

### 5.1 数据的组织和管理

信息是对客观世界中各种事物的运动状态和变化的反映。数据是信息的一种载体,是信息的一种表达方式,在计算机中信息是使用二进制进行编码的。数据是描述客观事物的数值、字符及能输入机器且能被处理的各种符号集合。简言之,数据就是计算机化的信息。

计算机的程序是对信息(数据)进行加工处理。可以说,程序=算法+数据组织和管理,程序的效率取决于两者的综合效果。随着信息量的增大,数据组织和管理变得非常重要,它直接影响程序的效率。

#### 5.1.1 数据结构

数据结构是计算机存储、组织数据的方式。数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。通常情况下,精心选择的数据结构可以带来更高的运行或者存储效率。数据结构往往同高效的检索算法和索引技术有关。

比如,一幅图像是由简单的数值组成的矩阵,一个图形中的几何坐标可以组成表,语言编译程序中使用的栈、符号表和语法树,操作系统中所用到的队列、树状目录等都是有结构的数据。

数据结构通常包括以下几个方面。

- (1) 数据的逻辑结构。由数据元素之间的逻辑关系构成。
- (2) 数据的存储结构。数据元素及其关系在计算机存储器中的存储表示,也称为数据结构的物理结构。
- (3) 数据的运算。施加在该数据上的操作。

##### 1. 逻辑结构

数据的逻辑结构是从逻辑关系上描述数据,它与数据的存储无关,是独立于计算机的,因此数据的逻辑结构可以看成从具体问题中抽象出来的数学模型。数据的逻辑结构有两个要素,一是数据元素,二是关系。根据数据元素之间关系的不同,数据通常有四类基本结构:集合结构、线性结构、树结构和图结构。

##### 1) 集合结构

集合是指比较简单的数据,即少量、相互间没有太大关系的数据。比如,在进行计算某方程组的解时,中间的计算结果数据可以存放在内存中以便以后调用。在程序设计语言中,往往用变量来实现。

## 2) 线性结构

线性结构是指该结构中的数据元素之间存在一对一的关系。其特点是开始元素和终端元素都是唯一的,除了开始元素和终端元素以外,其余元素都有且仅有一个前驱元素,有且仅有一个后继元素。典型的线性结构有线性表、栈和队列。

### (1) 线性表。

简单地说,线性数据是指同类的批量数据,也称线性表。比如,英文字母表(A,B,...,Z)、1000个学生的学号和成绩、3000个职工的姓名和工资、一年中的四个季节(春、夏、秋、冬)等。

### (2) 栈。

如果对线性数据操作增加如下规定:数据的插入和删除必须在同一端进行,每次只能插入或删除一个数据元素,则这种线性数据组织方式就称为栈结构。通常将表中允许进行插入、删除操作的一端称为栈顶(Top);同时表的另一端被称为栈底(Bottom)。当栈中没有元素时称为空栈。

栈的插入操作被形象地称为进栈或入栈,删除操作称为出栈或退栈。

栈是先进后出的结构(First In Last Out,FILO),如图 5.1(a)所示。日常生活中铁路调度就是栈的应用,如图 5.1(b)所示。

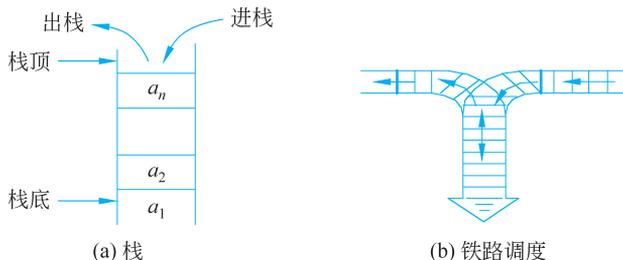


图 5.1 栈和栈的应用

比如,网络浏览器会将用户最近访问过的网址组织为一个栈。这样,用户每访问一个新页面,其地址就会被存放至栈顶;而用户每按下一次“后退”按钮,即可沿相反的次序访问此前刚访问过的页面。

