第 5 章　数据与算法

作者简介

吴及，清华大学电子工程系副系主任，长聘教授，博士生导师。清华大学精准医学研究院临床大数据中心共同主任。

1996 年和 2001 年在清华大学电子工程系获得工学学士和博士学位，2013—2015 年在美国佐治亚理工学院担任访问学者。主要从事人工智能、机器学习、自然语言处理、模式识别、数据挖掘等领域的研究工作。从 2006 起担任清华-讯飞联合研究中心主任。现在为 IEEE 高级会员，中国语音产业联盟技术工作组组长，认知智能国家重点实验室学术委员会委员，口腔数字化医疗技术和材料国家工程实验室第二届技术委员会委员，中国计算机学会语音对话与听觉专业组委员。还担任 2018—2022 年教育部电信类专业教学指导委员会副秘书长。

承担国家重点研发计划、863、国家自然科学基金、工信部电子发展基金等多项国家科研项目。参加的项目“智能语音交互关键技术及应用开发平台”于 2011 年获国家科技进步二等奖。负责的项目“面向海量语音数据的识别、检索和内容分析技术及其应用”获 2014 年度北京市科学技术奖一等奖。已在 *Nature Communications*，*IEEE Trans. on ASLP*，*AAAI*，*ACL* 等重要学术期刊和学术会议上发表论文约 130 篇。

5.1 数据

5.1.1 什么是数据

“数”是人们用来表示事物的量的基本数学概念。在人类发展的历史上,这种抽象的“数”的概念是从具体事物中逐步获得和建立起来的。例如,“一个苹果”“二个橘子”“三个香蕉”描述的是具体的事物,而“一”“二”“三”则是与具体事物无关的抽象的“数”。另一个相关的概念是“数字”。数字是人们用来计数的符号,如现在人们常用的阿拉伯数字“1”“2”“3”,又如中文的数字“一”“二”“三”和罗马数字“Ⅰ”“Ⅱ”“Ⅲ”。而我们在这里要讨论的“数据”,则是一个范围大得多的概念。

数据是客观事物的符号表示,往往是通过对客观事物的观察得到的未经加工的原始素材,是包含知识和信息的原始材料。在今天的信息社会中,数据可以说无处不在,其表现形式也是多种多样,例如:

(1) 文字和符号。文字和符号不仅普遍存在于书籍、报纸等传统的纸质媒介上,也广泛存在于计算机、手机、平板电脑等电子设备上;既包括今天人们使用的各种文字符号,也包括从远古时代遗存下来的象形文字和甲骨文等。

(2) 多媒体数据。计算机的图形界面、广播、电视、电影、数码相机(DC)和数码摄像机(DV),使得我们身处丰富多彩的多媒体时代。多媒体数据的采集,保存和播放已经非常方便;图像、音频、视频等各种媒体数据在我们的日常生活中随处可见。

(3) 通信信号。电信号和电磁波已经成为人类社会信息最方便快捷的传输方式,这些用于通信、控制和信息传输的电话信号、导航信号、手机信号、广播信号,无论是在发送端还是在接收端都是数据。

(4) 传感器采集的数据。通过各种各样的温度传感器、压力传感器,以及 CT、B 超、声呐等设备,人们可以采集到各种各样能够描述客观事物的数据。

(5) 社会性数据。人类社会生活的方方面面同样需要大量数据来描述,如社会普查数据、人口统计数据和民意调查数据等,著名的如美国总统大选期间盖洛普所做的候选人支持率的民意测验;也包括和我们日常生活有紧密联系的经济运行数据,如物价、收入等。随着社交网络的发展和普及,人们之间通过互联网和移动互联网的交互行为也成为重要的海量数据来源。

因此,可以很清楚地看到对数据的掌握和处理是当今社会的一个基本问题,在科研活动、经济活动、文化活动和政治活动中,我们随时都会面对各种各样的数据。数据和对数据的处理与我们每个人都息息相关。

我们在这里讨论的数据,进一步特指能够输入到计算机并被计算机所处理的数据。

5.1.2　数据处理技术

数据处理技术包括数据的获取、数据的存储、数据的传输，以及针对数据的计算。

数据是客观事物的表示和描述。人具有很强的获取数据的能力，如人对客观事物的观察、社会普查等；数据获取也可以通过多种多样的设备，如温度和压力等各种传感器，万用表和光谱仪等各种测量仪器，照相机和摄像机等图像视频采集设备，麦克风和录音机等声音采集设备，雷达接收机和卫星接收机等各种信号接收设备，等等。

传统的数据存储主要依靠纸质媒介，如书籍、报表和纸质文件等，典型的模拟存储介质有胶片和磁带。随着数字技术的发展，数字存储介质已经成为主流。从大型的磁盘存储系统，到容量越来越大的计算机硬盘，再到便携的移动硬盘、U 盘、光盘和闪存卡；存储容量不断增大，而且价格越来越便宜。

语言交流和书信曾是人类历史上数据传输和信息交互的主要手段。电磁波和电信号的发现和利用，造就了电话、电报等快捷的数据传输方式。互联网、移动通信以及 USB 和 DisplayPort 等高速率数据传输技术的发展，使数据传输的快速、高效和方便达到了前所未有的程度。

面向数据的计算涵盖了对数据的分析、管理和利用。其中既包括以处理器性能为代表的计算能力，又包括对数据进行处理以实现信息抽取和知识发现的技术方法。

随着信息技术的飞速发展，人类在数据采集、数据存储和数据传输方面的能力得到了巨大的发展。我们都知道，二进制是数字计算机的基础，计算机存储容量的基本单位是字节(Byte)，每一字节包含 8 个二进制位。为了描述不同规模的数据，人们定义了一系列的数据计量单位：

Bytes→ Kilobyte(2^{10}Bytes) → Megabyte(2^{20}Bytes) → Gigabyte(2^{30}Bytes) → Terabyte(2^{40}Bytes) → Petabyte(2^{50}Bytes) → Exabyte(2^{60}Bytes) → Zettabyte(2^{70}Bytes) → Yottabyte(2^{80}Bytes)

其中我们比较熟悉的有千字节(KB)、兆字节(MB)和吉字节(GB)。我们甚至难以想象更大的数据量单位意味着什么。美国国会图书馆所有藏书约为 10TB。按照 2001 年的数据估算，美国国家航空航天局地球观测系统(Earth Observing System)三年的数据总和约为 1PB[1]。据称 1ZB 大概相当于全世界所有海滩上的沙子总和，而 1YB 大概相当 7000 人体内的原子数总和[2]。如果以 1MB/min 的速度不间断播放 MP3 格式的歌曲，1ZB 存储的歌曲可以让人听上 19 亿年。

