

数学建模竞赛学习笔记

从理论到实战

谷文军

数学建模。

前言

目 录

序	i
前言	iii
 第一部分 数学建模基础	 1
第一章 绪论	3
1.1 绪论	3
第二章 数学建模的思想与方法	5
2.1 建模步骤	5
2.1.1 问题提出	5
2.1.2 量的分析	5
2.1.3 模型假设	5
2.1.4 模型建立	5
2.1.5 模型求解	5
2.1.6 模型分析	6
2.1.7 模型检验	6
2.1.8 模型应用	6
2.2 建模方法	6
2.2.1 机理分析法	6
2.2.2 系统识别建模法	6
2.2.3 仿真建模法	6
2.2.4 相似类比建模法	6
2.3 分类与特点	7
2.3.1 数学建模的分类	7
2.3.2 数学建模的特点	7
2.4 数学建模能力的培养	8

第二部分 数学建模方法

9

第三章 初等方法建模

11

3.1	勾股定理与黄金分割率	11
3.1.1	黄金分割应用于高跟鞋问题	11
3.1.2	黄金分割在其他领域的应用	11
3.2	九宫图	11
3.2.1	九宫图问题的提出	11
3.2.2	九宫图问题的求解	12
3.3	椅子稳定问题	12
3.3.1	问题引入与建模准备	12
3.3.2	模型假设	13
3.3.3	模型建立	13
3.3.4	模型求解	14
3.4	商人过河问题	14
3.4.1	问题引入	14
3.4.2	模型分析	15
3.4.3	模型建立	15
3.4.4	模型求解	15
3.5	图论方法与网络模型	16
3.5.1	图论的起源	16
3.5.2	图的概念	16
3.5.3	哥尼斯堡七桥问题	16
3.6	层次分析方法	17
3.6.1	引子	17
3.6.2	层次分析法	18
3.6.3	层次分析法的基本步骤	18
3.6.4	假期旅游案例	21
3.7	双层玻璃问题	22
3.7.1	问题的提出	22
3.7.2	量的分析	22
3.7.3	模型假设	23
3.7.4	模型建立	23
3.7.5	模型分析与求解	24

第四章 微积分与微分方程方法建模

27

4.1	Malthus 人口模型	27
4.1.1	Malthus 人口论	27

4.1.2	基本概念	27
4.1.3	模型假设	28
4.1.4	模型建立与求解	28
4.1.5	应用案例	28
4.1.6	Logistic 阻滞增长模型	29
4.2	细菌的繁殖数学模型	31
4.2.1	问题的提出	31
4.2.2	模型假设	31
4.2.3	模型建立	31
4.2.4	模型求解	32
4.3	传染病流行的控制模型	32
4.3.1	传统的传染病流行控制模型	32
4.3.2	改进的传染病流行控制模型 (SI 模型)	33
4.3.3	病人得到治愈的 SIS 模型	34
4.4	价格数学模型	35
4.4.1	价格数学模型的定义	35
4.4.2	价格数学模型分类	35
4.4.3	价格数学模型的分析	35
4.4.4	价格数学模型的建立	35
4.4.5	结论	36
4.5	湖泊污染减退模型	36
4.5.1	模型的背景介绍	36
4.5.2		37
4.6	狼捕食兔子模型	37
4.7	导弹追踪目标问题	37
4.8	森林救火模型	37
第五章	线性规划模型	39
5.1	线性规划模型实例	39
5.2	线性规划问题	39
5.3	求解线性规划问题的基本思想	39
5.4	线性规划问题的几何解释和图解法	39
5.5	整数线性规划问题	39
5.6	线性规划的对偶理论	39
5.7	非对称形式的对偶线性规划问题	39
5.8	两铁路平板车的装货问题	39

第六章 对策模型 41

6.1 对策模型的引入 41

6.2 对策模型的基本理论 41

6.3 矩阵对策模型 41

6.4 矩阵对策模型实例分析 41

6.5 鞍点存在定理 41

6.6 混合对策模型 41

第七章 决策模型 43

7.1 决策模型的概念及分类 43

7.2 风险型决策 43

7.3 不确定型决策 43

第三部分 算方案例 45

第四部分 数据预处理 47

第五部分 数据可视化 49

第八章 可视化基础 51

8.1 绪论 51

8.2 字体 51

8.3 颜色 51

第九章 统计图表 53

9.1 线型图 53

9.1.1 简单的线型图 53

9.1.2 自定义虚线样式 54

9.1.3 误差限制选择线 55

9.1.4 误差带曲线 57

9.2 柱状图 58

9.2.1 简单的柱状图 58

9.2.2 水平条形图 59

9.2.3 带有误差线的柱状图 60

9.2.4 带标签的分组柱状图 61

9.2.5 离散分布条形图 63

9.2.6 破碎的条形图 64

9.3 散点图	66
9.4 饼图	66
9.5 地图	66
9.6 热力图	66
9.7 表格	66
9.8 形象化描述	66
9.9 地图	66
9.10 热力图	66
 第十章 数值参数	 67
 第六部分 LATEX 论文写作	 69
 附录 A Angular CLI	 71

PART I

第一部分

数学建模基础

1.1 绪论

数学建模的思想与方法

2.1 建模步骤

8 步建模法

2.1.1 问题提出

- 了解实际问题的背景 (属于哪一个领域)
- 明确数学建模的目的 (解决什么问题)

2.1.2 量的分析

- 收集数学建模的必要信息 (相关数据和参考资料)
- 分析研究对象的主要特征 (内在机理或输入输出)

2.1.3 模型假设

根据所研究的对象特征及建模目的,抓住问题本质,忽略次要因素,对问题做出合理的简化假设 (基本符合实际情况,能够用数学语言描述问题)。

2.1.4 模型建立

根据假设,用数学语言、符号描述出研究对象的内在规律,并建立包含常量、变量等的数学模型,可以是函数表达公式、数学方程、数据表格、算法或图形甚至是一段文字描述。

2.1.5 模型求解

采用各种计算方法对所建立的数学模型进行求解,可能是求函数的极值、求方程的解、算法或图形的实现等。

2.1.6 模型分析

- 对求解结果进行数学上的分析
- 包括结果的误差分析（误差是否在允许范围内）
- 统计分析（结果是否符合特定的统计规律）
- 模型对数据的灵敏度分析（模型的结果是否会因数据的微小变化而发生大的变化）
- 对假设的鲁棒性分析（robustness）（模型的结果是否对某一假设非常依赖？）等

2.1.7 模型检验

- 将求解结果和分析结果翻译回到实际问题之中，与实际现象、实际数据进行比较，检验是否与实际吻合。
- 如果吻合较好，则模型及其结果可以应用于实际。如果吻合不好，则需要对模型进行修正。
- 此时问题常常出现在模型假设上（假设是否合理，是否忽略了重要因素而保留了次要因素），所以应对模型假设进行修正或补充，然后重新建模。

2.1.8 模型应用

当模型经过检验已成为一个具有合理性和实用性的模型后，即可以用来解决实际问题。

2.2 建模方法

2.2.1 机理分析法

在对研究对象内部机理分析的基础上，利用建模假设所给出的建模信息或前提条件及相关领域知识、相应的数学工具来构造模型。

2.2.2 系统识别建模法

对系统内部机理不清楚的情况下，利用建模假设或对系统进行实际测试所得到的数据信息，再运用数学方法确定模型形式，借助于概率论和数理统计来辨识参数，构造模型。

2.2.3 仿真建模法

利用各种仿真方法建模。

2.2.4 相似类比建模法

借助于相似原理和事物之间的类比关系进行建模的方法，即根据不同对象之间的某些相似性，借用移植领域的数学模型构造数学模型的方法。

2.3 分类与特点

2.3.1 数学建模的分类

- 按建模的数学方法划分

初等模型、数学规划模型、微分方程模型、差分方程模型、概率统计模型、图论模型、模糊模型和灰色模型等。

- 按建模中变量特点划分

确定性模型与随机性模型、静态模型与动态模型、线性模型与非线性模型、离散模型与连续模型等

- 按应用领域划分

人口模型、交通模型、环境模型、规划模型、生态模型、资源模型等

- 按建模目的划分 描述模型、预测模型、优化模型、决策模型、控制模型等

- 按照对问题的了解程度划分

白箱模型、黑箱模型、灰箱模型

2.3.2 数学建模的特点

- 逼真性和可行性

模型越逼真就越复杂,应用起来费用就越高,常与取得的效益不成正比,所以需要对逼真性与可行性进行折中

- 渐进性

数学模型通常不会是一次就成功的,往往需要反复修正,逐渐完善

- 健壮性

对于已经建好的数学模型,当观测数据有微小的改变或者模型结构及参数发生微小变化时,模型求解的结果也随之发生微小的变化

- 可移植性

数学模型是现实对对象抽象化的产物,它可能与其他领域或其他事物有共性,常常好多领域不同事物却有几乎相同的数学模型

- 非预制性

建模时遇到的问题往往事先没有答案,因此必须创新,产生新方法、新概念

- 条理性

从建模角度出发,人们对现实对象分析应该全面、深入,更具有条理性。即使建模失败,对解决研究实际问题也是有利的

- 技艺性

建模与其说是一门技术,不如说是一种技艺很强的技巧艺术,建模期间经验、想象力、洞察力、判断力以及直觉灵感起的作用往往比数学知识更大

- 局限性

由于建模时往往把显示对象简化、近似、假设,因此当模型应用到实际时就必须考虑被忽略的简化因

素,于是结论往往是相对的、近似的。另外,由于人类认识能力受科学技术以及数学本身发展水平的限制,至今还有不少实际问题没有建立起有价值实用的数学模型

2.4 数学建模能力的培养

- 数学知识的积累
- 多看、多学数学建模案例
- 培养观察能力和用数学解决问题的思想
- 需要丰富的想象力与敏锐、深刻的洞察力
- 兴趣是学习的动力,努力培养建模兴趣
- 与计算机紧密关联,学会使用相关软件
- 注意团队意识和团结协作
- 学会类比,做到“由此及彼和由彼及此”,培养发散思维能力
- 培养自学能力,能快速获取新知识,并能学以致用
- 从杂乱无章的各种信息中快速挑选收集有用信息,利用图书馆、网络查找相关资料

PART II

第二部分

数学建模方法

初等方法建模

3.1 勾股定理与黄金分割率

3.1.1 黄金分割应用于高跟鞋问题

原理:人的下肢长占身高的 0.618 时,满足最佳黄金比例。

计算黄金分割最简单的方法:

设某人身高为 h cm, 下肢长 h_1 cm, 选择高跟鞋的鞋跟高度为 x cm, 则由建模原理得:

$$\frac{x + h_1}{x + h} = 0.618 \quad (3.1)$$

解得:

$$x = \frac{0.618h - h_1}{0.382} \quad (3.2)$$

3.1.2 黄金分割在其他领域的应用

医学与 0.618 的联系

- 外界环境温度为人体温度的 0.618 倍时,人会感到最舒适
- 养生之道:动与静是一个 0.618 的比例关系,即四分动六分静是人们认为的最佳养生方式
- 吃饭六七成饱,几乎不生胃病

建筑学与 0.618 的联系

黄金矩形

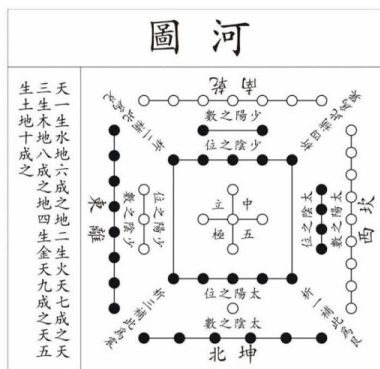
3.2 九宫图

3.2.1 九宫图问题的提出

九宫图又称为三阶幻方,出自西汉(公元前 206-公元 25)学者戴德编纂的《大戴礼》,源自河图洛书。



(a) 九宫图



(b) 河图



(c) 洛书

图 3.1: 九宫图及其起源

河图由十种花点组成,分别代表 1 – 10 这 10 个数,两种花点构成一组,分别布局在东、西、南、北、中五个位置上,每组花点所表示的两个数的差都是 5。

洛书由九种花点组成,分别代表 1 – 9 这 9 个数,各数的位置排列奇巧,奇偶交替变化,纵横六线及两条对角线上三数之和都为 15。

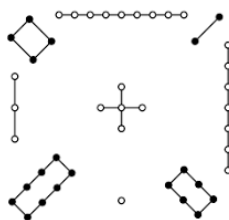
3.2.2 九宫图问题的求解

口诀一

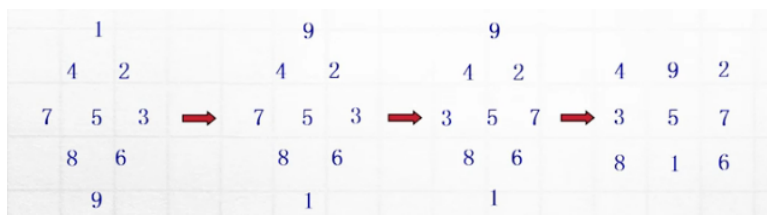
九宫者,法以灵龟。二四为肩,六八为足。左三右七,戴九履一,五居中央。

口诀二

九子斜列,上下对易,左右相更。四维挺出。



(a) 九宫图解一



(b) 九宫图解二

图 3.2: 九宫图解

3.3 椅子稳定问题

3.3.1 问题引入与建模准备

一把四条腿的椅子放在不很平坦的地面上,是否是稳定的?