类似地,主流的文本编辑器也大都支持编辑操作的历史记录功能(Ctrl+Z:撤销, Ctrl+Y:恢复),用户的编辑操作被依次记录在一个栈中。一旦出现误操作,用户只需按下“撤销”按钮,即可取消最近一次操作并回到此前的编辑状态。

### (3) 队列。

如果对线性数据操作增加如下规定:只允许在表的一端插入元素,而在另一端删除元素,则这种线性数据组织方式就称为队列,如图 5.2 所示。

队列具有先进先出(First In First Out,FIFO)的特性。在队列中,允许插入的一端称为队尾(Rear),允许删除的一端则称为队头(Front)。

队列运算包括:入队运算——从队尾插入一个元素;退队运算——从队头删除一个元素。

日常生活中排队就是队列的应用。计算机及其网络自身内部的各种计算资源,无论是多进程共享的 CPU 时间,还是多用户共享的打印机,都需要借助队列结构实现合理和优化

的分配。

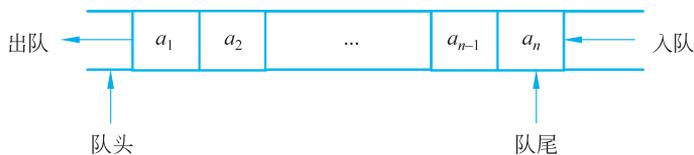


图 5.2 队列

### 3) 树结构

如果要组织和处理的数据具有明显的层次特性,比如,家庭成员间的辈分关系、一个学校的组织图,这时可以采用层次数据的组织方法,也形象地称为树结构。

层次模型是数据库系统中最早出现的数据模型,是用树结构来表示各类实体以及实体间的联系。层次数据库是将数据组织成树结构,并用“一对多”的关系联结不同层次的数据库。

严格地讲,满足下面两个条件的基本层次联系的集合称为树状数据模型或层次数据模型(见图 5.3):

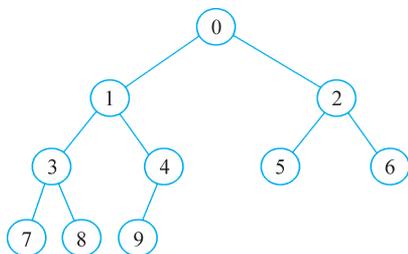


图 5.3 树状数据模型

- 有且只有一个结点没有双亲结点,这个结点称为根结点;
- 根以外的其他结点有且只有一个双亲结点。

在第 1 章例 1-3 中,提到了国际象棋世界冠军“深蓝”。国际象棋、西洋跳棋与围棋、中国象棋一样都属于双人完备博弈。所谓双人完备博弈就是两位选手对垒,轮流走步,其中一方完全知道另一方已经

走过的棋步以及未来可能的走步,对弈的结果要么是一方赢(另一方输),要么是和局。

对于任何一种双人完备博弈,都可以用一棵博弈树(与或树)来描述,并通过博弈树搜索策略寻找最佳解。博弈树类似于状态图和问题求解搜索中使用的搜索树。搜索树上的第一个结点对应一个棋局,树的分支表示棋的走步,根结点表示棋局的开始,叶结点表示棋局的结束。一个棋局的结果可以是赢、输或者和局。

树在计算机领域也有着广泛的应用。例如,在编译程序中,用树来表示源程序的语法结构;在数据库系统中,可用树来组织信息;在分析算法的行为时,可用树来描述其执行过程。

### 4) 图结构

有时,还会遇到更复杂一些的数据关系,满足下面两个条件的基本层次联系的集合称为图状数据模型或网状数据模型:

- 允许一个以上的结点无双亲;
- 一个结点可以有多个的双亲。

比如,在例 4-1 国际会议排座位问题中,可以将问题转化为在图  $G$  中找到一条哈密顿回路的问题。

#### 【例 5-1】哥尼斯堡七桥问题。

1736 年,29 岁的欧拉向圣彼得堡科学院递交了《哥尼斯堡的七座桥》的论文,在解答问题的同时,开创了数学的一个新的分支——图论与几何拓扑。

哥尼斯堡七桥问题:18 世纪的东普鲁士有一座哥尼斯堡城,城中有一座奈佛夫岛,普

雷格尔河的两条支流环绕其旁,并将整个城市分成北区、东区、南区和岛区 4 个区域,全城共有 7 座桥将 4 个城区相连起来,人们可以通过这 7 座桥到各城区游玩,如图 5.4 所示。

