前言

这些笔记以及相关的大一课程是关于信息的。 虽然你可能对信息有一个大致的概念，但你可能没有意识到你处理的信息是可以量化的。 没错，你经常可以测量信息的数量并使用关于信息行为的一般原则。 我们将考虑应用于计算和通信的应用，并且我们还将研究其他科学和工程领域的一般规律。

其中一个这些一般规律就是热力学第二定律。虽然热力学是物理学的一个分支，但在这里，我们将第二定律作为自然和工程系统中信息处理的一个例子来处理。 传统上，热力学中的第二定律涉及一种称为“熵”的物理量。每个人都听说过熵，但很少有人真正理解它。直到最近，熵才被广泛接受为一种信息形式。

第二定律无疑是科学中最辉煌的成就之一，但通常通过物理系统和模型（如理想气体）来教授，很难在初级水平上欣赏它。 另一方面，适用于计算和通信（其中处理信息）的第二定律形式更容易理解，特别是在信息革命即将开始的今天。这些笔记和基于它们的课程是为大一的大学生设计的。 虽然它们是在麻省理工学院开发的，这是一所专门从事科学和工程的大学，并且要求学生学习一整年的微积分，但在这些笔记
中几乎不使用微积分。 大多数例子都来自离散系统，而不是连续系统，因此可以使用求和和差分，而不是积分和导数。 换句话说，我们可以使用代数而不是微积分。 这门课程对于不学习工程或科学的大学生来说应该是可接近的，甚至对于准备充分的高中生来说也是如此。 实际上，当与关于能量的类似课程结合在一起时，这门课程可以为文科学生提供出色的科学背景。

信息和能量之间的类比很有趣。 在高中和大学一年级的物理课程中，学生们了解到有一种被称为能量的物理量，它是守恒的（不能被创造或销毁），但可以以不同的形式出现（势能、动能、电能、化学能等）。能量可以存在于空间的一个区域或另一个区域，可以从一个地方流向另一个地方，可以储存以备将来使用，并且可以从一种形式转换为另一种形式。 从许多方面来看，工业革命就是为了利用能量进行有用的目的。 但最重要的是，能量是守恒的——尽管它被移动、储存和转换，但在一天结束时，总能量的数量仍然完全相同。

能量守恒原理有时被称为热力学第一定律。它被证明是如此重要和基础，以至于每当发现一个“泄漏”时，理论就会通过定义一种新形式的能量来挽救。 这种情况的一个例子发生在1905年，当阿尔伯特·爱因斯坦认识到质量是一种能量形式时，正如他著名的公式所表达的那样 $E = mc^2$。 这种理解后来使得开发出能够将能量从质量形式转换为其他形式的设备（原子弹和核电站）成为可能。

但是信息呢？ 如果熵真的是一种信息形式，那么应该有一个理论

涵盖两者并描述信息如何转化为熵或反之亦然。 这样的理论目前还不够完善，有几个历史原因。 然而，这正是简化教学和理解基本概念所需要的。

这些笔记提供了信息的统一视角，其中熵是一种信息的形式，但还有其他种类的信息。 像能量一样，信息可以存在于一个地方或另一个地方，可以通过空间传输，并可以储存以供以后使用。但与能量不同，信息不是守恒的：第二定律表明熵随着时间的推移永远不会减少，通常会增加，但在特殊情况下可能保持不变。 此外，信息本质上是主观的，因为它涉及你知道和不知道的事情（熵作为一种信息形式也是主观的，这一点使一些物理学家感到不安）。 这两个事实使得信息理论与处理能量等守恒量的理论不同，也更加有趣。

这里提出的统一框架从未专门为新生开发过。 它与各个学科中的传统思维并不完全一致，这些学科是分别发展起来的。 事实上，我们可能还没有完全理解。 在初级教育中的一个结果是，近在咫尺的未解之谜令人不安，并且解决它们需要对基础知识进行新的研究。

试图以严谨而简单的方式解释事物通常需要新的组织原则和新的方法。
在当前情况下，新的方法是从信息开始，然后从那里工作到熵，而新的组织原则是信息的统一理论。

这将是一次令人兴奋的旅程。欢迎加入！

第1章

比特

信息的度量单位是比特，就像长度以米为单位，时间以秒为单位一样。 当然，知道信息的数量，以比特为单位，并不等同于了解信息本身，它的意义或者暗示。 在这些笔记中，我们不考虑信息的内容或意义，只关注数量。

在不同的情况下，需要不同尺度的长度。 有时我们想用公里来测量长度，有时用英寸，有时用埃安斯特朗。 同样，在信息的其他尺度上，除了比特，有时也会使用其他尺度；在物理系统的背景下，信息通常以每开尔文焦耳为单位来衡量。

信息如何量化？ 考虑一个可能有几种可能结果的情况或实验。 例如，抛硬币（2种结果，正面或反面）或从一副扑克牌中选择一张牌（52种可能结果）。 一个人（通常被称为爱丽丝）如何简洁地告诉另一个人（鲍勃）这样一个实验或观察的结果？

首先考虑抛硬币的两种结果的情况，并假设它们是等可能的。
如果爱丽丝想告诉鲍勃硬币抛掷的结果，她可以使用几种可能的技术，但从信息传达的数量上来说，它们都等效于说“正面”或“反面”或说0或1。 我们称这样传达的信息为一比特。

如果爱丽丝抛两个硬币，她可以通过说0或1两次来表明实际发生的四种可能结果。 类似地，具有八种等可能结果的实验的结果可以用三个比特来传达，而更一般地，具有2^n个结果的实验可以用 n个比特来传达。 因此，信息的数量是等可能结果的对数（以2为底）。

请注意，传达信息需要两个阶段。 首先是“设置”阶段，在此阶段中，爱丽丝和鲍勃就他们将要交流的内容以及每个比特序列的确切含义达成一致。 这种共同的理解被称为编码。 例如，为了传达从一副牌中选择的牌的花色，他们的编码可能是00表示梅花，01表示方块，10表示红心，11表示黑桃。 在任何观察之前，就可以达成关于编码的一致。因此，尚未发送任何信息。 设置阶段可以包括通知接收者有新信息的情况。 然后，进入“结果”阶段，发送实际的0和1序列，表示结果。 这些序列是数据。 使用约定的编码，爱丽丝抽取牌，并通过发送两个数据比特告诉鲍勃花色。 她可以重复这样做，进行多次实验，使用相同的编码。

在鲍勃得知一张卡被抽出来之后，但在收到爱丽丝的消息之前，他对花色感到不确定。
他的不确定性或缺乏信息可以用比特来表示。 听到结果后，他接收到的信息减少了他的不确定性。 鲍勃的不确定性在设置阶段上升，然后在结果阶段下降。

请注意关于信息的一些重要事项，其中一些在此示例中说明：

- 信息可以通过观察、实验或测量来学习
- 信息是主观的，或者说是“观察者相关的”。爱丽丝知道的与鲍勃知道的不同（如果信息不是主观的，就不需要传达它）
- 一个人的不确定性可以在得知可能有信息可用的观察后增加，然后通过接收该信息来减少
- 信息可以丢失，无论是通过数据本身的丢失还是通过代码的丢失
- 信息的物理形式是局限在空间和时间中的。因此，
  - –信息可以从一个地方发送到另一个地方
  - –信息可以被存储然后以后检索

1.1 布尔比特

正如我们所见，信息可以通过0和1的序列进行传递。 通过仅使用0和 1，我们可以处理来自许多不同类型来源的数据，并不关心数据的含义。 因此，我们使用的是抽象而不是具体的值。 这种方法使我们能够忽略与具体信息处理和传输系统相关的许多混乱细节。

比特是简单的，只有两个可能的值。 用于表示和操作单个
比特的数学并不困难。 它被称为布尔代数，以数学家乔治·布尔（1815-1864）命名。
在某些方面，布尔代数类似于在高中教授的整数或实数代数，但在其他方面又有所不同。

代数是数学的一个分支，处理具有一定可能值的变量和当给定一个或多个变量时返回具有一定可能值的结果的函数。 在布尔代数的情况下，可能的值是0和1。

首先考虑返回单个值的单变量布尔函数。 共有四个这样的函数。 其中一个被称为恒等函数，简单地返回其参数。 另一个被称为非（或否定、反转或补充）将0变为1，将1变为0。 另外两个无论参数如何都只返回0或1。 下面是显示这四个函数的表格：

x	$f(x)$			
参数	恒等	非	零	一
0	0	1	0	1
1	1	0	0	1

表1.1：单变量布尔函数

请注意，布尔代数比处理整数或实数的代数更简单，每个都有无限多个单变量函数。

接下来，考虑具有两个输入变量 A和 B以及一个输出值 C的布尔函数。有多少个？ 这两个参数中的每一个都可以取两个值中的任意一个，因此有四种可能的输入模式（00、01、10和11）。 将两个变量的每个布尔函数视为长度等于可能的输入模式数量（即4）的布尔值字符串，即0和1。 有确切的16种（2^4）不同的组合这样的字符串的方式，因此有确切的16种不同的两个变量的布尔函数。 在这16个函数中，有两个函数忽略输入，四个函数将输出分配为 A或 B或它们的补集，其他十个函数则依赖于两个参数。 最常用的是 AND、 OR、 XOR（异或）、 $NAND$（非与）

x	$f(x)$				
参数	AND	$NAND$	OR	$N\ OR$	异或
00	0	1	0	1	0
01	0	1	1	0	1
10	0	1	1	0	1
11	1	0	1	0	0

表1.2：五个 16 可能的两个变量的布尔函数

和)，和非或 (非或)，如表1.2所示。(类似地，因为三输入布尔函数有8种可能的输入模式，所以有2^8或256个不同的三变量布尔函数。)诱人的是将布尔值0和1看作整数0和1。然后与对应于乘法，或对应于加法，有点像。然而，普通代数的熟知结果在布尔代数中根本不成立，因此这样的类比是危险的。重要的是要区分整数0和1与布尔值0和1；它们并不相同。

布尔代数中有一种标准符号。(这种符号有时令人困惑，但其他不那么令人困惑的符号在实践中很别扭。)与函数用相同的方式表示乘法，通过将两个布尔值写在一起或用一个点隔开： $A\ AND\ B$写作 AB或 $A\ B$。 与函数使用加号表示： $A + B$表示$A\ OR\ B$。否定，或者$N\ OT$函数，用一个横线覆盖符号或表达式表示，所以$N\ OT\ A$是 A。 最后，异或函数 $X\overline{O}R$用一个圆圈内有一个加号的符号表示， $A \oplus B$。

$$
\begin{array}{rl}
N\ OT & \overline{A} \\
AND & A \cdot B \\
NAND & \overline{A \cdot B} \\
OR & A + B \\
N\ OR & \overline{A + B} \\
XOR & A \oplus B
\end{array}
$$

表1.3：布尔逻辑符号

布尔代数还有其他可能的表示方法。 这里使用的是最常见的一种。 有时候 AND、 OR和$N\ OT$以 $AND(A, B)$、 $OR(A, B)$和 $NOT(A)$的形式表示。 有时候使用中缀表示法，其中 $A \wedge B$表示 $A\ B$， $A \vee B$表示 $A + B$，而 $\sim A$表示 A。布尔代数在数学逻辑中也很有用，其中符号 $A \wedge B$表示 $A\ B$，符号 $A \vee B$表示$\cdot A + B$，符号 A表示 A $\neg$ 常用。

布尔函数的几个一般性质很有用。 这些性质可以通过简单地证明它们对所有可能的输入值都成立来证明。 例如，如果在知道输出的情况下可以确定输入，则称该函数是可逆的。 在这个意义上，单变量的四个函数中有两个是可逆的（实际上它们是自反的）。 显然，任何两个（或更多）输入的函数本身都不可逆，因为输入变量比输出变量多。 然而，如果组合的结果具有与输入相同数量的输出，那么某些两个或更多这样的函数的组合可以是可逆的；例如，很容易证明异或函数 $A \oplus B$在增加返回第一个参数的函数的情况下是可逆的——换句话说，更准确地说，是具有两个输出的两个变量的函数，一个是 $A \oplus B$，另一个是 A，是可逆的。

对于两个变量的函数，有许多要考虑的属性。 例如，如果一个由两个变量A和B组成的函数在交换A和B的位置时其值不变，则称其为可交换的，即如果$f(A, B) = f(B, A)$。 因此，函数AND是可交换的，因为$A\ B = B\ A$。其他15个函数也是可交换的。布尔函数的一些其他属性在表1.4中有所说明。 布尔函数的一些其他属性在表1.4中有所说明。

幂等性:	$A \cdot A = A$ $A + A = A$	吸收:	$A \cdot (A + B) = A$ $A + (A \cdot B) = A$
互补性:	$A \cdot \overline{A} = 0$ $A + \overline{A} = 1$ $A \oplus A = 0$ $A \oplus \overline{A} = 1$	结合性:	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$ $A + (B + C) = (A + B) + C$ $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
最小值:	$A \cdot 1 = A$ $A \cdot 0 = 0$	未命名定理:	$A \cdot (\overline{A} + B) = A \cdot B$ $A + (\overline{A} \cdot B) = A + B$
最大值:	$A + 0 = A$ $A + 1 = 1$	德摩根定理:	$\overline{A} \cdot \overline{B} = \overline{A + B}$ $\overline{A} + \overline{B} = \overline{A \cdot B}$
交换律:	$A \cdot B = B \cdot A$ $A + B = B + A$ $A \oplus B = B \oplus A$ $\overline{A \cdot B} = \overline{B \cdot A}$ $\overline{A + B} = \overline{B + A}$	分配律:	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $A + (B \cdot C) = (A + B) \cdot (A + C)$

表1.4：布尔代数的属性
这些公式对于所有的 A, B和 C都是有效的。

布尔位具有可以复制的属性（也可以丢弃）。 在布尔代数中，通过给位赋予一个名称，然后多次使用该名称来进行复制。 由于这个属性，布尔位不是量子力学系统的好模型。 一个不同的模型，量子位，将在下面描述。

1.2 电路比特

组合逻辑电路是一种以图形方式表示布尔表达式的方法。 每个布尔函数（非，与，异或等）对应于一个具有一个或两个输入和一个输出的“组合门”，如图1.1所示。不同类型的门具有不同的形状。 线用于连接一个门的输出到一个或多个门的输入，如图1.2中的电路所示。逻辑电路广泛用于建模数字电子电路，其中门代表集成电路的部分，线代表信号线路。

电路比特可以被复制（通过将一个门的输出连接到两个或多个门的输入）和

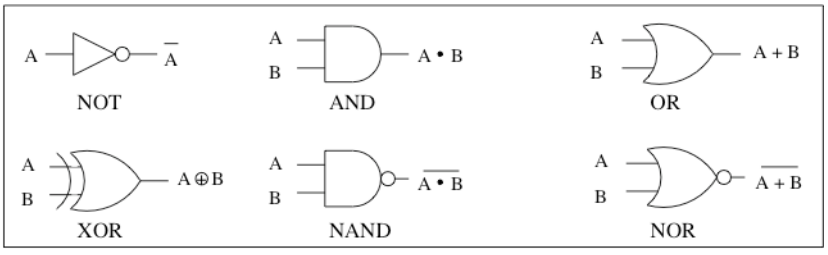


图 1.1：逻辑门对应的布尔函数$N\ OT$, AND, OR, XOR, $NAND$和$N\ OR$

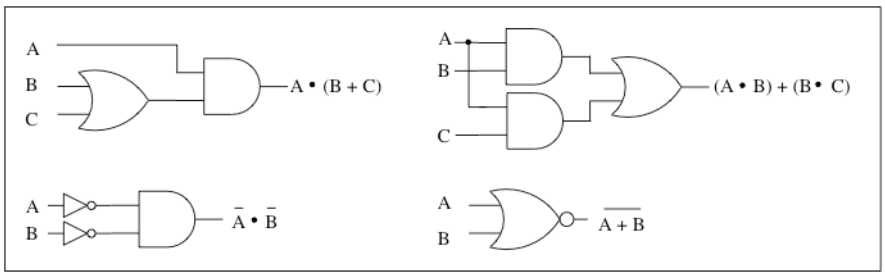


图1.2：一些组合逻辑电路和相应的布尔表达式

被丢弃（通过保持输出未连接）。

组合电路具有这样的特性，即从一个门的输出到达的输入永远不会反馈到最终将输入馈送到第一个门的任何门的输入上。 换句话说，电路中没有循环。 具有循环的电路被称为时序逻辑，布尔代数无法描述它们。 例如，考虑图1.3中最简单的电路。反相器（NOT门）的输出连接到其输入。 通过布尔代数的分析导致矛盾。 如果输入为1，则输出为0，因此输入为0。 根据布尔代数的规则，没有可能的状态。

另一方面，考虑图1.3中的电路，其中有两个反相器。 这个电路有两种可能的状态。
如果底部电路有偶数个门，它有两个稳定状态；如果有奇数个门，则没有稳定状态。

需要一个比布尔代数更复杂的模型来描述时序逻辑电路的行为。 例如，门或连接线（或两者）可以用时间延迟建模。 图1.3底部的电路（例如，具有13或15个门）通常被称为环形振荡器，并用于半导体工艺开发中测试使用新工艺制造的电路的速度。

1.3 控制比特

在计算机程序中，布尔表达式经常用于确定控制流，即执行哪些语句。 例如，假设一个变量 x为负数，另一个变量 y为正数，则第三个变量 z 应该设置为零。 在Scheme语言中，以下语句可以实现这一点：`(if (and (< x 0) (> y 0)) (define z 0))`（其他语言有自己表达相同内容的方式）。

控制位的代数与布尔代数类似，但有一个有趣的区别：控制表达式中不影响结果的部分可以被忽略。 在上面的情况下（假设 `and`的参数从左到右进行评估），如果 x被发现为正，则 `and`操作的结果无论 y的值如何都为0，因此无需查看 y是否为正甚至评估 y。 因此，程序可以运行得更快，并且与评估 y相关的副作用不会发生。

1.4 物理位

如果要存储或传输位，它必须具有物理形式。 存储位的任何对象都有两个不同的状态，其中一个被解释为0，另一个被解释为1。 通过将对象放入这些状态之一来存储位，并且在需要位时测量对象的状态。 如果对象在不改变其状态的情况下从一个位置移动到另一个位置，则发生了通信。 如果对象在一段时间内保持其相同的状态，则它充当了一种记忆。 如果对象的状态以随机方式改变，则其原始值已被遗忘。

与工程要求相一致（使其更小、更快、更强、更智能、更安全、更便宜）

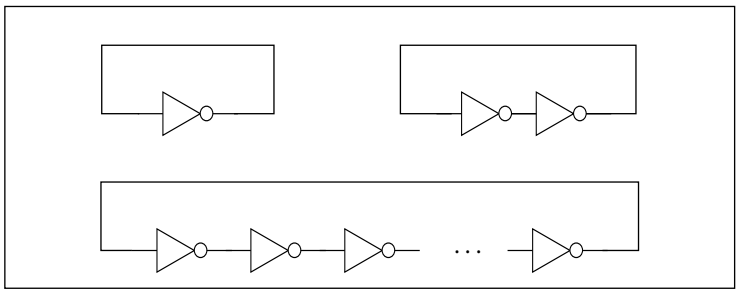


图1.3：一些时序逻辑电路

我们对尺寸较小的物体特别感兴趣。 一个物体能够存储一比特信息的最小尺寸限制来自于量子力学。 量子比特，或称量子位（qubit），是一个能够存储单个比特信息的模型，但其尺寸非常小，受到量子力学对测量的限制。

1.5 量子比特

根据量子力学，一个小物体可以具有两种可以测量的状态。 这听起来非常适合存储比特，结果通常被称为量子位（qubit），发音为“cue-bit”。这两个值通常用$|0\rangle$和$|1\rangle$表示，而不是0和1，因为这种表示法可以推广到表示多个量子位所需的情况，并且避免与实数0和1混淆。 量子力学有三个特征，即可逆性、叠加性和纠缠性，这使得量子位或量子位集合与布尔比特不同。

可逆性：量子系统的一个特性是，如果一个状态可以通过某种转换导致另一个状态，那么反向转换也是可能的。 因此，量子位的数学中的所有函数都是可逆的，函数的输出不能被丢弃，因为那将是不可逆的。 然而，在量子系统中至少有两个重要的不可逆性来源。 首先，如果量子系统与其环境相互作用，而环境的状态是未知的，那么系统中的一些信息将会丢失。 其次，测量系统状态的行为本身是不可逆的。

叠加：假设一个量子力学对象被准备成具有其两个状态的组合，即介于两个状态之间的状态。 在这种情况下，将会测量到什么？

在经典的非量子环境中，测量可以确定这种组合是什么。此外，为了更大的精度，可以重复测量，并对多个结果进行平均。 然而，在量子环境中情况不同。 在量子测量中，所问的问题是对象是否处于某个特定状态，答案只能是“是”或“否”，永远不会是“可能”或者例如“27%是，73%否”。此外，在测量之后，系统将处于与答案对应的状态，因此进一步的测量将不会提供额外的信息。 任何特定测量的结果都无法预测，但答案的可能性可以用概率来表示。量子力学的这种奇特性质既限制了单个量子位所能携带的信息量，也为设计能够充分利用这些特性的系统提供了机会。

我们将用一个例子来说明量子位。 让我们把光子作为我们的量子位，光子是电磁辐射的基本粒子，包括无线电、电视和光。 光子是从一个地方传递信息到另一个地方的良好候选者。 它很小，而且传播速度很快。

光子具有同时振荡的电场和磁场。 电场的方向被称为偏振方向（我们这里不考虑圆偏振光子）。因此，如果一个光子

朝着 z-方向前进，它的电场可以在 x-方向、 y-方向，或者实际上在 x-y平面的任何方向，有时被称为“水平-垂直平面”。

偏振可以用来存储一位信息。 因此，如果位是 $|0\rangle$，爱丽丝可以准备一个具有水平偏振的光子，如果位是 $|1\rangle$，她可以准备一个具有垂直偏振的光子。 然后当鲍勃得到光子时，他可以测量它的垂直偏振（即询问偏振是否垂直）。 如果答案是“是”，那么他推断位是 $|1\rangle$。

可能会认为一个光子的极化可以传输多于一个比特的信息。 为什么爱丽丝不能使用与水平和垂直不同的极化角度发送两个比特呢？ 为什么不能使用水平、垂直、倾斜向右的中间角度和倾斜向左的中间角度呢？ 问题在于鲍勃必须决定要测量的角度。 由于量子力学的限制，他不能询问“极化角度是多少”，而只能询问“极化是否在我选择的方向上”。他的测量结果只能是“是”或“否”，换句话说，只有一个比特。然后，在测量之后，光子要么停留在他测量的平面上（如果结果是“是”），要么垂直于它（如果结果是“否”）。

如果鲍勃想更准确地测量极化角度，为什么他不能重复测量多次并取平均值呢？ 这不起作用，因为进行第一次测量的行为会将极化角度重置为他测量的角度或垂直于它的角度。因此，随后的测量结果都将相同。

或者鲍勃可能决定制作光子的多个副本，然后对每个副本进行测量。 这种方法也不起作用。 他唯一能够复制光子的方法是测量其属性，然后创建一个具有完全相同属性的新光子。 他创建的所有光子都将是相同的。

如果爱丽丝用任意角度准备了光子，鲍勃测量的是什么？ 或者如果光子的偏振角度因途中的随机相互作用而改变了呢？ 或者如果光子在某个其他角度被邪恶的窃听者（通常称为伊夫）测量，并因此被重置为该角度呢？ 在这些情况下，无论鲍勃选择测量的偏振方向如何，他总是得到一个“是”或“否”的答案，实际偏振与该方向越接近，答案是“是”的可能性就越大。具体而言，答案是“是”的概率是鲍勃的测量角度与爱丽丝的准备角度之间余弦的平方。 无法预测鲍勃的任何一次测量结果。 这种固有的随机性是量子力学中不可避免的一部分。

纠缠：两个或更多个量子比特可以以特定的方式一起准备。 一个性质，我们现在不会进一步讨论，被称为“纠缠”。例如，两个光子可能具有相同的偏振（水平或垂直）。 然后它们可能分别前往不同的地方，但保持它们纠缠的偏振。 它们在物理位置上是分开的，但在偏振上不是分开的。 如果你将它们视为两个独立的光子，你可能会想知道为什么测量一个光子的偏振会影响到远处的另一个光子的偏振的测量。

请注意，量子系统并不总是表现出与叠加和纠缠相关的奇特性。 例如，光子可以独立准备（因此没有纠缠），并且偏振角可以被限制为水平和垂直（没有叠加）。 在这种情况下，量子比特的行为类似于布尔比特。

1.5.1 量子比特的优势

在量子环境中可以做一些经典环境无法做到的事情，其中一些是有优势的。
这里有一个例子：

再次考虑艾丽斯试图使用极化光子向鲍勃发送信息。 她准备了具有水平或垂直极化的光子，并在设置阶段告诉鲍勃。 现在假设一个破坏者山姆想要通过在艾丽斯和鲍勃之间的某个位置处理光子来破坏这种通信。他使用一台简单地测量所选角度的极化的机器。 如果他选择$45°$，每个光子最终都会具有该角度或垂直于该角度的极化。

无论其原始极化如何。 然后鲍勃进行垂直测量，无论爱丽丝发送了什么，他将一半的时间测量为0，一半的时间测量为1。

爱丽丝了解了山姆的计划，并希望重新与鲍勃建立可靠的通信。她能做什么?