根据 IDC 的统计和预测，2007 年全球数据量约为 161EB；2008 年激增到 487EB；金融危机的 2009 年，全球数据量达到 0.8ZB，增长 62%；2010 年进一步增长到 1.2ZB，约为 2007 年的 8 倍；而到 2020 年，这一数字将达到 35ZB。人类所拥有的数据量还在以更快的速度增长，2010 年 3 月，视频网站 YouTube 宣布每分钟就会有 24 小时的视频被上传，而到了 2010 年 11 月，每分钟上传至 YouTube 的视频长度已达 35 小时。根据 YouTube 产品管理负责人的计算："如果美国三大电视网每天播放 24 小时，一周 7 天，一年 365 天

不间断播放 60 年,那么这些视频内容才与 YouTube 每 30 天增加的内容一样多。”而到了 2014 年,每分钟上传的视频长度已经超过 300 小时,YouTube 上已经有了超过 1 万亿个视频。

根据《2018 年全球数字报告》,到 2018 年年初:

- 全世界互联网用户已经超过 40 亿,2017 年增加超过 2.5 亿;
- 全球活跃的社交媒体用户将近 32 亿,2017 年平均每天新增 100 万用户;
- 全世界有 51 亿的手机用户,移动互联网用户超过 37 亿;
- 占据社交领域头把交椅的 Facebook 的月活用户达到 21.7 亿;
- 微信的月活用户为 10 亿,排名全球第五,每天发出消息 380 亿条;
- YouTube 每分钟上传视频 300 小时,每天观看次数达到 50 亿次;
- 互联网用户每日平均在线 6 小时,互联网已经占据了人们清醒时间的 1/3,人类一年的总在线时长已突破 10 亿年。

很显然,人类获取和生产数据的能力已经十分惊人,我们已经生活在一个“数据爆炸”的时代。为了应对数据爆炸性的增长,最近 20 年以来,人类在数据存储能力上的进步极为迅速。20 年前,我们使用的个人计算机往往只有 40MB 的硬盘,数据交换依靠 720KB 的 5 英寸软盘和 1.44MB 的 3.5 英寸软盘。今天的个人计算机,1TB 的硬盘几乎成为标准配置,用于数据交换的移动存储设备也是 500GB 以上的移动硬盘和 8GB 以上的 U 盘。个人数据存储产品的容量 20 年间增大了成千上万倍。数据中心更是从萌芽走向成熟,当今的数据中心的存储规模往往能达到 PB 量级,并且在能效、安全、接入和管理等方面有了越来越完善的考虑和设计。

数据传输技术的发展同样迅猛。依赖于移动存储介质的数据交换,除了存储量增大以外,传输速率也飞速增长。传统的 1.44MB 软盘的传输速率为 62.5KB/s,计算机串口的传输速率为 14.4KB/s。CD 光盘的读取速度为 7.5MB/s,DVD 光盘的读取速度为 16.6MB/s。现在得到广泛应用的 USB 2.0 理论传输速率为 60MB/s,实际传输速率能达到 20～30MB/s;2008 年年底发布的 USB 3.0 标准理论传输速率已经达到了 600MB/s,而 2013 年 12 月发布的 USB Type-C 的最大数据传输速率达到了 10Gb/s。因此基于移动存储介质的传输速率在 20 年间也得到数百倍乃至数千倍的提升。互联网的发展使得数据传输不再受到地理位置的约束。早期 Modem 拨号上网的速率为 56Kb/s;现在 ADSL 接入的下行速率可以达到 1Mb/s,目前家庭常用的速率为 512Kb～2Mb/s。而局域网的传输速率可以达到 100Mb/s 甚至 1000Mb/s。而基于无线传输的 4G 移动互联网理论上也可以提供高达 100Mb/s 的下行速率。随着互联网,特别是移动互联网的发展,人们将继续向随时随地快速传输数据的目标前进。

数据的计算需要强大的处理能力,其中处理器和随机存储器起着至关重要的作用。20 年前的个人计算机,Intel 80386 的典型配置是 33MHz 主频和 1MB 内存;Intel Core2 的典型配置是主频 3GHz,64KB 的一级缓存(L1 Cache)和 6MB 的二级缓存(L2 Cache);而 Intel Core-i 系列进一步引入了三级缓存,并实现了 CPU 与图形处理单元 GPU 的整合封装。因此今天的处理器,计算能力已经不可同日而语。然而单处理器计算能力的提高仍然远远不能满足数据处理的需要,因此各种并行计算技术风起云涌,从多核处理器、并

行程序设计技术如 OpenMP、MPI，到分布式计算、网格计算和声名显赫的云计算，给数据计算提供了前所未有的强大能力。随着深度学习技术的迅猛发展，原来用于图形处理的 GPU 一跃成为人工智能第三波浪潮中最为重要的推动力量之一。

然而数据的计算除了计算能力之外，同样甚至更为重要的是计算的方法，因此近年来以机器学习、数据挖掘为代表的海量数据处理技术得到了普遍重视和迅速发展。而 2006 年提出的深度神经网络，在随后十多年中在语音、图像、机器翻译和不同扩展的更多领域得到了成功应用，直接推动了人工智能第三波浪潮的汹涌而来。

数据的重要性导致了数据采集能力和生产速度不断提高，爆炸性的数据增长推动了数据存储、数据传输和数据计算能力与计算方法的飞速发展，而数据处理技术的发展进一步提升了数据的可用性和重要性，这就形成了一个正反馈，从而促使数据和数据处理相关的领域成为当今社会最有活力的发展方向。同时应该看到，相比于数据的采集、传输和存储，数据的计算能使人们更充分更有效地发挥数据的价值，因此我们有理由期待数据的计算有着更为广阔的发展空间。

5.1.3　数据的重要性

我们在 5.1.2 节中将数据的重要性作为论述的基础，在这一节我们试图去回答数据为什么是重要的。数据是客观事物的符号表示，人类通过观察获得数据，通过数据积累和分析去获取知识，人类发展的历史同时也是一个数据积累和知识增长的过程。

远古时代，我们的祖先就为了生存去观察和适应环境，但在很长时间里人们缺乏描述客观事物的有效手段，观察的积累主要依靠个体进行。人们的知识主要来自直接的生活经验，信息的交流和保存非常困难，知识的积累缓慢而艰难。随着语言、文字的出现，以及保存文字的介质（如泥版和纸）的发明，人们对客观事物的观察和认识能够以数据的形式保存、积累和交流，人类文明也进入了新的发展阶段。然而在很长的时期，学习、掌握、传承和发展新的知识仍然主要依靠人类社会中的某些特殊群体，社会整体仍然处于愚昧落后的状态。17 世纪义务教育开始出现，18 世纪印刷术开始普及，书籍和报纸逐渐变得普遍起来，19 世纪电报和电话的诞生，20 世纪计算机和互联网的崛起，这些进步对于人类文明具有重要意义，它们推动了数据的产生、存储和传输，直接促进了信息和知识的分享。人们越来越多的知识来自于对数据的处理，而不是直接的生活经验。一个非常典型的例子是：16 世纪的丹麦天文学家第谷穷毕生精力，积累了大量准确细致的天文观测数据，在这些数据的基础上，他的学生、德国天体物理学家开普勒提出了著名的行星运动三大定律。可以说，正是第谷长期积累的精确数据加上开普勒的创新思想，对原始观测数据的尊重和有效利用，共同铸就了这一辉煌成就。