人们常通过这 7 座桥到各城区游玩,于是产生了一个有趣的数学难题:寻找走遍这 7 座桥,且只许走过每座桥一次,最后又回到原出发点的路径。该问题就是著名的“哥尼斯堡七桥问题”。

和例 4-1 一样,欧拉抽象出问题最本质的东西,忽视问题非本质的东西(如桥的长度等),把每一块陆地考虑成一个点,连接两块陆地的桥以线表示,并由此得到了如图 5.5 所示的几何图形。

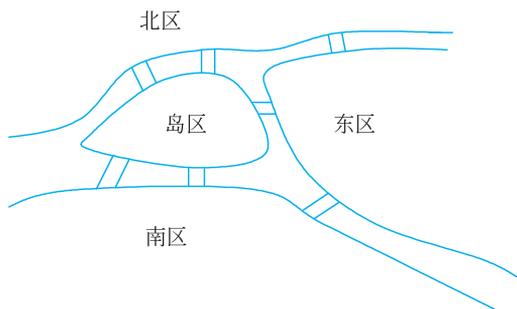


图 5.4 哥尼斯堡七桥

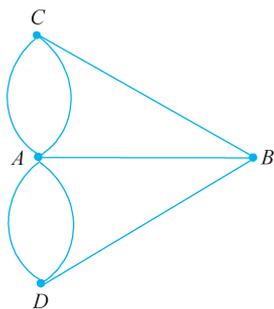


图 5.5 哥尼斯堡七桥的几何图形

若我们分别用 A、B、C、D 四个点表示哥尼斯堡的四个区域。这样著名的“七桥问题”便转化为是否能够用一笔画不重复的画出过此七条线的一笔画问题了。

欧拉不仅解决了此问题,且给出了一笔画问题的必需条件:图形必须是连通的;途中的奇点个数是 0 或 2(奇点是指连到一点的边的数目是奇数条)。

由此判断“七桥问题”中 4 个点全是奇点,可知图不能一笔画出,也就是不存在不重复地通过所有桥的路径。

“哈密尔顿回路问题”与“欧拉回路问题”的不同点:“哈密尔顿回路问题”是访问每个结点一次,而“欧拉回路问题”是访问每条边一次。

欧拉的论文为图论的形成奠定了基础。图论是对现实问题进行抽象的一个强有力的数学工具,已广泛地应用于计算学科、运筹学、信息论、控制论等学科。

在实际应用中,有时图的边或弧上往往与具有一定意义的数有关,即每一条边都有与它相关的数,称为权,这些权可以表示从一个顶点到另一个顶点的距离或耗费等信息。我们将这种带权的图称为赋权图或网,如图 5.6 所示。

可以利用算法求出图中的最短路径、关键路径等,因此图可以用来解决多类问题:电路网络分析、线路的敷设、交通网络管理、工程项目进度安排、商业活动安排等,是一种应用极为广泛的数据结构。

网状模型与层次模型的区别在于:网状模型允许多个结点没有双亲结点;网状模型允许结点有多个双亲结点;网状模型允许两个结点之间有多种联系(复合联系);网状模型可以更直接地去描述现实世界;层次模型实际上是网状模型的一个特例。

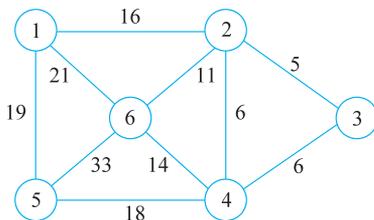


图 5.6 赋权图示例

## 2. 存储结构

数据对象在计算机中的存储表示称为数据的存储结构,也称为物理结构。把数据对象存储到计算机时,通常要求既要存储各数据元素的数据,又要存储数据元素之间的逻辑关系,数据元素在计算机内用一个结点来表示,数据元素在计算机中由两种基本的存储结构,分别是顺序存储结构和链式存储。