不很平坦的定义: 肉眼可见的地面是平坦的

椅子不稳定的原因:

- 椅子的四条腿的长度可能不一样
- 地面不平坦
- 椅子的四个底角与地面之间有距离 $h, h > 0$

三个点确定一个平面,对于相对比较平坦的地面来讲,总可以保证三条腿同时着地,因而只需再使其余的一条腿也能完全着地即可。

量的分析

- 椅子的脚和地面的距离作为变量。变量为 0,意味着椅子的脚与地面没有距离,椅子稳定。
- 设计旋转角度的连续函数

3.3.2 模型假设

假设 1: 椅子四条腿一样长,椅脚与地面接触处视为一个点,四脚的连线呈正方形 $ABCD$,四个椅脚的坐标中对称;

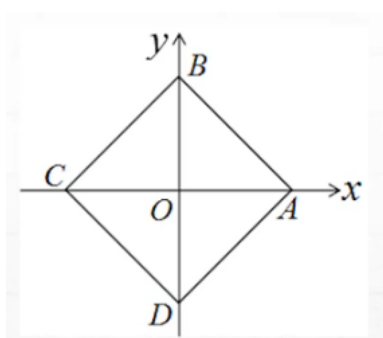


图 3.3: 椅脚坐标系

假设 2: 地面高度是连续变化的,沿任何方向都不会出现间断,如台阶,即地面可视为数学上的连续曲面;

假设 3: 对于椅脚的间距和椅腿长度而言,地面是相对平坦的,使椅子在任何位置至少有三只脚同时着地。

3.3.3 模型建立

构造表示距离的函数:

- 设 $f(\theta)$ 是 A, C 两椅脚与地面的距离之和
- $g(\theta)$ 是 B, D 两椅脚与地面的距离之和
- θ 是椅子绕中心点 O 旋转角度

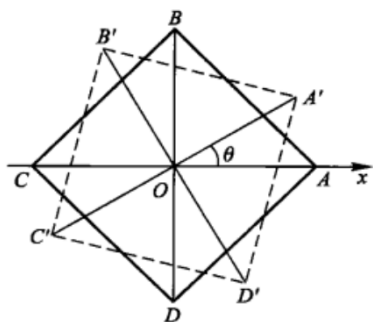


图 3.4: 椅脚坐标系

- 由假设 2, $f(\theta)$ 和 $g(\theta)$ 都是连续函数;
- 由假设 3, 椅子在任何位置至少有三只脚同时着地, 即对任意的 θ , $f(\theta)$ 和 $g(\theta)$ 中至少有一个为零函数。
- 在椅子处于初始不稳定的位置时, 即 $t = 0$ 时, 不妨设 $g(0) = 0, f(0) > 0$ 。

椅子在不很平坦地面上稳定问题的数学命题:

已知 $f(\theta)$ 和 $g(\theta)$ 是 t 的连续函数, 对任意 θ , 满足条件 $f(\theta) \cdot g(\theta) = 0$, 且 $g(0) = 0, f(0) > 0$, 则至少存在一个 θ_0 , 使 $f(\theta_0) = g(\theta_0) = 0$, 即在旋转椅子的时候, 可能在多个位置上都是稳定的。

3.3.4 模型求解

证明:

将椅子逆时针旋转 $\frac{\pi}{2}$, 则对角线 AC 和 BD 的位置相互交换, 且 $f(\theta)$ 和 $g(\theta)$ 是闭区间 $[0, \frac{\pi}{2}]$ 上的连续函数。

由上面数学命题, 一定有下列式成立:

$$\begin{cases} g(0) = 0, f(0) > 0 \\ g(\frac{\pi}{2}) > 0, f(\frac{\pi}{2}) = 0 \end{cases} \quad (3.3)$$

令 $h(\theta) = f(\theta) - g(\theta)$, 显然 $h(\theta)$ 也是闭区间 $[0, \frac{\pi}{2}]$ 上的连续函数,

容易检验: $h(\theta)$ 满足条件 $h(0) > 0, h(\frac{\pi}{2}) < 0$ 。

由闭区间上连续函数的零点存在定理, 至少存在 θ_0 , 使得 $h(\theta_0) = 0$, 即 $f(\theta_0) = g(\theta_0)$ 。

再由命题中的条件“对任意 $\theta, f(\theta) \cdot g(\theta) = 0$ ”, 得知 $f(\theta_0) = g(\theta_0) = 0$ 。

3.4 商人过河问题

3.4.1 问题引入

三个商人各带一名随从渡河, 随从们密约: 在河的任一岸, 一旦随从人数比商人多, 就杀人越货。问题: 商人该如何设计渡河方案以安全渡河?

3.4.2 模型分析

允许状态集合

X_k : 第 k 次渡河前此岸的商人数, $X_k, Y_k = 0, 1, 2, 3$;

Y_k : 第 k 次渡河前此岸的随从数, $k = 1, 2, \dots, n$;

渡河过程的状态: $S_k = (X_k, Y_k)$

允许状态集合: $S = \{(x, y) | x = 0, y = 0, 1, 2, 3; x = 3, y = 0, 1, 2, 3; x = y = 1, 2\}$

渡河状态集合

U_k, V_k : 分别任第 k 次渡河船上的商人数与随从数; $(U_k, V_k = 0, 1, 2; k = 1, 2, \dots, n)$

决策: $d_k = (U_k, V_k)$

允许决策集合: $D = (u, v) | u + v = 1, 2$

状态转移律

$$S_{k+1} = S_k + (-1)^k d_k$$

3.4.3 模型建立

制定安全渡河方案归结为如下的多步决策模型:

求 $d_k \in D(k = 1, 2, \dots, n)$, 使 $s_k \in S$ 按状态转移律由 $S_1 = (3, 3)$ 经有限步 n 到达 $S_{n+1} = (0, 0)$

3.4.4 模型求解

使用图解法, 一共 16 个状态格点, 其中允许状态 S 有 10 个点, 允许决策 D 表示沿方格移动 1 或 2 格。

- (1). k 为奇数, 由此岸去彼岸, 这时此岸人数减少。移动方向为左/下
- (2). k 为偶数, 由彼岸到此岸, 这时此岸人数增多。移动方向为右/上
- (3). d_k 移动一格表示船上有 1 个人, 移动两格表示船上有 2 个人
- (4). 用实线表示从此岸去彼岸, 虚线表示从彼岸回到此岸

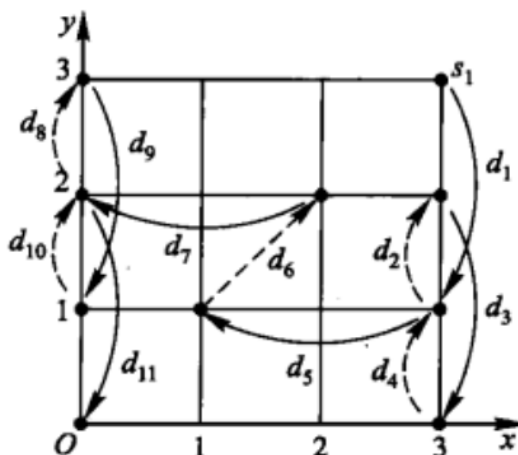


图 3.5: 安全渡河问题图解法

3.5 图论方法与网络模型

3.5.1 图论的起源

图论使组合数学的一个分支,起源于 1736 年欧拉的第一篇关于图论的论文,这篇论文解决了著名的“哥尼斯堡七桥问题”,从而使欧拉成为图论的创始人。

3.5.2 图的概念

图的定义

图是一个有序二元组 $G = V(G), E(G)$, 其中 $V(G) = v_1, v_2, \dots, v_n$ 为顶点集 $V(G)$ 中的元素 v_i 称为顶点, $E(G) = e_1, e_2, \dots, e_m$ 称为边集, $E(G)$ 中的元素 e_k 叫做边。

- 顶点总数记位 $|V(G)|$, 边的总数记位 $|E(G)|$
- 若 $|V(G)| = n$, 则称 G 为 n 阶图
- 若顶点总数 $|V(G)|$ 与边的总数 $|E(G)|$ 均为有限数, 则称 G 为有限图

有向图的定义

若顶点集合 $V(G) \neq \emptyset$, 边集 $E(G) \cap V(G) = \emptyset$, 则称 $G = \{V(G), E(G), \emptyset\}$ 为一个有向图。

关联函数

若 $\Phi_G(e) = uv, \Phi_G(e)$ 称为关联函数, 表示边 e 与顶点 u 与 v 相关联, 又称顶点 u 与 v 相邻, u 是 e 的尾, v 是 e 的头, 即由 u 指向 v 。

边 e 也成为弧, 是由两个顶点组成的有序对, 通常用尖括号表示。例如: $\langle v_i, v_j \rangle, v_i$ 称为弧尾, v_j 称为弧头。 $\langle v_i, v_j \rangle$ 和 $\langle v_j, v_i \rangle$ 是两条不同的有向边。

无向图的定义

若 G 的每条边头尾部分, 即 $\Phi_G(e) = uv = vu$, 则称 G 为无向图, 无向图中每条边均是顶点的无序对, 通常用圆括号表示, 例如: (v_i, v_j) 表示一条无向边, 且有 $(v_i, v_j) = (v_j, v_i)$ 。

3.5.3 哥尼斯堡七桥问题

柯尼斯堡七桥问题 (Seven Bridges of Königsberg) 是图论中的著名问题。这个问题是基于一个现实生活中的事例: 当时东普鲁士柯尼斯堡 (今日俄罗斯加里宁格勒) 市区跨普列戈利亚河两岸, 河中心有两个小岛。小岛与河的两岸有七条桥连接。在所有桥都只能走一遍的前提下, 如何才能把这个地方所有的桥都走遍?