她告诉鲍勃（使用山姆无法听到的路径）她将以45°和135°的角度发送光子，所以他应该在这些角度之一进行测量。 山姆的机器不会改变光子。 当然，如果山姆发现爱丽丝在做什么，他可以将他的机器旋转回垂直位置。或者还可以采取其他措施和对策。

这种情况依赖于光子的量子性质，以及单个光子除了沿着特定的偏振角度测量之外，山姆无法测量。 因此，爱丽丝阻止山姆的技术在经典位上是不可能的。

1.6 经典比特

由于量子测量通常会改变被测量的对象，量子位不能被第二次测量。 另一方面，如果一个位由具有相同属性的许多对象表示，那么在测量之后，可以保留足够的对象不变，以便可以再次测量相同的位。

在今天的电子系统中，信息的一位由许多以相同方式准备的对象携带（或者至少这是一种方便的看法）。 因此，在半导体存储器中，一个单独的位由大约60,000个电子的存在或不存在来表示（存储在充电至$1V$的10 fF电容器上）。 类似地，在无线通信中使用大量的光子。

因为涉及到许多物体，对它们的测量不仅仅局限于简单的是或否，而是可以涵盖一系列的值。 因此，半导体逻辑元件上的电压可以在从$0V$到$1V$的范围内的任何位置。 电压可以被解释为允许一定的误差范围，因此在$0V$到$0.2V$之间的电压表示逻辑0，在$0.8V$到$1V$之间的电压表示逻辑1。 电路不能保证正确解释$0.2V$到$0.8V$之间的电压。 如果电路中的噪声始终小于$0.2V$，并且每个电路门的输出要么是$0V$，要么是$1V$，那么电压始终可以被解释为没有错误的位。

这种电路显示出所谓的“恢复逻辑”，因为在处理信息时，电压与理想值$0V$和$1V$的微小偏差被消除。 现代计算机的稳健性取决于恢复逻辑的使用。

经典比特是一种抽象，可以在不干扰其状态的情况下测量比特。因此，可以制作经典比特的副本。 这个模型适用于使用恢复逻辑的电路。

由于所有物理系统最终都遵循量子力学，经典比特始终是对现实的近似。 然而，即使使用最现代、最小的设备，它仍然是一个很好的近似。一个有趣的问题是，随着半导体技术的进步，组件的尺寸是否会继续缩小，经典比特的近似是否仍然有用。 最终，当我们试图用少量的原子或光子来表示或控制比特时，量子力学的限制作用将变得重要。 很难准确预测这将在何时发生，但有些人认为这将在2015年之前发生。

1.7 总结

有几种比特模型，在不同的上下文中很有用。 这些模型并不都相同。 在接下来的笔记中，布尔比特将经常使用，但有时需要量子比特。

第二章

编码

在前一章中，我们研究了信息的基本单位比特，以及它的各种抽象表示：布尔比特（及其相关的布尔代数和组合逻辑电路实现）、控制比特、量子比特和经典比特。

如果一个问题只有两个可能的答案，那么单个比特是有用的。 例如，抛硬币的结果（正面或反面）、一个人的性别（男性或女性）、陪审团的裁决（有罪或无罪）以及一个命题的真假。 生活中的大多数情况都更加复杂。 本章讨论的是如何用比特数组来表示复杂对象。

方便起见，我们可以关注一个非常简单的系统模型，如图2.1所示。在这个模型中，输入是预先确定的一组对象或“符号”，所选择的特定符号的身份被编码在一组位数组中，这些位通过空间或时间传输，然后在稍后的时间或不同的地方解码，以确定最初选择的符号。 在后面的章节中，我们将扩展这个模型，以处理鲁棒性和效率问题。

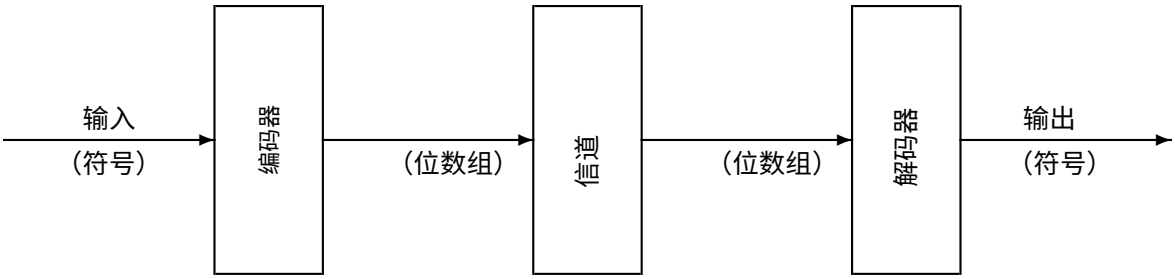


图2.1：通信系统的简单模型

在本章中，我们将研究代码设计的几个方面，并展示一些设计得好或不太好的示例。 各个部分将描述说明重要观点的代码。 可能需要代码的一些对象包括：

- 字母： BCD、 EBCDIC、 ASCII、 Unicode、莫尔斯电码
- 整数：二进制、格雷码、二补数

- 数字：浮点数
- 蛋白质：遗传密码
- 电话：NANP、国际代码
- 主机：以太网、IP地址、域名
- 图像：TIFF、GIF和JPEG
- 音频：MP3
- 视频：MPEG

2.1 符号空间大小

首先要解决的问题是需要编码的符号数量。 这被称为符号空间大小。 我们将考虑不同大小的符号空间：

- 1
- 2
- 2的整数次幂
- 有限的
- 无限可数的
- 无限不可数的

如果符号数量为2，则可以用一个比特来编码选择。如果可能的符号数量为4、8、16、32、64或其他2的整数次幂，则选择可以用等于符号空间大小的以2为底的对数的比特数来编码。 因此，2个比特可以指定一张扑克牌的花色（梅花、方块、红心或黑桃），而5个比特可以编码在32个学生中选择一个学生。 作为特例，如果只有一个符号，则不需要任何比特来指定它。陀螺是一种带有希伯来字母标记的四面体玩具，像陀螺一样在儿童游戏中旋转，尤其是在光明节。每次旋转的结果可以用2个比特来编码。

如果符号的数量是有限的，但不是2的整数幂，则可以使用下一个更高的2的整数幂所需的位数来编码选择，但会有一些未使用的位模式。 例如，10个数字，一个立方体骰子的六个面，一副扑克牌的13个面值和英文字母表的26个字母。 在每种情况下，都有多余的容量（4位数字表示中的6个未使用模式，3位骰子表示中的2个未使用模式等）。如何处理这些多余的容量是一个重要的设计问题，将在下一节中讨论。

如果符号的数量是无限的但可数的（能够与整数建立一对一的关系），那么给定长度的比特串只能表示这个无限集合中的有限数量的项。因此，用于非负整数的4位编码可能指定从0到15的整数，但无法处理超出此范围的整数。 如果由于某种计算需要表示更大的数字，则必须以某种方式处理这种“溢出”条件。

如果符号的数量是无限且不可数的（例如物理量如电压或声压的值），那么必须使用某种“离散化”技术，将可能的值替换为一些近似相同的有限数量的选定值。 例如，如果0到1之间的数字是符号，并且可用于编码表示的位数为2位，一种方法可能是通过数字0.125来近似表示0到0.25之间的所有数字，通过数字0.375来近似表示0.25到0.5之间的所有数字，依此类推。这样的近似是否足够取决于解码数据的使用方式。

这种近似是不可逆的，因为没有解码器可以仅凭近似值的编码来恢复原始符号。 然而，如果可用的位数足够多，那么对于许多目的来说，解码器可以提供一个足够接近的数字。 计算机中实数的浮点表示就是基于这个原理的。

2.2 备用容量的使用

在许多情况下，存在一些未使用的代码模式，因为符号的数量不是2的整数次幂。 有许多处理这个问题的策略。以下是一些例子：

- 忽略
- 映射到其他值
- 保留以备将来扩展
- 用于控制代码
- 用于常见缩写

这些方法将通过常见代码示例进行说明。

2.2.1 二进制编码十进制（BCD）

表示数字0-9的常见方法是使用表2.1中显示的十个四位模式。有六个位模式（例如1010）没有使用，问题是如何处理它们。 以下是一些想法。

首先，可以简单地忽略未使用的位模式。 如果解码器遇到一个未使用的位模式，可能是由于传输错误或编码错误导致的，它可以返回空值，或者可能会发出输出错误信号。
其次，未使用的模式可以映射到合法的值。 例如，未使用的模式可以全部转换为9，理论上它们表示10、11、12、13、14或15，而最接近的数字是9。
或者它们可以被解码为2、3、4、5、6或7，通过将初始位设置为0，在理论上，第一个位可能已经损坏。 这些理论都不是特别吸引人，但在使用BCD的系统设计中，必须提供一些这样的操作。

数字	代码
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

表2.1：二进制编码十进制

2.2.2 遗传密码

遗传密码提供了将未使用的模式映射为合法值的另一个例子，该密码在第2.7节中描述。蛋白质由20种不同类型的氨基酸组成，每种氨基酸含有10至27个原子。 生物体有数百万种不同的蛋白质，人们认为所有细胞活动都涉及蛋白质。 蛋白质必须作为生命过程的一部分制造出来，但很难想象有数百万个专门用于制造每种蛋白质的化学制造单元。 相反，一个通用的机制通过DNA（脱氧核糖核酸）和RNA（核糖核酸）分子中包含的描述（将其视为蓝图）来组装蛋白质。 DNA和RNA都是由小的“核苷酸”线性链组成的；一个DNA分子可能由一亿多个这样的核苷酸组成。 在DNA中，有四种类型的核苷酸，每种核苷酸都由一种共同的结构和四种不同的碱基（腺嘌呤、胞嘧啶、鸟嘌呤和胸腺嘧啶）组成。在RNA中，结构类似，只是胸腺嘧啶被尿嘧啶取代。

遗传密码是描述核苷酸序列如何指定氨基酸的方式。 鉴于这种关系，整个蛋白质可以由一系列核苷酸线性序列来指定。 请注意，蛋白质的编码描述本身并不比蛋白质本身更小或更简单；实际上，用于指定蛋白质的原子数量大于蛋白质本身的原子数量。 标准化表示的价值在于它允许相同的装配设备在不同的时间制造不同的蛋白质。

由于有四种不同的核苷酸，其中之一最多可以指定四种不同的氨基酸。 两个序列可以指定16种不同的氨基酸。 但这还不够-蛋白质中使用了20种不同的氨基酸-因此需要一个由三个序列组成的序列。 这样的序列被称为密码子。 有64个不同的密码子，远远超过了指定20种氨基酸所需的数量。 多余的容量用于为大多数氨基酸提供多种组合，从而提供了一定程度的稳健性。 例如，氨基酸丙氨酸有4个编码，包括所有以GC开头的编码；因此第三个核苷酸可以忽略，因此改变它的突变不会损害任何生物功能。 实际上，20种氨基酸中有8种具有相同的特性，即第三个核苷酸是“不关心”的。（碰巧第三个核苷酸在转录过程中更容易受损，这是一种被称为“摇摆”的效应。）

对遗传密码的研究发现，三个密码子（UAA、UAG和UGA）不指定任何氨基酸。 这三个密码子表示蛋白质的结束。 这种“停止密码”是必要的，因为不同的蛋白质长度不同。 密码子AUG指定氨基酸甲硫氨酸，并且表示蛋白质的开始；所有蛋白质链都以甲硫氨酸开始。许多人造代码具有这个特性，即一些位序列用于指定数据，而少数位序列用于控制信息。

2.2.3 电话区号

利用备用容量的第三种方式是将其保留用于未来扩展。 当AT&T在1947年开始为美国和加拿大使用电话区号（在1951年对公众开放使用）时，区号包含三个数字，并有三个限制。

- 第一个数字不能为0或1，以避免与0连接到操作符产生冲突，以及1成为粘性旋转拨号器或未知原因的临时断路的意外效果（或者今天是拨号人承认呼叫可能是收费呼叫的信号）

- 中间的数字只能是0或1（0表示只有一个区号的州和省，1表示有多个区号的州和省） 这个限制使得区号可以与交换机（交换机是当时用一个词的前两个字母和一个数字表示的设备，用于切换多达10,000个电话号码；今天交换机用三位数表示）区分开来

- 最后两位数字不能相同（形式为 abb的数字更容易记住，因此更有价值）—因此 $x11$拨号序列如911（紧急呼叫），411（目录

对于本地服务，411（信息查询），511（交通信息），611（维修服务）受到保护。这也允许后来采用500（跟随我），600（加拿大无线），700（互联服务），800（免费电话）和900（增值信息服务）。

结果只有144个区号是可能的。 最初使用了86个区号，并且分配给了拨号盘上更快拨号的地区（例如曼哈顿的212区号）。剩下的58个区号保留以供以后分配。

这58个新区号的数量足够使用四十多年。 最后，当需要超过144个区号时，通过放宽中间数字只能为0或1的限制来创建新的区号。 1995年1月15日，第一个中间数字不为0或1的区号在阿拉巴马州投入使用。目前对区号的限制是第一个数字不能为0或1，中间数字不能为9，最后两位数字不能相同。 截至2000年初，已经启动了108个新的区号，这种巨大需求部分是由于电话网络用于传真和手机等其他服务的扩展使用，部分是由于加勒比岛屿等地方政府的政治压力，他们希望拥有自己的区号，部分是由于大量新的电话公司提供服务，因此每个费率计费区至少需要一个完整的交换机。

有些人认为北美编号计划（NANP）在2025年之前将耗尽区号，并且有各种提议来解决这个问题。

考虑到北美的每个电话交换所都需要升级，无论是在修订软件方面还是在某些情况下的新硬件方面，1995年的过渡非常顺利。 总体而言，公众并没有意识到这个变化的重要性。 这是由于北美电话服务的整体高质量以及行业的紧密协调。 唯一的问题似乎是一些由独立供应商设计的PBX（私人分支交换机）没有及时升级。 自1995年以来，北美的电信行业发生了很大变化：现在它拥有较少的中央控制、更多的竞争以及更多种类的服务提供。 未来编号计划的变化肯定会给公众带来更大的混乱和不便。

2.2.4 IP地址

需要为将来的使用保留容量的另一个例子是IP（Internet Protocol）地址，在第 2.8节中有描述。这些地址（在版本 4中）的格式为 x.x.x.x其中每个 x是 0到255之间的数字，包括0和255。 因此，每个互联网地址可以用总共32位编码。 IP地址由互联网分配号码管理机构（IANA）分配，网址为http://www.iana.org/。

对于互联网的兴趣爆炸式增长，导致对IP地址的需求大幅增加，参与互联网发展的组织，他们被分配了大量的号码，开始感觉自己在囤积一种有价值的资源。 这些组织包括AT&T，BBN，IBM，Xerox，HP，DEC，Apple，MIT，Ford，Stanford，BNR，Prudential，duPont，Merck，美国邮政服务，以及几个美国国防部机构（见第2.8节）。美国电力行业以EPRI（Electric Power Research Institute）的形式，请求了大量的互联网地址，用于每个可计费的家庭或办公套房，以供远程抄表设备最终使用。 互联网工程任务组（IETF），网址为http://www.ietf.org/，意识到互联网地址在更广泛的范围内是必需的。

并且比最初设想的更细致的尺度上，例如，当这些设备连接到互联网时，冰箱、烤箱、电话和炉子等家电将需要地址，每辆汽车和卡车内可能需要多个地址，每个微处理器和传感器可能需要一个地址。 结果就是发展出了第6版IPv6，其中每个地址仍然是形如 x.x.x.x的格式，但每个 x现在是一个介于0和4,294,967,295之间的32位数字。 因此，新的互联网地址将需要128位。 现有的地址不需要改变，但所有的网络设备都需要改变以适应更长的地址。 新的分配包括为未来扩展保留的大块地址，据说（幽默地）还有一些地址块专门留给其他星球使用。 地址空间的大小足够容纳每台个人电脑的唯一硬件标识符，一些隐私倡导者指出IPv6可能会使匿名上网变得不可能。

2.2.5 ASCII

代码中备用容量的第四种用途是用于表示格式或控制操作。
许多代码包含的代码模式不是数据，而是控制代码。 例如，遗传密码中有三种64种模式作为停止代码来终止蛋白质的产生。

用于文本字符的最常用的代码是ASCII（美国信息交换标准代码，在第2.5节中描述），它明确保留了其128个代码中的33个用于控制，而仅有95个用于字符。 这95个字符包括英文字母的26个大写字母和26个小写字母，10个数字，空格和32个标点符号。

2.3 码的扩展

许多代码是由人类设计的。 有时代码非常强大，简单易用，并且可扩展。 有时它们是脆弱的，晦涩的，复杂的，甚至无法简单概括。
通常，为表示少量项目而开发了一个简单实用的代码，其成功引起了人们的关注，并开始在原始上下文之外使用它，以表示更大类别的对象，用于最初未预料到的目的。

常规化的代码往往会带有其原始上下文中的意外偏见。 有时候结果只是有趣，但在其他情况下，这些偏见会使代码难以处理。

一个相对无害的偏见的例子是ASCII有两个最初被忽略的字符。 ASCII最初是纸带上的7位孔模式，用于在电传打字机之间传输信息。 纸带最初没有孔（除了一系列始终存在的小孔，用于对齐和送纸），并通过冲孔机传送。 纸带可以通过接收到的传输或人工键盘输入进行冲孔。 这种冲孔操作产生的碎屑被称为“碎屑”。领导者（纸带的第一部分）没有被冲孔，因此实际上代表了一系列未确定长度的字符。 0000000当然，在读取纸带时应该忽略领导者，所以按照惯例字符应该被忽略。 0000000被称为NUL并被忽略。

后来，当ASCII在计算机中使用时，不同的系统对NUL的处理方式也不同。 Unix在某些情况下将NUL视为单词的结尾，这种用法会干扰字符被赋予数值解释的应用程序。 另一个最初被忽略的ASCII码是DEL。 1111111. 这种约定对于打字员来说非常有帮助，他们可以通过倒带磁带并打出每个孔来“擦除”错误。 在现代环境中，DEL通常被视为破坏性的退格键，但过去的一些文本编辑器使用DEL作为向前删除字符，有时它也会被忽略。

ASCII所带来的一个更严重的偏见是使用两个字符CR（回车）和LF（换行）来换行。 电传打字机中的物理机制具有单独的硬件来将纸张（在连续卷上）向上移动，并将打印元素重新定位到左边距。 设计演变为ASCII的代码的工程师们肯定认为通过允许单独调用这些操作，他们正在做一件好事。 他们无法想象ASCII适应具有不同硬件且不需要按照CR或LF单独调用打印点的情况时，给后代带来的痛苦。 不同的计算系统以不同的方式处理事情 - Unix使用LF进行换行并忽略CR，Macintosh（至少在OS X之前）使用CR并忽略LF，而DOS / Windows则需要两者。 这种不兼容性是持续的、严重的挫折和错误的根源。 例如，在使用FTP（文件传输协议）传输文件时，CR和LF应该根据文本文件的目标平台进行转换，但对于二进制文件则不需要。 一些FTP程序可以根据文件扩展名（文件名中最后一个句点后的部分）推断文件类型（文本或二进制）。

其他人会查看文件并计算“有趣字符”的数量。其他人则依赖于人工输入。 这些技术通常有效，但并非总是如此。 文件扩展名的约定并非普遍遵循。人们会犯错误。 如果文件的一部分是文本，一部分是二进制，会怎么样？

2.4 固定长度和可变长度编码

在设计编码时，必须在早期做出一个决策，即是将所有符号用相同位数的编码（固定长度）表示，还是让一些符号使用比其他符号更短的编码（可变长度）。 这两种方案都有优势。

固定长度编码通常更容易处理，因为编码器和解码器都事先知道涉及的位数，只需设置或读取值即可。对于可变长度编码，解码器需要一种方法来确定一个符号的编码何时结束，下一个符号的编码何时开始。

固定长度编码可以通过并行传输来支持，即位从编码器同时传输到解码器，例如使用多根导线传输电压。 这种方法应与串行传输编码信息相对比，在串行传输中，单根导线发送一串位，解码器必须决定一个符号的位何时结束，下一个符号的位何时开始。 如果解码器混淆了或在开始后查看了一串位，它可能无法知道。 这被称为“帧错误”。为了消除帧错误，在符号之间通常发送停止位；通常在串行线路上发送的ASCII码有1或2个停止位，通常赋值为0。 因此，如果解码器不同步，它最终会在它认为应该是停止位的位置找到一个1，并尝试重新同步。

虽然理论上帧错误可能会持续很长时间，但在实践中使用停止位效果很好。

2.4.1 莫尔斯电码

一个变长编码的例子是为电报开发的莫尔斯电码。 字母、数字和标点的编码是由点和短间隔的破折号序列组成的。参见第2.9节。

解码器通过记录下一个点或破折号之前的时间长度来确定单个字符的编码结束。 字符内部的间隔是一个点的长度，字符之间的间隔更长，是一个破折号的长度。 单词之间的间隔更长。

2.5 详细信息：ASCII

ASCII代表“美国信息交换标准代码”，由美国国家标准学会（ANSI）于1963年引入。 它是最常用的字符编码。

ASCII是一个七位代码，表示表2.2中的33个控制字符和95个打印字符（包括空格）。控制字符用于表示特殊条件，如表2.3所述。

控制字符			
十六进制	十进制	字符	控制键
00	0	NUL	^@
01	1	SOH	^A
02	2	STX	^B
03	3	ETX	^C
04	4	EOT	^D
05	5	ENQ	^E
06	6	ACK	^F
07	7	BEL	^G
08	8	BS	^H
09	9	HT	^I
0A	10	LF	^J
0B	11	VT	^K
0C	12	FF	^L
0D	13	回车	^M
0E	14	SO	^N
0F	15	SI	^O
10	16	DLE	^P
11	17	DC1	^Q
12	18	DC2	^R
13	19	DC3	^S
14	20	DC4	^T
15	21	NAK	^U
16	22	SYN	^V
17	23	ETB	^W
18	24	CAN	^X
19	25	EM	^Y
1A	26	SUB	^Z
1B	27	ESC	^[
1C	28	FS	^\
1D	29	GS	^]
1E	30	RS	^^
1F	31	US	^_

数字		
十六进制	十进制	字符
20	32	SP
21	33	!
22	34	"
23	35	#
24	36	$
25	37	%
26	38	&
27	39	’
28	40	(
29	41	)
2A	42	*
2B	43	+
2C	44	,
2D	45	-
2E	46	.
2F	47	/
30	48	0
31	49	1
32	50	2
33	51	3
34	52	4
35	53	5
36	54	6
37	55	7
38	56	8
39	57	9
3A	58	:
3B	59	;
3C	60	¡
3D	61	=
3E	62	>
3F	63	?

大写字母		
十六进制	十进制	字符
40	64	@
41	65	A
42	66	B
43	67	C
44	68	D
45	69	E
46	70	F
47	71	G
48	72	H
49	73	I
4A	74	J
4B	75	K
4C	76	L
4D	77	M
4E	78	N
4F	79	O
50	80	P
51	81	Q
52	82	R
53	83	S
54	84	T
55	85	U
56	86	V
57	87	W
58	88	X
59	89	Y
5A	90	Z
5B	91	[
5C	92	\
5D	93	]
5E	94	^
5F	95	_

小写字母		
十六进制	十进制	字符
60	96	‘
61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g
68	104	h
69	105	i
6A	106	j
6B	107	k
6C	108	l
6D	109	m
6E	110	n
6F	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u
76	118	v
77	119	w
78	120	x
79	121	y
7A	122	z
7B	123	{
7C	124	—
7D	125	}
7E	126	~
7F	127	DEL

表2.2：ASCII字符集

进入8位

在8位环境中，ASCII字符前面有一个0，因此可以将其视为更大代码的“底部一半”。 由HEX 80和HEX F F之间的代码表示的128个字符（有时错误地称为“高ASCII”或“扩展ASCII”）在不同的上下文中有不同的定义。在许多操作系统中，它们包括带重音的西欧字母和其他附加字符。

HEX	DEC	CHR	Ctrl	含义
00	0	NUL	^@	NULl空白纸带上的前导符号；通常被忽略
01	1	SOH	^A	开始标题
02	2	STX	^B	开始文本
03	3	ETX	^C	结束文本；与STX匹配
04	4	EOT	^D	传输结束
05	5	ENQ	^E	询问
06	6	ACK	^F	ACKnowledge; 对ENQ的肯定回应
07	7	BEL	^G	BELl; 可听到的信号，早期机器上的铃声
08	8	BS	^H	BackSpace; 非破坏性，在左边界处被忽略
09	9	HT	^I	Horizontal Tab
0A	10	LF	^J	Line Feed; 纸张向上或打印头向下；Unix系统上的换行
0B	11	VT	^K	Vertical Tab
0C	12	FF	^L	Form Feed; 开始新页面
0D	13	回车	^M	Carriage Return; 打印头回到左边界；Macs系统上的换行
0E	14	SO	^N	Shift Out; 开始使用备用字符集
0F	15	SI	^O	Shift In; 恢复使用默认字符集
10	16	DLE	^P	Data Link Escape; 改变下一个字符的含义
11	17	DC1	^Q	Device Control 1; 如果使用流控制，XON，可以发送
12	18	DC2	^R	Device Control 2
13	19	DC3	^S	Device Control 3; 如果使用流控制，XOFF，停止发送
14	20	DC4	^T	Device Control 4
15	21	NAK	^U	Negative AcKnowledge; 对ENQ的回应
16	22	SYN	^V	SYNchronous idle
17	23	ETB	^W	End of Transmission Block
18	24	CAN	^X	CANcel; 忽略前一个块
19	25	EM	^Y	End of Medium
1A	26	SUB	^Z	SUBstitute
1B	27	ESC	^[	ESCape; 改变下一个字符的含义
1C	28	FS	^\	File Separator; 最粗的刻度
1D	29	GS	^]	Group Separator; 粗刻度
1E	30	RS	^^	Record Separator; 细刻度
1F	31	US	^_	Unit Separator; 最细的刻度
20	32	SP		空格；通常不被视为控制字符
7F	127	DEL		删除；最初被忽略；有时会导致退格

表2.3：ASCII控制字符

标点符号。 在IBM PC上，它们包括绘图字符。 Mac使用（并且仍在使用）不同的编码。

幸运的是，人们现在认识到计算机平台的互操作性的重要性，因此更普遍的标准正在受到青睐。 Web页面中最常用的编码是ISO-8859-1（ISO-Latin），它使用HEX A0到HEX FF之间的96个代码表示西欧语言的重音字母、标点符号和一些其他符号。 ISO-8859-1中HEX 80到HEX 9F之间的32个字符被保留为控制字符。

自然厌恶真空。 大多数人不希望有32个额外的控制字符（事实上，在7位ASCII中的33个控制字符中，只有大约十个在文本中经常使用）。 因此，对于使用HEX 80到HEX 9F的想法层出不穷。 最广泛使用的约定是微软的Windows Code Page 1252（Latin I），它与ISO-8859-1（ISO-Latin）相同，只是将32个控制码中的27个分配给了可打印字符，其中一个是HEX 80，即欧元货币符号。 并非所有平台和操作系统都识别CP-1252，因此文档，特别是Web页面，需要特别注意。

超越8位

为了表示亚洲语言，需要更多的字符。 目前正在积极开发适当的标准，普遍认为需要表示的字符总数少于65,536个。 这是幸运的，因为16位或2字节可以表示许多不同的字符。 为了保持在这个数字范围内，一些汉语方言的书面版本必须共享外观相似的符号。

今天最有可能成为2字节标准字符编码的候选者被称为Unicode。

参考资料

有许多网页提供ASCII表，包括所有世界语言的扩展。
其中一些更有用的是：

- Jim Price的PC和Windows 8位表格，以及其他几个链接
  http://www.jimprice.com/jim-asc.shtml
- 字符编码简史，讨论了对亚洲语言的扩展
  http://tronweb.super-nova.co.jp/characcodehist.html
- Unicode主页
  http://www.unicode.org/
- Windows CP-1252标准，明确的
  http://www.microsoft.com/globaldev/reference/sbcs/1252.htm
- 与CP-1252相比：
  - Unicode
    http://ftp.unicode.org/Public/MAPPINGS/VENDORS/MICSFT/WINDOWS/CP1252.TXT
  - Unicode/HTML
    http://www.alanwood.net/demos/ansi.html
  - ISO-8859-1/Mac OS
    http://www.jwz.org/doc/charsets.html

2.6详细信息：整数编码

有许多方法可以将整数表示为位模式。 所有这些都无法在固定位数中表示任意大的整数。 产生超出范围结果的计算被称为溢出。

最常用的表示方法是无符号整数的二进制代码（例如，内存地址），有符号整数的2的补码（例如，普通算术），以及测量变化数量的二进制格雷码的仪器。

下表给出了五个4位整数代码的示例。 最高有效位（**MSB**）在左侧，最低有效位（**LSB**）在右侧。

	无符号整数		有符号整数		
	二进制码	二进制格雷码	二进制补码	符号/幅度码	二进制反码
范围 →	[0, 15]	[0, 15]	[-8, 7]	[-7,7]	[-7,7]
-8			1 0 0 0		
-7			1 0 0 1	1 1 1 1	1 0 0 0
-6			1 0 1 0	1 1 1 0	1 0 0 1
-5			1 0 1 1	1 1 0 1	1 0 1 0
-4			1 1 0 0	1 1 0 0	1 0 1 1
-3			1 1 0 1	1 0 1 1	1 1 0 0
-2			1 1 1 0	1 0 1 0	1 1 0 1
-1			1 1 1 1	1 0 0 1	1 1 1 0
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1	0 0 1 0	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 0	0 0 1 1	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 1 0	0 1 0 0	0 1 0 0	0 1 0 0
5	0 1 0 1	0 1 1 1	0 1 0 1	0 1 0 1	0 1 0 1
6	0 1 1 0	0 1 0 1	0 1 1 0	0 1 1 0	0 1 1 0
7	0 1 1 1	0 1 0 0	0 1 1 1	0 1 1 1	0 1 1 1
8	1 0 0 0	1 1 0 0			
9	1 0 0 1	1 1 0 1			
10	1 0 1 0	1 1 1 1			
11	1 0 1 1	1 1 1 0			
12	1 1 0 0	1 0 1 0			
13	1 1 0 1	1 0 1 1			
14	1 1 1 0	1 0 0 1			
15	1 1 1 1	1 0 0 0			

