能减少误差。
- 统计问题中的误差有模型误差、观测误差和数值计算误差，在统计计算研究中主要解决的是如何减少数值计算误差的问题。

误差的分类

- **模型误差** 从实际问题建立的数学模型往往都忽略了许多次要的因素,因此产生的误差称为模型误差.
- **观测（测量、实验）误差** 一般数学问题包含若干参数,他们是通过观测得到的,受观测方式、仪器精度以及外部观测条件等多种因素，不可能获得精确值，由此而来产生的误差称为观测误差。

- **截断误差** 在求解过程中，往往以近似替代，化繁为简，这样产生的误差称为截断误差。
- **舍入误差** 在计算机上运算时受机器字长的限制，一般必须进行舍入，此时产生的误差称为舍入误差。

误差和有效数字

定义1. 设x^*为准确数x的一个近似数,称

$$e(x^*) = x^* - x$$

和

$$e_r(x^*) = \frac{e(x^*)}{x} \quad (x \neq 0)$$

为近似数x^*的绝对误差和相对误差。

当$e(x^*)>0$时,称为过剩绝对误差;
当$e(x^*)<0$时,称为不足绝对误差。

绝对误差是做为衡量x^*的精度高低,比较直观,但无法衡量精度的好坏。

而用相对误差,也称百分比误差,衡量精度的好坏更合理。

误差估计

- 由于准确值在一般情况下是未知的，因此绝对误差和相对误差常常是无法计算的，但有可能给出估计。误差界就是用于误差估计的。

误差估计

定义1.2.2　设x^*是精确数x的一个近似数,

若有正数ε和ε_r满足:

$$|e(x^*)|=|x^*-x|<\varepsilon$$

$$|e_r(x^*)|=\frac{|x^*-x|}{|x|}<\varepsilon_r$$

则称ε和ε_r为近似数x^*的绝对误差界和相对误差界。

在实际计算绝对误差和相对误差时，由于准确值 x 未知，因此常用

$$e_r(x^*) = \frac{e(x^*)}{x^*}$$

表示 $e_r(x^*)$。

有效数字

- 在实践当中，误差的概念就转化为有效数字。

例如　$\pi = 3.14159265.....$的近似数$\pi^* = 3.1416$

则　$e(\pi^*) = 3.1416 - 3.14159265...$

$$= 0.00000734..... \le \frac{1}{2} \times 10^{-4}$$

称$\pi^* = 3.1416$具有五位有效数字的近似数。

定义2. 设近似数x^*有规格化形式

$$x^* = \pm 10^m \times 0.a_1 a_2 a_3 ... a_n ...$$

其中m和$a_i (i = 1, 2, ..., n, ...)$是整数且

$a_1 \neq 0, 0 \leq a_i \leq 9$。如果$x^*$的绝对误差满足

$$|e(x^*)| = |x^* - x| \leq \frac{1}{2} \times 10^{m-n}$$

则称x^*为x的具有n位有效数的近似数。

- 绝对误差，相对误差，有效数是度量近似数精度的常用三种。实际计算时最终结果均以有效数给出。同时也就隐含了绝对误差和相对误差界。

如 $x=\sqrt{2}, x^{*}=1.4142, m=1, n=5$

则x^{*}的绝对误差界$\varepsilon=\frac{1}{2}\times 10^{-4}$

算法的优劣

- 算法优劣的标准
  - 从截断误差观点看,算法必须是截断误差小,收敛敛速要快。即运算量小,机器用时少.
  - 从舍入误差观点看,舍入误差在计算过程中要能控制,即算法的数值要稳定.
  - 从实现算法的观点看,算法的逻辑结构不宜太复杂，便于程序编制和上机实现.

- 设计算法时应遵循的原则
  - 要有数值要稳定性,即能控制误差的传播.
  - 避免大数吃小数,即两数相加时,防止较小的数加不到较大的数上.
  - 避免两相近的数相减,以免有效数字的大量丢失.
  - 避免分母很小(或乘法因子很大),以免产生溢出.