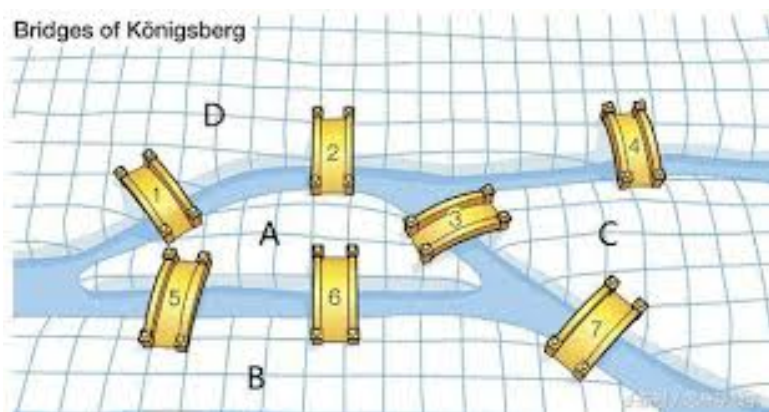


图 3.6: 哥尼斯堡七桥问题

我们将七桥抽象为无向图中的边, 四片陆地抽象为无向图中的点, 该无向图可以表示为 $G = \{V(G), E(G), \Phi_G\}$, 其中 $V(G) = \{A, B, C, D\}$, $E(G) = \{a, b, c, d, e, f, g\}$, $\Phi_G(a) = AB$, $\Phi_G(b) = BC$, $\Phi_G(b) = BC$, $\Phi_G(c) = \Phi_G(d) = AC$, $\Phi_G(e) = \Phi_G(f) = AD$, $\Phi_G(g) = BD$ 。

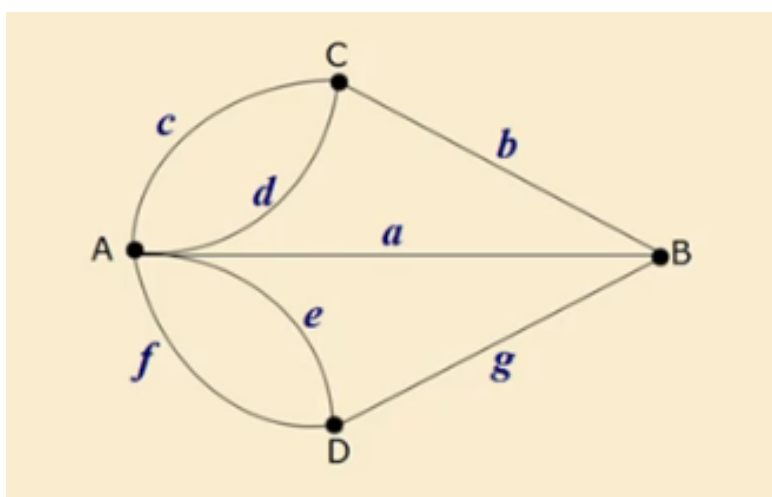


图 3.7: 哥尼斯堡七桥问题图结构

七桥问题的图论阐述

从七桥问题无向图中某个顶点出发, 遍历每条边恰好一次, 最后能否还回到原来的顶点处 (出发点)?

3.6 层次分析方法

3.6.1 引子

生活中一些问题很难用完全定量的数学模型来解决, 对这种复杂决策问题, 运用层次分析方法能够找到最佳的解决办法, 给出最优的决策。层次分析法 (Analytic Hierarchy Process 简称 AHP) 是将与决策有关的元素分解成目标、准则、方案等层次, 在此基础上进行定性和定量分析的决策方法。其基本思路与人对一个复杂的决策问题的思维、判断过程大体上是一样的。

3.6.2 层次分析法

将决策问题按总目标、各层子目标、评价准则直至具体的备选方案的顺序分解为不同的层次结构,然后使用求解判断矩阵特征向量的方法求得每一层次的各元素对上一层次某元素的优先权重,最后再运用加权的方法求出各备选方案对总目标的最终权重向量。其中权重最大者即为最优方案。

层次分析法中每一层的权重设置到最后都会直接或间接影响到结果,而且在每个层次中的每个因素对结果的影响程度都是量化的,非常清晰明确。这种方法擅长于对无结构特性的系统评价以及多目标、多准则、多时期等的系统评价。

层次分析法也具有**局限性**:

- 不能为决策提供新方案,层次分析法只能从原有的备选方案中选取最佳方案,因此可能会产生一个由于主观性或者自身创造能力不够而不是最优的方案。
- 定量数据少,主观因素多,不易令人信服。
- 指标过多时数据统计量大,且权重难以确定。当我们希望解决一个具有普适性的问题时,指标的选取数量很可能会随之增加。指标的增加意味着需要构造的层次更深、数量更多、规模更庞大的判断矩阵。
- 特征值和特征向量的精确求法比较复杂,随着指标的增加,阶数也随之增加,在计算上也变得越来越困难。

3.6.3 层次分析法的基本步骤

1. 建立层次结构模型

在深入分析实际问题的基础上,将有关的各因素按照不同属性自上而下地分解成若干层次,同一层的各个因素因为从属于上一层因素,因此对上层因素会产生较大影响,同时又会支配下一层因素或受到下层因素的影响。

在层次结构模型中,最上层为目标层,通常只有一个因素,最下层通常为方案层,中间可以有一个或几个层次,通常为准则或指标层。当准则过多时(例:> 9),应进一步分解出子准则层。准则层不宜过少,通常5-7个左右。

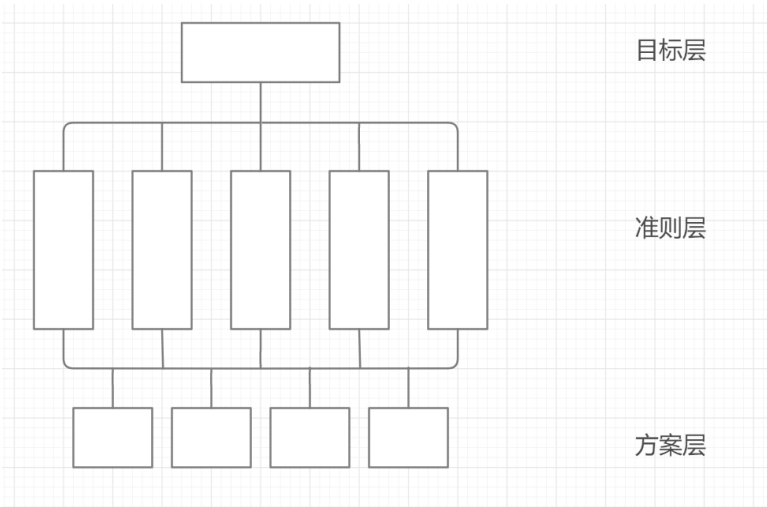


图 3.8: 层次结构模型

2. 构造成对比较矩阵

这一步是要比较层次结构模型的第二层各个因素对上一层因素的影响,从而确定它们对上层因素的影响作用中所占的权重。

设有 n 个因素 x_1, x_2, \dots, x_n 对上一层目标有影响直接确定它们对目标的影响程度不是很容易,所以每次取两个因素 x_i 与 x_j 比较。用 a_{ij} 表示 x_j 和 x_i 对上层目标的影响比,则 $A = (a_{ij})_{n \times n}$ 称为成对比较矩阵,又称为正互反矩阵。

$$A = \begin{bmatrix} \frac{x_1}{x_1} & \frac{x_1}{x_2} & \dots & \frac{x_1}{x_n} \\ \frac{x_2}{x_1} & \frac{x_2}{x_2} & \dots & \frac{x_2}{x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{x_n}{x_1} & \frac{x_n}{x_2} & \dots & \frac{x_n}{x_n} \end{bmatrix}_{n \times n}, a_{ij} > 0, a_{ii} = 1, a_{ij} = \frac{1}{a_{ji}} (i = 1, 2, \dots, n) \quad (3.4)$$

成对比较矩阵中,每一个 a_{ij} 的取值都是有一定的尺度和规范的,按照 Satty 的提议, a_{ij} 在 1-9 及其倒数中间取值。例如:

- $a_{ij} = 1$, 元素 i 与元素 j 对上一层次因素的重要性相同
- $a_{ij} = 3$, 元素 i 比元素 j 略重要
- $a_{ij} = 5$, 元素 i 比元素 j 重要
- $a_{ij} = 7$, 元素 i 比元素 j 重要的多
- $a_{ij} = 9$, 元素 i 比元素 j 及其重要
- $a_{ij} = 2, 4, 6, 8$, 元素 i 与 j 的重要性介于 1, 3, 5, 7, 9 之间

3. 计算权向量及一致性检验

对于每一个成对比较阵计算其最大特征根 $\lambda_{\max}(A)$ 及对应特征向量,利用一致性指标、平均随机一致性指标和一致性比率做一致性检验。

若检验通过,那么标准化特征向量即为权向量,若不通过,需要重新构造成对比较阵 A 。

由于成对比较矩阵是我们对复杂事物采取两两比较得到的矩阵,构造过程具有明显的主观性,不可能做到判断具有完全的一致性,难免有误差。所以需要成对比较矩阵进行一致性检验。

定义 1

如果一个正互反矩阵 A 满足 $a_{ij}a_{jk} = a_{ik} (i, j, k = 1, 2, \dots, n)$ 则称 A 为一致矩阵,简称一致阵

性质 1

如果矩阵 A 是一致阵,那么它的秩为 1,唯一非零特征根为 n 。

性质 2

一致阵的任一列(行)向量都是对应于特征根 n 的特征向量。

判别法

判别一个 n 阶矩阵 A 是否为一致阵,只要计算 A 的最大特征根即可。如果 A 不是一致阵,则可以证明 $\lambda_{\max}(A) > n$ 而且 $\lambda_{\max}(A) - n$ 越大,说明不一致程度越严重。

定义 2

设 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)^T$ 为正向量,称 α' 为 α 的标准化向量,其中

$$\alpha' = \left(\frac{\alpha_1}{\sum_{i=1}^n \alpha_i}, \frac{\alpha_2}{\sum_{i=1}^n \alpha_i}, \dots, \frac{\alpha_i}{\sum_{i=1}^n \alpha_i} \right) \quad (3.5)$$

一致性检验

设 A 为 n 阶成对比较矩阵

- 一致性指标: $CI = \frac{\lambda - n}{n-1}$ 来表征一致性程度, 当 $CI = 0$ 时为一致阵, CI 越大, A 的不一致程度越严重。
- 平均随机一致性指标: RI 来确定 A 的不一致程度的容许范围。对于固定 n , 随机构造成对比较矩阵 $A' = (a'_{ij})$ 其中 a'_{ij} 是从 $1, 2, \dots, 9$ 和 $1, \frac{1}{2}, \dots, \frac{1}{9}$ 中随机抽取的。这样构造的 A' 是不一致的, 它的 CI 相当大。如此构造相当多的 CI , 然后算出这些 A' 的平均值作为平均随机一致性指标的 RI

表 3.1: 随机一致性指标 RI 数值表

n	1	2	3	4	5	6	7	8	9	10	11
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51

- 一致性比率: $CR = \frac{CI}{RI}$, 当 $CR < 0.1$ 时 A 的不一致程度在容许范围内, 可用其标准的特征向量 α' 作为权向量, 否则需要重新调整判断矩阵 A

权向量的计算方法

- (1). 如果成对比较矩阵是一致矩阵, 则把它的列向量 (最大特征根对应的特征向量) 标准化得到的向量即为各个因素对上一层目标影响大小的权向量。
- (2). 如果成对比较矩阵不是一致矩阵, 而它的不一致程度又在容许的范围内, 则计算成对比较矩阵的最大特征根及其相对应的特征向量, 然后将其标准化, 令它作为各个因素对上一层目标影响大小的权向量。
- (3). 当成对比较矩阵的不一致程度很严重时, 需要重新构造或修正成对比较矩阵。

运用方根法近似计算最大特征值与相应的特征向量

- (1). 计算判断矩阵每一行元素的乘积 $W_i = \prod_{j=1}^n a_{ij} (i, j = 1, 2, \dots, n)$
- (2). 计算 W_i 的 n 次方根 $\bar{W}_i = \sqrt[n]{W_i}$
- (3). 对向量 $\bar{W}_i = (\bar{W}_1, \bar{W}_2, \dots, \bar{W}_n)$ 做标准化处理 $a_i = \bar{W}_i \div \sum_{j=1}^n \bar{W}_j (i = 1, 2, \dots, n)$, 得到 $B = (a_1, a_2, \dots, a_n)^T$ 即为所求的特征向量
- (4). 计算最大特征值 $\lambda_{\max}(A) = \frac{1}{n} \sum_{i=1}^n \frac{(AB)_i}{a_i}$, 其中

$$AB = \begin{bmatrix} (AB)_1 \\ (AB)_2 \\ \vdots \\ (AB)_N \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