表2.4：四位整数编码

二进制码

该编码用于非负整数。 对于长度为 n的编码， 2^n个模式表示从 0到 $2^n - 1$的整数。 LSB（最低有效位）为偶数时为 0，1 为奇数时为1。

二进制格雷码

这个编码适用于非负整数。 对于长度为 n的编码，2^n个模式表示从0到2^n-1的整数。 相邻整数的两个位模式只相差一个位。 这个特性使得编码在测量过程中整数可能发生变化的传感器中非常有用。 以下匿名致敬出现在马丁·加德纳的专栏“数学游戏”中，发表于《科学美国人》1972年8月，但实际上早就有人知道。

二进制格雷码很有趣，
因为它可以做出奇怪的事情...
十五，如你所知，
是一零零零，
而十是一一一一。

二进制补码

这个编码适用于正数和负数的整数。 对于长度为 n的编码，2^n个模式表示整数 -2^{n-1}到$2^{n-1}-1$。 最低有效位（LSB）对于偶数为0，对于奇数为1。 在它们重叠的部分，这个编码与二进制编码相同。 这个编码被广泛使用。

符号/大小

这个编码适用于正数和负数的整数。 对于长度为 n的编码，2^n个模式表示整数 $-(2^{n-1}-1)$到$2^{n-1}-1$。 最高有效位（MSB）对于正数为0，对于负数为1；其他位表示数值大小。 在它们重叠的部分，这个编码与二进制编码相同。
尽管在概念上很简单，但在实践中这段代码很笨拙。 它对于$+0$和-0的单独表示通常没有用处。

1的补码

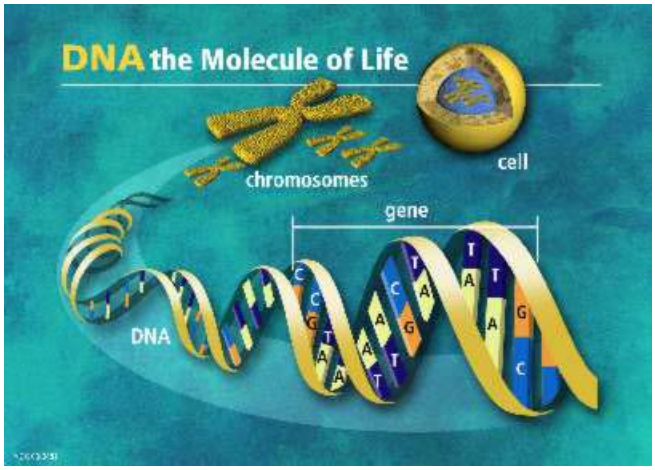
这段代码适用于正数和负数。 对于长度为 n的代码，2^n个模式表示从整数 $-(2^{n-1}-1)$到$2^{n-1}-1$。 最高有效位（MSB）为0表示正整数；负整数通过对应正整数的每一位取反得到。 在重叠的部分，这段代码与二进制代码相同。 这段代码笨拙且很少使用。 它对于$+0$和-0的单独表示通常没有用处。

2.7 详细信息：遗传密码*

你身体的基本构建单元是细胞。 两个或多个细胞组成组织，如骨骼或肌肉；组织组织起来形成器官，如心脏或大脑；器官形成器官系统，如循环系统或神经系统；器官系统共同构成你，也就是有机体。 细胞可以被分类为真核细胞或原核细胞-分别具有或不具有细胞核。 构成你的身体以及所有动物、植物和真菌的细胞都是真核细胞。 原核生物是细菌和蓝藻细菌。

细胞核与细胞体的其余部分形成了一个独立的隔间；这个隔间作为真核细胞的所有遗传信息的中央储存中心。 构成生命之书的所有遗传信息都存储在细胞核内的单个染色体上。 在健康人体内，有23对染色体（共46条）。 每个染色体都包含一条线状的脱氧核糖核酸（DNA）分子。 基因是沿着这些DNA链的功能区域，是从一代传递到下一代的基本物理单位。

在原核生物中，染色体在细胞体内自由漂浮，因为没有细胞核。



由美国能源部基因组计划的基因组管理信息系统提供，htt
p://genomics.energy.gov。

图2.2：细胞内DNA的位置

DNA分子由两个相互连接的核苷酸链组成，形成一条DNA链。
每个核苷酸由一个糖、一个磷酸和四种碱基组成。 碱基包括腺嘌呤、鸟嘌呤、胞嘧啶和胸腺嘧啶。 为了方便起见，每个核苷酸都以其碱基来引用；当提到单个核苷酸时，我们会简单地说鸟嘌呤（或G）。 因此，我们可以写CCACCA来表示一串相互连接的胞嘧啶-胞嘧啶-腺嘌呤-胞嘧啶-胞嘧啶-腺嘌呤核苷酸。

通过将核苷酸碱基配对连接在一起，个体核苷酸链形成了一个双螺旋结构。 配对规则是胞嘧啶总是与鸟嘌呤配对，胸腺嘧啶总是与腺嘌呤配对。 这些DNA链在体细胞分裂（即除了命定成为性细胞的细胞外的细胞分裂）期间进行复制，并将完整的遗传信息传递给产生的细胞。

基因是染色体的一部分，并在DNA链上进行编码。 线状DNA的个体功能区段称为基因。 基因中编码的信息指导细胞和有机体的维持和发展。 这些信息沿着一条路径传播：DNA（基因） ⇒mRNA（信使核糖核酸） ⇒核糖体/tRNA ⇒蛋白质。实质上，蛋白质是从基因中生成的最终输出，基因作为个体蛋白质的蓝图。

*本节基于Tim Wagner的笔记

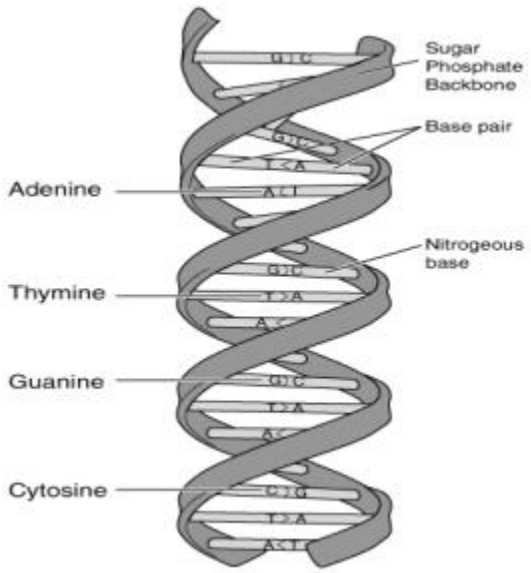


图2.3：显示DNA螺旋结构的示意图

蛋白质本身可以是你身体的结构组成部分（如肌肉纤维），也可以是功能组分（酶，帮助调节你身体中的成千上万个生化过程）。 蛋白质是由多肽链构建的，多肽链只是氨基酸的字符串（单个多肽链构成一个蛋白质，但通常功能蛋白质由多个多肽链组成）。

遗传信息通过信使RNA从细胞核的DNA传递到核外的核糖体（核糖体是帮助最终构建最终蛋白质的细胞组分）。
转录是从DNA生成信使RNA的过程。信使RNA是单个核苷酸链的一部分的复制。 它是单链，与DNA完全相同，除了核苷酸糖和胸腺嘧啶被尿嘧啶替换外。 信使RNA的形成与DNA的碱基配对规则相同，只是T被U替换（C对G，U对A）。

这个信使RNA在细胞体内通过核糖体和tRNA的帮助下被翻译成一串氨基酸（蛋白质）。 核糖体将信使RNA固定在位置上，而转运RNA则将适当的氨基酸放入正在形成的蛋白质中，如图2.4所示。信使RNA首先与核糖体结合，然后被翻译成蛋白质。 启动子tRNA结合到核糖体上，对应于mRNA链上的起始密
码子 - 在人类中，这对应于AUG密码子。 这个tRNA分子携带了密码子要求的适当的氨基酸，并在其核苷酸链上的另一个位置与mRNA链匹配，称为反密码子。 这些键通过mRNA和DNA的相同碱基配对规则形成（为简单起见，将忽略一些配对异常）。 然后，第二个tRNA分子将停靠在相邻位置的核糖体上，该位置由下一个密码子指示。 它还将携带密码子要求的相应氨基酸。一旦两个tRNA分子停靠在核糖体上，它们携带的氨基酸将结合在一起。 最初的tRNA分子将分离，将其氨基酸留在一个不断增长的氨基酸链上。 然后，核糖体将在mRNA链上向后移动一个位置，为另一个tRNA分子停靠带来另一个氨基酸。 这个过程将继续进行，直到mRNA上读取到一个终止密码子; 在人类中，终止因子是UAG，UAA和UGA。当读取到终止密码子时，氨基酸链（蛋白质）将从核糖体结构中释放出来。

什么是氨基酸？ 它们是有机化合物，其中心碳原子上通过共价键连接

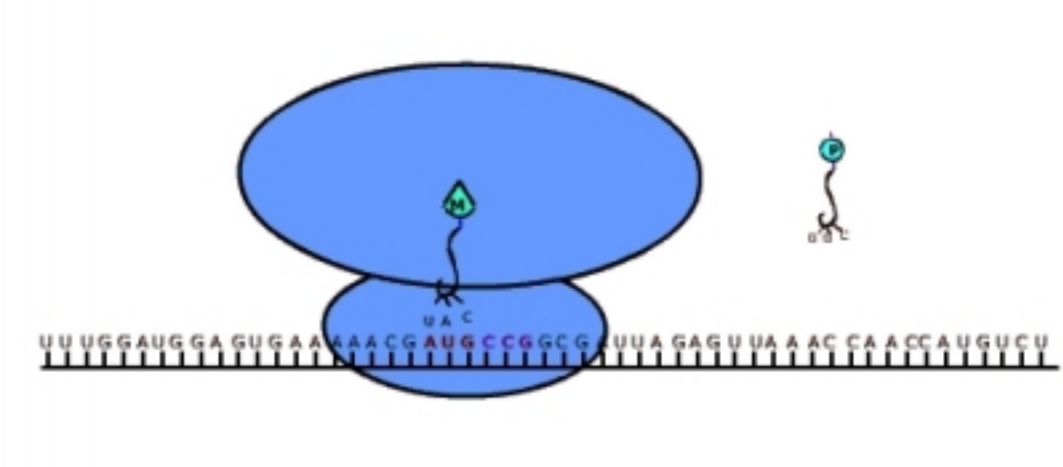


图2.4：RNA到蛋白质的转录（点击图像以在线动画形式查看
http://www.mtl.mit.edu/Courses/6.050/2008/notes/rna-to-proteins.html)

- 一个氢原子 H
- 一个氨基团 NH_2
- 一个羧基 COOH
- 一个侧链，每个氨基酸的侧链都不同

侧链的复杂程度从一个氢原子（对于氨基酸甘氨酸）到包含多达18个原子的结构（精氨酸）不等。 因此，每个氨基酸含有10到27个原子。 在上述蛋白质的生产中，一共使用了二十种不同的氨基酸（有时被称为“常见氨基酸”）。 其中十种被认为是“必需的”，因为它们在人体内无法合成，因此必须通过饮食获得（精氨酸对婴儿和生长中的儿童是必需的）。 九种氨基酸是亲水性（水溶性），八种是疏水性（另外三种被称为“特殊”）。 在亲水性氨基酸中，两种侧链带有净负电荷，因此是酸性的，三种带有净正电荷，因此是碱性的；而四种则是无电荷的。 通常，侧链完全由氢、氮、碳和氧原子组成，尽管两种氨基酸（半胱氨酸和蛋氨酸）还含有硫。

有二十个不同的常见氨基酸需要编码，而只有四种不同的碱基。这是如何实现的？ 作为单个实体，核苷酸（A、C、T或G）只能编码四种氨基酸，显然不够。 作为一对，它们可以编码16个（4^2）氨基酸，仍然不够。 通过三联体，我们可以编码64个（4^3）可能的氨基酸 - 这实际上是身体中的实际操作方式，三个核苷酸的字符串称为密码子。为什么要这样做？ 进化如何发展出如此低效的编码，具有如此多的冗余性？ 有多个密码子对应于一个氨基酸，有两个主要的生物学原因：存在具有不同反密码子的多种tRNA物种，将特定的氨基酸带到核糖体，以及在翻译过程中可能发生错误/松散配对（这称为摇摆）。

密码子，由三个核苷酸组成的字符串，编码氨基酸。 下表中是遗传密码表，从信使RNA密码子到氨基酸的对应关系，以及氨基酸的各种性质[1]在下表中，*代表(U、C、A或G)，因此CU*可以是CUU、CUC、CUA或CUG。

[1]显示了每个氨基酸的一字母缩写、分子量和一些性质，这些信息来自 H. Lodish、 D. Baltimore、A. Berk、S. L. Zipursky、P. Matsudaira和J. Darnell的《分子细胞生物学》第三版，W. H. Freeman and Company，纽约，纽约；1995年。

mRNA密码子的第二个核苷酸碱基

mRNA密码子的第一个核苷酸碱基	尿嘧啶	胞嘧啶	腺嘌呤	鸟嘌呤
尿嘧啶	UUU = 苯丙氨酸 UUC = 苯丙氨酸 UUA = 亮氨酸 UUG = 亮氨酸	UC* = Ser	UAU = 酪氨酸 UAC = 酪氨酸 UAA = 停止 UAG = 停止	UGU = 胱氨酸 UGC = 胱氨酸 UGA = 停止 UGG = 色氨酸
胞嘧啶	CU* = 亮氨酸	CC* = 脯氨酸	CAU = 组氨酸 CAC = 组氨酸 CAA = 谷氨酰胺 CAG = 谷氨酰胺	CG* = 精氨酸
腺嘌呤	AUU = 异亮氨酸 AUC = 异亮氨酸 AUA = 异亮氨酸 AUG = 蛋氨酸（起始）	AC* = 苏氨酸	AAU = 天冬酰胺 AAC = 天冬酰胺 AAA = 赖氨酸 AAG = 赖氨酸	AGU = 丝氨酸 AGC = 丝氨酸 AGA = 精氨酸 AGG = 精氨酸
鸟嘌呤	GU* = 缬氨酸	GC* = 丙氨酸	GAU = 天冬酰胺酸 GAC = 天冬酰胺酸 GAA = 谷氨酸 GAG = 谷氨酸	GG* = 甘氨酸

表2.5：氨基酸的简化图

符号		氨基酸	分子量	性质		密码子
丙氨酸	腺嘌呤	丙氨酸	89.09	非必需的	疏水性	GC*
精氨酸	R	精氨酸	174.20	必需的	亲水性，碱性	CG* AGA AGG
天冬氨酸	N	天冬氨酸	132.12	非必需的	亲水性，不带电荷	AAU AAC
天冬酰胺	D	天冬酰胺	133.10	非必需的	亲水性，酸性	GAU GAC
半胱氨酸	胞嘧啶	半胱氨酸	121.15	非必需的	特殊的	UGU UGC
谷氨酰胺	Q	谷氨酰胺	146.15	非必需的	亲水性，不带电荷	CAA CAG
谷氨酸	E	谷氨酸	147.13	非必需的	亲水性，酸性	GAA GAG
甘氨酸	鸟嘌呤	甘氨酸	75.07	非必需的	特殊的	GG*
组氨酸	H	组氨酸	155.16	必需的	亲水性，碱性	CAU CAC
异亮氨酸	I	异亮氨酸	131.17	必需的	疏水性	AUU AUC AUA
亮氨酸	L	亮氨酸	131.17	必需的	疏水性	UUA UUG CU*
赖氨酸	K	赖氨酸	146.19	必需的	亲水性，碱性	AAA AAG
甲硫氨酸	M	甲硫氨酸	149.21	必需的	疏水性	AUG
苯丙氨酸	F	苯丙氨酸	165.19	必需的	疏水性	UUU UUC
专业	P	脯氨酸	115.13	非必需的	特殊的	CC*
丝氨酸	S	丝氨酸	105.09	非必需的	亲水性，不带电荷	UC* AGU AGC
苏氨酸	T	苏氨酸	119.12	必需的	亲水性，不带电荷	AC*
色氨酸	W	色氨酸	204.23	必需的	疏水性	UGG
酪氨酸	Y	酪氨酸	181.19	非必需的	疏水性	UAU UAC
缬氨酸	V	缬氨酸	117.15	必需的	疏水性	GU*
开始		甲硫氨酸				AUG
停止						UAA UAG UGA

表2.6：氨基酸及其一些性质

2.8 详细信息：IP地址

表2.7是IPv4的摘录，http://www.iana.org/assignments/ipv4-address-space（版本4，正在逐步被版本6取代）。IP地址由互联网编号分配机构（IANA）分配，http://www.iana.org/。

IANA负责互联网上的所有“唯一参数”，包括IP（Internet Protocol）地址。
每个域名都与一个唯一的IP地址相关联，这是一个由四个块组成的数字名称，每个块最多包含三个数字，例如204.146.46.8，系统使用它来通过网络传递信息。

互联网协议地址空间

将互联网协议版本4（IPv4）的地址空间分配给各个注册机构的情况在这里列出。
最初，所有的IPv4地址空间都由IANA直接管理。后来，地址空间的部分被分配给其他注册机构，以管理特定的目的或世界各地的区域。RFC 1466记录了这些分配的大部分情况。

地址块	注册表 - 目的	日期
000/8	IANA - 保留	九月 81
001/8	IANA - 保留	九月 81
002/8	IANA - 保留	九月 81
003/8	通用电气公司	五月 94
004/8	Bolt Beranek和Newman公司	十二月 92
005/8	IANA - 保留	七月 95
006/8	陆军信息系统中心	二月 94
007/8	IANA - 保留	四月 95
008/8	Bolt Beranek和Newman公司	92年12月
009/8	IBM	92年八月
010/8	IANA - 私有使用	六月 95
011/8	国防部情报信息系统	93年五月
012/8	AT＆T贝尔实验室	六月 95
013/8	施乐公司	91年九月
014/8	IANA - 公共数据网络	91年六月
015/8	惠普公司	94年七月
016/8	数字设备公司	94年11月
017/8	苹果电脑公司	92年7月
018/8	麻省理工学院	94年1月
019/8	福特汽车公司	五月 95
020/8	计算机科学公司	94年10月
021/8	DDN-RVN	91年7月
022/8	国防信息系统局	93年5月
023/8	IANA - 保留	七月 95
024/8	IANA - 电缆区块	七月 95
025/8	皇家信号和雷达研究所 1月	95
⋮		

表2.7：IP地址分配 - 部分列表

2.9 详细信息：莫尔斯电码

塞缪尔F.B.莫尔斯（1791-1872）是来自马萨诸塞州查尔斯顿的风景和肖像画家。他经常从他在纽约市的工作室到全国各地的客户那里工作。 1825年，他的妻子卢克莉莎在华盛顿特区突然死于心力衰竭。 莫尔斯尽可能快地得知了这一事件，通过一封从纽约寄往华盛顿的信件，但他已经来不及赶回去参加她的葬礼了。

作为一位画家，莫尔斯只取得了适度的成功。 虽然他的画作现在可以在主要博物馆中找到——波士顿美术馆有七幅——但他对当代艺术并没有产生重要影响。他以发明家的身份最为人所知。 （他以一种有趣的方式将对技术的兴趣和对艺术的热情结合在一起：1839年，他学会了制作达盖尔银版照片的法国技术，并在接下来的几年里通过教授这项技术来维持生计。）

1832年，他从欧洲返回时，碰巧遇到了一位乘客，后者曾参观过伟大的欧洲物理实验室。 他了解到安培、富兰克林等人进行的实验中，电流可以在任意长度的导线上瞬间传输。 莫尔斯意识到这意味着情报可以通过电流瞬间传输。 他从妻子去世的情况中意识到了快速通信的需求。 在他的船甚至到达纽约之前，他发明了今天被称为莫尔斯电码的第一个版本。他后来的发明包括手键和一些接收设备。 1844年，他从华盛顿向巴尔的摩发送了他著名的信息“WHAT HATH GOD WROUGHT”。这一事件引起了公众的热情，产生了类似于150年后的互联网狂热的国家性兴奋。

莫尔斯电码由一系列短脉冲或音调（点和划线）组成，它们之间由短暂的静默间隔分隔。 一个人通过在手动键上建立和断开电连接来生成莫尔斯电码，而线路另一端的人则听取点和划线的序列，并将其转换为字母、空格和标点符号。 现代莫尔斯电码的形式如表2.8所示。@符号是在2004年添加的，以适应电子邮件地址。 其中两个控制码如图所示。

非英文字母和一些不常用的标点符号被省略。

脉搏吟	·–	K	–·–	屏蔽哭	···–	0	–––––	问号	··––··
B	–···	L	·–··	V	···–	1	·––––	撇号	·–––––·
C	–·–·	M	––	W	·––	2	··–––	括号	–·––·–
D	–··	N	–·	X	–··–	3	···––	引号	·–··–·
E	·	O	–––	Y	–·––	4	····–	分数线	–··–·
F	··–·	P	·––·	Z	––··	5	·····	等于号	–···–
G	––·	Q	––·–	周期	·–·–·–	6	–····	斜杠	–··–·
H	····	R	·–·	逗号	––··––	7	––···	At符号	·––·–·
I	··	S	···	连字符	–····–	8	––––··	删除前一个单词	·····
J	·–––	T	–	冒号	–––···	9	––––·	传输结束	·–·–·

表2.8：莫尔斯码

如果一个点的持续时间被视为一个时间单位，那么一个划的持续时间为三个单位。 一个字符内部的点和划之间的间隔为一个单位，字符之间的间隔为三个单位，单词之间的间隔为七个单位。 空格不被视为一个字符，就像在ASCII中一样。

与ASCII不同，莫尔斯码是一种可变长度的编码。 莫尔斯意识到英语字母中有些字母的使用频率比其他字母高，并给予它们较短的编码。 因此，平均而言，消息传输速度更快，而不是所有字母的长度都相等。表2.9显示了英语书面文字中每个字母的频率（每1000个字母平均出现的次数）。

莫尔斯电码在电报上设计得很好，在无线电通信中也得到了应用，直到AM收音机能够传送语音之前。直到1999年，莫尔斯电码是海洋船只的必需通信方式，尽管很少使用（理论上是因为一些老旧船只可能没有转换到更现代的通信设备）。 发送和接收莫尔斯电码仍然是美国的要求。

132	E	61	S	24	U
104	T	53	H	20	G, P, Y
82	A	38	D	19	W
80	O	34	L	14	B
71	N	29	F	9	V
68	R	27	C	4	K
63	I	25	M	1	X, J, Q, Z

表2.9：英语书面中字母的相对频率

希望获得某些类型的业余无线电执照的公民。

由于莫尔斯电码是设计用来听的，而不是看的，表2.8只有较小的用处。 你不能通过看纸上的点和划线来学习莫尔斯电码；你必须听到它们。 如果你想在你选择的文本上听到它，可以尝试互联网上的合成器，比如

- http://morsecode.scphillips.com/jtranslator.html

通过比较表2.8和2.9，可以发现莫尔斯在为常见字母分配短序列方面做得相当不错。 据报道，他并不是通过数书报中的字母来做到这一点的，而是通过参观印刷店。 当时的印刷机使用可移动的活字，人们将单独的字母组装成行。 每个字母都有多个字体和大小的副本，以铅块的形式存在。 莫尔斯只是简单地计算了每个字母的可用铅块数量，假设印刷工人知道他们的工作，并且用正确的数量储存了每个字母。

木质字盒分为两行，大写字母在上面一行，小写字母在下面一行。印刷工人称字盒上面一行的字母为“大写字母”。

第三章

压缩

在第一章中，我们研究了信息的基本单位——比特，以及它的各种抽象表示形式：布尔比特、电路比特、控制比特、物理比特、量子比特和经典比特。我们对改进的追求使我们希望能够用更小、更快、更强、更智能、更安全、更便宜的方式表示单个比特。

在第2章中，我们考虑了通过位数组（此时为布尔位）来表示复杂对象的一些问题。 用于表示要表示的对象（符号）和用于此目的的位数组之间的映射被称为编码。 我们自然希望得到更强大且更小的编码，即导致对象表示更小且更不容易出错的编码。

在本章中，我们将考虑用于生成特别高效表示的压缩技术。 在第4章中，我们将研究避免错误的技术。

在第2章中，我们考虑了图3.1所示的系统，其中符号被编码为位字符串，然后被传输（在空间和/或时间上）到解码器，解码器然后重新创建原始符号。

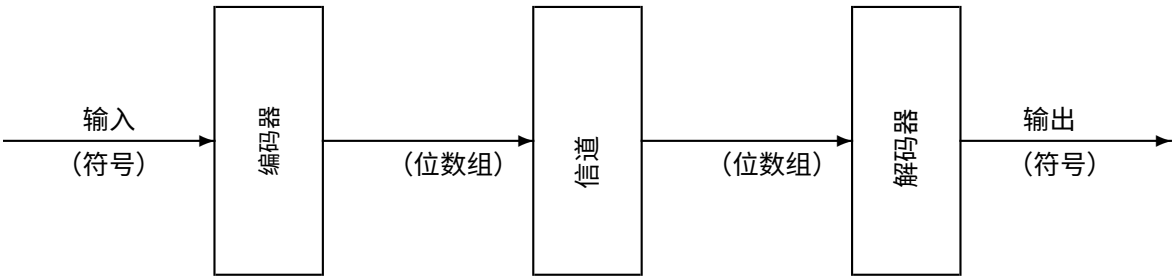


图3.1：广义通信系统

通常，相同的编码被用于连续的符号序列，一个接一个。 数据压缩的作用是将表示一系列符号的位字符串转换为更短的字符串，以实现更经济的传输、存储或处理。 结果是图3.2中的系统，包括压缩器和扩展器。 理想情况下，扩展器应该完全逆转压缩器的操作，以便编码器和解码器保持不变。

一开始可能会觉得这种方法很令人惊讶。 为什么有理由相信相同的原因

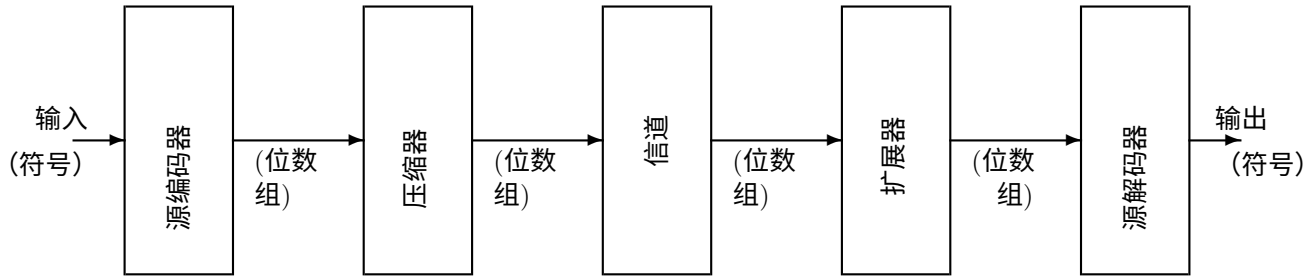


图3.2：更复杂的通信系统

信息可以包含在更少的位数中吗？ 我们将研究两种类型的压缩，使用不同的方法：

- 无损或可逆压缩（仅当原始代码效率低下时才能进行，例如存在未使用的位模式，或者未充分利用某些符号比其他符号更频繁使用的事实）

- 有损或不可逆压缩，其中原始符号或其编码表示无法从较小的版本中完全重构，而是扩展器生成一个足够“好”的近似值

下面描述了六种令人惊讶地有效的数据文件压缩技术。 前五种是可逆的，最后一种是不可逆的。 每种技术都有一些特别适用的情况（最佳情况）和一些不适合的情况（最差情况）

3.1 可变长度编码

在第2章中，莫尔斯码被讨论为一个源代码的例子，其中英文字母中出现频率较高的字母用较短的码字表示，而出现频率较低的字母用较长的码字表示。 平均而言，使用这些码字发送的消息比如果所有码字具有相同长度要短。 可变长度编码可以在源编码器或压缩器中进行。 可变长度编码的一般过程将在第5章中给出，因此对这种技术的讨论将推迟到该章节。