随着信息技术的发展和对社会的巨大影响，恐怕已经没有人会置疑信息的重要性，然而信息是不能单独存在的，数据是信息的载体。同样，数据尽管非常重要，但需要人们具有从数据中获取信息的能力，才可能有效地利用数据，真正发挥数据的作用。一方面，人们在传输数据以实现信息传递时，需要先验性的知识和约定，才能相互理解；另一方面，

即使同样的数据，不同人从不同的视角也能得到不同的信息和知识。有人说读书实际上是一个二次创作的过程，鲁迅先生在评价古典名著《红楼梦》时写道："经学家看见《易》，道学家看见淫，才子看见缠绵，革命家看见排满，流言家看见宫闱秘事……"，恰好说明了这种现象。

人们从对自然和社会的观察中获得数据，在这些数据中包含了人们知道和不知道的各种知识和规律，而这些未知的部分就是科学研究试图去发现的。

在以往使用数据的过程中，人们往往是通过数据的观察产生假设，然后用数据进行验证。但在数据爆炸性增长的今天，很多时候数据中蕴含的规律很难通过观察直接获得，因此从海量数据自动发现其中蕴含规律和知识的数据挖掘技术开始崭露头角。

通过对客观事物的观察获得数据，依靠存储和传输积累数据，针对数据进行分析和思考总结规律，使用更多数据进行验证，这是人类认知的基本途径。因此数据和数据处理极为重要，数据的积累和数据处理能力的提高是人类文明发展的重要阶梯。

5.2 数学模型

5.2.1 什么是数学模型

人们在长期的生产实践中遇到各种各样的问题，现实中的问题由于涉及很多因素而变得十分复杂，并不容易解决。对这些问题进行抽象和简化，保留主要矛盾，摒弃影响较小的次要因素就成为非常重要的思想方法。杠杆是人类最早使用的简单机械之一，早在公元前，东西方文明都已经认识到了"二重物平衡时，它们离支点的距离与重量呈反比"的杠杆原理。阿基米德曾经说过一句流传千古的名言："给我一个支点，我可以撬动地球"。但要知道"动力×动力臂＝阻力×阻力臂"的平衡条件是基于两个假设的，杠杆是无重量的刚体，而支点也要求是刚体。但正是这些并不完美的模型，推动了人们对各种自然现象和社会现象的理解，并且成为人类工程技术成就的重要基石。

数学模型是对于客观世界的现实对象，根据其内在规律，经过简化得到的数学结构。数据是客观事物的符号表示，但如果不能描述数据之间的内在联系，孤立的数据本身可能是片面的、冗余的，甚至是相互冲突的。数学模型由于抓住了客观事物的内在规律和主要矛盾，因而成为数据处理技术中的重要环节。针对研究对象，采集数据，根据对事物的认识和拥有的数据建立数学模型，形式化定义问题，然后设计和优化算法进行求解，再进行测试验证和反馈完善，已经成为科学研究的一般化方法。因此在我们讨论数据和算法的关系和相互作用时，也必然会涉及数学模型。

5.2.2　数学模型的种类

由于面对的问题纷繁复杂，数学模型也就必然多种多样。

在对电路进行分析时，根据欧姆定律和基尔霍夫定律就可以得到一组线性方程，求解这组线性方程就可以得到电路中的电流和电压参数。线性方程组就是重要而基本的数学模型。当方程数目超过变量数目（称为“超定”）时，实验数据的曲线拟合就是具有普遍性的超定方程求解问题。当方程数目少于变量数目（称为“欠定”）时，在实际工作中经常会转换成为线性规划问题进行求解。

客观世界中很多物理量之间的关系不是线性的，例如物体运动距离和加速度之间的关系，两个天体之间的万有引力与它们之间距离的关系等，非线性方程和非线性方程组就是描述这些物理规律的数学模型。

在对天体运动，摆线运动轨迹和热传导等问题的研究中，人们发展出了常微分方程和偏微分方程。微分方程模型已经成为一类重要的数学模型，用于描述电磁场的著名的麦克斯韦方程组就是由 4 个偏微分方程组成的。微分方程离散化就是差分方程，随着数字化的发展，特别是计算机的广泛应用，差分方程模型已经变得越来越重要。

概率论最早的起源是人们对赌博的研究及其内在规律的思考，概率和统计现在已经成为数学最重要的分支之一。人们可以用概率模型来描述彩票、保险、天气预报等社会生活中的很多问题。统计模型的经典例子包括用于进行人口预测的阻滞增长模型（Logistic model），用于描述无后效随机过程的马尔可夫模型，用于描述随机服务的排队论模型等。

集合上的序关系是数学中最基本的抽象结构之一，在二元关系的基础上，数据元素可以形成线性结构、树结构和图结构这些基本数据结构，如图 5.2.1 所示。对于集合 M，笛卡儿积 $M\times M$ 的子集 R 称为 M 上的二元关系：$R=\{(a,b),a,b\in M\}$。二元关系 R 中的每一个元素 (a,b)，a 称为前件，也是 b 的前驱；b 称为后件，也是 a 的后继。对于线性结构，除了第一个元素外，每个元素都有唯一的前驱；除了最后一个元素外，每个元素都有唯一的后继。对于树结构，有唯一的称为根的元素，除了根元素外，每个元素都有唯一的前驱；所有元素都可以有多个后继。对于图结构，每个元素都可以有多个前驱和多个后继。这些数据结构是很重要的数学模型，可以用于描述很多实际问题，如学生信息、家族谱系和互联网。很容易想象，线性结构最简单，但描述能力相对较弱；而图结构最复杂，同时描述能力也最强。由于这些数据结构描述的数据对象都是不连续的，因此称为离散模型。

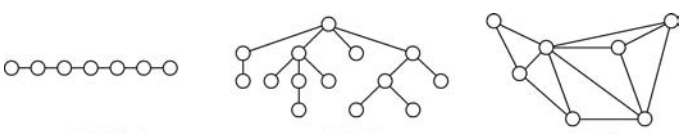

图 5.2.1　基本数据结构

我们还会遇到其他的一些模型，例如我们在研究经典力学中的直线运动、弹性碰撞、机械振动等问题时，一般都把物体视为刚体或者质点，这就是刚体模型和质点模型。这一类的模型由于只是出于特定的目的对客观事物加以简化得到的模型，一般称之为物理模型，我们还需要引入数学语言对其进行形式化描述才能得到数学模型。

当我们采用数学模型来描述客观世界时，客观世界的问题也就转换为数学模型上的问题。通过对数学模型上问题的求解，就可能得到客观世界中原始问题的解。当然这个过程并不能保证解的正确性，仍然需要通过客观世界的验证。如果解不能反映客观真实，那么既有可能是由于数学模型上问题求解不正确导致的，也可能是由于采用的数学模型对客观世界描述不够准确所导致。