- 显然算法不稳定，理论上成立的算法，在计算时，由于初值的误差在计算过程中的传播，而导致结果的失真，这是我们统计计算要研究的一个重要话题。

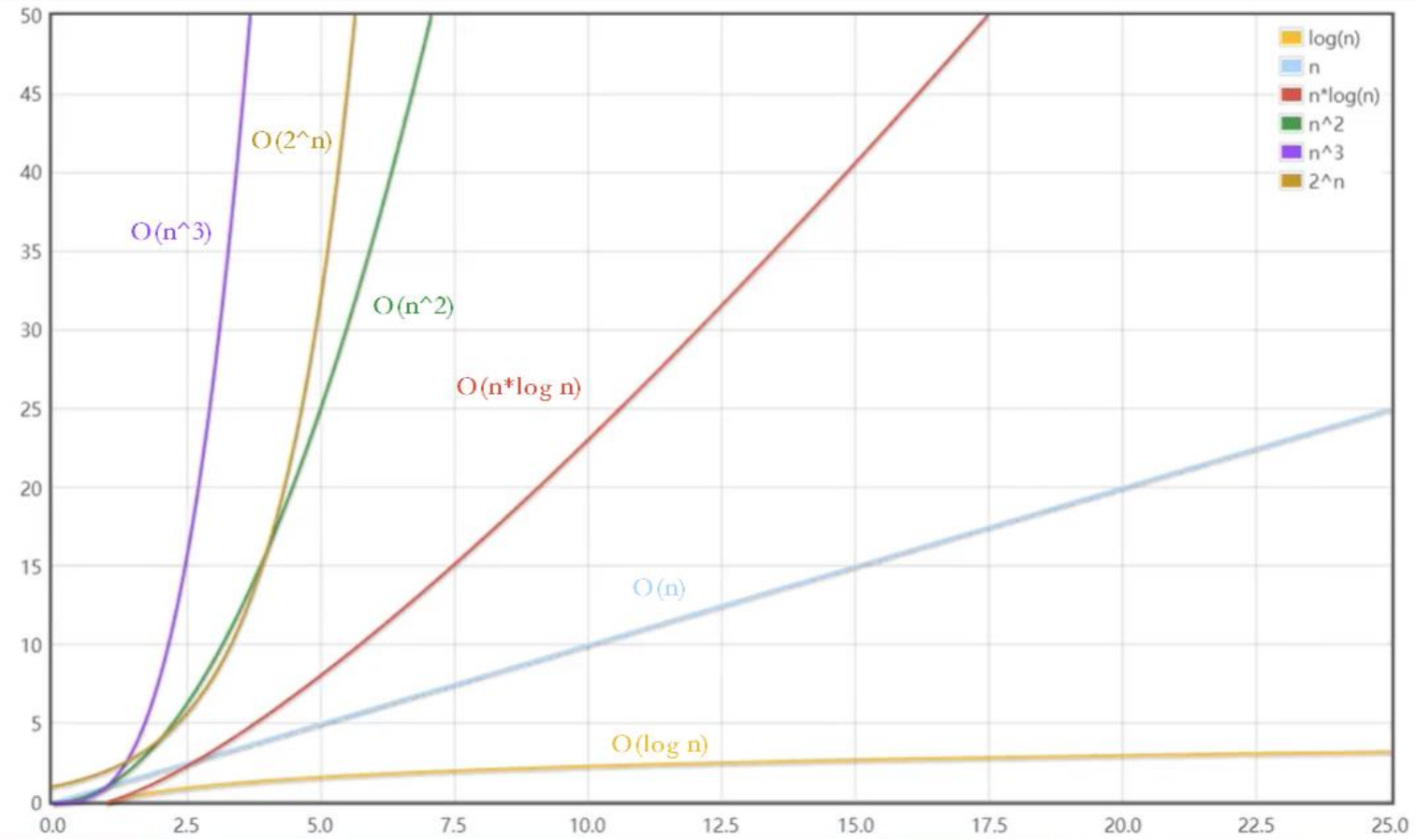
算法的复杂度

- 正确性
- 可读性
- 健壮性：对不合理输入的反应能力和处理能力。
- **时间复杂度（time complexity）：** 估算程序指令的执行次数（执行时间）。
- **空间复杂度（space complexity）：** 估算所需占用的存储空间。
- 评价一个算法的效率主要是看它的时间复杂度和空间复杂度情况。