3.2 运行长度编码

假设一条消息由少量符号或字符的长序列组成。 那么，该消息可以被编码为符号和其出现次数的列表。 例如，消息“a B B B B B a a a B B a a a a”可以被编码为“a 1 B 5 a 3 B 2 a 4”。 这种技术在相对较少的情况下非常有效。 一个例子是德国国旗，可以被编码为许多黑色像素、许多红色像素和许多黄色像素，相比于指定每个像素来说节省了很多。 另一个例子来自传真技术，其中文档被扫描，长串的白色（或黑色）像素仅以像素数量传输（由于只有白色和黑色像素，甚至不需要指定颜色，因为可以假设为另一种颜色）。

对于没有重复序列的消息，运行长度编码效果不好。
例如，它可能对绘画甚至是黑白扫描图像效果好，但对于其他情况则不行。

图3.3：德国国旗（顶部黑色条纹，中部红色，底部黄色）

对于照片来说效果不好，因为从一个像素到下一个像素的细微变化会导致需要定义很多符号。

3.3 静态字典

如果编码中有未使用的编码词，可以将其分配为常见序列的缩写。 然后，这些序列可以使用的比单个符号所需的位数更少进行编码。 例如，如果使用ASCII编码英文文本，并且DEL字符被认为是不必要的，则将其编码为127可能对常见单词“the”有意义。 实际编码提供了许多这种技术的示例。 编码词及其含义的列表称为编码表或字典。 这里考虑的压缩技术使用的字典在每个消息之间是静态的，即不会改变。

一个例子将说明这个技术。 在电报出现之前，还有其他传输消息的方案。 威尔逊详细描述了一种机械电报，英国海军于[1]796年在伦敦总部和各个港口之间建立了这种电报系统，包括普利茅斯、雅茅斯和迪尔。它由一系列的小屋组成，每个小屋都可以看到下一个小屋，有六个大百叶窗，可以旋转成水平（打开）或垂直（关闭）的状态。见图3.4。在运行中，所有百叶窗都处于打开状态，直到发送消息。 然后所有百叶窗都关闭，表示开始发送消息（小屋的操作员应该每五分钟查看新消息）。 然后以一系列的百叶窗模式发送消息，以全开的模式结束。

有六个百叶窗，因此有64种（2^6）模式。 其中两种是控制码（全开表示“开始”和“停止”，全闭表示“空闲”）。 其他62种模式可用于表示26个字母（如果包括J和U，则为26个字母），10个数字，一个单词结束标记和一个页面结束标记。 这样就有20个未使用的模式，用于常见词语或短语。 使用的特定缩写会随时间而变化，但包括常见词语如“the”，地点如“Portsmouth”，以及其他重要词语如“French”，“Admiral”，“east”和“frigate”。 它还包括短语如“Commander of the West India Fleet”和“To sail, the first fair wind, to the southward”。

在密码本中，最引人注目的条目可能是“军事法庭开庭”和“军事法庭判决执行”。 军事法庭真的足够常见，而且关于它们的消息足够频繁，以至于有必要将64个代码模式中的两个用于它们吗？

正如这个例子所示，通过选择一组合适的缩写词，长消息可以大大缩短。 然而，这也存在固有的风险：错误的影响可能会更大。 如果传输完整的文本，一个错误会导致拼写错误，或者最坏的情况下是错误的单词，而人类通常可以检测和纠正。 使用缩写词，一个错误的位可能会导致一个可能但不是预期的含义："east"可能会变成"south"，或者一个字母可能会变成"军事法庭判决执行"，带来重大后果。

这个电报系统运行良好。 在一次演示中，一条消息从伦敦到普利茅斯，往返500英里，只用了三分钟。这是声音速度的13倍。

如果使用缩写来压缩信息，必须有一个编码手册显示所有使用的缩写，在第一条消息发送之前分发。 因为它只分发一次，所以每条消息的成本很低。

[1] Geoffrey Wilson，“The Old Telegraphs，”Phillimore and Co.，Ltd.，London and Chichester，U.K.；1976；pp. 11-32。

图3.4：1797年的英国百叶窗电报舱，显示六个百叶窗关闭，并有一个打开的窗口用望远镜观察另一个舱（摘自T. Standage，“The Victorian Internet，”Berkley Books，New York；1998；p. 15）。

每条消息的分发量很低。 但它必须经过精心构建，以适应所有预期的消息，并且不能根据个别消息的需求进行更改。

这种技术对于非常相似的消息集合效果很好，就像18世纪海军通信可能是这种情况一样。 它不适用于更多样化的消息。 这种机械电报从未被用于商业或公共用途，这将使它具有更多样化的消息集合，而不需要那么多共同的词语或短语。

3.4 半自适应字典

静态字典方法需要一个预先定义的字典，适用于所有消息。 如果可以为每个消息定义一个新的字典，压缩效果可能会更好，因为可以将消息中特定的符号序列作为字典条目。

然而，这样做会有几个缺点。 首先，新字典必须与编码的消息一起传输，导致开销增加。 其次，必须分析消息以发现最佳的字典条目，因此整个消息必须在任何部分被编码之前都可用进行分析（即，该技术具有较大的延迟）。 第三，计算字典的计算机需要足够的内存来存储整个消息。

这些缺点限制了半自适应字典压缩方案的使用。

3.5 动态字典

对于许多应用程序来说，最好的编码方案是在处理消息时即时计算字典，不需要随消息一起传输，并且可以在消息结束之前使用。 初步考虑，这似乎是不可能的。

然而，这样的方案是已知的并且被广泛使用的。它是LZW压缩技术，以Abraham Lempel、Jacob Ziv和Terry Welch命名。 Lempel和Ziv实际上有一系列的技术，有时

被称为LZ77和LZ78，但是1984年Welch的修改赋予了该技术所有期望的特性。

Welch希望减少发送到磁盘驱动器的位数，部分是为了增加磁盘的有效容量，部分是为了提高数据传输的速度。 他的方案在这里描述，包括编码器和解码器。 它已经被广泛应用于许多情境中，以实现数据的可逆压缩。 当应用于典型计算机上的文本文件时，编码文件通常只有原始文件大小的一半。 它被用于流行的压缩产品，如Stuffit或Disk Doubler。当用于具有大面积完全相同颜色的绘图的彩色图像时，它可以导致比每个像素都存储的文件格式更小的文件。 常用的GIF图像格式使用LZW压缩。 当用于具有渐变颜色的照片时，节省的空间要小得多。

因为这是一种可逆的压缩技术，原始数据可以通过解码器精确地重建，而不需要近似。

LZW技术似乎有许多优点。 然而，它有一个主要的缺点。 它不是免费提供的-它被专利保护。美国专利号为4,558,302，于2003年6月20日到期，其他国家的专利于2004年6月到期，但在此之前，它引发了一场争议，至今仍是许多人不愉快的回忆，因为处理方式不当。

3.5.1 LZW专利

韦尔奇当时在斯佩里研究中心工作，他发表了一篇描述该技术的论文。 人们很快认识到它的优点，并将其用于各种压缩方案，包括1987年由CompuServe（一家国家互联网服务提供商）开发的图形交换格式GIF，目的是减小计算机中图像文件的大小。

那些定义GIF的人没有意识到GIF基于的LZW算法是有专利的。 Welch的文章没有警告说有专利正在申请中。 万维网在20世纪90年代初开始崭露头角，第一批图形浏览器接受GIF图像。 因此，网站开发者通常使用GIF图像，认为这项技术是公共领域的，这是CompuServe的意图。

到1994年，Sperry的继任公司Unisys意识到这项专利的价值，并决定试图从中获利。他们与CompuServe接触，起初CompuServe并没有太在意，显然没有认为威胁是真实存在的。最后，CompuServe认真对待了Unisys的要求，两家公司于1994年12月24日共同宣布，任何编写创建或读取GIF图像的软件的开发者都必须从Unisys获得许可。网站开发者不确定他们使用GIF图像是否需要支付版权费，他们对为网站上的每个GIF图像付费的想法并不高兴。 不久之后，Unisys意识到这可能会引发公关灾难，他们放弃了他们的要求。 1995年1月27日，他们宣布他们不会试图收取现有图像的使用费，也不会收取在1994年底之前分发的工具生成的图像的使用费，但坚持要求从1995年开始许可图形工具。 通过许可的工具生成的图像将允许在网络上使用而无需额外付费。

1999年，Unisys决定从可能包含未经许可的图像的个人网站收集信息，每个网站收费5000美元，非营利组织没有例外，小型低流量网站也没有更低的许可费用。 目前不知道有多少个网站实际支付了这笔费用；据报道，在头8个月中只有一个网站支付了。许多网站开发者感到沮丧和愤怒。 尽管Unisys在1995年避免了一场公关灾难，但在1999年却遇到了一场。

自由奔放的网络社区和以赚钱为目的的商业社区之间存在着非常真实的文化差异。

为了取代GIF，创建了一种非侵权的公共领域图像压缩标准PNG，但浏览器制造商对采用另一种图像格式的速度较慢。 此外，每个人都知道这些专利很快就会过期。 争议现在已经消失，只存在于少数人的记忆中，他们感到

[2] Welch，T.A. “高性能数据压缩技术”， IEEE计算机，第 17卷，第 6期，第 8-19页； 1984年。

特别强烈。 它仍然被引用为支持或反对专利制度变革，甚至是软件专利概念的理由。

至于PNG，它提供了一些技术优势（特别是更好的透明特性），并且截至2004年，几乎所有浏览器都对其提供了良好的支持，唯一的例外是Windows版的Microsoft Internet Explorer。

3.5.2 LZW是如何工作的?

该技术的编码和解码都在第3.7节中用文本示例进行了说明。

3.6 不可逆技术

本节尚未提供。抱歉。

它将包括浮点数、JPEG图像压缩和MP3音频压缩作为示例。 JPEG压缩中使用的离散余弦变换（DCT）在第3.8节中进行了讨论。MP3压缩的演示可在http://www.mtl.mit.edu/Courses/6.050/2007/notes/mp3.html上找到。

3.7 详细信息：LZW 压缩

下面描述了 LZW 压缩技术，并应用于两个示例。 同时考虑编码器和解码器。 LZW 压缩算法是“可逆的”，意味着它不会丢失任何信息-解码器可以完全重构原始消息。

3.7.1 LZW 算法，示例 1

考虑文本消息的编码和解码

```
itty bitty bit bin
```

（这个奇特的短语被设计成具有重复字符串，以便字典快速建立）。

初始字典条目是 8 位字符代码，代码点为 0-255，ASCII 作为前 128 个字符，包括上面字符串中的表 3.1 中的字符。 字典条目 256 定义为“清除字典”或“开始”，257 定义为“传输结束”或“停止”。编码消息是一个数字序列，代码表示字典条目。 最初，大多数字典条目由单个字符组成，但随着消息的分析，定义了代表两个或更多字符的新条目。 结果总结在表 3.2 中。

32	个空格	116	个t
98	个b	121	y
105	i	256	开始
110	个n	257	停止

表3.1：LZW示例1起始字典

编码算法：定义一个用于构建新字典条目的位置，并称之为新条目。 从新条目为空开始，并发送起始代码。 然后逐个从被压缩的字符串中添加字符到新条目中。 一旦新条目无法匹配任何现有的字典条目，就使用下一个可用的代码点将新条目放入字典中，并发送不包含最后一个字符的字符串的代码（该条目已经在字典中）。 然后使用接收到的最后一个字符作为下一个新条目的第一个字符。 当输入字符串结束时，发送新条目中的代码，后跟停止代码。 就是这样。

为了那些喜欢看到像计算机程序一样编写的算法的人的利益，该编码算法在图3.5中显示。当将此过程应用于所讨论的字符串时，第一个字符

```
 0  # 编码算法
 1  清空字典
 2  发送起始代码
 3  对于每个字符 {
 4     如果新条目加上字符不在字典中 {
 5        发送新条目的代码
 6        将新条目加上字符作为新的字典条目
 7        将新条目置空
 8     }
 9     将字符添加到新条目中
10  }
11  发送新条目的代码
12  发送停止代码
```

图3.5：LZW编码算法

编码				传输		解码		
输入		新字典条目		9位字符传输		新字典条目		输出
105	i	–	–	256	(起始)	–	–	–
116	t	258	i t	105	i	–	–	i
116	t	259	t t	116	t	258	i t	t
121	y	260	t y	116	t	259	t t	t
32	空格	261	y 空格	121	y	260	t y	y
98	b	262	空格 b	32	空格	261	y 空格	空格
105	i	263	b i	98	b	262	空格 b	b
116	t	–	–	–	–	–	–	–
116	t	264	i t t	258	i t	263	b i	i t
121	y	–	–	–	–	–	–	–
32	空格	265	t y 空格	260	t y	264	i t t	t y
98	b		–	–	–	–	–	–
105	i	266	空格-b i	262	空格 b	265	t y 空格	空格 b
116	t	–	–	–		–	–	–
32	空格	267	i t 空格	258	i t	266	空格 b i	i t
98	b	–	–	–		–	–	–
105	i	–	–	–		–	–	–
110	n	268	空格 b i n	266	空格 b i	267	i t 空格	空格 b i
–	–	–	–	110	n	268	空格 b i n	n
–	–	–	–	257	(停止)	–	–	–

8位字符输入

表3.2：LZW示例1传输摘要

是“i”，并且只包含该字符的字符串已经在字典中。 因此，下一个字符被附加到新条目中，结果是“it”，该条目不在字典中。 因此，发送了字典中的字符串“i”，并将字符串“i t”添加到字典中的下一个可用位置，即258。 新条目被重置为只是最后一个未发送的字符，因此它是“t”。 下一个字符“t”被附加，结果是“tt”，该结果不在字典中。 该过程重复直到达到字符串的末尾。

在开始的一段时间里，额外的字典条目都是由两个字符组成的字符串，并且每遇到一个新字符就会传输一个字符串。 然而，当这些两个字符的字符串中的一个重复出现时，它的代码就会被发送（使用比分开发送两个字符所需的更少的位数），并定义一个新的三个字符的字典条目。 在这个例子中，它发生在字符串“i t t”上（这个消息的设计是为了比正常文本预期更早地发生这种情况）。

在这个例子中的后面，一个三个字符的字符串的代码被发送，并定义了一个四个字符的字典条目。

在这个例子中，代码被发送到一个接收器，该接收器预计解码消息并产生原始字符串作为输出。 接收器无法访问编码器的字典，因此解码器必须建立自己的副本。

解码算法：如果接收到起始代码，则清除字典并设置新条目为空。 对于接收到的下一个代码，输出由代码表示的字符，并将其放入新条目中。 然后，对于接收到的后续代码，将代码表示的字符串的第一个字符附加到新条目中，将结果插入字典中，然后输出接收到的代码的字符串，并将其放入新条目中以开始下一个字典条目。 当接收到停止代码时，不需要执行任何操作；新条目可以被丢弃。

该算法在图3.6中以程序格式显示。

```
 0  # 解码算法
 1  对于每个接收到的代码，直到停止代码为止 {
 2    如果代码是起始代码 {
 3      清除字典
 4      获取下一个代码
 5      从字典中获取字符串 # 它将是一个单个字符
 6    }
 7    否则 {
 8      从字典中获取字符串
 9      通过将字符串的第一个字符附加到最后一个字典条目来更新最后一个字典条目
10    }
11    将字符串作为新的字典条目添加
12    输出字符串
13  }
```

图3.6：LZW解码算法

请注意，编码器和解码器都会即时创建字典；因此，字典不需要显式传输，并且编码器在单次传递中处理文本。

这个方法有效吗？即传输所需的位数是否减少了？ 我们在14次9位传输（126位）中发送了18个8位字符（144位），这个非常短的例子节省了12.5%。 对于典型的文本，字符串长度不到500字节时减少的并不多。 较大的文本文件通常可以压缩2倍，而绘图则更多。

3.7.2 LZW算法，示例2

编码和解码文本消息

```
itty bitty nitty grrritty bit bin
```

（同样，这个奇特的短语被设计成具有重复的字符串，以便字典迅速形成；它还有一个三个字符长的序列 `rrr`，这说明了该算法的一个方面）。

初始字典条目包括表3.3中的字符，这些字符在字符串中找到，以及用于开始和停止的控制字符。

32	空格	114	r
98	b	116	t
103	g	121	y
105	i	256	开始
110	n	257	停止

表3.3：LZW示例2起始字典

在这里可以应用与示例1相同的算法。 结果显示在表3.4中。请注意，字典建立得非常迅速，并且有一个四字符字典条目的实例被传输。 这种压缩有效吗？绝对有效。总共发送了33个8位字符（264位）在22个9位传输（198位，甚至包括开始和停止字符）中，节省了25%的位数。

在这个例子中，有一个地方需要解码器做一些不寻常的事情。通常，在接收到传输的编码字后，解码器可以在字典中查找其字符串，然后输出它并使用其第一个字符来完成部分形成的最后一个字典条目，然后开始下一个字典条目。 因此只需要进行一次字典查找。 然而，上面介绍的算法使用的是

两次查找，一次是为了第一个字符，另一次是为了整个字符串。 为什么不只使用一次查找
以提高效率？

在这个例子中，有一个特殊情况，即代码271的传输，其中对应于接收到的代码的字符串不完整。 可以找到第一个字符，但在检索到整个字符串之前，必须完成输入。 当一个字符或字符串连续出现三次时，就会发生这种情况，因此很少见。 上述算法的工作是正确的，但需要额外的查找，这很少需要，并且可能会减慢算法的速度。 只有在检测到这种情况并将其视为特殊情况时，具有单个字典查找的更快算法才能可靠地工作。

编码				传输		解码		
输入		新字典条目				新字典条目		输出
105	i	–	–	256	(起始)	–	–	–
116	t	258	i t	105	i	–	–	i
116	t	259	t t	116	t	258	i t	t
121	y	260	t y	116	t	259	t t	t
32	空格	261	y 空格	121	y	260	t y	y
98	b	262	空格 b	32	空格	261	y 空格	空格
105	i	263	b i	98	个 b	262	空格 b	b
116	t	–	–	–	–	–	–	–
116	t	264	i t t	258	i t	263	b i	i t
121	y	–	–	–	–	–	–	–
32	空格	265	t y 空格	260	t y	264	i t t	t y
110	n	266	空格 n	32	空格	265	t y 空格	空格
105	i	267	n i	110	n	266	空格 n	n
116	t	–	–	–	–	–	–	–
116	t	–	–	–	–	–	–	–
121	y	268	i t t y	264	i t t	267	n i	i t t
32	空格	–	–	–	–	–	–	–
103	g	269	y 空格 g	261	y 空格	268	i t t y	y 空格
114	r	270	g r	103	g	269	y 空格 g	g
114	r	271	r r	114	r	270	g r	r
114	r	–	–	–	–	–	–	–
105	i	272	r r i	271	r r	271	r r	r r
116	t	–	–	–	–	–	–	–
116	t	–	–	–	–	–	–	–
121	y	–	–	–	–	–	–	–
32	空格	273	i t t y 空格	268	i t t y	272	r r i	i t t y
98	b	–	–	–	–	–	–	–
105	i	274	空格 b i	262	空格 b	273	i t t y 空格	空格 b
116	t	–	–	–	–	–	–	–
32	空格	275	i t 空格	258	i t	274	空格 b i	it
98	b	–	–	–	–	–	–	–
105	i	–	–	–	–	–	–	–
110	n	276	空格 b i n	274	空格 b i	275	i t 空格	空间 b i
–	–	–	–	110	n	276	空格 b i n	n
–	–	–	–	257	(停止)	–	–	–

表3.4：LZW示例2传输摘要

3.8 详细信息：2-D离散余弦变换

本节基于Luis P´erez-Breva于2005年2月3日的笔记和Joseph C. Huang于2000年2月25日的笔记。

离散余弦变换（DCT）是JPEG（联合图像专家组）压缩算法的一个重要组成部分。 DCT用于将像素阵列中的信息转换为最相关于人类感知的形式，并保留该信息，而较不相关的信息则可以被丢弃。

DCT是许多离散线性变换中用于图像压缩的一种。 它的优点是可以使用快速算法（与FFT，快速傅里叶变换相关）进行计算。

有多种数学符号可以用来描述DCT，其中最简洁的是使用向量和矩阵。 向量是一维数组，包含数字（或其他内容）。 它可以用单个字符表示，也可以用大方括号表示，并将各个元素显示为垂直列（也可以使用行表示，但通常被视为向量的转置）。 在这些笔记中，我们将使用粗体字母（$\mathbf{V}$）表示向量。 矩阵是二维数组，包含数字（或其他内容），同样可以用单个字符或大方括号表示。 在这些笔记中，我们将使用一种名为“黑板粗体”（$\mathbb{M}$）的字体表示矩阵。 当需要指示向量或矩阵的特定元素时，使用带有一个或两个下标的符号。 对于向量，单个下标是从0到$n-1$的整数，其中n是向量中的元素数。 对于矩阵，第一个下标表示行，第二个下标表示列，每个下标都是从0到$n-1$的整数，其中n是行数或列数，只有在矩阵是方阵时才相同。

3.8.1 离散线性变换

一般来说，离散线性变换将一个向量作为输入，并返回相同大小的另一个向量。 输出向量的元素是输入向量元素的线性组合，因此可以通过矩阵乘法进行该变换。

例如，考虑以下矩阵乘法：

$$\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}4\\ 1\end{bmatrix} \longrightarrow \begin{bmatrix}5\\ 3\end{bmatrix}. \tag{3.1}$$

如果此方程中的向量和矩阵被命名为

$$\mathbb{C}=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix},\quad \mathbf{I}=\begin{bmatrix}4\\ 1\end{bmatrix},\quad \mathbf{O}=\begin{bmatrix}5\\ 3\end{bmatrix},$$

那么方程3.1变为

$$\mathbf{O}=\mathbb{C}\mathbf{I}. \tag{3.2}$$

现在我们可以将 $\mathbb{C}$视为将输入向量 $\mathbf{I}$转换为输出向量 $\mathbf{O}$的离散线性变换。 顺便说一下，这个特定的变换 $\mathbb{C}$是将输入向量 $\mathbf{I}$转换为包含其分量之和 (5)和差 (3)的向量。[3]对于作用在3个元素向量上的3× 3矩阵，过程是相同的：

$$\begin{bmatrix}c_{1,1} & c_{1,2} & c_{1,3}\\ c_{2,1} & c_{2,2} & c_{2,3}\\ c_{3,1} & c_{3,2} & c_{3,3}\end{bmatrix}\begin{bmatrix}i_1\\ i_2\\ i_3\end{bmatrix} \longrightarrow \begin{bmatrix}o_1 & =\sum_j c_{1,j}i_j\\ o_2 & =\sum_j c_{2,j}i_j\\ o_3 & =\sum_j c_{3,j}i_j\end{bmatrix} \tag{3.3}$$

这又可以用简洁的形式写成

$$\mathbf{O}=\mathbb{C}\mathbf{I}. \tag{3.4}$$

[3]碰巧 $\mathbb{C}$是 $\sqrt{\ }$ 2倍于第 3.8.2节中定义的2×2离散余弦变换矩阵。

现在向量的大小为 3，矩阵为 3×3。

一般来说，对于这种形式的变换，如果矩阵 $\mathbb{C}$有逆矩阵 $\mathbb{C}^{-1}$，则可以通过变换的逆过程重构向量 $\mathbf{I}$。

$$\text{我} = \mathbb{C}^{-\mathbf{1}}\mathbf{O}. \qquad (3.5)$$

方程3.3和3.1说明了当输入为列向量时的线性变换。 行向量的过程类似，但向量和矩阵的顺序相反，并且变换矩阵被转置。[4] 这种变化与将行向量视为列向量的转置一致。例如：

$$\begin{matrix} \curvearrowright \\ 4 \quad 1 \end{matrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \\ \longrightarrow \end{bmatrix} \begin{matrix} \curvearrowright \\ 5 \quad 3 \end{matrix}. \qquad (3.6)$$

向量在处理具有一维特性的对象时非常有用，例如有限次采样的声波形。 图像本质上是二维的，使用矩阵来表示像素的属性是很自然的。 视频本质上是三维的（两个空间维度和一个时间维度），使用三维数组来表示其数据是很自然的。 这里给出的简洁的向量-矩阵表示法在二维系统中可以很好地扩展，但在更高维度上（可以使用其他数学表示法）。

将线性变换扩展到作用于矩阵而不仅仅是向量并不困难。 例如，
考虑一个非常小的图像，有六个像素，每行三个像素，或者每列三个像素。
表示每个像素某种属性的数字（例如在0到1的亮度尺度上的亮度）可以形成一个
3×2矩阵：

$$\begin{bmatrix} i_{1,1} & i_{1,2} \\ i_{2,1} & i_{2,2} \\ i_{3,1} & i_{3,2} \end{bmatrix} \qquad (3.7)$$

导致一个3×2输出矩阵的最一般的线性变换需要36个系数。 当矩阵中元素的排列反映了所表示的基础对象时，一组更少一般的线性变换，分别对行和列进行操作，使用不同的矩阵 $\mathbb{C}$和 $\mathbb{D}$，可能会有用：

$$\begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \begin{bmatrix} i_{1,1} & i_{1,2} \\ i_{2,1} & i_{2,2} \\ i_{3,1} & i_{3,2} \end{bmatrix} \begin{bmatrix} d_{1,1} & d_{1,2} \\ d_{2,1} & d_{2,2} \end{bmatrix} \begin{bmatrix} \\ \longrightarrow \end{bmatrix} \begin{bmatrix} o_{1,1} & o_{1,2} \\ o_{2,1} & o_{2,2} \\ o_{3,1} & o_{3,2} \end{bmatrix} \qquad (3.8)$$

或者，用矩阵表示，

$$\mathbb{O} = \mathbb{C}\mathbb{I}\mathbb{D}, \qquad (3.9)$$

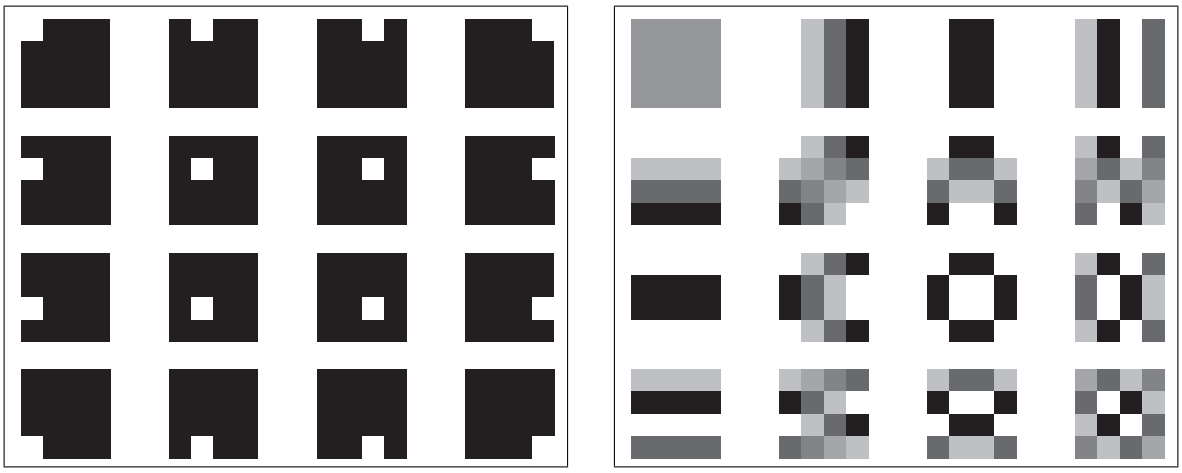
注意，此处左边的矩阵 $\mathbb{C}$和右边的矩阵 $\mathbb{D}$通常具有不同的大小，并且可能或可能不具有相同的一般特性。 (一个重要的特殊情况是当 $\mathbb{I}$是方阵时，即它包含相同数量的行和列。在这种情况下，输出矩阵 $\mathbb{O}$也是方阵，而 $\mathbb{C}$和 $\mathbb{D}$具有相同的大小。)