5.2.3 数学模型与计算机

数学问题的一个重要来源，就是人们在社会生活和生产实际中遇到的各种问题。人类最初的数学大致来自于土地丈量、天文历法、工程建筑和贸易的实际需要。16 世纪以后，随着航海、天文学和地理学的发展，引发了一系列在理论和实践上都非常重要的课题，例如经纬度的测量，时间的准确测定，物体运动的瞬时速度，炮弹的最大射程，曲线的线长和面积，行星的运动描述，热传导规律等。这些问题导致很多重要数学分支的诞生：解析几何、微积分、级数、微分方程等。从社会实践中的问题抽象出数学问题的过程，实际上就是数学模型建立的过程。数学问题求解上的进展又会在社会实践中得到应用，从而提高人们的实践能力。因此，数学建模是实际应用和抽象数学之间的桥梁。

对很多复杂的数学问题，人们经过研究发现有些问题是没有解析解的，如五次以上高次方程；有些问题能够找到精确或者近似的解法，但在实际应用中代价过大。

第二次世界大战期间，研制和开发新型大炮和导弹的需求十分迫切，为此美国陆军军械部在马里兰州的阿伯丁设立了“弹道研究实验室”。宾夕法尼亚大学莫尔学院电子系和阿伯丁弹道研究实验室共同负责为陆军每天提供 6 张火力表。这项任务非常困难和紧迫，因为每张表都要计算几百条弹道，而每条弹道的数学模型是一组非常复杂的非线性方程组。这些方程组是无法求出准确解的，只能用数值方法近似地进行计算。而一个熟练的计算员计算一条飞行时间 60 秒的弹道要花 20 小时。尽管他们改进了微分分析仪，聘用了 200 多名计算员，一张火力表仍要计算两三个月。

这么慢的速度显然不能满足军方和战争的需求，这就成为电子计算机诞生的最重要驱动力。从 1943 年开始研制，到 1946 年 2 月，第一台得到实际应用的电子计算机 ENIAC 诞生了。ENIAC 的体积约为 90 平方米，占地 170 平方米，总重量达到 30 吨。它拥有电子管 18000 个，继电器 1500 个，耗电 150 千瓦，每秒运算 5000 次。尽管 ENIAC 体积庞大，耗电惊人，运算速度也不过每秒几千次，但它已经比当时的计算装置要快 1000 倍，而且可以按照事先编制好的程序自动执行运算。ENIAC 宣告了一个新时代的开始。[4-6]

图 5.2.2 中正在操作计算机的女士们可以被称为最早的计算机程序员。

对于实际问题求解的需求导致了计算机的出现和不断发展，而计算机的出现给人们

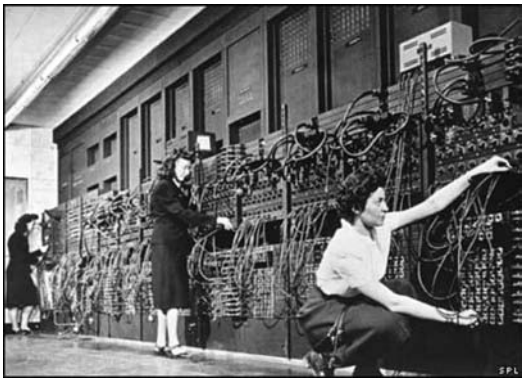

图 5.2.2　第一台通用电子计算机 ENIAC

提供了前所未有的计算能力和存储能力。这不但使人们具备了更有效的计算工具，并且推动了数学建模和计算机算法的迅猛发展。通过数学模型描述客观事物，并解决相应的数学问题成为科学研究和工程实践的有效途径，人们认识世界和改造世界的能力不断得到增强。

5.3　算法

5.3.1　什么是算法

算法描述的是解决特定问题的方法。算法的中文名称出自《周髀算经》；而英文名称 Algorithm 来自于 9 世纪波斯数学家花拉子米(al-Khwarizmi)。算法给出的不是问题的答案，而是描述如何获得答案的过程，因此算法的创意、设计、构造和分析能让人们获得解决问题的有效途径，这要比仅给出某个问题的答案重要得多。

我们在这里讨论的算法是由一系列确定性的步骤组成的，因此更为严格地说，算法是用以求解问题的有限长度的指令序列，或者说算法是特定问题的程序化解决方案。一个非常古老而经典的算法是用于求两个整数最大公因子的欧几里得算法，也称辗转相除法。

对于问题：求整数 m 和 n 的最大公因子，欧几里得算法描述如下：

(1) 用 n 去除 m，将余数赋给 r；

(2) 将 n 的值赋给 m，将 r 的值赋给 n；

(3) 如果 $n=0$；返回 m 的值作为结果，过程结束；否则，返回第(1)步。

可以尝试用欧几里得算法来求 60 和 24 的最大公因子，按照上述步骤可以得到结果为 12。

虽然现在的算法绝大多数情况下都是通过计算机来实现的，但算法本身并不依赖于

计算机。计算机只是由于其强大的计算能力和存储容量而成为算法研究和实现的最重要工具；而用计算机语言所写成的程序就成为算法最常见的载体。

著名计算机科学家、图灵奖获得者、美国斯坦福大学计算机系荣誉退休教授高德纳(Donald Knuth)在他的经典名著《计算机程序设计艺术》(*The Art of Computer Programming*)一书中，给出了算法的5个基本特征[7]。

有穷性：一个算法必须在执行有穷步之后就能够结束，并且其中每一步都可在有穷时间内完成；

确定性：算法的描述必须无歧义，以保证算法的实际执行结果是精确地符合要求或期望，通常要求算法的实际执行结果是确定的；

可行性：算法中描述的指令都可以通过已经实现的基本操作运算的有限次执行来实现；

输入：一个算法有零个或多个输入，这些输入取自某个特定的对象集；

输出：一个算法有一个或多个输出，输出量是算法计算的结果。

在现代社会中，算法已经成为一种一般性的智能工具，并且在绝大多数的科学、商业和技术领域都得到了广泛的应用。但算法并不能解决所有问题，例如我们无法找到一个使人生活愉快的算法，也不存在使人富有和出名的算法[10]。

5.3.2 问题与解

在为某个问题寻找求解的算法之前，我们首先关心的是问题是否有解，如果有解，解是否唯一。例如线性方程组 $\boldsymbol{A}\boldsymbol{x}=b$，如果系数矩阵 $\boldsymbol{A}$ 是满秩的，则线性方程组有唯一解；如果系数矩阵 $\boldsymbol{A}$ 是缺秩的，那么线性方程组可能有无穷多个解，也可能没有解。采用随机访问的方式遍历一个数据集合，会得到很多个不同的数据序列；但是对一个给定的实数序列进行排序的结果是唯一的。因此，对于待求解的问题，解的存在性和唯一性是我们在设计具体的求解算法之前就需要关心的问题。

我们用计算机处理和解决的问题一般可以分为两类，分别是数值问题和非数值问题。用于求解数值问题的算法称为数值算法，这个学术方向也称为数值分析或者科学计算；而用于求解非数值问题的算法称为非数值算法。