上式中, $(AB)_i$ 表示向量 AB 的第 i 个元素

4. 计算组合权向量并做组合一致性检验

这是层次分析法的最后一个步骤, 需要计算最下层对最上层目标的组合权向量, 并做组合一致性检验。若检验通过, 则可按照组合权向量表示的结果进行决策, 否则需要重新考虑模型或重新构造那些一致性比率较大的成对比较矩阵。

3.6.4 假期旅游案例

问题描述

假如你在准备一趟假期旅游,有三个景点 A、B、C 供你选择,你将根据景点费用、居住条件、饮食和旅途等准则比较这三个候选景点,并最终决策选择其中一个景点。

层次结构模型

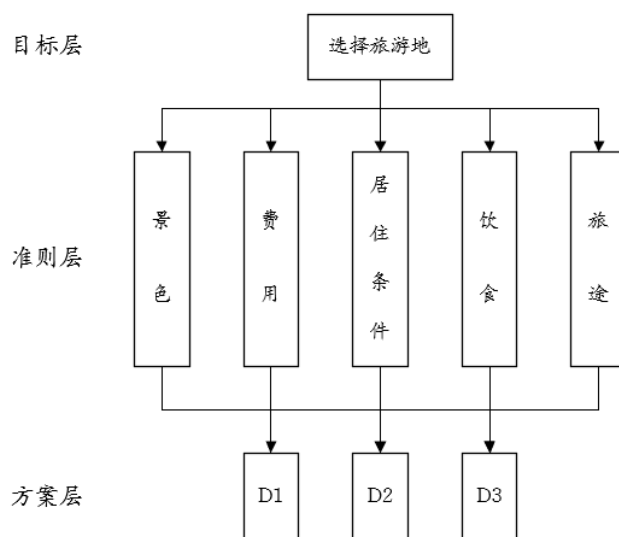


图 3.9: 旅游地选择层次分析模型

构造成对比较矩阵

设景色、费用、居住条件、饮食和交通便利五个因素分别为 X_1, X_2, X_3, X_4, X_5 , 假设成对比较矩阵如下:

$$A = \begin{bmatrix} 1 & \frac{1}{2} & 4 & 3 & 3 \\ 2 & 1 & 7 & 5 & 5 \\ \frac{1}{4} & \frac{1}{7} & 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{5} & 2 & 1 & 1 \\ \frac{1}{3} & \frac{1}{5} & 3 & 1 & 1 \end{bmatrix}$$

其中 $a_{21} = 2$ 表示费用 X_2 与景色 X_1 对选择旅游地这个目标的比是 2 : 1, 说明对费用看的更重要一些, 其他同理。

计算权向量与一致性检验

运用上一节的方法计算可得最大特征值 $\lambda_{\max} = 5.073$, 一致性指标 $CI = \frac{\lambda_{\max} - n}{n - 1} = \frac{5.073 - 5}{5 - 1} = 0.018$, 查表得 $RI = 1.12$, 最终可得一致性比率 $CR = \frac{CI}{RI} = \frac{0.018}{1.12} = 0.016 < 0.1$, 因此通过了一致性检验。

计算特征根 $\lambda_{\max} = 5.073$ 对应的特征向量并对其标准化, 得 $\alpha = \alpha^{(2)} = (0.263, 0.475, 0.055, 0.099, 0.110)^T$, 为了跟后面得步骤进行区分, 这里记位 $\alpha^{(2)}$ 。

从向量中可以看出费用最为重要, 其次是景色和旅途, 再次是饮食和居住条件。

计算组合权向量并做一致性检验

三个方案分别针对五个准则构造成对比较矩阵,如下:

$$B_1 = \begin{bmatrix} 1 & 2 & 5 \\ \frac{1}{2} & 1 & 2 \\ \frac{1}{5} & \frac{1}{2} & 1 \end{bmatrix} \quad B_2 = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{8} \\ 3 & 1 & \frac{1}{3} \\ 8 & 3 & 1 \end{bmatrix} \quad B_3 = \begin{bmatrix} 1 & 1 & 3 \\ 1 & 1 & 3 \\ \frac{1}{3} & \frac{1}{3} & 1 \end{bmatrix} \quad B_4 = \begin{bmatrix} 1 & 1 & 4 \\ \frac{1}{3} & 1 & 1 \\ \frac{1}{4} & 1 & 1 \end{bmatrix} \quad B_5 = \begin{bmatrix} 1 & 1 & \frac{1}{4} \\ 1 & 1 & \frac{1}{4} \\ 4 & 4 & 1 \end{bmatrix}$$

B_1, B_2, B_3, B_4, B_5 表示三种旅游方案 D_1, D_2, D_3 分别对准则层中的五个因素(景色、费用、居住条件、饮食和旅途)的影响程度的比较矩阵。例如 B_3 中的 $b_{23} = 3$ 表示方案 D_2 与 D_3 相比居住条件好坏的程度之比。

接着分别计算各矩阵的最大特征根以及相应的权向量,再经过标准化得到标准化向量,然后我们还需要对他们分别进行一致性检验,结果如下表:

表 3.2: 数值表

k	1	2	3	4	5
$\alpha_k^{(3)}$	0.595	0.082	0.429	0.633	0.166
	0.277	0.236	0.429	0.193	0.166
	0.129	0.682	0.142	0.175	0.668
λ_k	3.005	3.002	3	3.009	3
CI_k	0.003	0.001	0	0.005	0
CR_k	0.052	0.002	0	0.009	0

显然,这 5 个矩阵都通过了一致性检验,其中 $\alpha_k^{(3)}$ 即为组合权向量,表示第二层对第一层以及第三层对第二层各个因素的综合影响。

最终我们需要的组合权向量 $\alpha^{(3)} = (\alpha_1^{(3)}, \alpha_2^{(3)}, \alpha_3^{(3)})^T = (0.300, 0.246, 0.456)^T$, 即表示三个景点 D_1, D_2, D_3 中分别占的比重,所以可知方案 D_3 占有更高的比重,因此应该选择方案 D_3 。

3.7 双层玻璃问题

3.7.1 问题的提出

来自北方的同学可能会知道,身边很多的玻璃窗都是双层的,两层玻璃之间是真空的空隙。那么这样的工艺有什么好处呢? 它可以有效的阻止室内温度想室外的扩散。那么当建筑物室内外的热传递过程处于热力学平衡状态时,这种构造形式的双层玻璃窗究竟能比单层玻璃窗阻止多少热量损失? 两层玻璃窗之间的距离控制在多少为好?

3.7.2 量的分析

符号说明

表 3.3: 符号表

符号	说明
T_1	室内温度
T_2	室外温度
d	单层玻璃厚度
l	两层玻璃之间的空气厚度
T_a	内层玻璃的外侧温度
T_b	外层玻璃的内侧温度
k	热传导系数
Q	热量损失

3.7.3 模型假设

1. 热量的传播过程只有传导没有对流,即窗户的密封性能良好,两层玻璃之间的空气是不流动的
2. 室内温度和室外温度保持不变,热传导过程已处于稳定状态即沿热传导方向,单位时间内通过单位面积的热量是常熟
3. 玻璃材料均匀,热传导系数是常数

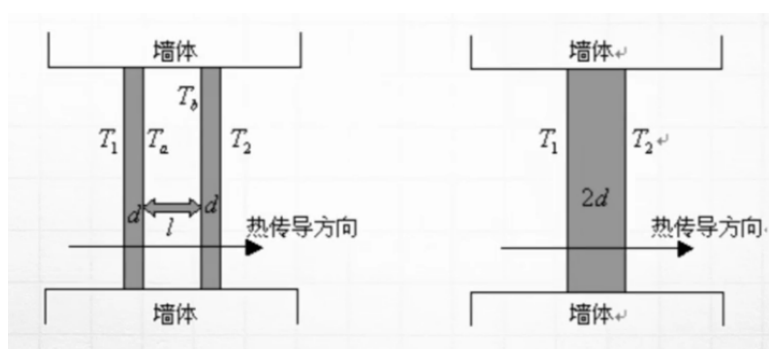


图 3.10: 玻璃热传导示意图

3.7.4 模型建立

热力学传导定律

厚度为 d 的均匀介质,两侧温度差为 ΔT ,则单位时间由温度高的一侧向温度低的一侧通过单位面积的热量 Q 与温度差 ΔT 成正比,与介质的厚度 d 成反比,即

$$Q = k \frac{\Delta T}{d} \quad (3.6)$$

其中, k 为热传导系数。

双层玻璃的热量流失

由热传导方程遵从的物理定律得知

$$Q = k_1 \frac{T_1 - T_a}{d} = k_2 \frac{T_a - T_b}{l} = k_1 \frac{T_b - T_2}{d} \quad (3.7)$$

由 (3.7) 式, 可知

$$T_1 - T_a = T_b - T_2 = \frac{dQ}{k_1}, T_a - T_b = \frac{lQ}{k_2} \quad (3.8)$$

从而得到

$$T_1 - T_2 = (T_1 - T_a) + (T_a - T_b) + (T_b - T_2) = \left(\frac{2d}{k_1} + \frac{l}{k_2}\right)Q \quad (3.9)$$

由此可知

$$Q = \frac{k_1(T_1 - T_2)}{d(s+2)}, s = h \frac{k_1}{k_2}, h = \frac{l}{d} \quad (3.10)$$

单层玻璃的热量流失

对于厚度为 $2d$ 的单层玻璃窗户, 其热量流失为

$$Q' = k_1 \frac{T_1 - T_2}{2d} \quad (3.11)$$

单层玻璃窗和双层玻璃窗热量流失比较

由 (3.7) 式和 (3.11) 式, 可知:

$$\frac{Q}{Q'} = \frac{2}{s+2} < 1 \quad (3.12)$$

由此可见, 双层玻璃窗热量流失一定是小于单层玻璃窗的。

已知不流通、干燥空气的热传导系数是 $k_2 = 2.5 \times 10^{-4}(\text{J}/\text{cm} \cdot \text{s} \cdot \text{度})$, 常用玻璃的热传导系数是 $k_1 = 4 \times 10^{-3} - 8 \times 10^{-3}(\text{J}/\text{cm} \cdot \text{s} \cdot \text{度})$, 于是 $\frac{k_1}{k_2} \in [16, 32]$ 。

在分析双层玻璃窗比单层玻璃窗可减少多少热量损失时, 我们做最保守的估计, 即取 $\frac{k_1}{k_2}$ 的最小值 16, 由 (3.7) 式和 (3.11) 式可得:

$$\frac{Q}{Q'} = \frac{1}{8h+1}, h = \frac{l}{d} \quad (3.13)$$

3.7.5 模型分析与求解

比值 $\frac{Q}{Q'} = (8h+1)^{-1}$ 反映了双层玻璃窗在减少热量损失的功效, 它只与 $h = \frac{l}{d}$ 有关。

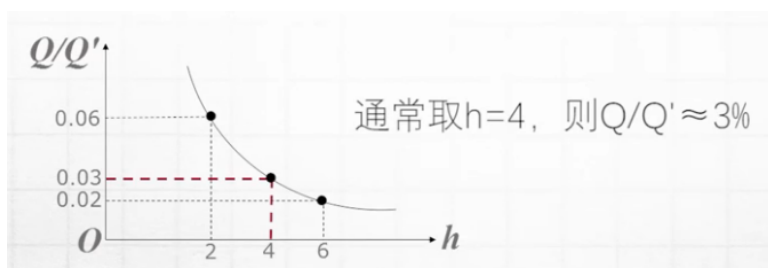


图 3.11: 热量损失比重示意图

通过描述曲线, 我们可以观察到 $h = 4$ 时, $\frac{Q}{Q'} \approx 3\%$, 即 $Q \approx 3\%Q'$, 也就是说双层玻璃窗热量损耗是单层玻璃窗的 3%, 换句话说双层玻璃窗能够阻止 97% 的热量, 要比单层玻璃窗的功效更好。从图中可以看到, 从