3.8.2 离散余弦变换

用线性代数的语言来说，公式

$$\mathbb{Y} = \mathbb{C}\mathbb{X}\mathbb{D} \qquad (3.10)$$

[4]转置矩阵意味着将其元素沿主对角线翻转，所以转置的i, j元素是原始矩阵的j, i元素。 转置矩阵用上标 T表示，如 $\mathbb{C}^T$。 一般来说，两个矩阵（或向量）的乘积的转置是两个转置矩阵或向量的乘积，顺序相反：$(\mathbb{A}\mathbb{B})^T = \mathbb{B}^T\mathbb{A}^T$。



（a）变换矩阵 $\mathbb{Y}$ （b）基矩阵 $\mathbb{X}$

图3.7：（a）表示出现在方程3.15中的矩阵 $\mathbb{Y}$中的系数的4×4像素图像。以及，（b）相应的逆离散余弦变换，这些ICDT可以解释为与 $\mathbb{Y}$的系数对应的基本图像。

表示将矩阵 $\mathbb{X}$转化为系数矩阵 $\mathbb{Y}$的转换。 假设转换矩阵 $\mathbb{C}$和 $\mathbb{D}$都有逆矩阵 $\mathbb{C}^{-1}$和 $\mathbb{D}^{-1}$，原始矩阵可以通过逆转换从系数中重构出来：

$$\mathbb{X} = \mathbb{C}^{-1}\mathbb{Y}\mathbb{D}^{-1}。 \tag{3.11}$$

将 $\mathbb{Y}$解释为重构 $\mathbb{X}$的系数是离散余弦变换特别有用的解释。

离散余弦变换是上述类型的离散线性变换。

$$\mathbb{Y} = \mathbb{C}^T\mathbb{X}\mathbb{C}, \tag{3.12}$$

其中所有矩阵的大小都为 $N\times N$，两个转换矩阵互为转置。
这个变换被称为余弦变换，因为矩阵 $\mathbb{C}$被定义为

$$\{\mathbb{C}\}_{m,n} = k_n \cos\left[\frac{(2m+1)n\pi}{2N}\right] \text{其中} \quad k_n = \begin{cases} \sqrt{1/N} & \text{if } n = 0 \\ \sqrt{2/N} & \text{otherwise} \end{cases} \tag{3.13}$$

其中m, n= $0, 1, \ldots (N-1)$. 这个矩阵 $\mathbb{C}$有一个等于其转置的逆矩阵：

$$\mathbb{C}^{-1} = \mathbb{C}^T. \tag{3.14}$$

使用方程式3.12和方程式3.13中定义的 $\mathbb{C}$，我们可以计算任意矩阵$\mathbb{X}$的DCT $\mathbb{Y}$，其中矩阵 $\mathbb{X}$可以表示给定图像的像素。 在DCT的背景下，方程式3.11中概述的逆过程被称为逆离散余弦变换（IDCT）：

$$\mathbb{X} = \mathbb{C}\mathbb{Y}\mathbb{C}^T. \tag{3.15}$$使用这个方程式，我们可以计算DCT的一组基本矩阵，即：通过DCT与 $\mathbb{Y}$的每个元素对应的矩阵集合。 让我们构建所有可能的图像集合每个集合只有一个非零像素。 这些图像将代表矩阵 $\mathbb{Y}$的个别系数。

图3.7(a)显示了4× 4像素图像的集合。 图3.7(b)显示了将IDCT应用于图3.7(a)中的图像的结果。图3.7(b)中的图像集被称为基础，因为它们的DCT会产生一个只有一个非零系数的矩阵 $\mathbb{Y}$，因此它们代表了DCT“分解”任何输入图像的基本图像。

回顾我们上面对离散线性变换的概述，如果我们想从其DCT $\mathbb{Y}$恢复图像$\mathbb{X}$，我们只需将 $\mathbb{Y}$的每个元素与3.7(b)中相应的矩阵相乘。确实，图3.7(b)介绍了DCT基础的一个非常显著的特性：它编码了空间频率。 通过忽略具有较小DCT系数的空间频率可以实现压缩。 想象一下国际象棋棋盘的图像-它具有高空间频率分量，几乎可以去除所有低频分量。 相反，模糊图像往往具有较少的较高空间频率分量，然后可以将高频分量（图3.7(b)中的右下角）设置为零作为“可接受的近似”。 这是JPEG不可逆压缩背后的原理。

图3.8显示了用于生成4× 4、8× 8和16×16 DCT的基础图像的MATLAB代码。

```
% N是进行DCT的NxN图像的大小。
% 此代码将考虑4x4、8x8和16x16图像
% 并构建DCT变换的基础
 N = [4 8 16]的基础。
    % 创建变换矩阵C
    C = zeros(N,N);
    mrange=0:(N-1);             % m表示行
    nrange=mrange;              % n表示列
    k=ones(N,N)*sqrt(2/N);      % 创建归一化矩阵
    k(:,1)=sqrt(1/N);           % 注意第一列的不同归一化
    C=k.*cos((2*mrange+1)'*nrange*pi/(2*N));     % 构建变换矩阵
    % 注意我们对逆离散余弦变换感兴趣
    % 因此我们必须反转（即转置）C
    C=C';
    % 获取基础矩阵
    figure;colormap('gray');%打开图像并将颜色设置为灰度
    for m = mrange
        for n = nrange
            %创建一个只有一个像素的矩阵
            Y=zeros(N,N);
            Y(n+1,m+1)=1;

            X = C'*Y*C;         % X是转换后的矩阵。
            subplot(N,N,m*N+n+1);imagesc(X);%绘制X
            axis square;        %调整坐标轴的刻度
            axis off;           %隐藏坐标轴
        end
    end
end
```

图3.8：基础矩阵生成器

第4章

错误

在第2章中，我们看到了如何用位数组表示符号的示例。 在第3章中，我们研究了一些压缩这些符号的位表示或一系列这些符号的技术，以便需要更少的位来表示它们。 如果在保留所有原始信息的同时进行压缩，则称为无损压缩或可逆压缩，但如果在丢失（可能是不重要的）信息的同时进行压缩，则称为有损压缩或不可逆压缩。 通常，源编码和压缩被合并为一个操作。

由于压缩，承载相同信息的比特位更少，因此每个比特位更重要，而单个比特位的错误会带来更严重的后果。 所有实际系统都会引入对其处理的信息的错误（当然，某些系统比其他系统更多）。 在本章中，我们将研究确保这种不可避免的错误不会造成任何损害的技术。

4.1 系统模型的扩展

我们的信息处理模型将扩展以包括“信道编码”。新的信道编码器会向消息中添加比特位，以便在某种方式下它被损坏时，信道编码器将知道并且可能甚至能够修复损坏。

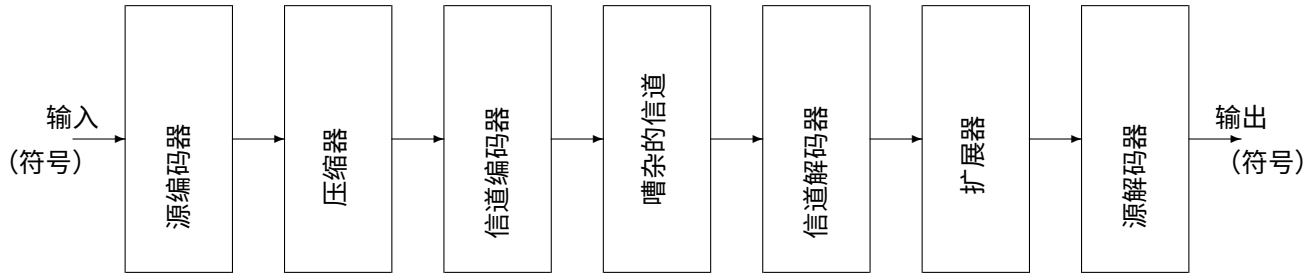


图4.1：带有错误的通信系统

4.2 错误是如何发生的?

上面所示的模型非常通用，因为其目的可能是从
一个地方传输信息到另一个地方（通信），将其存储以供以后使用（数据存储），甚至是处理它以使
输出不是输入的忠实副本（计算）。 不同的系统涉及不同的
物理设备作为信道（例如通信链路，软盘或计算机）。 许多
物理效应会导致错误。 CD或DVD可能会被刮伤。存储单元可能会失效。 电话线可能
会有噪音。 计算机门可能会对电源电压的意外波动做出响应。

为了我们的目的，我们将把所有这样的错误建模为一个或多个位从1变为0或反之的变化。
在通常情况下，一条消息由多个位组成，我们通常假设不同的位独立地发生错误，但在某些情况下，相邻位的错误不是彼此独立的，而是有一个共同的潜在原因（即错误可能会连续发生）。

4.3 检测与纠正

处理错误有两种方法。 一种是检测错误，然后让使用输出的人或系统知道发生了错误。 另一种方法是让信道解码器尝试修复消息并纠正错误。 在这两种情况下，都会向消息中添加额外的位，使其变长。 结果是消息包含冗余信息 - 如果没有冗余信息，每个可能的位模式都将是一个合法的消息，而错误只会将一个有效的消息变为另一个有效的消息。 通过改变事物，使得许多（实际上，大多数）位模式不对应合法的消息，错误的影响通常是将消息改变为非法模式之一；信道解码器可以检测到错误并采取适当的措施。 实际上，如果每个非法模式在某种意义上比任何其他模式更接近一个合法的消息，解码器可以替换最接近的合法消息，从而修复损坏。

在日常生活中，错误检测和纠正经常发生。 书面和口头交流是用英语等自然语言进行的，并且有足够的冗余（估计为50%），即使省略了几个字母、声音甚至单词，人们仍然能理解信息。

请注意，信道编码器由于添加了位模式，通常会保留所有原始信息，因此是可逆的。 信道通过允许错误发生，实际上引入了信息（确切哪些位发生了变化的细节）。 解码器是不可逆的，因为它丢弃了一些信息，但如果设计良好，它会丢弃由错误引起的“坏”信息，并保留原始信息。 在后面的章节中，我们将定量分析这种系统中的信息流动。

4.4 海明距离

我们需要一些技术来判断两个位模式的相似程度。对于长度等物理量，将两个测量视为接近或近似相等是很自然的。是否存在一种类似的方式，可以判断两个位模式是否接近?

起初，人们很容易认为，如果两个位模式表示相邻的整数或接近的浮点数，它们就是相似的。 然而，这个概念并不有用，因为它基于对位模式赋予的特定含义。 很难说两个位模式在第一个位上的差异比在最后一个位上的差异更大或更小。两个位模式之间的差异更有用的定义是它们之间不同的位数。这被称为汉明距离，以Richard W. Hamming (1915 – 1998)的名字命名[1]。 因此，0110和1110之间的汉明距离为一。

相同的模式之间的汉明距离为零。

[1]在 http://www-groups.dcs.st-andrews.ac.uk/%7Ehistory/Biographies/Hamming.html上查看 Hamming的传记

请注意，汉明距离只能在具有相同位数的两个位模式之间定义。 讨论单个比特串的汉明距离或不同长度的两个比特串的汉明距离是没有意义的。

使用这个定义，信道引入的错误的影响可以通过输入信道和输出信道的两个位模式之间的汉明距离来描述。 没有错误意味着汉明距离为零，而单个错误意味着汉明距离为一。如果发生两个错误，通常意味着汉明距离为二。（然而，请注意，如果两个错误发生在同一位上，第二个错误将取消第一个错误，实际上汉明距离将为零。）编码器的作用也可以用汉明距离来理解。 为了提供错误检测，编码器必须产生的位模式使得任意两个不同的输入在输出中至少相隔两个汉明距离，否则一个单个错误可能会将一个合法的码
字转换为另一个。 为了提供双重错误保护，任意两个有效码字之间的分离距离必须至少为三。 为了实现单个错误纠正，所有有效码字之间必须至少相隔三个汉明距离。

4.5 单个比特

传输一个比特可能看起来不重要，但它确实引出了一些常用的错误检测和纠正技术。

保护一个单独的比特的方法是多次发送它，并期望大多数情况下每个发送的比特都不会改变。 最简单的情况是发送两次。 因此，通道编码器将消息0替换为00，将1替换为11。 如果两个比特不同（这只能是由于错误导致的），解码器可以发出警报。 但是有一个微妙的问题。 如果有两个错误会怎样？ 如果两个错误都发生在同一个比特上，那么该比特将恢复到其原始值，就好像没有发生错误一样。 但是如果两个错误发生在不同的比特上，它们最终会变得相同，尽管是错误的，并且错误将无法检测。 如果有更多的错误，那么未检测到的更改的可能性就会变得很大（奇数个错误会被检测到，但偶数个错误不会被检测到）。

如果可能出现多个错误，增加冗余可以有所帮助。 因此，为了检测双重错误，可以将单个位发送三次。除非通道解码器接收到的三个位都相同，否则就知道发生了错误，但不知道可能有多少个错误。 当然，三重错误可能不会被检测到。

那么如何使解码器能够纠正错误，而不仅仅是检测错误呢？ 如果已知最多只有一个错误，并且发送三次相同的位，则通道解码器可以判断是否发生了错误（如果三个位不完全相同），并且还可以确定原始值是什么-这个过程有时被称为“多数逻辑”（选择出现最频繁的位）。 这种称为“三重冗余”的技术可以用于保护通信通道、存储器或任意计算。

请注意，三重冗余可以用于纠正单个错误或检测双重错误，但不能同时实现两者。 如果需要同时实现两者，可以使用四重冗余-发送四个相同的位。两个重要问题是这些技术的效率和有效性。至于效率，
方便地将编码前的位数除以编码后的位数定义为编码率。 因此，编码率介于0和1之间。 双重冗余导致编码率为0.5，三重冗余为0.33。至于有效性，如果错误非常不可能发生，忽略两个错误非常接近的情况可能是合理的。 如果是这样，三重冗余非常有效。 另一方面，一些物理错误源可能会导致大量数据丢失（想象一下CD上的物理划痕），在这种情况下，即使错误不太可能，也很可能伴随着相邻位上的类似错误，因此三重冗余将不会起作用。

图4.2和4.3说明了三重冗余如何保护单个错误，但如果有两个错误则可能失败。

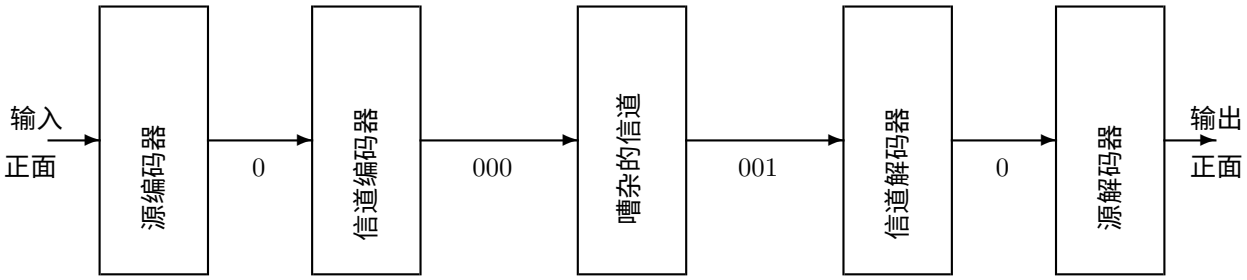


图4.2：三重冗余信道编码和单错误纠正解码，用于引入一个位错误的信道。

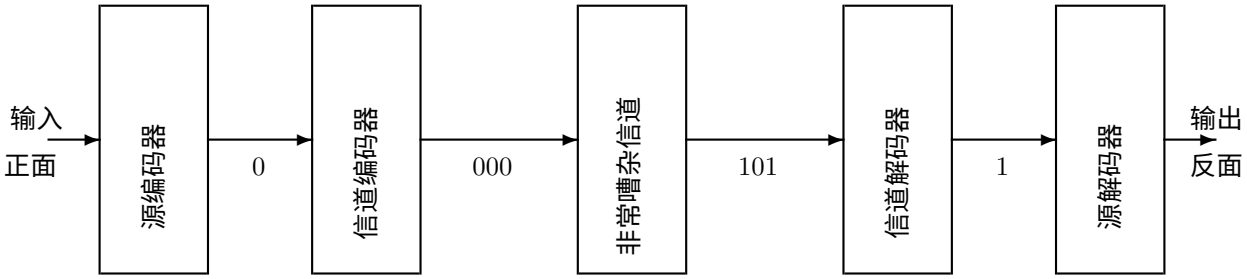


图4.3：三重冗余信道编码和单错误纠正解码，用于引入两个位错误的信道。

4.6 多位

为了检测位序列中的错误，可以使用几种技术。 有些技术可以进行错误纠正和检测。

4.6.1 奇偶校验

考虑一个字节，它由8位组成。 为了能够检测单个错误，可以添加一个“奇偶校验”位（也称为“检查位”），将8位字符串变为9位。 如果位数等于1的位数是奇数，则添加的位为1，否则为0。 因此，9位字符串始终具有偶数个位数等于1。 然后解码器只需计算1位的数量，如果是奇数，则知道存在错误（或者，更一般地，存在奇数个错误）。 解码器无法修复损坏，并且实际上无法确定损坏是否可能偶然发生在奇偶校验位上，如果是这种情况，则数据位仍然正确。 它也无法检测双重错误（或者更一般地说，偶数个错误）。 奇偶校验位的使用是高效的，因为编码率为$8/9$，但有效性有限。 它无法处理信道表示计算的情况，因此输出不打算与输入相同。 它通常在错误的可能性非常小，并且没有理由认为相邻位的错误会一起发生，并且接收器能够请求重新传输数据时使用。

有时候即使无法重新传输，也会使用奇偶校验。 在早期的IBM个人电脑上，内存引用受到单比特奇偶校验的保护。 当检测到错误时（非常少见），计算机会崩溃。

纠错比错误检测更有用，但需要更多的比特，因此效率较低。
下面讨论两种较常见的方法。

如果被传输的数据更方便地被视为对象的序列，这些对象本身可以用多个比特进行编码，例如字母或数字，则可以通过添加类似的对象而不是奇偶校验位来实现奇偶校验位的优势。 例如，可以将校验位添加到数字数组中。
第4.9节讨论了校验位的一些常见用途。

4.6.2 矩形码

矩形码可以同时提供单错误纠正和双错误检测。 假设我们希望保护一个字节的信息，即八个数据位 D0 D1 D2 D3 D4 D5 D6 D7。让我们将它们排列在一个矩形表中，并为每行和每列添加奇偶校验位：

D0	D1	D2	D3	PR0
D4	D5	D6	D7	PR1
PC0	PC1	PC2	PC3	P

表4.1：奇偶校验位

这个想法是，每个奇偶校验位PR0 PR1 PC0 PC1 PC2 PC3都被设置为使得特定行或列的总奇偶校验位为偶数。 然后，总奇偶校验位P被设置为使得只由奇偶校验位组成的右侧列本身具有偶校验—这保证了底行也具有偶校验。
这15位可以通过信道发送，解码器分析接收到的位。 它对三行和五列进行了总共8次奇偶校验。 如果任何一个位有单个错误，那么三个行奇偶校验位和五个列奇偶校验位中的一个将是错误的。 因此，可以确定出错的位（它位于具有错误奇偶校验的行和列的交叉点）并进行更改。

如果有两个错误，就会出现不同的奇偶校验失败模式；双重错误可以被检测出来，但不能被纠正。 三重错误可能会模拟出一个无辜位的单一错误。可以设计其他基于几何的编码，例如将位排列成三角形、立
方体、金字塔、楔形或更高维结构。

4.6.3 汉明码

假设我们希望纠正单一错误，并愿意忽略多个错误的可能性。 Richard Hamming发明了一组带有最少额外奇偶校验位的编码。
信道编码器添加的每个额外位允许解码器进行一次奇偶校验，并因此提供一位可用于帮助确定错误位置的信息。 例如，如果使用了三个额外位，这三个测试可以识别出多达八种错误条件。 其中之一将是“无错误”，因此剩下的七个将用于确定模式中最多七个位置的错误位置。 因此，数据块可以是七位长。 其中三位用于错误检测，剩下四位用于有效载荷数据。 类似地，如果有四个奇偶校验位，数据块可以是15位长，剩下11位用于有效载荷。

对于给定数量的校验位，具有尽可能大有效载荷的编码有时被称为“完美”。当然，也可能有较小的有效载荷，这种情况下得到的海明码的编码率会较低。 例如，由于许多数据处理集中在每个8位长的字节上，一个方便的海明码可以使用四个校验位来保护八个数据位。 因此，可以在三个字节中处理两个字节的数据，以及校验位。

表4.2列出了一些海明码。 没有显示出只有1个校验位的特殊情况，因为没有空间容纳任何数据。 第一个条目很简单，我们已经见过了。 这是三重冗余，其中三个位的块用于一个数据位。正如我们之前看到的，这种方案能够进行单错误纠正或双错误检测，但不能同时进行（这对所有海明码都成立）。第二个条目是非常有趣的，因为它是具有合理效率的最简单的海明码。

让我们设计一个$(7,4,3)$汉明码。 有几种方法可以做到这一点，但最简单的方法可能是从解码器开始。解码器接收七个位并对其中的组进行三个奇偶校验。

奇偶校验位	块大小	负载	码率	块码类型
2	3	1	0.33	(3, 1, 3)
3	7	4	0.57	(7, 4, 3)
4	15	11	0.73	(15, 11, 3)
5	31	26	0.84	(31, 26, 3)
6	63	57	0.90	(63, 57, 3)
7	127	120	0.94	(127, 120, 3)
8	255	247	0.97	(255, 247, 3)

表4.2：完美的汉明码

通过标记位1到7，我们可以识别出错误发生的位置。 如果结果都是偶校验，解码器就会得出没有发生错误的结论。 否则，通过知道哪些奇偶校验操作失败，可以推断出发生了哪个位的改变。 在具有三个奇偶校验位的完美汉明码中，有一个特别优雅的码：

- 第一个奇偶校验使用位 4、 5、 6或 7，因此如果其中一个发生改变，它就会失败
- 第二个奇偶校验使用位 2、 3、 6或 7，因此如果其中一个发生改变，它就会失败
- 第三个奇偶校验使用位 1、 3、 5或 7，因此如果其中一个位发生改变，则校验失败

这些规则很容易记住。 这三个奇偶校验是有关错误位的二进制表示的一部分 - 例如，整数6的二进制表示为 1 1 0，对应于第一和第二个奇偶校验失败，但第三个奇偶校验没有失败。

现在考虑编码器。 在这七个位中，有四个是原始数据，另外三个是编码器添加的。 如果原始数据位是3 5 6 7，编码器可以根据上面给出的规则轻松计算出位1 2 4 - 例如，位2被设置为使位2 3 6 7的奇偶校验为偶数所需的值，这意味着如果位3 6 7的奇偶校验已经是偶数，则为0，否则为1。 编码器计算奇偶校验位并按所需顺序排列所有位。 然后解码器在必要时纠正一个位，提取数据位并丢弃奇偶校验位，因为它们已经完成了它们的工作并且不再需要。

4.7 块码

方便起见，可以考虑为一定量的数据提供纠错保护，然后将结果以长度为n的块发送。 如果该块中的数据位数为k，则奇偶校验位数为n-k，并且习惯上将这样的编码称为(n, k)块编码。 因此，刚刚描述的海明码是$(7, 4)$。

惯例上（在本讲义中也是如此），括号中还包括任意两个有效码字或原始数据项之间的最小海明距离d，形式为$(n, k, d\ )$。 刚刚描述的海明码可以被归类为$(7, 4, 3)$块编码。

4.8 高级编码

最小海明距离大于3的块编码是可能的。 它们可以处理多个错误。 一些被称为Bose-Chaudhuri-Hocquenhem（BCH）编码。 今天商业上非常感兴趣的是1960年由MIT林肯实验室的Irving S. Reed和Gustave Solomon宣布的一类字节数据编码，而不是位。 $(256, 224, 5)$和$(224, 192, 5)$的Reed-Solomon码用于CD播放器，可以一起防止长时间的错误突发。

更先进的信道编码利用过去的数据块和当前的数据块。 这种编码的编码器和解码器都需要本地存储器，但不一定需要很多。数据处理

对于这种先进的编码来说是非常具有挑战性的。 开发一个高效、能够防止大量错误、易于编程和执行速度快的代码并不容易。 一个重要的编码类别被称为卷积码，其中一个重要的子类是格子码，它们常用于数据调制解调器。

4.9 详细信息：检查位数

错误检测常常用于减少人为错误。很多时候，人们必须处理长串的序列号或字符序列，这些序列号要么被大声读出，要么在键盘上输入。例如信用卡号码、社会安全号码和软件注册码。这些操作容易出错。可以包含额外的数字或字符来检测错误，就像在位串中包含奇偶校验位一样。通常情况下，这已经足够了，因为当检测到错误时，操作可以方便地重复执行。

在其他情况下，例如电话号码或电子邮件地址，不使用校验字符，因此
任何序列都可能是有效的。显然，在使用这些信息时需要更加小心，以避免拨打错误的
号码或向错误的人发送电子邮件或传真。

信用卡

信用卡号码有一个额外的校验位，是根据IBM于1954年指定的一种方式计算的。它旨在防止一种常见的错误类型，即两个相邻数字的调位。

信用卡号码通常包含15或16位数字（Luhn算法实际上适用于任意位数的数字）。前六位数字表示发卡机构。金融行业不鼓励公开披露这些代码，尽管大多数已经广为人知，特别是那些严肃考虑欺诈的人。在这六位数字中，第一位表示与卡片相关的经济部门，例如1和2表示航空公司，3表示旅行和娱乐，4、5和6表示银行和商店。最后一位是校验位，其他数字表示个人卡账户。

根据这个方案，信用卡发行商已被分配了自己的前缀。例如，美国运通卡的号码以34或38开头，Visa卡以4开头，MasterCard卡以51、52、53、54或55开头，Discover卡以6011或65开头。

Luhn过程用于验证信用卡号码（包括校验位）是否有效。首先，选择卡号中从倒数第二位开始的交替位置上的数字。
例如，如果卡号是1234 4567 7891，那些数字将是9、7、6、4、3和1。注意有多少个数字大于4（在这个例子中有3个）。然后将这些数字相加（例如，$9+7+6+4+3+1=30$）。然后将卡号中的所有数字相加（在这个例子中为57）。查看这三个数字的总和（在这个例子中为$3+30+57=90$）。如果结果是10的倍数，就像这个例子一样，那么该卡号通过了测试，可能是有效的。否则，它是无效的。

该过程可以检测所有个位数错误，几乎可以检测出所有相邻数字的置换错误（例如，输入“1243”而不是“1234”），但是还有许多其他可能的转录错误无法被检测到，例如“3412”而不是“1234”。它具有较高的编码率（只在14或15个有效数字中添加一个校验位），并且使用起来很简单。它不能用于纠正错误，因此只在其他手段用于纠正的情况下才有价值。

ISBN

国际标准书号（ISBN）是一个13位数字，用于唯一标识一本书或类似书籍。同一本书的不同版本可能具有不同的ISBN。这本书可以是印刷品，也可以是电子书、音频磁带或软件。ISBN对消费者来说并不是很重要，但对书店、图书馆、作者、出版商和分销商来说是有用的。

该系统是由英国书商W.H.史密斯于1966年创建的，使用9位数字，然后在1970年进行了升级，通过在现有数字前面添加0来进行国际使用，然后在2007年进行了升级，通过在前面添加978并重新计算校验位来使用13位数字。

一本书的ISBN出现在以“ISBN”字母开头的数字后面，通常出现在封面背面或平装书的背面。通常它靠近一些条形码，并经常以机器可读的字体呈现。

ISBN由五个部分组成（2007年之前为四个部分），长度可变，由连字符分隔。首先是前缀978（2007年之前没有）。当使用该前缀的数字用尽时，将使用前缀979。接下来是一个国家标识符（或共享相同语言的国家或地区的群体）。

接下来是一个用于识别特定出版商的数字。 然后是标题的标识符，最后是单个校验位。 国家标识符由位于柏林的国际ISBN机构分配。出版商标识符由所代表的国家或地区内分配，而标题标识符由出版商分配。 校验位的计算方法如下所述。