典型的数值问题包括求解线性方程组、非线性方程、拟合和插值、矩阵运算、数值微积分和很多规划问题。这类问题的解空间是连续的，一般是 n 维实数空间 $\mathbb{R}^n$ 或者是其子集 $S\subseteq\mathbb{R}^n$。这些问题中很多要么不存在解析解，要么求解过程的代价很大，因此利用计算机实现数值算法进行求解就成为最有效的手段。为了提高计算效率，在计算机上建立模型或者求解过程中需要控制复杂度，这就会引入“截断误差”，例如利用台劳展开时往往会截取到一阶导数项或者二阶导数项，而把更高阶的项直接丢弃。除此之外，计算机是一个离散的数字系统，其中能表示的数是非常有限的，因此在数值问题的描述和求解过程中总是需要用计算机能表示的数值近似替代实际的数值，这就引入了“舍入误差”。截断误差和舍入误差都是导致计算误差的原因，除了计算误差外，误差还会在计算过程中传递，称为

传播误差。因此计算机用于描述和求解数值问题时，误差成为难以回避的关键问题。我们很少有机会利用计算机求得问题的精确解，而是只能满足于得到一个误差足够小的近似解。

对于数值问题，我们还很关心问题的病态性。一个数值问题被称为病态的，是指当问题对于输入参数非常敏感，只要输入参数有微小的变化时，问题的解就会发生非常大的改变。例如对于线性方程组：

$$\begin{cases} x+\alpha y=1 \\ \alpha x+y=0 \end{cases}$$

当 $\alpha=1$ 时，问题无解；当 $\alpha\neq 1$ 时，解为：$\begin{cases} x=1/(1-\alpha^2) \\ y=-\alpha/(1-\alpha^2) \end{cases}$。因此当 $\alpha\approx 1$ 时，问题的解就对 α 的取值非常敏感，例如当 $\alpha=0.999$ 时，$x=500.25$；而当 $\tilde{\alpha}=0.998$ 时，$\tilde{x}=250.25$；问题的参数值变化了千分之一，解的数值就随之变化了一倍。因此，当 $\alpha\approx 1$ 时，这个线性方程组是病态的。

由于计算机表示数值问题时，误差是不可避免的，而病态问题的解是不可信的。我们还可以进一步定义数值算法的病态性。但值得注意的是，如果一个数值算法不是病态的，我们有可能通过寻找更稳健的算法来得到可靠的解；但如果一个数值问题是病态的，那无论采用什么算法，都不能改善其病态性。

虽然计算机的发明首先来自数值计算的迫切需求，但由于计算机同样具有出色的逻辑运算和符号计算的能力，因此计算机已经越来越多地被用来解决非数值问题，包括描述集合、线性表、树和图等数据结构及其操作，查找和排序，以及各种组合优化问题。

尽管非数值问题的解空间也可以很大，但可行解一定是离散的，非数值问题的求解实际上是符号运算的过程。因此在使用计算机来表示和求解非数值问题时，一般来说误差不是关键问题。

数值问题和非数值问题具有不同特性，因此在求解时关注点也有所不同。然后在解决很多实际问题时，其求解方案中可能既有数值的部分，也有非数值的部分。这既说明了实际问题的复杂性，也提醒我们需要对这两类问题都有深刻的理解。

5.3.3　算法的分析与评价

解决一个问题可以采用很多种不同的算法，那么如何评价这些算法呢？算法最主要的评价标准有两个，其一是正确性，其二是算法效率。

算法要求对于符合条件的输入，能够得到符合预期的正确结果。这是对算法正确性的基本要求，但是这个看似简单的要求并不容易满足，因为一个算法对某些输入数据得到结果正确并不能保证它对于所有输入数据都能得到正确的结果，而要对所有符合条件的输入数据都进行测试是不现实的。因此我们希望通过一些精心选择的、典型、苛刻且带有刁难性的输入数据来对算法进行正确性测试。而对算法更高的要求是，如果输入不符合

条件，算法要能够妥善应对，特别是不能因为对输入数据的可能性考虑不周而引起程序崩溃，这样的算法才是稳健的。因此算法的正确性和稳健性测试是一项重要的、有很大难度的专门性工作。

需要与算法的正确与否严格区分的是精确和近似算法。能够得到问题精确解的算法就是精确算法，但在很多情况下得到问题的精确解或无可能或无必要，这时我们往往会采用近似算法。这其中又分为两种类型，其一对数值问题，如解方程组，求定积分，矩阵分解，函数插值和逼近等，这类问题研究的是连续性对象，问题的解也往往具有连续性，因此算法的目标就是给出符合精度要求的近似解。其二是有一些非常困难的问题，得到精确解的时间或者资源代价过高，如经典的旅行商问题（TSP）和图着色问题。因此在工程实践中，对于这类问题，我们通常满足于通过一个效率较高的近似算法得到原问题的次优解。

对于近似算法，在算法设计时就确定了算法求解的目标是能够满足需要的近似解。只要达到了预期的要求，近似算法就是正确的。这与由于算法设计和实现上的错误，导致输出结果不正确是完全不同的。

正确的算法并不一定就是好算法。算法评价的另一个重要标准是算法的效率，包括算法的时间效率和空间效率，即执行算法获得正确结果所需要耗费的时间和空间资源。尽管计算机所拥有的计算能力和存储能力增长非常迅速，但是算法的作用仍然不是计算机硬件性能提升所能替代的。我们用下面这个例子来说明算法效率的重要性。

排序是非常重要的基本算法，应用极为广泛。在基于比较的排序算法中，插入排序是一种时间效率较低的排序方法，它的执行时间与待排序序列规模 n 的关系为 C_1n^2；而归并排序是一种时间效率较高的排序方法，它的执行时间与待排序序列规模 n 的关系为 $C_2n\log n$；我们分别用运算速度不同的两台计算机 A 和 B 来运行这两种算法，具体情况如下：

排序方法	时间复杂度	系数取值	运行速度
插入排序	C_1n^2	$C_1=2$	计算机 A：10^9
归并排序	$C_2n\log n$	$C_2=50$	计算机 B：10^7

计算机 A 的运行速度是计算机 B 的 100 倍，我们用速度较快的计算机 A 来执行效率较低的插入排序算法，而用速度较慢的计算机 B 来执行效率较高的归并排序算法。我们希望通过这组实验来观察算法的效率和计算机的运行速度，分析在算法的实际执行中影响更大的因素。我们分别对序列规模为 10^4、10^6 和 10^7 的两个序列进行对比实验，结果如下：

排 序 方 法	序列规模 $n=10^4$		序列规模 $n=10^6$		序列规模 $n=10^7$	
	指令数	运行时间	指令数	运行时间	指令数	运行时间
计算机 A 运行插入排行	2×10^8	0.2s	2×10^{12}	2000s	2×10^{14}	2.3day
计算机 B 运行归并排行	6.64×10^6	0.66s	10^9	100s	1.17×10^{10}	20min