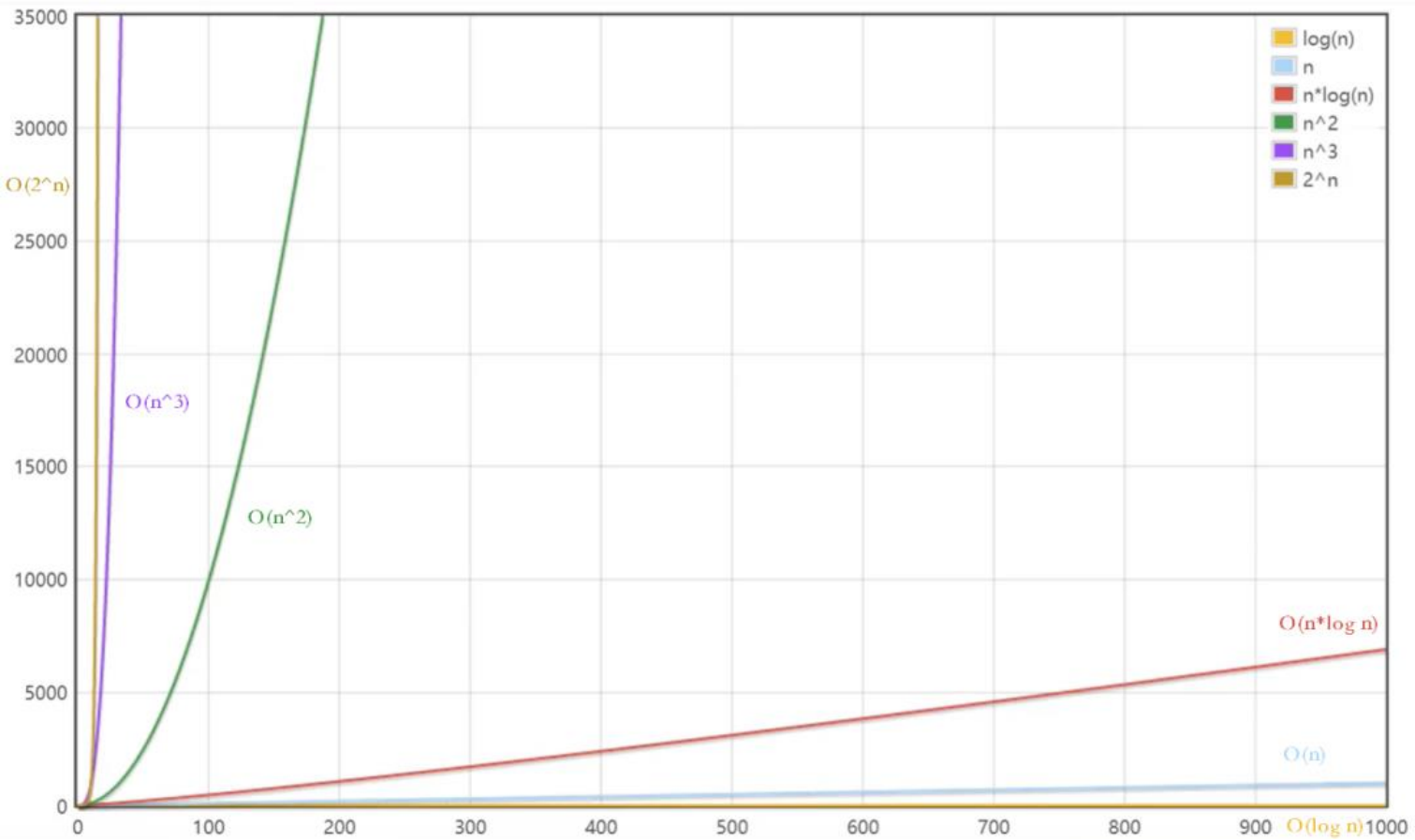
大O表示法

执行次数	复杂度	非正式术语
12	O(1)	常数阶
$2n+3$	O(n)	线性阶
$4n^2+2n+6$	$O(n^2)$	平方阶
$4\log_2 n+25$	O(logn)	对数阶
$3n+2n\log_3 n+15$	O(nlogn)	nlogn阶
$4n^3+3n^2+22n+100$	$O(n^3)$	立方阶
2^n	$O(2^n)$	指数阶

- **当数据规模较小时，** 各复杂度对应的曲线如下图所示。



- **当数据规模较大时，** 各复杂度对应的曲线如下图所示。



python™

谢谢Q/A