$h = 4$ 往后,曲线的走势趋于平缓,减少热量损失的功效不明显,因此实际操作时可以建议操作者,双层玻璃窗之间的厚度和玻璃之间的比值控制在 4 倍即可。

微积分与微分方程方法建模

4.1 Malthus 人口模型

4.1.1 Malthus 人口论

Malthus 指数增长模型

反映人口增加与食物增加速度之间的关系,是刻画单种群模型的典型代表,被世界公认为“马尔萨斯人口论”

马尔萨斯人口理论主要论点

生活资料按 $1, 2, 3, 4, 5, \dots$ 的算术级数增加,而人口是按 $1, 2, 4, 8, 16, \dots$ 几何级数增长,因此生活资料的增加赶不上人口的增长,这是自然的永恒的规律。

重要假设

大规模种群的个体数量是时间的连续可微函数。

设 $x(t)$ 是 t 时刻的人口总数量,种群个体总数的变化主要受出生、死亡、迁入和迁出等因素的影响。假设种群是孤立的,即不考虑迁入迁出因素,那么种群的繁衍发展就不受其他生物种群的影响。

4.1.2 基本概念

人口自然增长率

一定时期内人口自然增加数与同一时期平均人口之比,即:

$$\text{人口自然增长率} = \frac{\text{一年内人口自然增长率}}{\text{年平均人口数}} \quad (4.1)$$

人口平均增长率

一定时期内人口增加量与期初数和所用时间乘积之比,即:

$$\text{人口平均增长率} = \frac{\text{期末人口总数} - \text{期初人口总数}}{\text{期初人口数} \times \text{所用时间}} \quad (4.2)$$

人口增长率

人口平均增长率在所用时间趋于零时的极限,即:

$$r(t) = \lim_{\Delta t \rightarrow 0} \frac{x(t + \Delta t) - x(t)}{\Delta t \times x(t)} \quad (4.3)$$

其中 $x(t)$ 为 t 时刻的人口总数量。

4.1.3 模型假设

- 假设 1: 人口发展过程比较平稳 (这是建模工作的基本要求);
- 假设 2: 人口数量为时间的连续可微函数, 即人口的取值在整数集合上的离散变量, 不是连续的量。但是由于通常人口数量很庞大, 为了运用微积分工具, 将离散问题做连续化处理;
- 假设 3: 人口增长率是与时间 t 无关的常数 r , 该假设是对欧洲百余年人口数据做统计研究作出的, 是一种近似。

4.1.4 模型建立与求解

根据假设 (3), 人口增长率 r 为与时间无关的常数, 即任意时刻 t , 均有:

$$\lim_{\Delta t \rightarrow 0} \frac{x(t + \Delta t) - x(t)}{\Delta t \times x(t)} = r (r \text{ 为常数}) \quad (4.4)$$

再根据假设 (2), $x(t)$ 为时间的连续可微函数, 可得 $x(t)$ 满足以下微分方程:

$$\begin{cases} \frac{dx}{dt} = rx \\ x(0) = x_0 \end{cases} \quad (4.5)$$

运用分离变量法, 容易求得上述常微分方程的解为

$$x(t) = x_0 e^{rt} \quad (4.6)$$

容易看出, 当 $t \rightarrow \infty$ 时, 人口总数将趋于无穷大, 即

$$\lim_{t \rightarrow \infty} x(t) = \lim_{t \rightarrow \infty} x_0 e^{rt} = \infty \quad (4.7)$$

4.1.5 应用案例

以美国 1790 – 1930 年的人口统计资料为数据依据:

年份	实际人口数	Malthus指数增长模型计算结果	计算误差(%)
1790	3.9	—	—
1800	5.3	—	—
1810	7.2	7.2	0.0
1820	9.6	9.8	2.0
1830	12.9	13.3	3.1
1840	17.1	18.1	5.7
1850	23.2	24.6	5.9
1860	31.4	33.4	6.3
1870	38.6	45.4	17.5
1880	50.2	61.6	22.8
1890	62.9	83.8	33.2
1900	76.0	113.8	49.8
1910	92.0	154.7	68.1
1920	106.5	210.2	97.4
1930	123.2	285.7	131.9

图 4.1: 美国 1790-1930 年人口统计数据 单位:百万

通过这张表,我们可以看出,在短期内的表现 Malthus 人口模型的拟合还是比较吻合的,但是随着时间的推移,长期来看,误差越来越大。为什么会出现这样的误差呢? 我们慢慢来分析。

要想构造 Malthus 人口模型,我们首先就要通过给出的资料来计算人口增长 5 率。假设增长率 r 是常数,我们将 $x(0) = 3.9, x(1) = 5.3$,代入人口指数增长模型中,计算人口增长率 r :

$$r = \frac{\ln x(1) - \ln x(0)}{t} = \frac{\ln 5.3 - \ln 3.9}{10} = \frac{1.66771 - 1.36098}{10} \approx 0.03067 \quad (4.8)$$

结果分析:

种群规模和时间跨度小,符合实际;种群规模和时间跨度大,误差较大。若 $r > 0$,当 $t \rightarrow +\infty$ 时,有 $x(t) \rightarrow +\infty$,因此,该模型不能用于对人口的长期预测。

原因就出在假设 3, r 是常数没有考虑有限的资源对种群的增长会产生阻滞作用。因此如果要更好的预测长期的种群增长情况,我们一定要考虑地球有限的资源会给种群增长带来什么样的影响。因此我们需要对 Malthus 人口模型进行修正。

人口的增加必定会消耗有限的资源,因此有限的资源必定会限制人口的增长,这种现象必将促使人口增长率的下降,也就是说,人口增长率是人口数量的递减函数。我们将据此对 Malthus 模型进行修正,这种修正的结果是我们将得到一个另外的反映人口变动的模型,叫做 Logistic 模型。

4.1.6 Logistic 阻滞增长模型

P.F.Verhulst 引入环境的最大容量常数 X_{max} ,简记为 X_m ,表示自然资源和环境条件下所能容许的最大人口数量,而且为了进一步的考虑资源环境对种群的规模增长所产生的这种阻滞的影响作用,Verhulst 又增加了以下几条假设:

- 假设 4:确定的环境内的资源供给为常数,且对每个个体的分配是均等的,即表明当种群规模(密度)增大时,每个个体食物的平均分配量必然减少,从而导致种群增长率降低。
- 假设 5:种群个体的增长率是 $x(t)$ 的线性减函数,即为:

$$r - r \frac{x(t)}{x_m} = r(1 - \frac{x(t)}{x_m}) (x_m > x(t) > 0) \quad (4.9)$$

当 $x(t) = X_m$ 时,种群规模不再增大, X_m 代表环境所能容许的种群最大数量,得到种群增长的 Verhulst 阻滞增长模型:

$$\begin{cases} \frac{dx}{dt} = r \left[1 - \frac{x(t)}{x_m} \right] x(t) \\ x(0) = x_0 \end{cases} \tag{4.10}$$

其中 $1 - \frac{x(t)}{x_m}$ 叫做 Verhulst 因子, $-\frac{r}{x_m}x^2(t)$ 反映种群密度对种群规模增长的抑制作用,称为密度制约。不考虑密度制约时就是原来的 Malthus 模型

Logistic 模型的解

模型 (4.10) 即为 Logistic 模型,其解为:

$$x(t) = \frac{x_m}{1 + (\frac{x_m}{x_0} - 1)e^{-rt}} \tag{4.11}$$

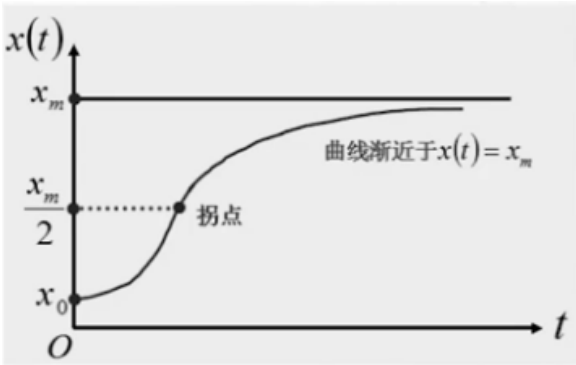


图 4.2: Verhulst 人口模型

人口总量与人口对时间变化率的关系:

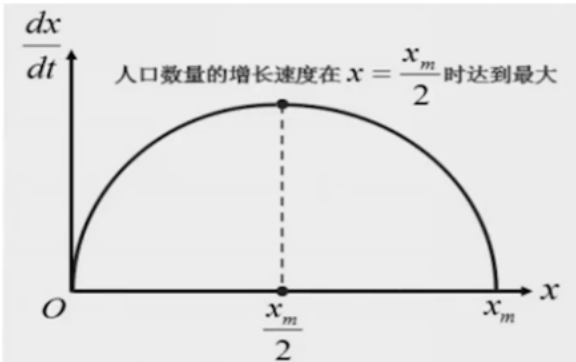


图 4.3: 人口总量与人口对时间变化率的关系

由图可知, $0 - \frac{X_m}{2}$ 时,增长率是单调递增的,当 $x = \frac{X_m}{2}$ 时,增长率达到最大,当 $x = X_m$ 时,增长率为 0。

4.2 细菌的繁殖数学模型

4.2.1 问题的提出

已知某时刻某种细菌的数量,预测在任意时刻 t 这种细菌的数量变化趋势,建立细菌繁殖的数学模型。

4.2.2 模型假设

由实验可知,细菌繁殖的速度应该和培养基是否充足。当培养基充足时,细菌繁殖的速度 V 与当时已有的数量 A_0 成正比,即 $y = kA_0$ ($k > 0$ 为比例常数)

- 假设某种细菌的个数按照指数方式增长,给出如下表:

表 4.1: 细菌繁殖数据表

天数	细菌个数
5	936
10	2190

- 假设在每一个小时间段 $[\frac{(i-1)t}{n}, \frac{it}{n}]$ ($i = 1, 2, \dots, n$) 内细菌的繁殖速度不变,且在各小段时间内只繁殖一次。

按照假设,我们要求的是:

- (1). 开始时细菌的个数可能是多少?
- (2). 若继续以现在速度增长下去,假定细菌无死亡,那么 60 天后细菌的个数应该是多少?

4.2.3 模型建立

首先我们要量化这个数学模型,设细菌总数为 y ,为了计算出 t 时刻细菌的个数,我们需要将时间间隔 $[0, t]$ 分成 N 等分,通过对细菌数量的变化来着手研究细菌的繁衍速度问题。

细菌的繁殖过程可视为连续变化的,在很短的一段时间内数量的变化很小,繁殖的速度可以近似地看作不变。由假设 (2) 在第一段时间 $[0, \frac{t}{n}]$ 内,细菌繁殖的数量为 $\frac{kA_0t}{n}$

第一段时间末细菌的数量为 $A_0(1 + \frac{kt}{n})$;

第二段时间末细菌的数量为 $A_0(1 + \frac{kt}{n})^2$;

依此类推,到最后一段时间末细菌的数量为

$$y = A_0(1 + k\frac{t}{n})^n \quad (4.12)$$

结论 时间间隔无限细分 (即 $n \rightarrow \infty$) 时,可求得精确值,经过时间 t 后细菌的总数是:

$$\begin{aligned} A_0 \lim_{n \rightarrow \infty} \left(1 + k \cdot \frac{t}{n}\right)^n &= A_0 \lim_{n \rightarrow \infty} \left(1 + \frac{kt}{n}\right)^n \\ &= A_0 \lim_{n \rightarrow \infty} \left[\left(1 + \frac{1}{x}\right)^x\right]^{kt} \\ &= A_0 \left[\lim_{x \rightarrow \infty} \left(1 + \frac{1}{x}\right)^x\right]^{kt} \\ &= A_0 e^{kt} \end{aligned} \quad (4.13)$$

至此我们得到细菌繁殖服从生长函数模型:

$$y = A_0 e^{kt} \quad (4.14)$$

其中, k 为繁衍比例系数, t 为时间, A_0 为细菌初始数量。

4.2.4 模型求解

由公式 (4.14) 及题目所给数据表 4.1 得:

$$\begin{cases} 936 = A_0 e^{5k} \\ 2190 = A_0 e^{10k} \end{cases} \quad (4.15)$$