可以看到，对于规模为 10^4 的序列，B 的运行时间约为 A 的 3.3 倍，但耗时都不到 1s；但随着序列规模的增大，B 采用高效算法的优势开始表现出来。对于规模为 10^6 的序列，A 的运行时间是 B 的 20 倍；对于规模为 10^7 的序列，A 的运行时间是 B 的 171 倍。因此，算法的作用要远远超出不同计算机硬件之间的性能差异，并且当处理问题的规模越大时，算法的作用会体现得越明显。而在当今这个数据爆炸的时代，算法处理的问题的规模在迅速膨胀。这也是算法的效率非常重要的原因。

5.3.4　算法的实现方式

算法的分类方式很多，根据算法的实现方式就可以有很多分类。

递归与迭代：递归是不断调用自身直到满足终止条件的算法实现方式，递归由于其易于理解和实现方便得到了广泛的应用。迭代是通过构造重复结构来求解问题的算法实现方式。人们有时希望通过实现递归算法的非递归化来提高算法的时间和空间效率，而引入迭代是消除递归的重要方式。我们前面提到了用于求解两个整数最大公因子的欧几里得算法，既可以用递归算法实现，也可以采用非递归的迭代算法来实现。

串行、并行和分布式算法：在一般的算法设计中都假设指令会被逐条执行，这样的算法称为串行算法，适合在顺序执行指令的计算机上运行。如果一台计算机可以有多个处理器同时处理一个任务，我们就可以设计相应的并行算法，通过同时利用多个处理器的计算能力来缩短算法的执行时间。如果存在用网络连接的多台计算机，我们就可以设计分布式算法来使这些计算机协同处理一个任务，Google 采用的搜索引擎算法就是典型的分布式算法。对于并行算法和分布式算法来说，不仅要考虑多台计算机和多个处理器的任务分配，还需要考虑计算机之间，以及处理器之间的通信和协同问题。

5.3.5　常用的算法设计思想

算法是求解问题的方法，因而算法设计就是寻找适合求解特定问题的有效策略。设计一个正确而高效的算法往往是一件困难的事情，因为没有办法告诉人们对于一个新的问题，如何才能设计出最合适的算法。因此算法的设计是一个创造性的过程，优秀的算法往往也是经过长期努力不断优化才得到的。排序、查找和模式匹配这些经典问题的算法发展历程就清楚地表明了这一点。

经过学者们长期不懈的努力，算法设计的技术体系逐步形成。这些算法设计技术已经成为计算机科学的核心内容，对于我们寻找解决问题的策略并进行优化具有重要的价值。常用的算法设计技术包括蛮力法、分治法、变治法、动态规划、贪心算法、线性规划、搜索算法和随机算法，我们在这里只进行简要的介绍。

蛮力法，顾名思义，就是不采用任何技巧，基于问题的描述简单直接地解决问题的方法。例如在求两个整数 m 和 n 的最大公因子时，可以连续检测从 2 到 m 和 n 中较小者，

从所有可以同时整除 m 和 n 的数选出最大的，即为 m 和 n 的最大公因子。又如传统的百元买百鸡问题：公鸡每只 5 元、母鸡每只 3 元、小鸡 3 只 1 元，百元买百鸡，问共有多少种买法？根据百元最多可以购买的公鸡、母鸡和小鸡数目，穷举所有可能的组合，寻找满足百鸡约束的方案。这种穷举的方法就是一种典型的蛮力法。一般来说，蛮力法的效率相对较低，但是蛮力法仍然是一种非常重要的算法设计技术。这是因为首先蛮力法的思路很直接，实现简单，也容易被理解，往往可以作为求解问题的基本方案，并用于衡量其他方法的正确性和效率。其次，对于一些规模较小的问题，花费很大精力去研究高效算法也许并不值得，这时采用蛮力法是一种很经济的做法。

分治法：这可能是最著名的算法设计技术，基本思想是把问题的一个实例分解成为属于同一问题的若干个规模较小的实例，重复这个过程直到规模较小的实例很容易求解，然后通过合并这些规模较小的实例来得到原始问题的解。分治法的例子很多，如快速排序、二叉树遍历等。有一类较为简单的分治法称为减治法，是指把问题转化为规模较小的子问题，通过求解子问题来得到原问题的解。分治法一般会把一个规模较大的问题分解成为若干个规模较小的问题进行求解；而减治法直接降低了问题的规模，问题的个数仍然是一个。减治法的典型例子是二分查找，读者可以通过这个例子来体会减治法和一般分治法的区别。

变治法：其基本思路是把一个求解困难的问题转换成为一个有已知解法的问题，并且这种转换的复杂度不超过求解目标问题的算法复杂度。如堆排序算法，将一个针对待排序序列的排序算法转化为先将这个待排序序列构建成最大堆，然后逐个输出堆顶元素的过程。又如查找问题，一种典型的动态查找算法是根据数据先构建出一棵查找树，然后在查找树上实现快速查找。

动态规划：这是一种求解多阶段决策过程最优化问题的通用方法，是在 20 世纪 50 年代由美国数学家 Richard Bellman 发明的。问题的整个过程被划分为若干个阶段，在每个阶段都需要做出决策从而进入下一个阶段。适于用动态规划求解的问题是由相互交叠的子问题组成的。如果这样的问题满足最优子结构和无后效性，采用动态规划求解就一定可以得到问题的最优解。著名的背包问题就可以采用动态规划算法求得最优解，用于卷积码最大似然译码的 Viterbi 算法和求解图的全源最短路径的 Floyd 算法都是动态规划的典型例子。

贪心算法：与动态规划一样，贪心算法也是用于求解多阶段决策问题，但是贪心算法在每个阶段只根据当前情况作出局部最优的决策，并不考虑解的全局最优性，这也是贪心算法得名的原因。贪心算法最著名的例子是找零问题，这是一个大家日常生活中经常遇到的问题。贪心算法具有很高的效率，但不能保证得到的解是最优解。但有一些问题确实存在能够得到最优解的贪心策略，如求图的最小生成树的 Prim 算法和 Kruskal 算法，寻找最优编码的 Huffman 算法。

线性规划：这是一种解决多变量最优决策的典型方法，一般是在一组不等式约束下对一个线形目标函数求最优解。典型的线性规划问题有生产计划安排、货运方案问题等；求有向图的最大流也是线形规划问题。线形规划问题的典型求解方法是单纯形法和内点法。

搜索算法：对于组合优化问题，可行解的集合可以用确定的解空间来描述，因此按照某种规则在解空间中进行搜索可以求解这类问题。搜索算法又可以分为遍历算法、回溯算法和分支界限法。对于很多问题，解空间的规模都相当大，因此如何提高搜索效率就成为提升算法性能的关键。我们希望能够找到有效的方法，在保证最优解存活的基础上通过剪枝以尽量减少对搜索空间不必要的访问。启发式搜索算法则希望通过对问题的理解、背景知识的利用和一些特殊的技巧，在较低的复杂度下得到问题的最优解或者次优解。虽然启发式算法并不能保证解的质量和求解效率，但在处理很多实际问题时，好的启发式算法确实在这两个方面都表现出了优异的性能。