### 1) 顺序存储结构

顺序存储结构是借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系,通常借助程序设计语言的数组类型来描述,如图 5.7 所示。

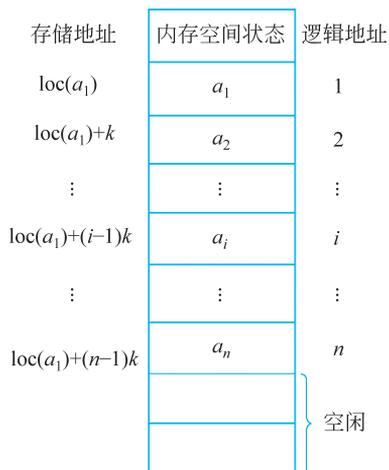


图 5.7 顺序存储

在此方式下,每当插入或删除一个数据,该数据后面的所有数据都必须向后或向前移动。因此,这种方式比较适合于数据相对固定的情况。

### 2) 链式存储结构

顺序存储结构要求所有的元素依次存放在一片连续的存储空间中,而链式存储结构无须占用整块存储空间,但为了表示结点之间的关系,需要给每个结点附加指针字段用于存放后继元素的存储地址。所以,链式存储结构通常借助于程序设计语言的指针类型来描述。

在链式存储方式中,每个结点由两部分组成:一部分用于存放数据元素的值,称为数据域;另一部分用于存放指针,称为指针域,用于指向该结点的前一个或后一个结点(前件或后件)。对于最后一个数据,就填上一个表示结束的特殊值,这种像链条一样的数据组织方法也称链表结构。设一个头指针  $\text{head}$  指向第一个结点。指定线性表中最后一个结点的指针域为“空”(NULL),如图 5.8 所示。

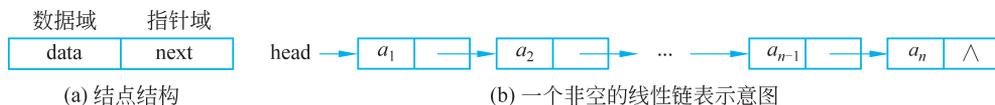


图 5.8 链表结构

**【例 5-2】** 线性表(A, B, C, D, E, F, G)的单链表存储结构,如图 5.9 所示。整个链表的存取需从头指针开始进行,依次顺着每个结点的指针域找到线性表的各个元素。

头指针 head 位置: 16

存储地址	数据域	指针域
1	D	55
8	B	22
22	C	1
37	F	25
16	A	8
25	G	NULL
55	E	37

图 5.9 线性表(A,B,C,D,E,F,G)的单链表存储结构

在此方式下,每当插入或删除一个数据,可以方便地通过修改相关数据的位置信息来完成。因此,这种方式比较适合于数据相对不固定的情况。

### 思考与探索

链式存储这种非连续方式中,每个数据都增加了存放位置信息的空间,所以是靠空间来换取数据频繁插入和删除等操作时间的设计,这种空间和时间的平衡问题是计算机中算法和方法设计中经常要考虑的问题。

### 3. 数据结构与算法

数据结构与算法之间存在着密切的关系。可以说不了解施加于数据上的算法需求就无法决定数据结构;反之,算法的结构设计和选择又依赖于作为其基础的数据结构,即数据结构为算法提供了工具。算法是利用这些工具来解决问题的最佳方案。

(1) 数据结构与算法的联系。数据结构是算法实现的基础,算法总是要依赖于某种数据结构来实现的。算法的操作对象是数据结构。算法的设计和选择要同时结合数据结构,简单地说,数据结构的设计就是选择存储方式,如确定问题中的信息是用普通的变量存储还是用其他更加复杂的数据结构。算法设计的实质就是对实际问题要处理的数据选择一种恰当的存储结构,并在选定的存储结构上设计一个好的算法。不同的数据结构的设计将导致差异很大的算法。数据结构是算法设计的基础。算法设计必须考虑数据结构,算法设计是不可能独立于数据结构的。另外,数据结构的设计和选择需要为算法服务。如果某种数据结构不利于算法实现,它将没有太大的实际意义。知道某种数据结构的典型操作才能设计出好的算法。