解此方程组,得 $A_0 = 400, k = 0.17$. 即开始时细菌个数为 400 个 按此速度增长下去,则 60 天后细菌个数为:

$$y(60) = 400e^{60 \times 0.17} \approx 10761200(\text{个})$$

4.3 传染病流行的控制模型

通常某种传染病的大面积流行,例如伤风、流感等,最常见的传播方式是带菌患者通过空气、食物和饮食等渠道,把病菌传播给健康人,形成传播锁链。

4.3.1 传统的传染病流行控制模型

假设 t 时刻的病人数目 $x(t)$ 是时间 t 的连续可微函数,每天每个病人有效接触 (足以使被接触者致病) 的人数为常数 k ,考察 t 到 $t + \Delta t$ 发病人数的增加,再假设 $t = t_0$ 时有 x_0 个病人,即得微分方程

$$x(t + \Delta t) - x(t) = k \cdot x(t) \cdot \Delta t \quad (4.16)$$

$$\frac{dx}{dt} = kx(t), x(t_0) = x_0 \quad (4.17)$$

将这个模型对比前面学习过得 Malthus 指数增长模型,可以看出二者非常相似,这个模型得解与 Malthus 指数增长模型的解是完全一样的

模型的结果分析

随着时间 t 的增加,患病人数 $x(t)$ 将无限增加,不符合实际,模型建立失败

原因分析

病人有效接触的人群中既有健康人也有已经发病的人,只有健康人才可以被传染而成为患病的人,建模需要区分人群。

4.3.2 改进的传染病流行控制模型 (SI 模型)

模型假设

- 在疾病传播期内,所考察地区的总人数 N 不变,既不考虑生死,也不考虑迁移。
- 把人群 N 分为易感染者和已感染者两类,简称健康者和病人。在 t 时刻这两类人在总人数中所占的比例分别记作 $x(t)$ 和 $y(t)$ 。
- 每个病人每天有效接触的平均人数是常数 k , k 称为日接触率。假设当病人与健康者进行有效接触时,可使健康者受到感染成为病人。

模型建立

每个病人每天可使 $kx(t)$ 人感染成为病人,病人数为 $N \cdot y(t)$,每天有 $kx(t)Ny(t)$ 个健康者被感染。

- 病人数 $N \cdot y(t)$ 对时间的增长率为

$$\frac{d(Ny)}{dt} = k \cdot x(t) \cdot N \cdot y(t) \quad (4.18)$$

- 初始时刻 $t = 0$,病人的比例记位 $y(0) = y_0$

得到改进的 SI 模型:

$$\begin{cases} \frac{dy}{dt} = k \cdot y(t) \cdot [N - y(t)] \\ y(0) = y_0 \\ N = x(t) + y(t) = 1 \end{cases} \quad (4.19)$$

这依然是一个微分方程的初值问题,其实它就是本章第一节所讲的改进的 Logistic 模型,即阻滞增长模型

模型求解

$$y(t) = \frac{N}{1 + \left(\frac{N}{y_0} - 1\right) \cdot e^{-kt}} \quad (4.20)$$

结果分析

负指数函数 e^{-kt} ($k > 0$) 起到了衰减作用,因此 t 越来越大(发病持续的时间逐渐增加)时, e^{-kt} 将越来越小,因此 $1 + \left(\frac{N}{y_0} - 1\right) \cdot e^{-kt}$ 也将越来越小,从而 $y(t)$ 随 t 的增大单调增加。因此当 $t \rightarrow \infty$ 时, $\lim_{t \rightarrow \infty} y(t) = N$ 。

- 当已感染者达到总人数的一半, 即 $y = 0.5$ 时, $\frac{dy}{dt}$ 达到最大值, 很容易求出这个时刻为 $t_m = k^{-1} \ln\left(\frac{1}{y_0-1}\right)$, 表明此时刻病人增加的速度最快。可以认为这是医院门诊量最大的时刻, 预示着传染病爆发高潮的到来, 需要医疗卫生部门高度关注。
- 由 $t_m = k^{-1} \ln\left(\frac{1}{y_0-1}\right)$ 得知, t_m 与 k 成反比例关系, 原因是模型中的日接触率表示该地区的医疗卫生水平的高低, k 越小, 表明卫生水平越高。因此, 改善保健设施, 提高卫生与防御水平可以推迟传染病爆发高潮的到来时间。
- 由 SI 模型解的表达式 (4.20) 可以看出, $y(t)$ 随 t 单调增加, $t \rightarrow \infty$ 时有 $\lim_{t \rightarrow \infty} y(t) = N$, 这说明所有的人最终都将被传染, 全部成为病人, 不符合实际情况。

4.3.3 病人得到治愈的 SIS 模型

在上一节的结果分析指出, SI 模型解的表达式不符合实际情况。为什么这么说呢? 因为 SI 模型没有考虑到病人还可以被治愈成为健康者, 存在思考上的缺陷。有些传染病被治愈后会使患者的免疫力非常低, 因此可以假设没有免疫性, 于是病人被治愈后成为健康者, 但是健康者还有可能再次被感染, 再次成为病人。这个模型称为 SIS 模型。

模型假设

- 在疾病传播期内, 所考察地区的总人数 N 不变, 既不考虑生死, 也不考虑迁移。
- 把人群 N 分为易感染者和已感染者两类, 简称健康者和病人。在 t 时刻这两类人在总人数种所占的比例分别记作 $x(t)$ 和 $y(t)$ 。
- 每天被治愈的病人数占病人总数的比例为常数 λ , 称为日治愈率。病人治愈后成为仍可被感染的健康者。 $\frac{1}{\lambda}$ 是这种传染病的平均传染期。

模型建立

$$\begin{cases} N \frac{dy}{dt} = k \cdot N \cdot x(t) \cdot y(t) - \lambda \cdot N \cdot y(t) \\ x(t) + y(t) = 1 \end{cases} \quad (4.21)$$

经过整理, 得到:

$$\begin{cases} \frac{dy}{dt} = k \cdot y(t) \cdot [N - y(t)] \\ y(0) = y_0 \end{cases} \quad (4.22)$$

定义 $\sigma = \frac{k}{\lambda}$, 其中 k 是每个病人有效接触 (足以使被接触者致病) 的人数, $\frac{1}{\lambda}$ 是这种传染病的平均传染期。 σ 的含义是整个传染期内的每个病人有效接触的平均人数, 简称接触数。

模型求解分析

利用定义 $\sigma = \frac{k}{\lambda}$, SIS 模型可以改写成:

$$\frac{dy}{dt} = -ky(t) \left[y(t) - \left(1 - \frac{1}{\sigma} \right) \right] \quad (4.23)$$

SIS 模型的结论

接触数 $\sigma = 1$ 是一个阈值 (临界值), 讨论如下:

- $\sigma > 1$ 时, $y(t)$ 的增减性取决于 y_0 的大小, 其极限值 $\lim_{t \rightarrow \infty} y(t) = 1 - \frac{1}{\sigma}$ 随 σ 的增加而增加。
- 当 $\sigma \leq 1$ 时, 病人比例 $y(t)$ 越来越小, 最终趋于零。这是由于传染期内经有效接触从而使健康者变为病人数不超过原来病人数的缘故。

4.4 价格数学模型

4.4.1 价格数学模型的定义

价格数学模型是运用数学的方法对价格形成和价格变动规律所作的描述。最常见的是投入-产出价格数学模型

4.4.2 价格数学模型的分类

- 从价格形成的内在机制上来描述经济变量间的数量关系
- 从经济事物表面现象上探索经济变量间的内在联系
- 组合模型

4.4.3 价格数学模型的分析

一些概念

- 均衡价格是指商品的供给与需求相等时的市场价格
- 实际的市场价格通常不会等于均衡价格, 并且随时间不断变化

假设在某一时刻 t , 某商品的价格与该商品的均衡价格有差别, 此时会存在供需差, 供需差必将促使价格变动

4.4.4 价格数学模型的建立

设某商品在时刻 t 的销售价格为 $p(t)$, 市场对该商品的需求量和它对市场的供给量分别为 $D(p)$ 、 $S(p)$, 它们均为价格 p 的函数。

在 t 时刻价格 $p(t)$ 对时间 t 的变化率 $\frac{dp}{dt}$ 与该商品在同一时刻市场超额需求量 $D(p) - S(p)$ 成正比, 即

$$\frac{dp}{dt} = k [D(p) - S(p)] \quad (k > 0, \text{为比例系数}) \quad (4.24)$$

由经济学原理可知, 商品供给量 $S(p)$ 是价格 p 的单调递增函数, 而市场需求量 $D(p)$ 是价格 p 的单调递减函数。假设

$$S(p) = a + bp, D(p) = c - dp \quad (4.25)$$

其中 a, b, c, d 均为正的常数。当供给量 $S(p)$ 与需求量 $D(p)$ 相等时,对应的价格便是均衡价格 p^* 。
那么当供给量与需求量相等时

$$a + bp = c - dp$$

这表示两条曲线 $S(p) = a + bp$ 与 $D(p) = c - dp$ 的交点恰好为供求平衡时的均衡价格 p^* ,即

$$p^* = (c - a) \div (d + b) \quad (4.26)$$

当商品供不应求时,即 $S(p) < D(p)$, 商品价格要上涨

当商品供过于求时,即 $S(p) > D(p)$, 商品价格要下降

将 (4.25) 式代入 (4.24) 式中,得

$$\frac{dp}{dt} = \alpha(p^* - p) \quad (4.27)$$

其中, $\alpha = k(b + d) > 0$, 另方程 (4.27) 得通解是:

$$p(t) = p^* + Ce^{-\alpha t} \quad (4.28)$$

假设初始价格为 $p(0) = p_0$, 代入上式,得 $C = p_0 - p^*$, 从而得到价格表达式

$$p(t) = p^* + (p_0 - p^*)e^{-\alpha t} \quad (4.29)$$

由 $\alpha > 0$ 时,当 $t \rightarrow \infty$ 时, $e^{-\alpha t} \rightarrow 0$, 因而价格将趋近于均衡价格,即 $p(t) \rightarrow p^*$

4.4.5 结论

随着时间不断延续,市场价格将逐渐趋近于均衡价格 p^*

4.5 湖泊污染减退模型

4.5.1 模型的背景介绍

随着世界经济的迅猛发展,水污染问题日益加剧,水体的污染最突出的问题就是富营养化。根据联合国环境规划署 (UNEP) 的一项调查表明全球范围内的湖泊和水库 30%-40% 都有不同程度的富营养化现象。

湖泊富营养化的特点

- 发展速度快
- 危害程度大
- 治理过程难
- 修复历时长

数学模型可以综合反映系统特征,因此建立该问题的数学模型是针对湖泊污染管理工作种特别有用的手段。

湖泊富营养化研究模型的种类

1. 经验回归模型
2. 营养盐模型
3. 浮游植物动力学模型
4. 生态系统动力学模型

4.5.2

4.6 狼捕食兔子模型

4.7 导弹追踪目标问题

4.8 森林救火模型

线性规划模型

5.1 线性规划模型实例

5.2 线性规划问题

5.3 求解线性规划问题的基本思想

5.4 线性规划问题的几何解释和图解法

5.5 整数线性规划问题

5.6 线性规划的对偶理论

5.7 非对称形式的对偶线性规划问题

5.8 两铁路平板车的装货问题

对策模型

6.1 对策模型的引入

6.2 对策模型的基本理论

6.3 矩阵对策模型

6.4 矩阵对策模型实例分析

6.5 鞍点存在定理

6.6 混合对策模型

决策模型

7.1 决策模型的概念及分类

7.2 风险型决策

7.3 不确定型决策

PART III

第三部分

算法案例

PART IV

第四部分

数据预处理

PART V

第五部分

数据可视化

可视化基础

8.1 绪论

8.2 字体

8.3 颜色

统计图表

9.1 线型图

9.1.1 简单的线型图

</> 代码 9.1: SimpleLine.py

```
1 import matplotlib
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 # Data for plotting
6 t = np.arange(0.0, 2.0, 0.01)
7 s = 1 + np.sin(2 * np.pi * t)
8 fig, ax = plt.subplots()
9 ax.plot(t, s)
10 ax.set(xlabel='time (s)', ylabel='voltage (mV)',
11        title='About as simple as it gets, folks')
12 fig.savefig("./test.png", dpi=600)
13 plt.show()
```