随机算法：随机算法希望在求解问题的过程中引入了随机性的选择，并且正是随机性的引入使我们有可能实现更好的性能。例如在快速排序中，划分元素是随机选择得到的，这使得快速排序在所有基于比较的排序方法中平均时间复杂度最低。这是舍伍德算法(Sherwood Algorithm)的一个例子。典型的随机算法还有拉斯维加斯算法(Las Vegas algorithm)和蒙特卡罗方法(Monte Carlo algorithms)。

时空平衡技术：在设计和优化算法过程中，时间效率和空间效率有时会成为一对矛盾，也就是说我们可以通过牺牲空间效率来提高时间效率，或者通过牺牲时间效率来提高空间效率。一个典型的例子是散列，在散列中我们建立了从关键字到存储地址的映射，并通过冗余的表空间来处理冲突，以降低空间效率来换取查找操作较高的时间效率，很好地反映了时空平衡的思想。在一些对时间效率要求很高而空间资源相对宽裕的场合，我们常常会利用查表法来满足应用需求，采用的是空间换时间的思想。

预构造技术：即为了后续操作的方便和高效，预先将数据构建成某种结构形态。例如，在动态查找算法中，可以将数据元素组织成二叉搜索树、B-树或红黑树等多种查找树，通过数据的逻辑结构体现数据之间的部分有序性，使得后续查找和维护操作都比较方便。另一个非常重要的例子是在搜索引擎中，为了对用户的搜索请求给出迅速而正确的回复，在网页搜寻和处理阶段就需要构建倒排索引，倒排索引是搜索引擎最重要的基本技术。

输入增强技术：在算法正式执行之前，我们可以对输入数据进行预先的处理，使得算法在实际执行过程中由于能够充分利用输入数据的特性从而取得更高的效率。例如，在字符串模式匹配中，KMP 算法和 Boyer-Moore 算法都是通过对子串进行预处理来把握子串的特性，并利用这种特性来提升算法的时间效率，这就是输入增强技术。

时空平衡技术、预构造技术和输入增强技术都是典型的算法优化技术。

值得注意的是，不同的算法设计和优化技术之间并不是泾渭分明的。例如，我们在待查找的数据集上建立二叉搜索树、B-树或红黑树等动态查找结构，高效率地实现查找和维护数据集等操作。这样的做法既可以认为是采用变治法的思想，也可以认为是引入了预构造技术。算法设计和优化技术是人们对长期以来形成了设计方法和优化思路经过总结归纳形成的一个体系，帮助我们在面对新的问题时有可能借助这些已有的算法设计思想找到好的解决方案，但同时我们也不能被现有的体系所束缚，算法设计和优化是一个非常需要创新性思维的研究领域。

5.4 数据与算法的相互作用

数据与算法是什么关系呢？数据是客观世界的描述，是信息的载体，是算法的处理对象；算法是解决问题的方法和步骤，是处理数据的系统。因此数据与算法的关系，本质上是信息载体与系统的关系。

今天，信息已经和物质、能源一起成为人类社会的三大基石。然而信息不能独立存在，必须要依附于某种载体；电势、比特和各种信号都可以是信息的载体。数据的广泛性使其成为信息最重要的载体。算法是处理数据的系统，按照数据形式、规模以及需求的不同，我们可以设计各种不同的算法来处理数据。

人类从自己的感知来感受，了解和认知我们所处的世界，因此科学的发展具有很强的经验主义传统。由于经验知识往往是零散而不完善的，因此在相当长一段时间里根据逻辑规则进行演绎成为数学研究的主要方式，人们试图构造出完美的公理系统。但是哥德尔不完备定律则证明这样的努力不可能取得成功。仅仅依靠演绎对于科学的发展是不够的，直觉和实验是引发科学猜想的重要源泉。特别是随着计算机的发展，计算和模拟在科学研究中的地位越来越重要。

基于这个认识，科学研究的一般方法如图 5.4.1 所示。

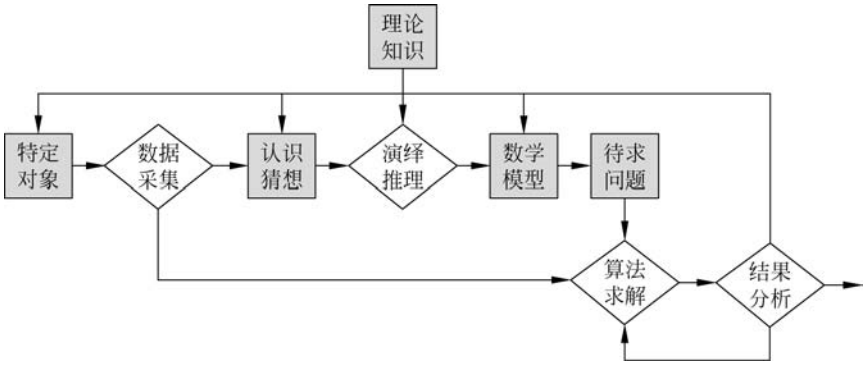

图 5.4.1 科学研究的一般方法

人们通过对客观世界中的特定对象的观察，或者设计各种特定的实验来采集数据。在这些数据的基础上形成对特定问题的认识和猜想，根据人们已经掌握的理论知识，经过演绎推理形成数学模型，同时现实问题也被抽象成为数学模型上的待求问题。设计相应的算法来求解这些问题，对求解得到的结果进行分析，判断是否符合实际情况，如果不符合，可能是对特定对象的认识和猜想有不正确的地方，也可能是没有形成足够好的数学模型，或者求解的算法存在问题；经过反馈验证，调整数学模型和求解算法，最终得到满足要求的结果。

在这个迭代过程中，数据包含了客观事物的信息，是算法处理的对象；算法是人们设计的处理数据的方法，包含了人们对数据和问题的认识和猜想。同时，算法的设计和优化

必须充分考虑到数据的性质和存在形式。不了解和利用数据的特性，不可能找到最有效的算法。这很好地体现了数据和算法之间相互作用的关系。

当人们面对的数据很少时，人们通过自身的观察和认知能力，以及对小规模数据的观察、分析和思考，来探寻其中蕴含的规律和知识。但随着数据规模越来越大，数据中蕴含的规律越来越复杂，人们仅依靠自身的能力往往不足以形成有效的认知。因此，在数据爆炸的今天，海量数据的处理能力对人类提出了越来越大的挑战。

随着社会的进步，人们对于人工智能给予了更多的关注。机器是否能具有智能？人造的机器能够比人更聪明吗？影响较大的尝试有 IBM 的深蓝、沃森（Waston）以及 Google 的 AlphaGo。

深蓝是一台 32 节点的 IBM RS/6000 SP 计算机，运行 AIX 操作系统，专为国际象棋设计，拥有强大的计算能力，每秒可检查超过 2 亿个棋步。1996 年 2 月 10 日，深蓝首次挑战卡斯帕罗夫，但以 2∶4 落败，1997 年 5 月，改良后的深蓝再次挑战卡斯帕罗夫，以 3.5∶2.5 击败卡斯帕罗夫，成为首个在标准比赛时限内击败国际象棋世界冠军的计算机系统。卡斯帕罗夫与深蓝对阵的第 6 局如图 5.4.2 所示。