总之,算法的设计同时伴有数据结构的设计,两者都是为最终解决问题服务的。

(2) 数据结构与算法的区别。数据结构关注的是数据的逻辑结构、存储结构以及基本操作,而算法更多的是关注如何在数据结构的基础上解决实际问题。算法是编程思想,数据结构则是这些思想的逻辑基础。

## 5.1.2 文件系统和数据库

### 1. 文件系统

在较为复杂的线性表中,数据元素(Data Elements)可由若干数据项组成,由若干数据项组成的数据元素称为记录(Record),由多个记录构成的线性表称为文件(File)。

以文件方式进行数据组织和管理,一般需要进行文件建立、文件使用、文件删除、文件复制和移动等基本操作,其中文件的使用必须经过打开、读、写、关闭这四个基本步骤。程序设

计语言一般都提供了文件管理功能。

## 2. 数据库系统

如果数据量非常大,关系也很复杂,这时可以考虑使用数据库技术来组织和管理。

数据管理技术是在 20 世纪 60 年代后期开始的,经历了人工管理、文件管理、数据库系统三个阶段,与前两个阶段相比,数据库系统具有以下特点。

(1) 数据结构化。在数据库系统中数据是面向整个组织的,具有整体的结构化。同时存取数据的方式可以很灵活,可以存取数据库中的某一个数据项、一组数据项、一个记录或者一组记录。

(2) 共享性高、冗余度低、易扩充。数据库系统中的数据不再面向某个应用而是面向整个系统,因而可以被多个用户、多个应用共享使用。使用数据库系统管理数据可以减少数据冗余度,并且数据库系统弹性大,易于扩充,可以适用各种用户的要求。

(3) 数据独立性高。数据独立性包括数据的物理独立性和数据的逻辑独立性。物理独立性是指用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的。数据的物理存储改变了,应用程序不用改变。逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的,数据的逻辑结构改变了,用户程序也可以不变。

利用数据库系统,可以有效地保存和管理数据,并利用这些数据得到各种有用的信息。

### 1) 数据库系统概述

数据库系统主要包括数据库(Database)和数据库管理系统(Database Management System,DBMS)等。

(1) 数据库。数据库是长期存储在计算机内的、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和存储,具有较小的冗余度、较高的数据独立性和易扩展性,并可为各种用户共享。

(2) 数据库管理系统。数据库管理系统具有建立、维护和使用数据库的功能;具有面向整个应用组织的数据结构、高度的程序与数据的独立性,数据共享性高、冗余度低、一致性好、可扩充性强、安全性和保密性好、数据管理灵活方便等特点;具有使用方便、高效的数据库编程语言的功能;能提供数据共享和安全性保障。

数据库系统包括两部分软件——应用层与数据库管理层。

应用层软件负责数据库与用户之间的交互,决定整个系统的外部特征,例如,采用问答或者填写表格的方式与用户交互,也可以采用文本或图形用户界面的方式等。

数据库管理层软件负责对数据进行操作,例如,数据的添加、修改等。它是位于用户与操作系统之间的一层数据管理软件,主要有以下几个功能。

- 数据定义功能。提供数据定义语言,以对数据库的结构进行描述。
- 数据操纵功能。提供数据操纵语言,用户通过它实现对数据库的查询、插入、修改和删除等操作。
- 数据库的运行管理功能。数据库在建立、运行和维护时由 DBMS 统一管理、控制,以保证数据的安全性、完整性、系统恢复性等。
- 数据库的建立和维护功能。数据库的建立、转换,数据的转储、恢复,数据库性能监视、分析等。