例如，考虑ISBN 0-9764731-0-0（这是在2007年之前的格式）。 语言区域代码0代表讲英语的国家。 出版商9764731是麻省理工学院电气工程与计算机科学系。项目标识符0代表书籍《电子与比特》。这个标识符为0是因为这本书是该出版商使用ISBN出版的第一本书。

出版商标识符使用7位数字来识别，而项目只使用了1位数字，这反映了这是一个非常小的出版商，可能不需要超过十个ISBN。ISBN可以以10个一组购买（价格为269.95美元，截至2007年），100个一组（914.95美元），1000个一组（1429.95美元）或10000个一组（3449.95美元），购买时分配的出版商标识符分别为7位、6位、5位或4位数字。 这种安排方便处理许多小型出版商和少数大型出版商。

有利于许多小型出版商而不是少数大型出版商的出版趋势可能会对ISBN系统造成压力。 小型出版商每次至少要购买10个号码，如果只出版一两本书，未使用的号码不能被其他出版商使用。 那些主要不是出版商但偶尔出版书籍的组织很可能会丢失未使用的ISBN，因为没有人记得它们上次放在哪里。

2007年及以后出版的书籍采用了13位数字的ISBN，该系统设计与广泛应用于商店的UPC（通用产品代码）条形码兼容。 查找UPC校验位的过程也适用于13位数字的ISBN。从12位数字（不包括校验位）开始。 将奇数位置上的第一、第三、第五位等数字相加，并将总和乘以3。 然后将结果与偶数位置上的数字（第2、第4、第6、第8、第10和第12位）的总和相加。 从比结果大的下一个10的倍数中减去该结果。 结果是一个介于0和9之间（包括0和9）的数字，即所需的校验位。

这种技术产生了一个具有较大码率（0.92）的代码，可以捕捉到所有的个位数错误，但不能捕捉到所有的换位错误。

对于2007年之前出版的书籍，可以通过以下步骤计算校验位。 从九位数开始（不包括校验位）。 将每个数字乘以其位置，最左边的位置为1，最右边的位置为9。 将这些乘积相加，并找到该和对11取余的结果（即，需要减去的数字，使结果成为11的倍数）。 结果是一个介于0和10之间的数字。 这就是校验位。 例如，对于ISBN 0-9764731-0-0，$1 \times 0 + 2 \times 9 + 3 \times 7 + 4 \times 6 + 5 \times 4 + 6 \times 7 + 7 \times 3 + 8 \times 1 + 9 \times 0 = 154$，它对11取余为0。

如果校验位小于十，则用于ISBN。如果校验位为10，则使用字母X代替（这是罗马数字十的表示）。 如果你看几本书，你会发现不时地出现校验位X。

这种技术产生了一个具有较高码率（0.9）的代码，能够有效地检测两个相邻数字的置换或任意单个数字的改变。

ISSN

国际标准连续出版物编号（ISSN）是一个8位数字，用于唯一标识印刷或非印刷连续出版物。 ISSN以ISSN 1234-5678的形式写在每期连续出版物上。它们通常不被一般公众注意到，但对出版商、分销商和图书馆很有用。

ISSN用于报纸、杂志和许多其他类型的期刊，包括期刊、学会交易、专著系列甚至博客。 ISSN适用于整个系列，预计会无限期地继续下去，而不是适用于单个期刊。 分配的ISSN是永久的 - 如果连续出版物停止出版，ISSN不会被回收，如果连续出版物更改名称，则需要一个新的ISSN。 在美国，ISSN是逐个发放的，不收费，由国会图书馆的一个办公室发放。与ISBN不同，ISSN的各部分没有特定的含义，除了前七位数字形成一个唯一的编号，最后一位是校验位。 除非更改格式，否则最多只能分配1000万个ISSN。 截至2006年，全球已分配了1,284,413个ISSN，其中包括当年发放的 57,356个。

校验位的计算过程如下。 从七位数开始（不包括校验位）。 将每个数字乘以其（反向）位置，最左边的位置为8，最右边的位置为2。 将这些乘积相加，并从下一个更高的11的倍数中减去。 结果是一个介于0和10之间的数字。 如果小于10，则该数字为校验位。 如果等于10，则校验位为X。校验位成为最终ISSN的第八位数字。

第5章

概率

我们一直在考虑一个信息处理系统的模型，其中输入的符号被编码为比特，然后通过“通道”发送到接收器，并解码回符号。
见图5.1。

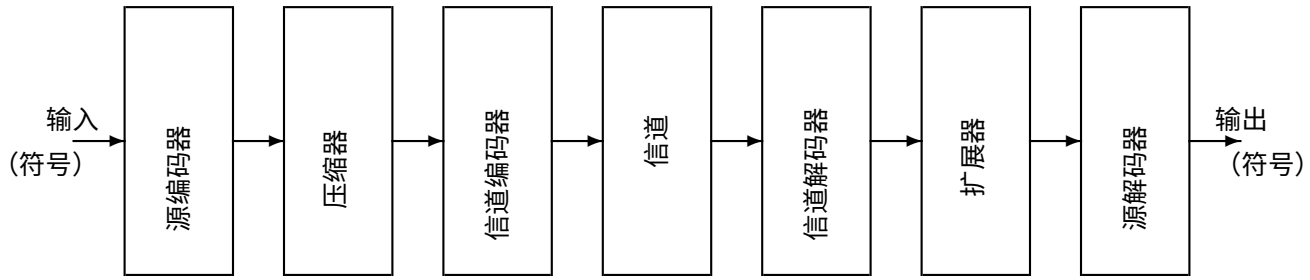


图5.1：通信系统

在本笔记的早期章节中，我们已经研究了这个模型中的各个组件。 现在我们回到源头，并更全面地对其进行建模，以概率分布为基础。

源提供了一个符号或一系列符号，从某个集合中选择。选择过程可能是一个实验，比如抛硬币或掷骰子。 或者它可能是观察到的不由观察者引起的行为。 或者符号序列可以来自某个对象的表示，比如文本中的字符或图像中的像素。

我们只考虑有限数量的可供选择的符号情况，以及符号既互斥（一次只能选择一个）又穷尽（实际上选择了一个）。 每个选择都构成一个“结果”，我们的目标是追踪结果的序列以及随之而来的信息，当信息从输入传输到输出时。 为了做到这一点，我们需要能够说明结果是什么，以及我们对结果的某些属性的知识。

如果我们知道结果，我们有一种完全有效的方式来表示结果。 我们只需命名所选择的符号，忽略其他未被选择的符号。 但是如果我们还不知道结果，或者对结果有任何程度的不确定性呢？ 如果存在不确定性，我们应该如何表达我们的知识状态？ 我们将使用概率数学来实现这个目的。

为了说明这个重要的概念，我们将使用基于MIT学生特点的例子。 MIT在2007年秋季的学生人数官方统计为[1]人，包括表5.[1]中的数据，该数据以Venn图的形式在图5.2中重现。

	女性	男性	新生
总数	496	577	1,07
3名本科生	1,857	2,315	4,17
2名研究生	1,822	4,226	6,04
8名学生总数	3,679	6,541	10,220

表5.1：MIT 2007年秋季的人口统计数据

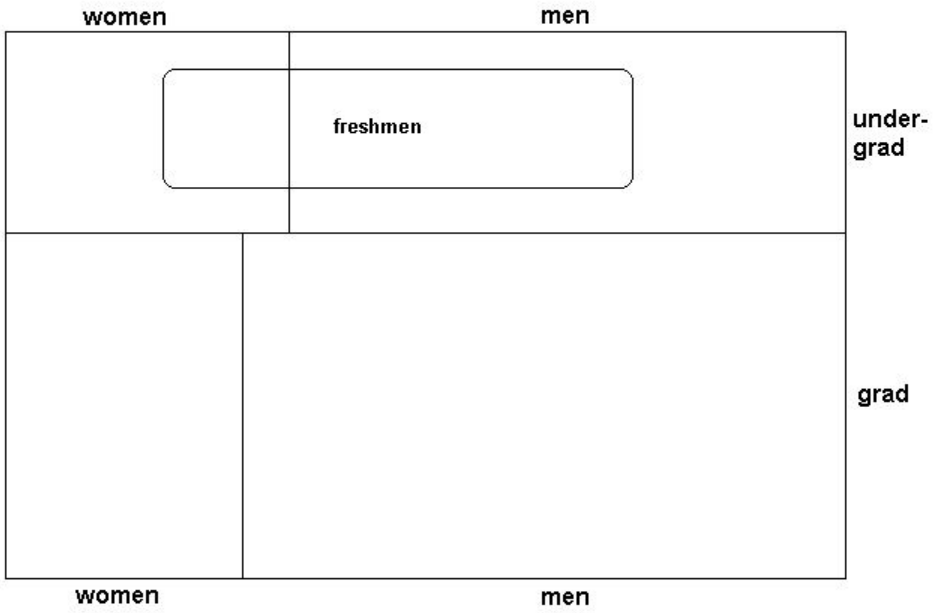


图5.2：MIT学生数据的Venn图，其中各个子群体的面积应与其规模成比例。

假设选择了一名MIT的新生（符号选择为一个个体学生，可能的符号集合为1073名新生），你不知道选择的是谁。你想知道他是女性还是男性。当然，如果你知道选择的学生的身份，你就会知道他的性别。

但如果不是这样，你如何描述你的知识？ 一个女人被选择的可能性，或概率是多少？

请注意，2007年新生班级中的46%（496/1, 073）是女性。 这是一个事实，或者是一个统计数据，它可能代表了被选择的新生是女性的概率。 如果你有理由相信所有新生被选择的可能性是相等的，你可能会认为是女性的概率是46%。 但是如果你被告知选择是在麦考密克大厅（一个女生宿舍）进行的呢？ 在这种情况下，被选择的新生是女性的概率可能高于46%。统计数据和概率都可以用相同的数学方法来描述（下面将介绍），但它们是不同的东西。

[1]所有学生：http://web.mit.edu/registrar/www/stats/yreportfinal.html，
所有女性：http://web.mit.edu/registrar/www/stats/womenfinal.html

5.1 事件

像数学或科学的许多分支一样，概率论有自己的术语，其中一个词可能与日常意义不同或更具体。 考虑两个词**event**，它有几个日常意义，和 **outcome**。 麦里姆-韦伯斯特学院词典给出了与概率论中的技术含义最接近的定义：

- **outcome**：作为结果或后果而发生的事情
- **event**：实验可能结果的子集

在我们的背景下， **outcome**是所选的符号，无论我们是否知道它。虽然在还没有进行选择的情况下谈论选择的结果是错误的，但是在考虑的选择的可能结果集合中谈论是可以的。 在我们的情况下，这是所有符号的集合。 至于 **event**这个术语，它最常见的日常意义是发生的事情。 我们的意思是上面引用的，它在词典中排在最后。 我们将以这种受限制的方式使用这个词，因为我们需要估计或描述符号各种属性的知识。 这些属性是每个符号是否适用的事物，一个方便的思考方式是将所有符号的集合分为两个子集，一个具有该属性，一个没有。 当进行选择时，会发生几个事件。 其中一个是结果本身。 这被称为**fundamentalevent**。 其他事件是选择具有特定属性的符号。

严格来说，一个事件是一组可能的结果，但在概率论中，通常将产生这些结果的实验称为事件。 因此，我们有时将选择称为事件。

例如，假设选择了一位麻省理工学院的新生。 具体选择的人就是结果。 基本事件将是那个人，或者选择那个人。 另一个事件可能是选择一位女性（或男性）。 另一个事件可能是选择来自加利福尼亚的人，或者选择年龄超过18岁的人，或者选择身高超过六英尺的人。 还可以考虑更复杂的事件，例如来自德克萨斯州的女性，或者来自密歇根州的男性，具有特定的SAT成绩。

选择任何符号的特殊事件是肯定会发生的。 我们将这个事件称为全事件，根据集合论中相应概念的名称。 选择没有符号的特殊“事件”被称为空事件。 空事件不可能发生，因为只有在进行选择后才能定义结果。

不同的事件可能重叠，也可能不重叠，也就是说两个或多个事件可能具有相同的结果。 一组不重叠的事件被称为互斥事件。 例如，选择的大一新生是（1）来自俄亥俄州，或者（2）来自加利福尼亚州，这两个事件是互斥的。

当选择任何符号时，几个事件可能具有至少一个事件发生的属性。
一组事件中，至少会发生其中一个事件，被称为穷尽事件。 例如，选择的大一新生是（1）年龄小于25岁，或者（2）年龄大于17岁，这两个事件是穷尽的，但不是互斥的。

既是互斥的又是穷尽的一组事件被称为划分事件。 包含所有基本事件的划分被称为基本划分。 在我们的例子中，选择女性和选择男性的两个事件形成了一个划分，与每个1073个个人选择相关的基本事件形成了基本划分。

由少量事件组成的分区，其中一些事件可能对应于许多符号，被称为粗粒度分区，而具有许多事件的分区则是细粒度分区。
基本分区与任何其他分区一样细粒度。由普遍事件和空事件组成的分区与任何其他分区一样粗粒度。

虽然我们描述事件时总是假设存在一个基本分区，但在实践中可以不使用该分区。

5.2 已知结果

一旦你知道一个结果，很容易表示它。你只需指定选择了哪个符号。 如果其他事件是根据符号来定义的，那么你就知道哪些事件已经发生了。 然而，在结果未知之前，你无法以这种方式表达你的知识状态。当然要记住，你的知识可能与其他人的知识不同，即知识是主观的，或者有些人可能说是“观察者依赖的”。

这里有一种更复杂的表示已知结果的方式，它很有用，因为它可以推广到结果尚未知晓的情况。 让 i成为一个在分区上运行的索引。 因为符号的数量是有限的，我们可以将这个索引从0到 $n-1$运行，其中 n是分区中的事件数。 然后对于分区中的任何特定事件 A_i，定义 $p(A_i)$为1（如果选择了相应的结果）或0（如果未选择）。 在任何分区中，都会有一个 i使得 $p(A_i)=1$，而所有其他 $p(A_i)$都为0。 这个相同的符号可以适用于不在分区中的事件——如果事件 A作为选择的结果发生，则 $p(A)=1$，否则$p(A)=0$。

从这个定义可以得出结论， $p(\text{普遍事件 }) = 1$， $p(\text{空事件 }) = 0$。

5.3 未知结果

如果符号尚未被选择，或者您尚不知道结果，则每个 $p(A)$ 可以被赋予0到1之间的数字，较高的数字表示对该事件发生的更大信念，较低的数字表示对该事件可能不会发生的信念。 如果您确定某个事件 A是不可能的，则 $p(A)=0$。 一旦结果被了解，每个 $p(A)$ 可以调整为0或1。 再次注意， $p(A)$ 取决于您的知识状态，因此是主观的。

这些数字应如何分配以最好地表达我们的知识将在后面的章节中进行讨论。 然而，我们要求它们遵守概率论的基本公理，并将它们称为概率（适用于一个分区的概率集合称为概率分布）。 根据定义，对于任何事件 A

$$0 \leq p(A) \leq 1 \qquad (5.1)$$在我们的例子中，我们可以用概率 $p(W)$来描述我们对一个尚未选择（或尚未知道）的大一新生的性别的理解。 同样， $p(CA)$可能表示被选择的人来自加利福尼亚的概率。

为了与概率论保持一致，如果某个事件 A只发生在某些互斥的其他事件 A_i的发生时（例如因为它们来自一个分区），那么$p(A)$是这些事件的各个 $p(A_i)$的和：

$$p(A) = \sum_i p(A_i) \tag{5.2}$$

其中 i是所讨论事件的索引。 这意味着对于任何分区，由于 $p(\text{全集事件 }) = 1$,

$$1 = \sum_i p(A_i) \tag{5.3}$$

其中这里的求和是对分区中所有事件进行的。

5.4 联合事件和条件概率

你可能对所选择的符号具有两个不同属性的概率感兴趣。 例如，

被选择的新生是来自德克萨斯州的女性的概率是多少？ 如果我们知道选择是女性的概率，$p(W)$，和选择来自德克萨斯州的概率，$p(\mathrm{TX})$，我们能找到这个概率$p(W, \mathrm{TX})$吗？

一般来说不行。可能有47%的新生是女性，可能有5%的新生来自德克萨斯州，但仅凭这些事实不能保证有任何来自德克萨斯州的女性新生，更不用说有多少了。

然而，如果已知或假设这两个事件是独立的（一个事件发生与否不依赖于另一个事件），那么可以找到联合事件（两个事件同时发生）的概率。

它是两个事件概率的乘积。 在我们的例子中，如果在来自德克萨斯州的新生中女性的比例与所有新生中女性的比例相同，那么 我们可以找到这个概率。

$$p(W, T X) = p(W)p(T X) \tag{5.4}$$

由于两个事件独立是不寻常的，需要一个更一般的联合事件公式。
这个公式利用了“条件概率”，即已知另一个事件发生的情况下，某一事件发生的概率。 在我们的例子中，选择为女性的条件概率，已知新生选择来自德克萨斯州，用符号表示为 $p(W \mid T X)$，其中竖线表示“给定”，分隔了两个事件——右边是条件事件，左边是被条件限制的事件。 如果两个事件是独立的，那么被条件限制的事件的概率与其正常的或“无条件”的概率相同。

就条件概率而言，联合事件的概率是其中一个事件发生的概率乘以另一个事件在第一个事件发生的情况下发生的概率：

$$\begin{aligned} p(A, B) &= p(B)p(A \mid B) \\ &= p(A)p(B \mid A) \end{aligned} \tag{5.5}$$

请注意，任何一个事件都可以作为条件事件，因此有两个公式可以计算这个联合概率。 使用这些公式，您可以从一个条件概率计算出另一个条件概率，即使您不关心联合概率。

这个公式被称为贝叶斯定理，以18世纪英国数学家托马斯·贝叶斯命名。我们经常使用贝叶斯定理。 这个定理具有显著的普遍性。 如果两个事件在物理上或逻辑上相关，则该定理成立；如果它们不相关，则该定理也成立。如果一个事件导致另一个事件发生，该定理成立；如果不是这种情况，该定理也成立。 如果结果已知，则该定理成立；如果结果未知，则该定理也成立。

因此，选择的学生是来自德克萨斯州的女性的概率 $p(W, T X)$是选择来自德克萨斯州的学生的概率 $p(T X)$乘以选择女性的概率 $p(W \mid T X)$。它也是选择女性的概率 $P(W)$乘以选择来自德克萨斯州的人的概率 $p(T X \mid W)$。

$$\begin{aligned} p(W, T X) &= p(T X)p(W \mid T X) \\ &= p(W)p(T X \mid W) \end{aligned} \tag{5.6}$$

作为另一个例子，考虑上面的学生表，并假设从整个学生群体中“随机”选择一个（意味着每个学生的概率相等）。 选择是男性研究生的概率 $p(M, G)$是什么？ 这是一个联合概率，如果我们能够找到必要的条件概率，我们可以使用贝叶斯定理。

在这种情况下，基本分区是10,206个基本事件，其中选择了一个特定的学生。 所有这些概率的总和为1，并且根据假设，所有概率都相等，因此每个概率为$1/10{,}220$，约为0.01%。

选择是研究生学生 $p(G)$的概率是与研究生学生相关的所有048个基本事件的概率之和，因此 $p(G) = 6{,}048/10{,}220$。

假设选择是研究生学生，那么选择是男性的条件概率是多少？ 现在我们来看看研究生学生的集合和其中的一个选择。 新的基本分区是6,048个可能的研究生学生选择，从上表中我们可以看出其中4,226个是男性。 可以通过以下方式找到这个新的（条件）选择的概率。 最初的选择是“随机的”，因此所有学生被选择的可能性相等。 特别地，所有研究生学生被选择的可能性相等，因此对于所有6,048个选择，新的概率将是相同的。

由于它们的总和为1，每个概率为$1/6{,}048$。 选择一个男人的事件与这些新的基本事件中的4,226个相关联，因此条件概率 $p(M \mid G) = 4{,}226/6,\ 048$。 因此根据贝叶斯定理：

$$\begin{aligned} p(M, G) &= p(G)p(M \mid G) \\ &= \frac{6{,}048}{10{,}220} \times \frac{4{,}226}{6{,}048} \\ &= \frac{4{,}226}{10{,}220} \end{aligned} \tag{5.7}$$

这个问题可以从另一个角度来解决：选择一个男人的概率是 $p(M) = 6{,}541/10{,}220$，而选择是研究生的概率是 $p(G \mid M) = 4{,}226/6{,}541$，所以（当然答案是一样的）

$$\begin{aligned} p(M, G) &= p(M)p(G \mid M) \\ &= \frac{6{,}541}{10{,}220} \times \frac{4{,}226}{6{,}541} \\ &= \frac{4{,}226}{10{,}220} \end{aligned} \tag{5.8}$$

5.5 平均值

假设我们对我们例子中选择的新生的身高感兴趣。如果我们知道谁被选择了，我们可以很容易地发现他或她的身高（假设每个新生的身高在某个数据库中可用）。 但是如果我们还没有了解到被选择的人的身份呢？ 我们还能估计身高吗？

起初，我们可能会说我们对身高一无所知，因为我们不知道谁被选择了。 但是这显然是不正确的，因为经验表明，绝大多数新生的身高在60英寸（5英尺）到78英寸（6英尺6英寸）之间，所以我们可能会安全地估计身高为70英寸。 至少我们不会将身高估计为

$$82 \text{ 英寸}。$$

通过概率，我们可以更精确地计算出一个不知道选择的身高估计。 而且，我们用于这个计算的公式在我们了解实际选择并相应调整概率后仍然有效。

假设我们有一个分区，其中的事件 A_i每个事件都有一个属性值，比如身高，比如 h_i。 然后，这个属性的平均值（也称为期望值） H_{av}可以通过与每个事件相关的概率来计算

$$H_{av} = \sum_i p(A_i)h_i \tag{5.9}$$

其中求和是针对分区的。

这种类型的公式可以用来计算许多属性的平均值，比如SAT成绩、体重、年龄或净财富。 对于非数值属性，比如性别、眼睛颜色、个性或预期的学术专业，这种公式是不适用的。

请注意，这个平均值的定义适用于每个分区中的每个事件都有一个属性值的情况，比如身高。 对于新生的身高来说，这只适用于基本分区。
我们希望能够以类似的方式计算其他分区的平均值，例如男性和女性的分区。 问题在于，并不是所有的男性都有相同的身高，所以在方程式5.9中不清楚应该使用什么值作为 h_i。

解决方案是根据一个更细粒度的分区（如基本分区）来定义男性的平均身高。 贝叶斯定理在这方面很有用。 请注意，已知选择的是一个男性的情况下，选择到新生 i的概率是

$$p(A_i \mid M) = \frac{p(A_i)p(M \mid A_i)}{p(M)} \tag{5.10}$$

其中 $p(M \mid A_i)$ 特别简单 - 它要么是1，要么是0，取决于新生 i是男性还是女性。 然后男性新生的平均身高是

$$H_{av}(M) = \sum_i p(A_i \mid M)h_i \tag{5.11}$$

同样地，对于女性也是如此，

$$\text{女性新生的平均身高是} \sum_i p(A_i \mid W)h_i \tag{5.12}$$

然后所有新生的平均身高由一个与方程5.9完全相同的公式给出：

$$H_{av} = p(M)H_{av}(M) + p(W)H_{av}(W) \tag{5.13}$$

如果所讨论的分区中的所有 $p(A_i)$都相等（例如，如果随机选择了一个新生），则这些平均值公式是有效的。 但是它们更一般化——它们也适用于任何概率分布 $p(A_i)$。唯一需要注意的是其中一个事件的概率等于零的情况，例如，如果你想要计算内华达州新生的平均身高，而恰好没有这样的新生。

5.6 信息

我们希望定量地表达我们对符号选择的信息或缺乏信息。 在我们了解结果之前，我们对所选择的符号或其各种属性以及可能发生的事件存在不确定性。 然而，在选择被做出之前或至少在我们了解结果之前，我们存在一些不确定性。有多少不确定性？

在我们了解结果之后，我们现在拥有的信息可以通过指定选择的符号来告诉另一个人。 如果有两个可能的符号（例如硬币正反面），那么可以用一个比特来表示。 如果有四个可能的事件（例如从一副牌中抽取的花色），结果可以用两个比特表示。 更一般地，如果有 n个可能的结果，则需要$\log_2 n$比特。

这里的概念是，听到结果后我们学到的信息量是可以用来告诉我们的最小比特数，即指定符号的比特数。 这种方法有一些优点，但也有两个缺点。

首先，通过比特序列实际指定一个符号需要整数比特数。 如果符号的数量不是2的整数次幂怎么办？ 对于单个选择，可能无法做太多事情，但如果源进行重复选择并且所有选择都需要指定，可以将它们分组在一起以恢复小数比特。 例如，如果有五个可能的符号，则一个符号需要三个比特，但两个符号的25种可能组合可以用五个比特（每个符号2.5比特）进行通信，三个符号的125种组合可以用七个比特（每个符号2.33比特）进行通信。 这与$\log_2(5)$相差不大，$\log 2(5)$为2.32比特。

其次，不同的事件可能具有不同的被选择的可能性。 我们已经看到如何用概率来建模我们的知识状态。 如果我们已经知道结果（一个 $p(A_i)$ 等于1，其他都等于0），那么不会获得进一步的信息，因为之前没有不确定性。 我们对信息的定义应该包括这种情况。

考虑一个由32名学生组成的班级，其中两名是女生，30名是男生。如果选择一个学生，并且我们的目标是知道是哪一个，我们最初的不确定性是五个比特，因为这是需要指定结果的位数。 如果随机选择一个学生，每个学生被选择的概率是$1/32$。 选择学生也会导致一个性别事件，要么是“选择女生”概率为 $p(W) = 2/32$，要么是“选择男生”概率为 $p(M) = 30/32$。

如果我们被告知选择是一个女人，但没有告诉是哪一个，我们会获得多少信息？ 我们的不确定性从五个比特减少到一个比特（足以指定是两个女人中的哪一个）。 因此，我们获得的信息量是四个比特。 如果我们被告知选择是一个男人，但没有告诉是哪一个，我们会获得多少信息？ 我们的不确定性从五个比特减少到$\log_2(30)$或4.91比特。 因此，我们学到了0.09比特的信息。

这里的重点是，如果我们有一个事件概率不同的分区，我们从不同的结果中学到不同的数量。 如果结果可能性较高，我们学到的信息较少，而如果结果可能性较低，我们学到的信息较多。
我们在一个情况中说明了这个原则，每个结果都没有解决从基本分区中选择一个事件，但即使我们不关心基本分区，这个原则也适用。 从结果 i学到的信息是 $\log_2(1/p(A_i))$。 从这个公式可以看出，如果对于某些$_i$ ， $p(A\ i) = 1$，则从该结果学到的信息为0，因为$\log_2(1) = 0$。 这与我们的预期一致。

如果我们想在学习结果之前量化我们的不确定性，我们不能使用任何特定结果获得的信息，因为我们不知道该使用哪个。相反，我们必须对所有可能的结果进行平均，即对具有非零概率的分区中的所有事件进行平均。 每个事件的平均信息是通过将每个事件 A_i的信息乘以 $p(A_i)$并对分区求和来找到的：

$$I = \sum_i p(A_i)\ \log_2\left(\frac{1}{p(A_i)}\right(\tag{5.14}$$

这个数量对于表征源的信息非常重要，被称为源的熵。 如果概率都相等，这个公式就适用；如果概率不相等，它也适用；在结果已知且概率调整为其中一个为1，其他都为0的情况下，它也适用；无论报告的事件是否来自基本分区，它都适用。

在这个和其他关于信息的公式中，必须小心处理概率为零的事件。
这些情况可以被视为具有非常小但非零的概率。 在这种情况下，对数虽然在参数趋近无穷大时趋近于无穷大，但增长非常缓慢。 该因子乘以概率的乘积趋近于零，因此这些项可以直接置零，即使公式可能暗示一个不确定的结果，或者计算过程可能出现“除以零”的错误。