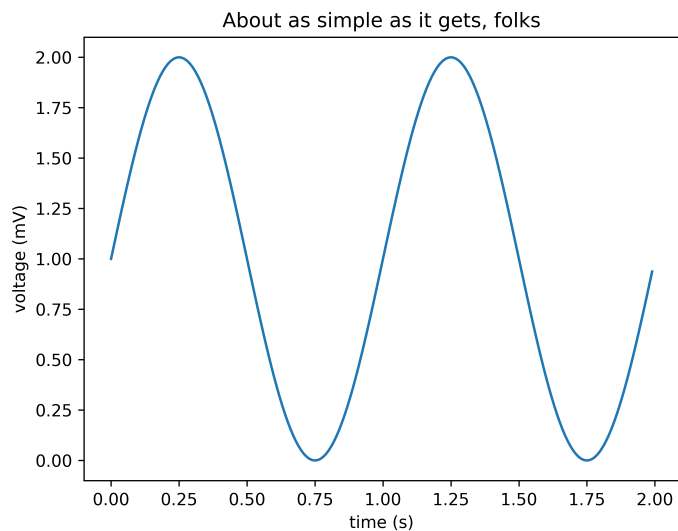


图 9.1: 简单的线型图

9.1.2 自定义虚线样式

</> 代码 9.2: **CustomizeLineStyle.py**

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.linspace(0, 10, 500)
5 y = np.sin(x)
6
7 fig, ax = plt.subplots()
8
9 # Using set_dashes() to modify dashing of an existing line
10 line1, = ax.plot(x, y, label='Using set_dashes()')
11 line1.set_dashes([2, 2, 10, 2]) # 空间长度: 2个点线长, 2个点分隔, 10个点线长, 2个点分隔
12
13 # Using plot(..., dashes=...) to set the dashing when creating a line
14 line2, = ax.plot(x, y - 0.2, dashes=[6, 2, 2, 4], #空间长度: 6个点线长, 2个点分隔, 2个点线长, 4个点分隔
15                 label='Using the dashes parameter')
16
17 ax.legend()
18 plt.savefig("./CustomizeLineStyle.png", dpi=600)
19 plt.show()
```

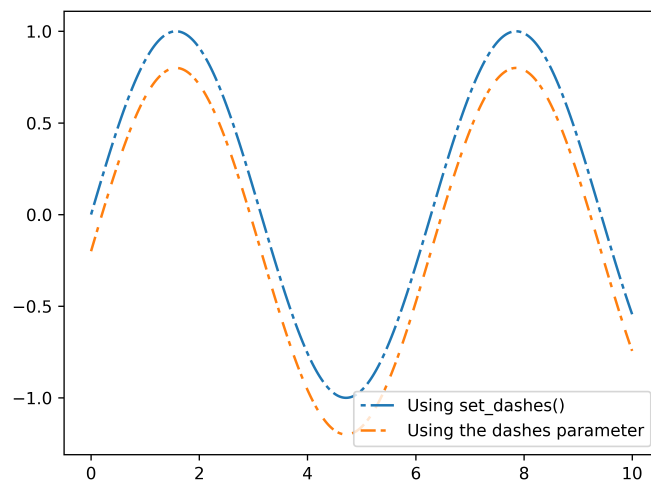


图 9.2: 自定义虚线样式

9.1.3 误差限制选择线

</> 代码 9.3: UpLowErrorBar.py

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4
5 fig = plt.figure()
6 x = np.arange(10)
7 y = 2.5 * np.sin(x / 20 * np.pi)
8 yerr = np.linspace(0.05, 0.2, 10)
9
10 plt.errorbar(x, y + 3, yerr=yerr, label='both limits (default)')
11
12 plt.errorbar(x, y + 2, yerr=yerr, uplims=True, label='uplims=True')
13
14 plt.errorbar(x, y + 1, yerr=yerr, uplims=True, lolims=True,
15             label='uplims=True, lolims=True')
16
17 upperlimits = [True, False] * 5
18 lowerlimits = [False, True] * 5
19 plt.errorbar(x, y, yerr=yerr, uplims=upperlimits, lolims=lowerlimits,
20             label='subsets of uplims and lolims')
21
22 plt.legend(loc='lower right')
23 plt.savefig("../UpLowErrorBar.png", dpi=600)

```

24 plt.show()

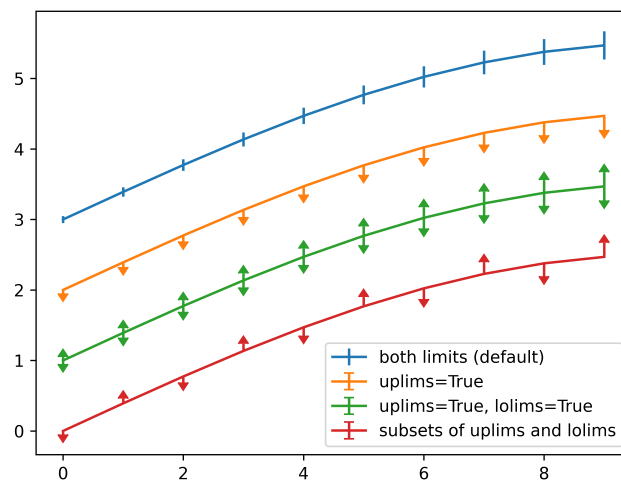


图 9.3: 竖直方向误差限制选择线

</> 代码 9.4: XYErrorBar.py

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 fig = plt.figure()
5 x = np.arange(10) / 10
6 y = (x + 0.1)**2
7
8 upperlimits = [True, False] * 5
9 lowerlimits = [False, True] * 5
10
11 plt.errorbar(x, y, xerr=0.1, xlolims=True, label='xlolims=True')
12 y = (x + 0.1)**3
13
14 plt.errorbar(x + 0.6, y, xerr=0.1, xuplims=upperlimits, xlolims=lowerlimits,
15             label='subsets of xuplims and xlolims')
16
17 y = (x + 0.1)**4
18 plt.errorbar(x + 1.2, y, xerr=0.1, xuplims=True, label='xuplims=True')
19
20 plt.legend()
21 plt.savefig("./XYErrorBar.png", dpi=600)
22 plt.show()
```

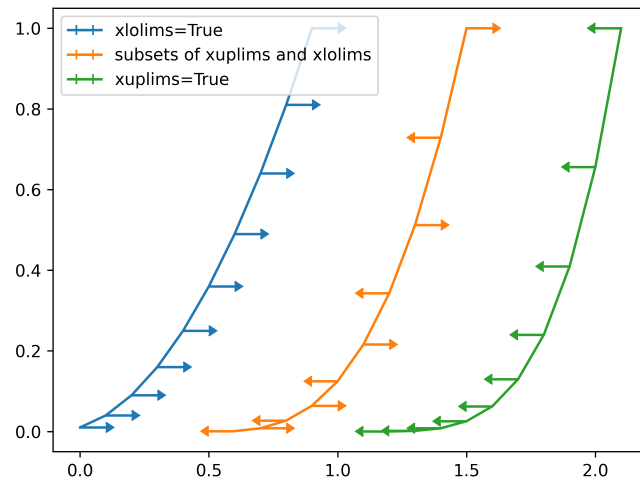


图 9.4: 水平方向误差限制选择线

9.1.4 误差带曲线

</> 代码 9.5: **ErrorBandCurve.py**

```
1 import numpy as np
2 from scipy.interpolate import splprep, splev
3
4 import matplotlib.pyplot as plt
5 from matplotlib.path import Path
6 from matplotlib.patches import PathPatch
7
8 N = 400
9 t = np.linspace(0, 2 * np.pi, N)
10 r = 0.5 + np.cos(t)
11 x, y = r * np.cos(t), r * np.sin(t)
12
13 # Error amplitudes depending on the curve parameter *t*
14 # (actual values are arbitrary and only for illustrative purposes):
15 err = 0.05 * np.sin(2 * t) ** 2 + 0.04 + 0.02 * np.cos(9 * t + 2)
16
17 # calculate normals via derivatives of splines
18 tck, u = splprep([x, y], s=0)
19 dx, dy = splev(u, tck, der=1)
20 l = np.hypot(dx, dy)
21 nx = dy / l
22 ny = -dx / l
23
```

```

24 # end points of errors
25 xp = x + nx * err
26 yp = y + ny * err
27 xn = x - nx * err
28 yn = y - ny * err
29
30 vertices = np.block([[xp, xn[::-1]],
31                      [yp, yn[::-1]]]).T
32 codes = Path.LINETO * np.ones( len(vertices), dtype=Path.code_type)
33 codes[0] = codes[ len(xp)] = Path.MOVETO
34 path = Path(vertices, codes)
35
36 patch = PathPatch(path, facecolor='C0', edgecolor='none', alpha=0.3)
37
38 fig, ax = plt.subplots()
39 ax.plot(x, y)
40 ax.add_patch(patch)
41 plt.savefig("./ErrorBandCurve.png",dpi=600)
42 plt.show()

```

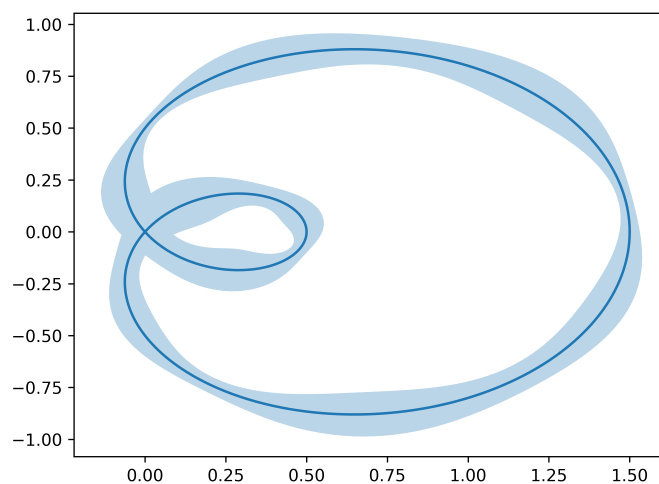


图 9.5: 误差带曲线

9.2 柱状图

9.2.1 简单的柱状图

</> 代码 9.6: **SimpleBar.py**

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4
5 labels = ['G1', 'G2', 'G3', 'G4', 'G5']
6 men_means = [20, 35, 30, 35, 27]
7 width = 0.35      # the width of the bars: can also be len(x) sequence
8
9 fig, ax = plt.subplots()
10
11 ax.bar(labels, men_means, width, label='Men')
12
13 ax.set_ylabel('Scores')
14 ax.set_title('Scores by gender')
15 ax.legend()
16 plt.savefig("./SimpleBar.png",dpi=600)
17 plt.show()

```

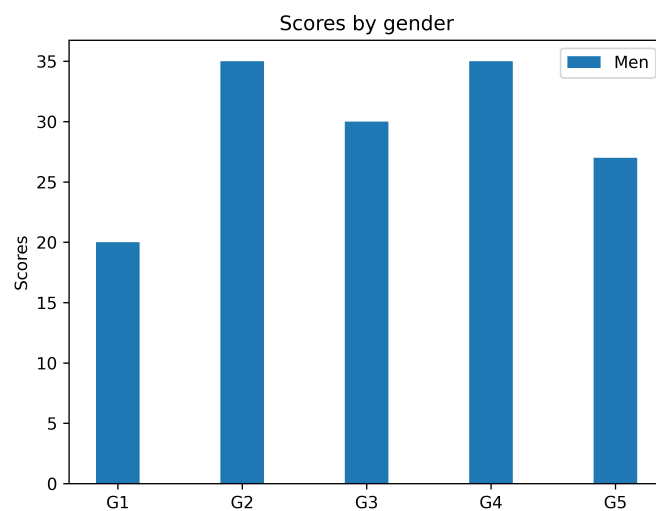


图 9.6: 简单的柱状图

9.2.2 水平条形图

</> 代码 9.7: HorizontalBar.py

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig, ax = plt.subplots()
5

```