(3) 数据库管理员。数据库和人力、物力、设备、资金等有形资源一样,是整个组织的基

本资源,具有全局性、共享性的特点,因此对数据库的规划、设计、协调、控制和维护等需要专门人员来统一管理,这些人员统称为数据库管理员。

## 2) 数据模型

各个数据及它们相互间的关系称为数据模型。数据库从结构上分主要有四种模型,即层次模型、网状模型、关系模型和面向对象模型。

关系模型是1970年由IBM公司的研究员E.F.Codd首次提出的,是目前最重要的一种数据模型,它建立于严格的数学概念基础上,具有严格的数学定义。20世纪80年代以来推出的数据库管理系统几乎都支持关系模型,关系数据库系统采用关系模型作为数据的组织方式。关系模型应用最为广泛,如SQL Server、MySQL、Oracle、Access、Sybase、Excel等都是常用的关系模型数据库管理系统。

关系模型是对关系的描述,由关系名及其所有属性名组成的集合。格式为

关系名(属性1,属性2,……)

例如,表5.1的学生成绩管理(学号,姓名,高数,英语,计算机)。

表 5.1 学生成绩管理

学 号	姓 名	高 数	英 语	计 算 机
130840101	张三	90	87	92
130740103	李四	77	88	96
130840102	王五	89	97	87
⋮				

关系模型中数据的逻辑结构实际上就是一个二维表,它应具备以下条件。

(1) 关系模型要求关系必须是规范化的。最基本的一个条件:关系的每一个分量必须是不可分的数据项。

(2) 表中每一列的名称必须唯一且每一列除列标题外,必须有相同的数据类型。

(3) 表中不允许有内容完全相同的元组(行)。

(4) 表中的行或列的位置可以任意排列,并不影响所表示的信息内容。

### 思考与探索

从抽象到具体的实现思想:数据库技术来源于现实世界的的数据及其关系的分析和描述。首先建立抽象的概念模型,然后将概念模型转换为适合计算机实现的逻辑数据模型,最后将数据模型映射为计算机内部具体的物理模型(存储结构)。

## 5.2 挖掘数据的潜在价值——数据挖掘与数据仓库

从人类认知的历史来看,最早了解自然规律的手段就是观察和归纳,人类最早就是从数据中获取知识的。现在随着信息技术的发展,获取数据的能力有了极大提高,进入了大数据时代,通过观察设备(传感器等)作用于各种自然现象、社会活动和人类行为,产生了大量的

数据。当采用大数据的分析方法和处理手段来解决问题时,我们得到了一系列对于世界的新认知。这些成果包括语音识别、图像判断、自动驾驶等领域。

## 5.2.1 大数据

### 1. 大数据的概念

研究机构 Gartner 给出了这样的定义:大数据(Big Data)是指无法在一定时间范围内用常规软件工具进行捕捉、管理和处理的数据集合,是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产。大数据不仅用来描述大量的数据,还涵盖了处理数据的速度。

麦肯锡全球研究所给出的定义:一种规模大到在获取、存储、管理、分析方面大大超出了传统数据库软件工具能力范围的数据集合,具有海量的数据规模、快速的数据流转、多样的数据类型和价值密度低四大特征。

当前对大数据的基本共识:大数据泛指无法在可容忍的时间内用传统信息技术和硬件工具对其进行获取、管理和处理的巨量数据集合,具有海量性、多样性、时效性及可变性等特征,需要可伸缩的计算体系结构以支持其存储、处理和分析。

大数据最根本的价值在于为人类提供了认识复杂系统的新思维和新手段,著名计算机科学家、图灵奖获得者 Jim Gray(吉姆·格雷)将数据密集型科研称为继实验观测、理论推导和计算模拟之后,人类探索未知、求解问题的“第四范式”,即数据驱动。基于数据,我们可以去触摸、理解和逼近现实复杂系统。

### 2. 大数据的特点

大数据具有五个层面的特点,可以用五个“V”来代表——Volume(大量)、Velocity(高速)、Variety(多样)、Veracity(真实)、Value(价值)。