5.7 信息的属性

方便起见，可以将物理量视为具有维度。 例如，速度的维度是长度除以时间，因此速度用米每秒表示。 以类似的方式，将信息视为具有维度的物理量是方便的。 也许这有点不太自然，因为概率本质上是无量纲的。 然而，请注意公式使用以2为底的对数。 选择底数相当于信息的比例因子。 原则上可以使用任何底数k，并通过以下等式与我们的定义相关联

$$\log_k(x) = \frac{\log_2(x)}{\log_2(k)} \tag{5.15}$$

使用以2为底的对数来表示信息的单位是比特。 稍后，我们将发现自然对数很有用。

如果分区中有两个事件，其概率分别为 p和 $(1-p)$，则每个符号的信息量为

$$I = p\log_2\left(\frac{1}{p}\right(+ (1-p)\log_2\left(\frac{1}{1-p}\right(\tag{5.16}$$

在图5.3中，该函数是 p的函数。当 p= 0. 5时，它最大（1比特）。 因此，当两个可能事件的概率相等时，信息量达到最大值。此外，在 p= 0.4和 p= 0.6之间的整个概率范围内，信息量接近1比特。当 $p = 0$和$p = 1$时，信息量为0。 这是合理的，因为对于这样的 p值，结果是确定的，因此学习它不会获得任何信息。

对于具有两个以上可能事件的分区，每个符号的信息量可能更高。 如果有n个可能事件，则每个符号的信息量介于0和$\log_2(n)$比特之间，当所有概率相等时，达到最大值。

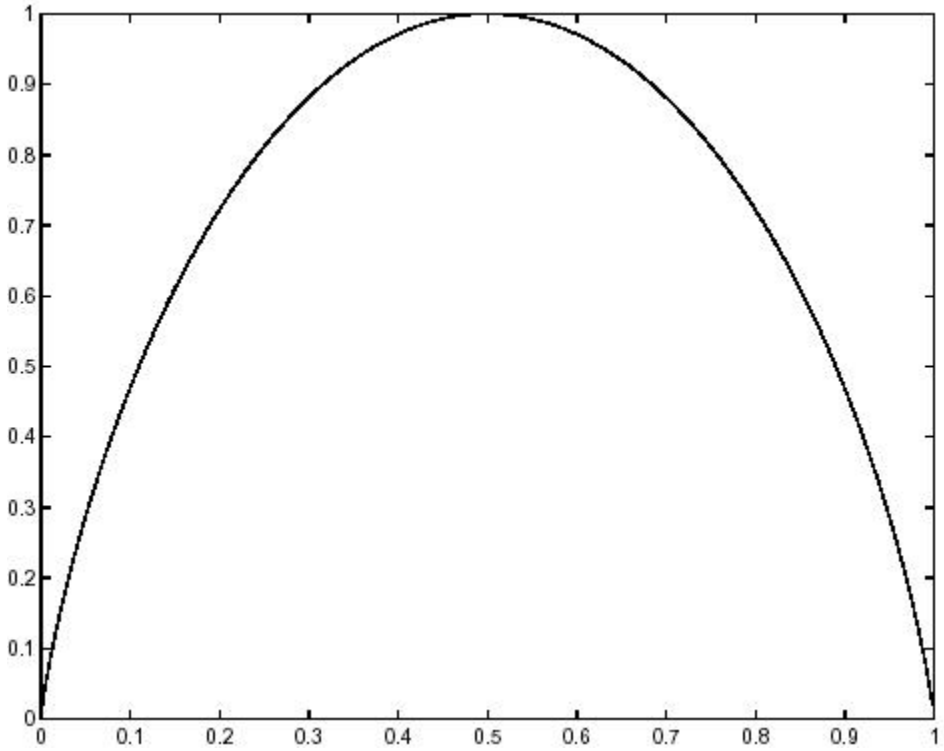


图5.3：作为两个概率之一的函数的源的熵

5.8 高效的源编码

如果一个源有n个可能的符号，那么对它进行固定长度编码将需要每个符号$\log_2(n)$（或更高的整数）比特。 每个符号的平均信息量 I不能超过这个值，但如果符号具有不同的概率，可能会更小。 是否可能通过使用对于更可能的符号使用较少的比特和对于较不可能的符号使用更多的比特的可变长度编码，平均使用较少的比特来对来自这样的源的符号流进行编码？

当然可以。 莫尔斯电码是一个很好的例子，它是一种可变长度编码。 有一种通用的过程可以构建这种非常高效的编码（事实上，即使 I远低于$\log_2(n)$，它们每个符号的平均比特数也少于 $I+1$比特）。这些编码被称为霍夫曼编码，以麻省理工学院的毕业生大卫·霍夫曼（1925-1999）命名，并广泛用于通信系统。请参见第5.10节。

5.9 详细信息：人寿保险

统计学和概率论在日常生活中的一个例子是它们在人寿保险中的应用。 我们在这里只考虑一年期保险（保险公司在营销更复杂的保单时非常有创意，结合了保险、储蓄、投资、退休收入和税收最小化的方面）。当您购买人寿保险时，您需要支付一定金额的保费，如果您在一年内去世，您的受益人将获得更大的金额。 人寿保险可以从多个角度来看待。

从赌徒的角度来看，您打赌自己会死亡，而保险公司则打赌您会活下来。 每个人都可以估计自己死亡的概率，由于概率是主观的，它们可能有足够大的差异，使得这样的赌注对双方都有利（例如，假设您了解到一种威胁性的医疗情况，并且没有向保险公司披露）。保险公司使用死亡率表，如表5.2（也显示在图5.4中）来确定其费率。

（有趣的是，保险公司也销售年金，从赌徒的角度来看，这些年金是反向的赌注——公司赌你会很快死去，而你赌你会活很久。）

另一种思考人寿保险的方式是将其视为一种金融投资。 由于保险公司平均支付的金额少于他们收取的金额（否则他们会破产），投资者通常会选择其他方式来投资他们的钱，例如将其存入银行。

当然，大多数购买人寿保险的人并不将其视为赌注或投资，而是将其视为一种安全网。他们知道如果他们死了，他们的收入将停止，他们希望为他们的受赡养者（通常是子女和配偶）提供部分补偿。 保费很低，因为在这种安全网重要的年份里，死亡的概率很低，但是在不太可能发生死亡的情况下，对受益人来说可能非常重要。 对于非常富有的人（他们可以承受收入损失）、没有受赡养者的单身人士或子女已经长大的老年人来说，这种安全网可能并不那么重要。

图5.4和表5.2显示了1988年出生的美国居民（来自伯克利死亡数据库[2]）在一年内死亡的概率与年龄的关系。

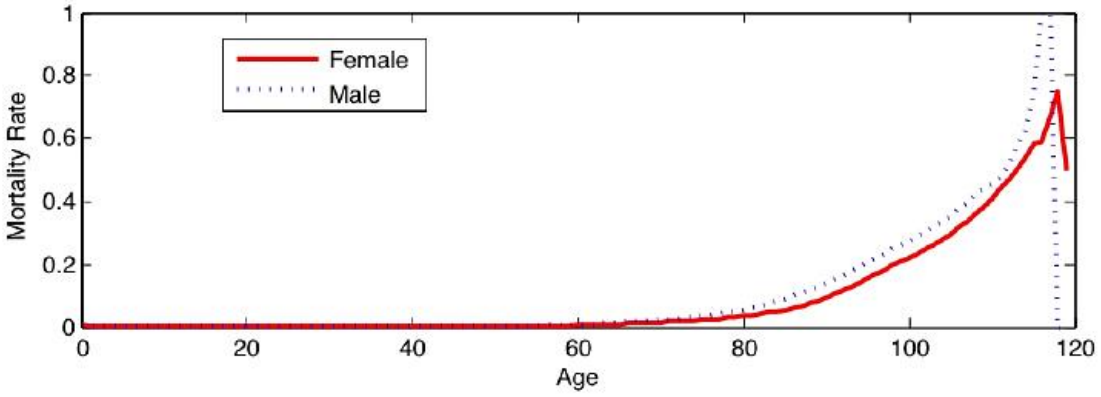


图5.4：1988年出生的美国居民一年内死亡的概率。

[2]伯克利死亡数据库可以在线访问：http://www.demog.berkeley.edu/ bmd/states.html

年龄	女性	男性	年龄	女性	男性	年龄	女性	男性
0	0.008969	0.011126	40	0.000945	0.002205	80	0.035107	0.055995
1	0.000727	0.000809	41	0.001007	0.002305	81	0.038323	0.061479
2	0.000384	0.000526	42	0.00107	0.002395	82	0.041973	0.067728
3	0.000323	0.000415	43	0.001144	0.002465	83	0.046087	0.074872
4	0.000222	0.000304	44	0.001238	0.002524	84	0.050745	0.082817
5	0.000212	0.000274	45	0.001343	0.002605	85	0.056048	0.091428
6	0.000182	0.000253	46	0.001469	0.002709	86	0.062068	0.100533
7	0.000162	0.000233	47	0.001616	0.002856	87	0.06888	0.110117
8	0.000172	0.000213	48	0.001785	0.003047	88	0.076551	0.120177
9	0.000152	0.000162	49	0.001975	0.003295	89	0.085096	0.130677
10	0.000142	0.000132	50	0.002198	0.003566	90	0.094583	0.141746
11	0.000142	0.000132	51	0.002454	0.003895	91	0.105042	0.153466
12	0.000162	0.000203	52	0.002743	0.004239	92	0.116464	0.165847
13	0.000202	0.000355	53	0.003055	0.00463	93	0.128961	0.179017
14	0.000263	0.000559	54	0.003402	0.00505	94	0.142521	0.193042
15	0.000324	0.000793	55	0.003795	0.005553	95	0.156269	0.207063
16	0.000395	0.001007	56	0.004245	0.006132	96	0.169964	0.221088
17	0.000426	0.001161	57	0.004701	0.006733	97	0.183378	0.234885
18	0.000436	0.001254	58	0.005153	0.007357	98	0.196114	0.248308
19	0.000426	0.001276	59	0.005644	0.008028	99	0.208034	0.261145
20	0.000406	0.001288	60	0.006133	0.008728	100	0.220629	0.274626
21	0.000386	0.00131	61	0.006706	0.009549	101	0.234167	0.289075
22	0.000386	0.001312	62	0.007479	0.010629	102	0.248567	0.304011
23	0.000396	0.001293	63	0.008491	0.012065	103	0.263996	0.319538
24	0.000417	0.001274	64	0.009686	0.013769	104	0.280461	0.337802
25	0.000447	0.001245	65	0.011028	0.015702	105	0.298313	0.354839
26	0.000468	0.001226	66	0.012368	0.017649	106	0.317585	0.375342
27	0.000488	0.001237	67	0.013559	0.019403	107	0.337284	0.395161
28	0.000519	0.001301	68	0.014525	0.020813	108	0.359638	0.420732
29	0.00055	0.001406	69	0.015363	0.022053	109	0.383459	0.439252
30	0.000581	0.001532	70	0.016237	0.023393	110	0.408964	0.455882
31	0.000612	0.001649	71	0.017299	0.025054	111	0.437768	0.47619
32	0.000643	0.001735	72	0.018526	0.027029	112	0.466216	0.52
33	0.000674	0.00179	73	0.019972	0.029387	113	0.494505	0.571429
34	0.000705	0.001824	74	0.02163	0.032149	114	0.537037	0.625
35	0.000747	0.001859	75	0.023551	0.035267	115	0.580645	0.75
36	0.000788	0.001904	76	0.02564	0.038735	116	0.588235	1
37	0.00083	0.001961	77	0.027809	0.042502	117	0.666667	1
38	0.000861	0.002028	78	0.030011	0.046592	118	0.75	0
39	0.000903	0.002105	79	0.032378	0.051093	119	0.5	0

表5.2：1988年出生的美国居民的死亡表

5.10 详细信息：高效的源代码

有时，在图5.1所示的通信系统的源编码和压缩可以同时进行（将源编码和信道编码结合是否有实际好处是一个悬而未决的问题）。 对于具有有限数量的符号，但在输入流中出现的概率不相等的源，存在一种优雅而简单的源编码技术，可以实现最小冗余。

有限源的例子

考虑一个生成符号的源，这些符号是MIT的等级字母，可能的值为A、B、C、D和F。你被要求设计一个系统，可以在每秒产生一个符号的速率下，通过一个只能传输两个布尔数字（每秒一个）的通信通道传输这些等级的流。

首先，不要对等级分布做任何假设。 要单独传输每个符号，你必须将每个符号编码为一系列位（布尔数字）。 使用7位ASCII码是浪费的；我们只有五个符号，而ASCII可以处理128个符号。 由于只有五个可能的值，每个符号可以用三位编码。 但是通道每秒只能处理两位。

然而，三位是多余的。 熵，假设没有关于概率的信息，最多为$\log_2(5) = 2.32$位。 这也是 $\sum_i p(A_i)\log_2(1/p(A_i))$，其中有五个这样的$p_i$，每个都等于$1/5$。 为什么在第一个案例中我们需要三位？ 因为我们没有办法传输部分位。为了做得更好，我们可以使用“块编码”。我们将符号分组成块，比如三个一组。

每个块中的信息是每个符号的三倍，即6.97比特。 因此，可以使用7个布尔比特来传输一个块（在7个比特中有125个不同的三个等级序列和128个可能的模式）。 当然，我们还需要一种表示结束的方式，以及一种表示最后一个传输的单词只有一个有效等级（或两个等级）而不是三个等级的方式。

但这对于信道来说仍然是太多的比特每秒。 所以让我们来看看符号的概率分布。 在一个典型的“以B为中心”的MIT课程中，学生优秀，成绩分布可能如表5.3所示。假设这是一个概率分布，每个符号的信息量是多少，平均每个符号的信息量是多少？ 这个计算结果如表5.4所示。每个符号的信息量为1.840比特。 由于这小于两个比特，也许可以对符号进行编码以使用这个信道。

A	B	C	D	F
25%	50%	12.5%	10%	2.5%

表5.3：典型MIT课程的成绩分布

符号概率信息			对平均值的贡献
	p	$\log \frac{1}{p}$	$\sum p \log \frac{1}{p}$
A	0.25	2位	0.5位
B	0.50	1位	0.5位
C	0.125	3位	0.375位
D	0.10	3.32位	0.332位
F	0.025	5.32位	0.133位
总计	1.00		1.840位

表5.4：MIT平均分布中的信息分布

[3]个布尔数字，或二进制数字，通常称为“位”。单词“位”也指信息单位。 当一个布尔数字携带一位信息时，可能不会有混淆。 但是低效的编码或冗余的编码可能导致布尔数字序列比最小值更长，因此每个位携带的信息少于一位。同样的混淆也适用于其他度量单位，例如米、秒等。

哈夫曼编码

大卫·A·哈夫曼（1925年8月9日 - 1999年10月6日）是麻省理工学院的研究生。为了解决一项来自罗伯特·M·法诺教授的课程作业，他设计了一种用于编码具有不同概率的符号的方法，以最小化冗余并且不需要特殊的符号框架，从而实现最紧凑的编码。 他在1962年9月的IRE会议论文中描述了这种方法。 他的算法非常简单。 目标是设计一个“码书”（每个符号对应一串比特），使得平均码长最小化。 假设不常见的符号会得到较长的编码，而常见的符号会得到较短的编码，就像摩尔斯电码一样。 算法如下（你可以参考表5.5）：

1.初始化：让每个符号的部分编码最初为空的比特串。 为每个符号定义一个相应的“符号集”，只包含该符号，并且概率等于该符号的概率。

2.循环测试：如果只有一个符号集（其概率必须为1），则完成。 码书由与该符号集中的每个符号相关联的编码组成。

3.循环操作：如果有两个或更多个符号集，请选择概率最低的两个（如果有多个选择，则选择任意两个）。 用0在一个符号集中添加代码，在另一个符号集中添加1。 定义一个新的符号集，该符号集是刚刚处理的两个符号集的并集，概率等于两个概率的和。 用新的符号集替换这两个符号集。符号集的数量因此减少了一个。重复此循环，包括循环测试，直到只剩下一个符号集。

请注意，此过程通常会产生可变长度的编码。 如果有n个不同的符号，则其中至少有两个符号的编码长度最长。

对于我们的例子，我们从五个符号集开始，每一步减少一个符号集，直到只剩下一个。步骤显示在表5.5中，最终的编码表显示在表5.6中。

开始：(A='-' p=0.25) (B='-' p=0.5) (C='-' p=0.125) (D='-' p=0.1) (F='-' p=0.025)
下一步：(A='-' p=0.25) (B='-' p=0.5) (C='-' p=0.125) (D='1' F='0' p=0.125)
下一步：(A='-' p=0.25) (B='-' p=0.5) (C='1' D='01' F='00' p=0.25)
下一步：(B='-' p=0.5) (A='1' C='01' D='001' F='000' p=0.5)
最终：(B='1' A='01' C='001' D='0001' F='0000' p=1.0)

表5.5：麻省理工学院课程成绩分布的哈夫曼编码，其中“-”表示空位字符串

	符号代码
A	0 1
B	1
C	0 0 1
D	0 0 0 1
F	0 0 0 0

表5.6：典型麻省理工学院成绩分布的哈夫曼编码表

这个代码真的很紧凑吗？ 最频繁出现的符号（B）被赋予最短的代码，而最不频繁的符号（D和F）被赋予最长的代码，因此根据假设的概率分布，输入流所需的平均比特数确实很短，如表5.7所示。

将这个表与之前的信息内容表5.4进行比较。请注意，每个符号的平均编码长度为1.875比特，大于每个符号的信息量，即1.840比特。 这是因为符号D和F无法用分数位进行编码。 如果考虑一个包含多个符号的块

符号	代码	概率	长度	代码贡献	平均
A	01	0.25	2	0.5	
B	1	0.50	1	0.5	
C	001	0.125	3	0.375	
D	0001	0.1	3.32	0.4	
F	0000	0.025	5.32	0.1	
总计		1.00		1.875位	

表5.7：典型MIT成绩分布的Huffman编码

综合起来，Huffman编码的平均长度可能更接近于每个符号的实际信息量，但不能低于它。

该信道每秒可以处理两个比特。 通过使用这个编码，你可以平均每秒传输略多于一个符号。 你可以实现你的设计目标。

关于Huffman编码，至少有六个实际考虑的事情：

- 可能会出现一连串的D或F等级。 编码器需要将这些比特存储起来，直到信道能够跟上。需要多大的缓冲区来存储？ 如果缓冲区溢出会发生什么？
- 由于缓冲区备份，输出可能会延迟。 输入和相关输出之间的时间称为“延迟”。对于交互式系统，您希望保持延迟低。每秒处理的比特数，即“吞吐量”，在其他非交互式应用中更为重要。
- 由于突发引起的延迟，输出不会以规律的间隔发生。 在某些实时应用程序（如音频）中，这可能很重要。
- 如果我们对假定的概率分布错误怎么办？ 一个大型课程可能会给出较少的A和B等级，而给出更多的C和D等级。我们的编码将是低效的，并且可能会发生缓冲区溢出。
- 解码器需要知道如何将比特流分解为单个代码。在这种情况下的规则是，在第一个'1'或'0000'之后进行分解，以先出现的为准。然而，对于上述算法中第3步中0和1的选择，存在许多可能的霍夫曼编码。 大多数编码没有这么简单的规则。 确定符号间断应该放在哪里可能很困难（尽管总是可能的）。 确定符号间断应该放在哪里可能很困难（尽管总是可能的）。
- 编码器和解码器都必须提前给出代码本身。 这可以通过一次性在信道上传输代码本来完成。

另一个例子

麻省理工学院的新生在校园的第一个学期采用“及格/无记录”制度。 A、B和C的成绩在成绩单上报告为P（及格），而D和F则不报告（为了我们的目的，我们将其指定为无记录，N）。让我们设计一个系统，以最快的平均速率将这些P和N符号发送到打印机。 不考虑概率，每个符号需要1比特。 但是概率（假设表5.3中的典型麻省理工学院成绩分布）为 $p(P) = p(A) + p(B) + p(C) = 0.875$，和 $p(N) = p(D) + p(F) = 0.125$。 因此，每个符号的信息量不是1比特，而只有$0.875 \times log_2(1/0.875) + 0.125 \times log_2(1/0.125) = 0.544$比特。 对单个符号进行Huffman编码没有帮助。 我们需要将一组位一起处理。 例如，作为一个块的十一个成绩将具有5.98比特的信息，并且原则上可以编码为只需要6比特发送到打印机。

第六章

通信

我们一直在考虑一个信息处理系统的模型，其中输入的符号被编码成比特，然后通过“通道”发送到接收器，并解码回符号，如图6.1所示。

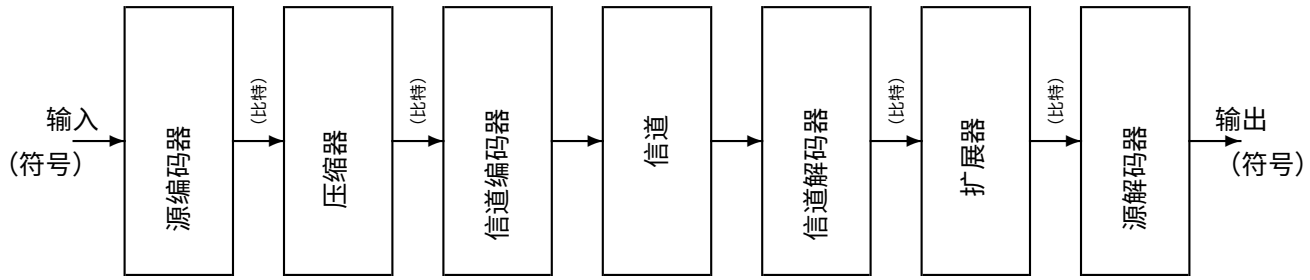


图6.1：通信系统

在本章中，我们关注的是识别符号的信息传输速度。 当然，并不传输符号本身，而只传输识别它们所需的信息。 这就是在输出端重新创建符号流所必需的。

我们将对源和通道进行更详细的建模，然后给出与源特性和通道容量相关的三个定理。

6.1 源模型

假设源以每秒产生 R个符号的速率。 每个符号是从一组可能的符号中选择的，索引 i在可能的符号范围内变化。 选择符号 i的事件将被表示为 A_i。

让我们假设每个事件 A_i（即选择符号 i）由一个不同的编码词C_i表示，其长度为 L_i。 对于固定长度的编码，如ASCII码，所有的 L_i都是相同的，而对于可变长度的编码，如霍夫曼编码，它们通常是不同的。 由于编码词是位的模式，每个长度可用的数量是有限的。

每个长度只有有限数量的可用编码。 例如，只有四个不同的两位编码词
可能，即00、01、10和11。

这种编码词的一个重要特性是，没有一个编码词可以与另一个更长的编码词的第一部分相同，否则相同的位模式可能会导致两个或多个不同的消息，从而产生歧义。 遵守这个特性的编码被称为前缀条件编码，有时也被称为瞬时编码。

6.1.1 Kraft不等式

由于短长度的不同编码数量有限，不是所有编码都可以是短的。 有些必须更长，但前缀条件进一步限制了可用的短编码。 $L.$ G. Kraft在他1949年的硕士论文中给出了对编码长度分布的重要限制。

这被称为Kraft不等式：

$$\sum_i \frac{1}{2^{L_i}} \leq 1 \tag{6.1}$$

任何有效的不同码字集合都遵守这个不等式，反之亦然，对于任何满足该不等式的提议的 L_i，都可以找到一个编码。

例如，假设一个编码由四个不同的两位码字00、01、10和11组成。 然后，每个 $L_i = 2$，并且方程6.1中的每个项都是$1/2^2 = 1/4$。 在这种情况下，方程式得到了相等，有很多不同的方法可以将这四个码字分配给四个不同的符号。

举个例子，假设只有三个符号，并且提出的编码词是00 01和11。

在这种情况下，克拉夫特不等式是一个不等式。 然而，由于总和小于1，可以通过用更短的编码词替换其中一个编码词来使代码更有效。特别地，如果由11表示的符号现在由1表示，则结果仍然是一个前缀条件码，但总和将为$(1/2^2)+(1/2^2)+(1/2) = 1$。

克拉夫特不等式可以很容易地证明。 设 L_{max}为前缀条件码的最长编码词的长度。 这种长度的0和1的模式恰好有$2^{L_{max}}$个。因此

$$\sum_i \frac{1}{2^{L_{max}}} = 1 \tag{6.2}$$

其中这个求和是在这些模式上进行的（这是一个不寻常的方程，因为被求和的数量不依赖于 i）。 这些模式中至少有一个是编码词，但除非这恰好是一个固定长度的编码，否则还有其他更短的编码词。 对于每个长度为 k（$k < L_{max}$）的较短编码词，恰好有$2^{L_{max}-k}$个以该编码词开头的模式，而其中没有一个是有效的编码词（因为该编码是一个前缀条件码）。 在方程6.2的求和中，用一个等于$1/2^k$的单个项替换那些模式对应的项。求和保持不变。 继续这个过程与其他较短的编码词。 当完成时，求和中有与每个编码词对应的项，且求和仍然等于1。 可能还有与不是编码词的模式对应的其他项 - 如果是这样，在方程6.2的求和中将它们排除。结果正好是方程6.1中的求和，且小于或等于1。证明完成。

6.2 源熵

作为源模型的一部分，我们假设每个符号选择与其他符号的选择是独立的，因此概率 $p(A_i)$不依赖于先前选择的符号（当然，这个模型可以以许多方式进行推广）。 下一个选择的符号的身份的不确定性 H是在下一个符号被知道时获得的平均信息：

$$H = \sum_i p(A_i) \log_2 \sum \frac{1}{p(A_i)} \sum \tag{6.3}$$

这个量也被称为源的熵，以比特每个符号的形式进行测量。 信息速率，以比特每秒为单位，是 $H\,R$其中 R是源选择符号的速率，
以每秒符号数为单位进行测量。

6.2.1 Gibbs不等式

在这里，我们介绍了Gibbs不等式，以美国物理学家J. Willard Gibbs (1839–1903)[1]命名，这在后面的证明中对我们很有用。 这个不等式表明熵小于或等于使用相同概率但不同对数函数形成的任何其他平均值。
具体来说，

$$\sum_i p(A_i)\,\log_2\frac{1}{p(A_i)} \leq \sum_i p(A_i)\,\log_2\frac{1}{p'(A_i)} \tag{6.4}$$

其中 $p(A_i)$ 是任何概率分布（我们将用它来表示源事件和其他分布），而
$p'(A_i)$ 是任何其他概率分布，或更一般地说是任何一组数，使得

$$0 \leq p'(A_i) \leq 1 \tag{6.5}$$

和

$$\sum_i p'(A_i) \leq 1. \tag{6.6}$$

对于所有概率分布都成立的是,

$$\sum_i p(A_i) = 1. \tag{6.7}$$

通过注意到自然对数具有以下性质可以证明方程6.4，即它小于或等于任何一点处切线的直线（例如图6.2中的点 $x = 1$）。这个性质有时被称为凹性或凸性。因此

$$\ln x \leq (x-1) \tag{6.8}$$

因此，通过将对数的底从e转换为2，我们有

$$\log_2 x \leq (x-1)\,\log_2 e \tag{6.9}$$

然后

$$\begin{aligned}
\sum_i p(A_i)\,\log_2\frac{1}{p(A_i)} - \sum_i p(A_i)\,\log_2\frac{1}{p'(A_i)} &= \sum_i p(A_i)\,\log_2\frac{p'(A_i)}{p(A_i)} \\
&\leq \log_2 e \sum_i p(A_i)\left[\frac{p'(A_i)}{p(A_i)} - 1\right[\\
&= \log_2 e \left(\sum_i p'(A_i) - \sum_i p(A_i)\right(\\
&= \log_2 e \left(\sum_i p'(A_i) - 1\right(\\
&\leq 0
\end{aligned} \tag{6.10}$$

[1]请参阅 Gibbs的传记，网址为 http://www-groups.dcs.st-andrews.ac.uk/%7Ehistory/Biographies/Gibbs.html