```

6 labels = ['G1', 'G2', 'G3', 'G4', 'G5']
7 men_means = [20, 35, 30, 35, 27]
8 y_pos = np.arange( len(labels))
9
10 ax.barh(y_pos, men_means, height=0.4, align='center')
11 ax.set_yticks(y_pos)
12 ax.set_yticklabels(labels)
13 ax.invert_yaxis() # labels read top-to-bottom
14 ax.set_xlabel('Men means')
15 ax.set_title('scores by gender?')
16
17 plt.savefig("HorizontalBar.png",dpi=600)
18 plt.show()

```

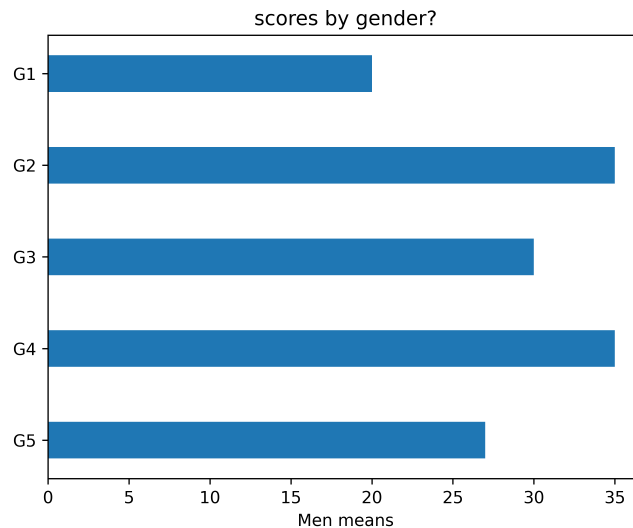


图 9.7: 水平条形图

9.2.3 带有误差线的柱状图

</> 代码 9.8: StackedBar.py

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4
5 labels = ['G1', 'G2', 'G3', 'G4', 'G5']
6 men_means = [20, 35, 30, 35, 27]
7 women_means = [25, 32, 34, 20, 25]
8 men_std = [2, 3, 4, 1, 2]
9 women_std = [3, 5, 2, 3, 3]

```



```

10 width = 0.35
11
12 fig, ax = plt.subplots()
13
14 ax.bar(labels, men_means, width, yerr=men_std, label='Men')
15 ax.bar(labels, women_means, width, yerr=women_std, bottom=men_means,
16       label='Women')
17
18 ax.set_ylabel('Scores')
19 ax.set_title('Scores by group and gender')
20 ax.legend()
21 plt.savefig("./StackedBar.png", dpi=600)
22 plt.show()

```

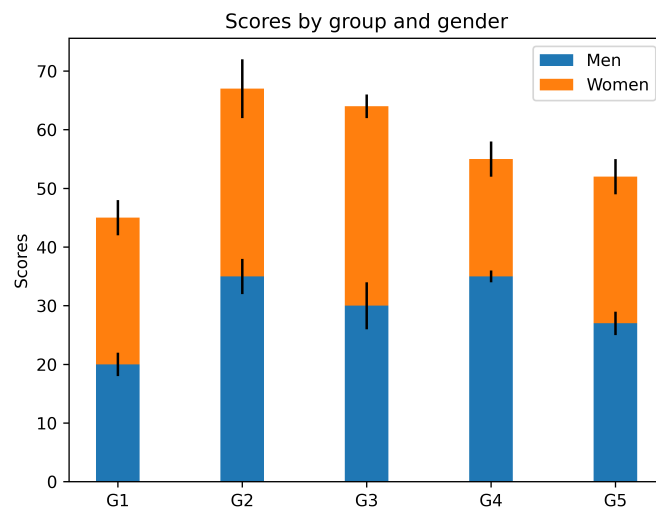


图 9.8: 带误差线的堆叠柱状图

9.2.4 带标签的分组柱状图

</> 代码 9.9: **GroupedBar.py**

```

1 import matplotlib
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5
6 labels = ['G1', 'G2', 'G3', 'G4', 'G5']
7 men_means = [20, 34, 30, 35, 27]
8 women_means = [25, 32, 34, 20, 25]
9

```

```

10 x = np.arange( len(labels)) # the label locations
11 width = 0.35 # the width of the bars
12
13 fig, ax = plt.subplots()
14 rects1 = ax.bar(x - width/2, men_means, width, label='Men')
15 rects2 = ax.bar(x + width/2, women_means, width, label='Women')
16
17 # Add some text for labels, title and custom x-axis tick labels, etc.
18 ax.set_ylabel('Scores')
19 ax.set_title('Scores by group and gender')
20 ax.set_xticks(x)
21 ax.set_xticklabels(labels)
22 ax.legend()
23
24
25 def autolabel(rects):
26     """Attach a text label above each bar in *rects*, displaying its height."""
27     for rect in rects:
28         height = rect.get_height()
29         ax.annotate('{}' . format(height),
30                     xy=(rect.get_x() + rect.get_width() / 2, height),
31                     xytext=(0, 3), # 3 points vertical offset
32                     textcoords="offset points",
33                     ha='center', va='bottom')
34
35
36 autolabel(rects1)
37 autolabel(rects2)
38
39 fig.tight_layout()
40 plt.savefig("./GroupedBar.png",dpi=600)
41 plt.show()

```

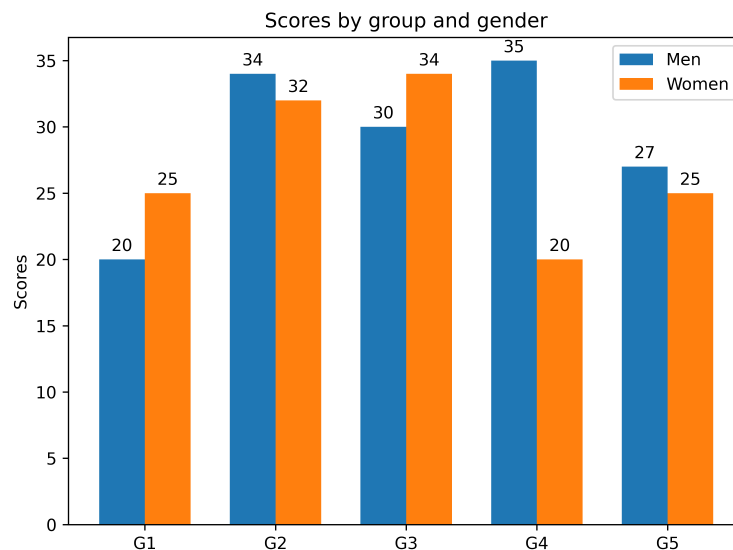


图 9.9: 带标签的分组柱状图

9.2.5 离散分布条形图

</> 代码 9.10: **DiscreteDistributionBar.py**

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4
5 category_names = ['Strongly disagree', 'Disagree',
6                   'Neither agree nor disagree', 'Agree', 'Strongly agree']
7 results = {
8     'Question 1': [10, 15, 17, 32, 26],
9     'Question 2': [26, 22, 29, 10, 13],
10    'Question 3': [35, 37, 7, 2, 19],
11    'Question 4': [32, 11, 9, 15, 33],
12    'Question 5': [21, 29, 5, 5, 40],
13    'Question 6': [8, 19, 5, 30, 38]
14 }
15
16 def survey(results, category_names):
17     labels = list(results.keys())
18     data = np.array( list(results.values()))
19     data_cum = data.cumsum(axis=1)
20     category_colors = plt.get_cmap('RdYlGn')(
21         np.linspace(0.15, 0.85, data.shape[1]))
22
23     fig, ax = plt.subplots(figsize=(9.2, 5))

```

```

24 ax.invert_yaxis()
25 ax.xaxis.set_visible(False)
26 ax.set_xlim(0, np. sum(data, axis=1). max())
27
28 for i, (colname, color) in enumerate( zip(category_names, category_colors)):
29     widths = data[:, i]
30     starts = data_cum[:, i] - widths
31     ax.barh(labels, widths, left=starts, height=0.5,
32             label=colname, color=color)
33     xcenters = starts + widths / 2
34
35     r, g, b, _ = color
36     text_color = 'white' if r * g * b < 0.5 else 'darkgrey'
37     for y, (x, c) in enumerate( zip(xcenters, widths)):
38         ax.text(x, y, str( int(c)), ha='center', va='center',
39                color=text_color)
40 ax.legend(ncol= len(category_names), bbox_to_anchor=(0, 1),
41          loc='lower left', fontsize='small')
42
43 return fig, ax
44
45 survey(results, category_names)
46 plt.savefig("./DiscreteDistributionBar.png", dpi=600)
47 plt.show()

```

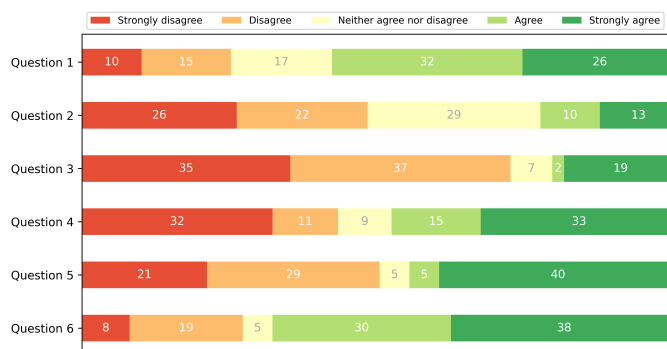


图 9.10: 离散分布条形图

9.2.6 破碎的条形图

</> 代码 9.11: **BrokenBar.py**

```

1 import matplotlib.pyplot as plt

```

```

2
3 fig, ax = plt.subplots()
4 ax.broken_barh([(110, 30), (150, 10)], (10, 9), facecolors='tab:blue')
5 ax.broken_barh([(10, 50), (100, 20), (130, 10)], (20, 9),
6                 facecolors=('tab:orange', 'tab:green', 'tab:red'))
7 ax.set_ylim(5, 35)
8 ax.set_xlim(0, 200)
9 ax.set_xlabel('seconds since start')
10 ax.set_yticks([15, 25])
11 ax.set_yticklabels(['Bill', 'Jim'])
12 ax.grid(True)
13 ax.annotate('race interrupted', (61, 25),
14            xytext=(0.8, 0.9), textcoords='axes fraction',
15            arrowprops=dict(facecolor='black', shrink=0.05),
16            fontsize=16,
17            horizontalalignment='right', verticalalignment='top')
18 plt.savefig("./BrokenBar.png",dpi=600)
19 plt.show()

```

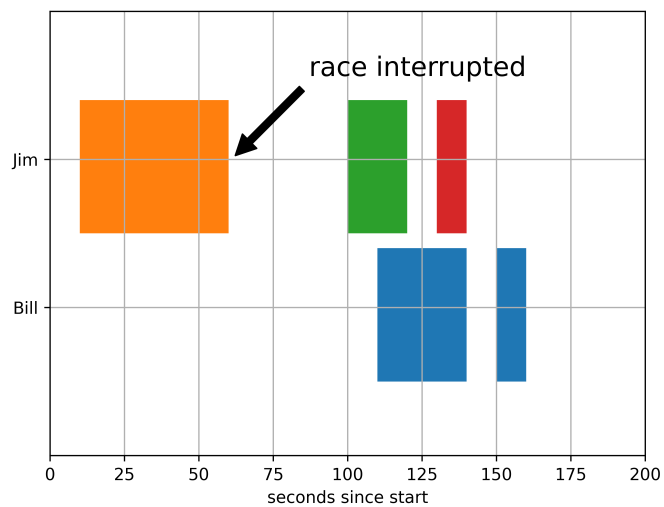


图 9.11: 破碎的条形图

9.3 散点图

9.4 饼图

9.5 地图

9.6 热力图

9.7 表格

9.8 形象化描述

9.9 地图

9.10 热力图

数值参数

PART VI

第六部分

LATEX 论文写作

Angular CLI

```
ng new appname --style scss --skip- install
```