(1) Volume(大量)。大数据的起始计量单位至少是 PB(1000 个 TB)、EB(100 万个 TB)或 ZB(10 亿个 TB)。

(2) Velocity(高速)。大数据的处理速度快,时效性要求高。这是大数据区别于传统数据挖掘最显著的特征。

(3) Variety(多样)。大数据的数据类型繁多,包括网络日志、音频、视频、图片、地理位置信息等,多类型的数据对数据的处理能力提出了更高的要求。

(4) Veracity(真实)。只有真实而准确的数据才能让对数据的管控和治理真正有意义。

(5) Value(价值)。大数据的数据价值密度相对较低。以视频为例,连续不间断监控过程中,可能有用的数据仅仅有一两秒。如何通过强大的机器算法更迅速地完成数据的价值“提纯”,是大数据时代亟待解决的难题。

大数据时代对人类的数据驾驭能力提出了新的挑战,也为人们获得更为深刻、全面的洞察能力提供了前所未有的空间与潜力。

简言之,从各种各样类型的数据中,快速获得有价值信息的能力,就是大数据技术。

### 3. 大数据的应用

#### 1) 大数据正在改善我们的生活

大数据不单单只是应用于企业和政府,同样也适用我们生活当中的每个人。比如,我们

可以利用穿戴的装备(如智能手表或者智能手环)生成最新的数据,根据热量消耗或睡眠模式提供的数据进行健康情况追踪。

#### 2) 业务流程优化

大数据正在帮助业务流程实现优化。其中应用最广泛的就是供应链及配送路线的优化:利用地理定位和无线电频率的识别追踪货物和送货车;利用实时交通路线数据制定更加优化的路线。

#### 3) 理解客户、满足客户服务需求

大数据的应用目前在这领域是最广为人知的。很多企业搜集社交方面的数据、浏览器的日志、传感器的数据等,并建立出数据模型进行预测,以更好地了解客户以及他们的爱好和行为。比如,通过大数据的应用,电信公司可以更好地预测出流失的客户;沃尔玛更加精准地预测哪个产品会热卖;汽车保险行业更精准地了解客户的需求和驾驶水平。

#### 4) 在体育行业的应用

现在很多运动员在训练的时候应用了大数据技术。比如,使用视频分析来追踪和分析比赛中每个球员的表现,使用智能技术来追踪运动员的营养状况及睡眠状况,并智能地给出战术策略和健康营养方面的建议。

#### 5) 疫情防控和医疗研发

大数据分析应用的计算能力可以让我们在几分钟内就解码整个 DNA,从而帮助我们制订出最佳的治疗方案,同时也可以更好地去理解和预测疾病。比如,面对突如其来的疫情,大数据技术凭借强大的数据采集、分析能力,并且结合互联网、区块链、云计算等技术,在疫情追踪、防控、预测等方面,做到了及时响应疫情突发、回应公共安全,为疫情提供了有力的技术支持。

#### 6) 金融交易

大数据在金融行业主要是应用金融交易。比如,现在很多股权的交易都是利用大数据算法进行的。

#### 7) 改善我们的城市

大数据还被应用到改善我们日常生活的城市中。比如,基于城市实时交通信息、利用社交网络和天气数据来优化最新的交通情况。

#### 8) 改善安全和执法

大数据现在已经广泛应用到安全和执法的过程当中。各国安全局利用大数据进行恐怖主义打击;而企业则应用大数据技术防御网络攻击;警察应用大数据工具捕捉罪犯;信用卡公司应用大数据工具来监测欺诈性交易。

#### 9) 优化机器和设备性能

大数据分析还可以让机器和设备在应用上更加智能化和自主化。比如,无人驾驶汽车、智能电话的优化等。

### 4. 大数据与云计算的关系

云计算是一种基于互联网的计算方式,通过这种方式,共享的软硬件资源和信息可以按需提供给计算机和其他设备。大数据与云计算的关系就像一枚硬币的正反面一样密不可分。大数据的特色在于对海量数据进行分布式数据挖掘,但它必须依托云计算的分布式处理、分布式数据库、云存储和虚拟化技术。