图 5.4.2　卡斯帕罗夫与深蓝对阵的第 6 局

沃森，拥有 90 个 IBM 处理器，是首台由标准服务器构成的超级计算机（4 路 8 核 32 线程 3.5GHz 处理器，共 2880 核），拥有 16TB 的存储容量，以 IBM 创始人托马斯·沃森命名。《危险边缘》（*Jeopardy*）是美国著名的智答竞赛节目，其问题设置的涵盖面非常广泛，涉及历史、文学、艺术、流行文化、科技、体育、地理、文字游戏等各个领域。2011 年 2 月 14 日到 16 日，沃森与《危险边缘》这个节目历史上最成功的两位传奇选手肯·詹宁斯和布拉德·鲁特展开了公开对决，结果沃森以大比分获胜，如图 5.4.3 所示。

AlphaGo（阿尔法围棋）是 Google DeepMind 开发的人工智能围棋程序。2015 年 10 月，AlphaGo 在五番棋比赛中以 5∶0 战胜了欧洲围棋冠军华裔法籍棋士樊麾二段。2016 年 3 月，AlphaGo 以 4∶1 战胜了顶尖职业棋手李世石，成为第一个无须让子而击败围棋职业九段棋士的计算机围棋程序，如图 5.4.4 所示。2017 年年初，AlphaGo 化名为 Master 在网络上取得对职业顶尖棋手的 60 连胜。2017 年 5 月，AlphaGo 以 3∶0 战胜世界排名第一的中国围棋选手柯洁。

图 5.4.3 机器智能系统 Waston 在《危险边缘》中击败人类选手

图 5.4.4 AlphaGo 在五番棋中战胜世界冠军李世石

以上三个例子说明了人类在人工智能领域取得的辉煌成就。在国际象棋、智力问答和围棋这三个特定领域，人们针对数据的表示和存储，针对特定目标的算法进行了大量的研究工作，使得在这三个任务上数据和算法之间达到十分融洽的程度。在这三个特定场景下，数学模型被精心的设计和建立起来，数据的表示和存储最便于被算法所处理，信息也能以最有效的方式被提取；同时一系列针对性的高性能算法被设计和优化，这些是深蓝、沃森和 AlphaGo 取得成功的根本原因。这样的例子还有很多，例如已经成为人们最重要信息来源的搜索引擎，试图将地球上大部分人都纳入其中的社交网络(SNS)。但是这些成功并不意味着机器已经比人更聪明，针对特定应用，经过特殊优化的计算机系统表现出来的能力，还不足以推广到一般情形。何况，即使是在比赛中获胜的沃森，也犯了一些在人看来不可思议的低级错误。因此，今天的机器，无论多么先进，都还远不足以拥有真正的智能。但这些成功案例让我们看到，数据和算法的完美融合，将会迸发出多么巨大的威力。人工智能技术研究的目的，并不应该是让机器取代人，而是看到机器和人类各自的优势和特点，从而充分地发挥机器的特长和能力，来更好地为人类服务。

作为客观事物符号表示的数据，是信息的载体；解决特定问题策略的算法，是处理数据和信息的系统。因此数据与算法，反映了信息载体与系统的相互作用。数据会具有更加多种多样的呈现形式，算法也可以组合形成更复杂的集成系统。数据与算法的关系，体

现了人类不断探索、实现认知的发展历程。每个人在工作和个人生活中，时时刻刻都能看到数据的身影，也随处可见算法的智慧。随着社会的信息化，面对触手可及的数据浪潮，人类已不仅在使用算法工具，有时人类自身也已经开始成为算法的一个部分，例如：Google 的 PageRank 算法在进行搜索结果排序时就会考虑外部链接的数量和质量，通过群体的协作实现更准确的评价；同时使用搜索引擎的用户点击又成为 Google 优化搜索引擎性能的有效信息。又如，在简单的协作和分享的机制下，维基百科取得了出乎很多人意料的成功，让更多的人参与进来，他们既是知识的提供者，又是知识的使用者，角色的模糊造成了信息流动的更复杂模式。时代在发展，人们对于数据与算法的认识还在不断进化，新的模式也需要用新的数学模型来描述。但毫无疑问，这是一个对人类生活有着巨大影响的前进方向。

参考文献

[1] Huggins J S. How Much Data Is That? http://www.jamesshuggins.com/h/tek1/how_big.htm.

[2] Smith B. Care for a Byte-Explaining Bits, Bytes and More. http://www.digital-photography-school.com/care-for-a-byte-explaining-bits-bytes-and-more.

[3] 莫里斯·克莱因. 古今数学思想. 张理京，张锦炎，江泽涵，译. 上海：上海科学技术出版社，2002.

[4] Weik M H. The ENIAC Story. Ordnance Ballistic Research Laboratories. Aberdeen Proving Ground, MD.

[5] Winegrad D, Akera A. A Short History of the Second American Revolution. http://www.upenn.edu/almanac/v42/n18/eniac.html.

[6] 张凤雏. 电子计算机之父——冯·诺依曼(上). 中国计算机报：1999，305(05) [1999-02-03].

[7] Knuth D E. 计算机程序设计艺术. 北京：人民邮电出版社，2010.

[8] Sedgewick R. C++算法——图算法. 林琪，译. 北京：清华大学出版社，2003.

[9] Sedgewick R. 算法Ⅰ-Ⅳ(C++实现)——基础、数据结构、排序和搜索. 张铭泽，赵剑云，等译. 北京：清华大学出版社，2004

[10] Levitin A. 算法设计与分析基础. 潘彦，译. 北京：清华大学出版社，2004.

[11] Cormen T H. Introduction to algorithms. Second Edition. MIT Press, 2001.

[12] 朱明方，吴及. 数据结构与算法教程. 北京：人民邮电出版社，2011.

[13] 朱明方，吴及. 数据结构与算法. 北京：清华大学出版社，2010.

[14] 殷人昆，等. 数据结构(用面向对象方法与 C++语言描述). 北京：清华大学出版社，2007.

[15] 朱明方，吴及. 数据结构教程. 北京：机械工业出版社，2007.

[16] Heath M T. 科学计算导论. 张威，贺华，冷爱萍，译. 北京：清华大学出版社，2005.

[17] 严蔚敏，吴伟民. 数据结构(C 语言版). 北京：清华大学出版社，1996.

[18] Newborn M. 旷世之战——IBM 深蓝夺冠之路. 邵谦谦，译. 北京：清华大学出版社，2004.

[19] 新浪科技. IBM 沃森超级电脑人机对战专题. http://tech.sina.com.cn/d/IBMWatson/，2011.

[20] 新浪科技. 谷歌人工智能破解围棋比赛. http://tech.sina.com.cn/d/AlphaGo/，2011.