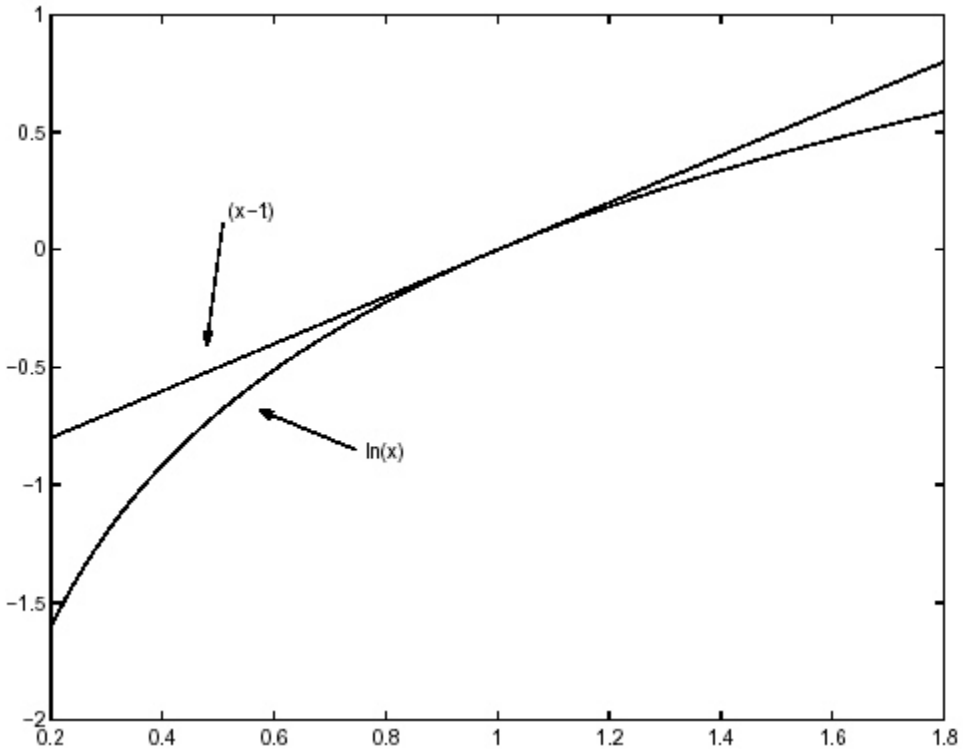


图 6.2：不等式 $\ln x \leq (x-1)$的图形

6.3源编码定理

现在回到源模型，注意码字的平均长度，以比特为单位
符号,

$$L = \sum_i p(A_i)L_i \tag{6.11}$$

为了达到最大速度，需要尽可能低的平均码字长度。 将高概率符号分配给短码字可以帮助使 L small。 Huffman编码是实现这一目的的最优编码。 然而，平均码字长度有一个限制。具体来说，源编码定理表明每个符号的平均信息总是小于或等于平均码字长度：

$$H \leq L \tag{6.12}$$

使用Gibbs和Kraft不等式很容易证明这个不等式。 使用Gibbs不等式和$p'(A_i) = 1/2^{L_i}$（Kraft不等式确保$p'(A_i)$除了是正数外，总和不超过1）。因此

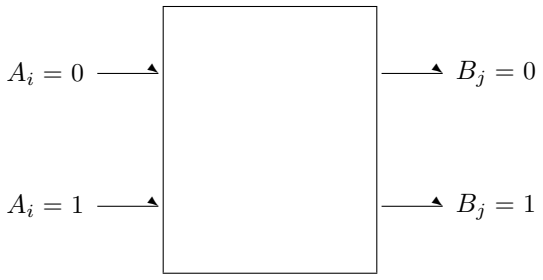


图6.3：二进制信道

$$\begin{aligned} H &= \sum_i p(A_i) \log_2 \frac{\sum 1}{p(A_i)} \sum \\ &\leq \sum_i p(A_i) \log_2 \frac{\sum 1}{p'(A_i)} \sum \\ &= \sum_i p(A_i) \log_2 2^{L_i} \\ &= \sum_i p(A_i) L_i \\ &= L \end{aligned} \tag{6.13}$$

源编码定理也可以用每秒比特数的传输速率来表示，通过将方程6.12乘以每秒符号数 R：

$$HR \leq LR \tag{6.14}$$

6.4声道模型

通信信道接受输入比特并产生输出比特。 我们将输入建模为选择有限数量的输入状态之一（对于最简单的信道，有两个这样的状态），输出则是类似的事件。 概率论的语言在描述信道时非常有用。 如果信道完全根据输入状态改变其输出状态，则称其为无噪声的，在这种情况下，除了输入之外，没有任何影响输出。 假设信道具有一定的最大速率W，其输出可以以该速率跟随输入的变化（就像源具有选择符号的速率 R 一样）。

我们将使用索引 i来遍历输入状态，并使用 j来索引输出状态。 我们将把输入事件称为 A_i，输出事件称为 B_j。 你可以将信道想象成图6.3中的具有输入和输出的东西，但请注意，输入不是正常的信号输入或系统的电气输入，而是互斥的事件，任何时候只有一个事件为真。 对于简单的信道，这样的图表很简单，因为可能的选择很少，但对于更复杂的结构，可能的输入太多，图表变得不实用（尽管它们可能作为概念模型有用）。 例如，一个具有三个输入的逻辑门，每个输入可以是0或1，在这种图表中将有八个输入。 二进制信道具有两个互斥的输入状态，并且是图6.3中所示的信道之一。

对于一个无噪声的信道，每个可能的输入状态都对应一个输出状态，每个新的输入状态（每秒R次）可以用$\log_2 n$比特来指定。 因此，对于二进制信道， $n = 2$，因此新状态可以用一个比特来指定。输入提供的信息的最大速率可以

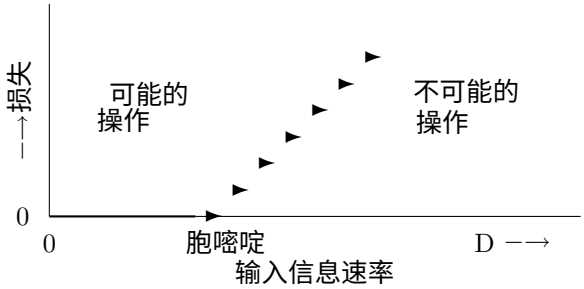


图6.4：信道损耗图。对于输入数据速率 D，无论小于还是大于信道容量 C，信息丢失的最小可能速率是0和 $D-C$中的较大值

影响输出的因素被称为信道容量 $C = W \log_2 n$比特每秒。 对于二进制信道，$C = W$。

如果输入速率 R小于 W（或者等价地，如果输入处提供的信息小于 C），则输出可以跟随输入，并且输出事件可以用于推断该速率下输入符号的身份。 如果试图更快地改变输入，则信道无法跟随（因为 W根据定义是输入变化影响输出的最大速率），并且一些输入信息会丢失。

6.5 无噪声信道定理

如果信道不引入任何错误，输入提供的信息与输出上可用的信息之间的关系非常简单。 假设输入信息速率为每秒比特数，用 D表示（例如， D可以是以比特为单位表示的源符号的熵每个符号的速率 H乘以源的速率 R每秒的符号数）。 如果 $D \leq C$，则输出上可用的信息可以达到 D（即输入上的信息），如果$D > C$，则输出上可用的信息不能超过 C，因此至少丢失了 $D-C$的数量。 这个结果在图6.4中显示。

请注意，这个结果限制了信息在给定信道上传输的速度。 它并不表明如何接近这个限制的结果。 然而，已经知道如何使用霍夫曼编码将符号流高效地表示为比特流。 如果信道是二进制信道，只需使用该比特流来改变输入即可。 对于其他具有多于两个可能输入状态的信道，接近极限的操作涉及使用多个比特来快速控制输入。

实现高通信速度可能需要用长码字表示一些不经常出现的符号（如Huffman码）。 因此，个别比特到达信道输入的速率可能会有所变化，即使平均速率可能是可接受的，如果巧合地出现几个低概率符号相邻，可能会出现较高速率的突发情况。 为了容纳这些突发情况，可能需要提供临时存储缓冲区，并且符号可能不会以均匀的速率出现在系统的输出端。 此外，为了高效地编码符号，可能需要考虑将它们组合在一起，这种情况下，第一个符号在输出端可能要等到输入端呈现了几个符号后才可用。 因此，高速操作可能导致较高的延迟。 不同的通信系统对延迟或突发情况有不同的容忍度；例如，在电话通话中，超过约100毫秒的延迟是令人讨厌的，而在电子邮件中，几分钟的延迟可能是可以容忍的。

一些实际通信系统的需求列表，在第6.9节中显示了速度、吞吐量、延迟等方面的广泛变化。

6.6 噪声信道

如果信道引入噪声，则输出不是输入的唯一函数。 我们将通过说对于每个可能的输入（即互斥状态由 i 索引）可能会有多个可能的输出结果来模拟这种情况。 实际发生的情况是偶然的，我们将通过输出事件 B_j 发生的概率集合来模拟信道，当每个可能的输入事件 A_i发生时。 这些转移概率 c_{ji}当然是概率，但它们是信道的属性，不依赖于输入的概率分布 $p(A_i)$。 像所有的概率一样，它们的值介于0和1之间。

$$0 \leq c_{ji} \leq 1 \tag{6.15}$$

可以看作是一个矩阵，其列数等于输入事件的数量，行数等于输出事件的数量。 因为每个输入事件必须导致恰好一个输出事件，

$$1 = \sum_j c_{ji} \tag{6.16}$$

对于每个 i。 （换句话说，每列 i中的c_{ji}的总和为 1。）如果信道没有噪声，对于每个值的 i，各种 c_{ji}中恰好有一个等于 1，其他全部为 0。

当信道由具有概率 p（A_i）的源驱动时，输出事件在给定输入事件的条件下的条件概率为

$$p(B_j \mid A_i) = c_{ji} \tag{6.17}$$

）每个输出事件的无条件概率 p（B_j）为

$$p(B_j) = \sum_i c_{ji} p(A_i) \tag{6.18}$$

可以使用贝叶斯定理找到反向条件概率 p（$A_i \mid B_j$）：

$$\begin{aligned} p(A_i, B_j) &= p(B_j)p(A_i \mid B_j) \\ &= p(A_i)p(B_j \mid A_i) \\ &= p(A_i)c_{ji} \end{aligned} \tag{6.19}$$

最简单的噪声信道是对称二进制信道，其中存在一个（希望很小的）错误概率 ε，所以

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{bmatrix} \tag{6.20}$$

这个二进制信道被称为对称信道，因为两个输入的错误概率是相同的。
如果 $\varepsilon = 0$，则该信道是无噪声的（如果 $\varepsilon = 1$，它的行为类似于反相器）。
如果在输入到输出的可能转换被显示出来，图6.3对于噪声信道会更有用，就像图6.5一样。

如果观察到输出 B_j处于其（互斥的）状态之一，能否确定导致它的输入 A_i？ 在没有噪声的情况下，是的；一旦知道输出，就没有关于输入的不确定性。 然而，由于噪声存在一些残余的不确定性。 我们将根据转移概率 c_{ji}计算这种不确定性，并将我们根据知道输出而学到的关于输入的信息定义为互信息。 从而我们将定义信道容量 C。

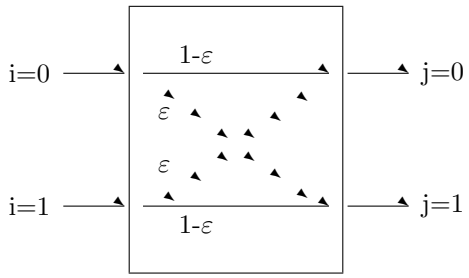


图6.5：对称二进制信道

在我们了解输出之前，关于输入事件的身份，我们的不确定性 U之前是多少？ 这是输入的熵：

$$U_{\text{之前}} = \sum_i p(A_i) \log_2 \frac{1}{p(A_i)} \tag{6.21}$$

在观察到某个特定的输出事件 B_j 之后，关于输入事件的剩余不确定性 U之后(B_j)是多少？ 类似的公式适用于条件反向概率 $p(A_i \mid B_j)$取代条件概率 $p(A_i)$：

$$U_{\text{之后}}(B_j) = \sum_i p(A_i \mid B_j) \log_2 \frac{1}{p(A_i \mid B_j)} \tag{6.22}$$

在这个特定的输出事件的情况下，我们学到的量是 U之前和 U之后(B_j)之间的差异。 互信息 M被定义为所有输出中学到的量的平均值。

$$M = U_{\text{之前}} - \sum_j p(B_j)_{\text{之后}}(B_j) \tag{6.23}$$

证明 $M \geq 0$并不难，即我们对输入的了解在平均情况下不会因为学习输出事件而变得更加不确定。 为了证明这一点，使用了吉布斯不等式，对于每个 j：

$$\begin{aligned} U_{\text{之后}}(B_j) &= \sum_i p(A_i \mid B_j) \log_2 \frac{1}{p(A_i \mid B_j)} \\ &\leq \sum_i p(A_i \mid B_j) \log_2 \frac{1}{p(A_i)} \end{aligned} \tag{6.24}$$

这种使用吉布斯不等式的方法是有效的，因为对于每个 j， $p(A_i \mid B_j)$是一个关于 i的概率分布，而 $p(A_i)$是另一个关于 i的概率分布，与进行平均的概率分布不同。 这个不等式对于每个 j的每个值都成立，因此对所有 j的平均值也成立：

$$\begin{aligned}\sum_j p(B_j)\text{在 }(B_j)\text{ 之后} &\le \sum_j p(B_j)\sum_i p(A_i \mid B_j)\log_2\frac{\sum 1}{p(A_i)}\sum \\ &= \sum_{ji} p(B_j)p(A_i \mid B_j)\log_2\frac{\sum 1}{p(A_i)}\sum \\ &= \sum_{ij} p(B_j \mid A_i)p(A_i)\log_2\frac{\sum 1}{p(A_i)}\sum \\ &= \sum_i p(A_i)\log_2\frac{\sum 1}{p(A_i)}\sum \\ &= \text{之前}\end{aligned} \tag{6.25}$$

现在我们可以根据输入概率分布和信道的特性来找到 M的表达式。 在方程6.23中进行替换和简化得到

$$M = \sum_j \left(\sum_i p(A_i)c_{ji}\left(\log_2\frac{\sum 1}{\sum_i p(A_i)c_{ji}}\sum - \sum_{ij} p(A_i)c_{ji}\log_2\frac{\sum 1}{c_{ji}}\sum\right.\right. \tag{6.26}$$

请注意，方程6.26是针对输入“引起”输出的情况推导出来的。 至少，描述是这样的。 然而，并不一定需要这种因果关系。 互信息这个术语（正确地）表明，将输出视为引起输入，或者完全忽略什么引起了什么，都是同样有效的。 两个替代的 M公式表明 M可以在任一方向上解释：

$$\begin{aligned}M &= \sum_i p(A_i)\log_2\frac{\sum 1}{p(A_i)}\sum - \sum_j p(B_j)\sum_i p(A_i \mid B_j)\log_2\frac{\sum 1}{p(A_i \mid B_j)}\sum \\ &= \sum_j p(B_j)\log_2\frac{\sum 1}{p(B_j)}\sum - \sum_i p(A_i)\sum_j p(B_j \mid A_i)\log_2\frac{\sum 1}{p(B_j \mid A_i)}\sum\end{aligned} \tag{6.27}$$

与其对这些或类似的公式给出一般解释，不如简单地看一下对称二进制信道。 在这种情况下， $p(A_i)$ 和 $p(B_j)$ 都等于0.5，因此方程6.26中 M的第一项为1，第二项可以用 ε表示：

$$M = 1 - \varepsilon\log_2\frac{\sum 1}{\varepsilon}\sum - (1-\varepsilon)\log_2\frac{\sum 1}{(1-\varepsilon)}\sum \tag{6.28}$$

这恰好是1比特减去具有概率 ε和$1-\varepsilon$的二进制源的熵。 这是一个杯状曲线，当 $\varepsilon = 0$时，值为1，当 $\varepsilon = 0.5$时，值为0，当$\varepsilon = 1$时，值再次为1。参见图6.6。这个结果的解释很直接。 当 $\varepsilon = 0$（或 $\varepsilon = 1$）时，只要已知输出，就可以准确确定输入，因此没有信息损失。 因此，互信息与输入信息相同，为1比特。 当 $\varepsilon = 0.5$时，每个输出都是等可能的，无论输入是什么，因此学习输出对输入没有任何信息。 互信息为0。

6.7 噪声信道容量定理

噪声信道的信道容量是通过互信息 M来定义的。 然而，一般来说 M不仅取决于信道（通过传输概率 c_{ji}），还取决于输入。

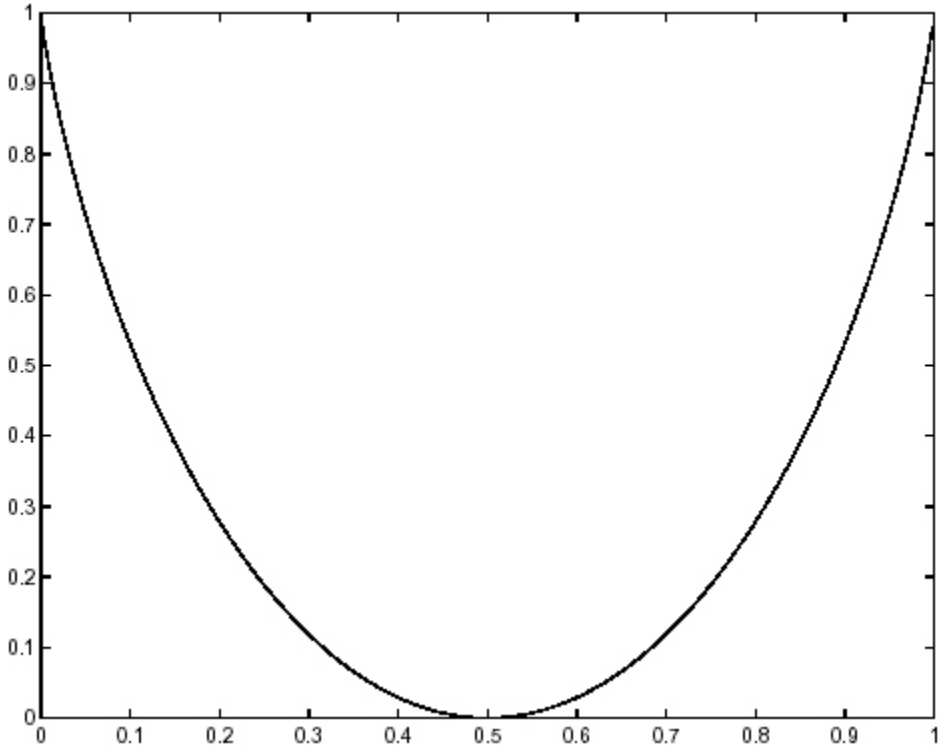


图6.6：互信息作为比特错误概率 ε的函数，以比特为单位。

概率分布 p（A_i）。 更有用的是，将信道容量定义为仅依赖于信道，因此使用最大互信息 $M_{\max}$，即由任何可能的输入概率分布导致的最大互信息。 在对称二进制信道的情况下，当两个输入概率相等时，达到最大值。一般来说，远离对称情况在工程系统中几乎没有任何优势，并且特别是本章中的定理所给出的基本限制不能通过这种技术来规避。 因此，对称情况给出了正确的直观理解。

信道容量被定义为

$$C = M_{\max} W \tag{6.29}$$

其中 W是输出状态能够跟随输入变化的最大速率。 因此 C以每秒比特为单位表示。

信道容量定理由香农在1948年首次证明。 它给出了通过信道传输信息的速率的基本限制。 如果输入信息速率以每秒比特为单位 D小于 C，则可以通过以任意低的错误率处理长序列的输入来编码数据。 另一方面，如果$D > C$，则不可能实现这一点；事实上，从学习输出中推断出关于输入的信息的最大速率是 C。 这个结果与无噪声信道的结果完全相同，如图6.4所示。这个结果真的非常了不起。 一个只取决于信道的容量数值被定义，然后定理表明可以找到一个性能与这个容量任意接近的编码。 结合源编码定理，它意味着可以分两个阶段设计通信信道——首先，对源进行编码，使得码字的平均长度等于其熵，然后，这串比特流可以以任意低的错误率传输，速率高达信道容量。 信道容量与输入能够改变的本地速率不同，而是由于噪声而降低了该值。

不幸的是，这个定理的证明（这里没有给出）并没有说明如何进行

找到这样的编码。 换句话说，这不是一个构造性的证明，其中通过显示编码来证明断言。 自从Shannon发表这个定理以来的半个世纪里，已经有很多发现了越来越好的编码，以满足各种高速数据通信的需求。 然而，目前还没有任何关于如何从零开始设计编码的通用理论（例如Huffman过程提供了源编码的方法）。

克劳德·香农（1916-[2]001）被公认为通信领域历史上最伟大的人物。他建立了今天被称为信息理论的整个科学研究领域。 他在贝尔实验室工作时完成了这项工作，在从麻省理工学院获得电气工程硕士学位和数学博士学位后。 正是他认识到二进制数字是所有通信中的基本要素。

1956年，他作为教职人员返回麻省理工学院。 在他晚年，他患有阿尔茨海默病，遗憾的是，他无法参加1998年在麻省理工学院举行的纪念他开创性论文50周年的研讨会。

6.8 可逆性

值得注意的是，迄今为止讨论的哪些操作涉及信息的丢失，哪些不涉及。一些布尔运算具有这样的特性，即无法从输出中推断出输入。 AND和 OR门就是例子。 其他操作是可逆的——当输出与两个输入之一相加时，$EXOR$门就是一个例子。

有些源可以被编码，使得所有可能的符号都由不同的编码字表示。 如果符号的数量是有限的，这总是可能的。 其他源具有无限数量的可能符号，这些符号无法被精确编码。 用于编码这些源的技术包括整数的二进制编码（存在溢出问题）和实数的浮点表示（存在溢出和下溢问题，同时也存在有限精度问题）。

一些压缩算法在输入和输出之间是可逆的，意味着输入可以完全恢复。 其中一种技术是LZW，它用于文本压缩和一些图像压缩等等。 其他算法以更高的效率为代价，会有一些信息的损失。

例如，JPEG压缩图像和MP3压缩音频。

现在我们已经看到一些通信信道是无噪声的，在这种情况下，可以以接近信道容量的速率进行完美传输。 其他信道存在噪声，完美的可逆通信是不可能的，尽管如果数据速率小于信道容量，错误率可以被无限小地减小。 对于更高的数据速率，信道必然是不可逆的。

在所有这些不可逆的情况下，信息会丢失（或者最多保持不变）。 在我们考虑的所有系统中，信息从未增加过。

这里是否有一个普遍的原则在起作用？

[2]请参阅Shannon的传记，网址为 http://www-groups.dcs.st-andrews.ac.uk/%7Ehistory/Biographies/Shannon.html

6.9 详细信息：通信系统要求

我们一直在开发的通信系统模型如图6.1所示。假设源发出一串符号。 通道可以是空间中不同点之间的物理通道，也可以是存储信息以便以后检索的存储器，甚至可以是以某种方式处理信息的计算。

当然，不同的通信系统，尽管它们都可以很好地描述我们的模型，但它们的要求是不同的。 下表试图说明现代系统合理的要求范围。 当然，这并不完整。

这些系统由四个指标来描述：吞吐量、延迟、容错性和非均匀速率（突发）容忍度。 吞吐量简单地指的是这样一个系统每秒应该容纳的比特数，以便成功。 延迟是消息的时间延迟；它可以被定义为源开始后输出开始的延迟，或者关于消息结束（或者关于消息中的任何特定特征）的类似数量。 吞吐量的数字，以MB为单位

(兆字节)或kb (千比特)是近似值。

	吞吐量 (每秒)	最大 延迟	错误 容忍的?	突发 容忍的?
计算机内存	许多MB	微秒	否	是
硬盘	MB或更高	毫秒	否	是
对话	??	50毫秒	是的；反馈错误控制	令人讨厌
电话	20 kb	100毫秒	容忍噪音	否
广播	??	秒	容忍一些噪音	否
即时消息	低	秒	否	是
光盘	1.4 MB	2秒	否	否
互联网	1 MB	5秒	否	是
打印队列	1 MB	30秒	否	是
传真	14.4 kb	分钟	容忍错误	是
快门电报站	??	5 min	否	是
电子邮件	不适用	1小时	否	是
隔夜递送	大	1天	否	是
包裹递送	大	天	否	是
普通邮件	大	天	否	是

表6.1：各种通信系统

第7章

过程

我们一直在开发的通信系统模型如图7.1所示。这个模型对于一些计算系统也很有用。 假设源发出一系列符号。 通道可以是空间中不同点之间的物理通道，也可以是存储信息以便以后检索的存储器，或者可以是以某种方式处理信息的计算。

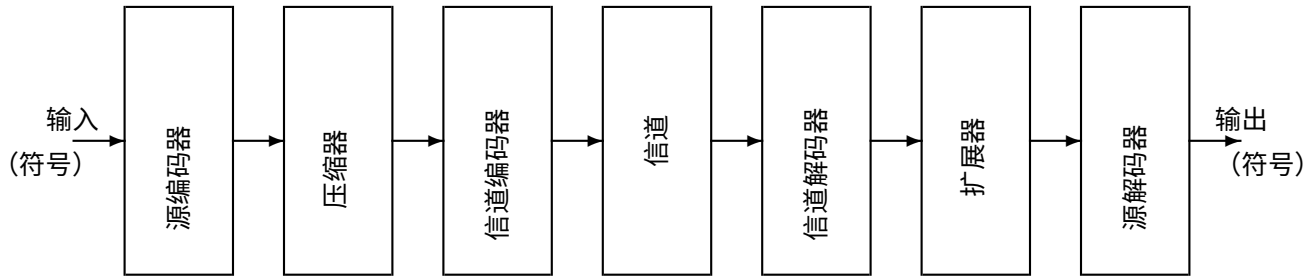


图7.1：通信系统

图7.1显示了模块的输入和输出以及它们的连接方式。 这样的图表非常有用，可以概述系统的运作，但其他表示方法也很有用。
在本章中，我们构建了两个抽象模型，这些模型足够通用，可以表示图7.1中的每个方框，并以定量方式显示信息的流动。

因为图7.1中的每个方框都以某种方式处理信息，所以被称为处理器，并且它所做的被称为过程。 我们在这里考虑的过程是

- 离散的：输入是一组互斥的可能性中的成员，每次只发生一个，输出是另一组互斥事件中的一个。
- 有限的：可能输入的集合数量有限，可能输出的集合数量也是有限的。
- 无记忆的：该过程在某个时间对输入进行处理，并根据该输入产生输出，忽略任何先前的输入。

- 非确定性：当给定相同的输入时，该过程可能会产生不同的输出（该模型也适用于确定性过程）。 由于该过程是非确定性的，输出可能包含随机噪声。

- 有损失性：可能无法通过观察输出来“看到”输入，即无法通过观察输出来确定输入（该模型也适用于无损失性过程）。 这样的过程被称为有损失性，因为在创建输出时丢失了关于输入的信息（该模型也适用于无损失性过程）。

7.1 过程图的类型

不同类型的过程图对于不同的目的是有用的。 我们在这里使用的四种图都是递归的，意味着一个过程可以用同类的其他更详细的过程来表示，并相互连接。 相反，两个或多个连接的过程可以由一个单一的高级过程表示，其中一些详细信息被抑制。 所表示的过程可以是确定性（无噪声）或非确定性（有噪声），也可以是无损失性或有损失性。

- 块图: 图7.1（上一页）是一个块图。 它展示了过程之间的连接，但很少涉及过程如何实现其目的，以及连接是如何建立的。 它对于以高度抽象的方式查看系统非常有用。 块图中的互连可以表示许多位。

- 电路图: 如果系统由逻辑门组成，一个有用的图是显示这些门之间互连的图。 例如，图7.2是一个 AND 门。 每个输入和输出代表一个具有单一逻辑值的电线，例如，高电压表示1，低电压表示0。 逻辑门的可能位模式数量大于物理电线的数量；每根电线可以有两种可能的电压，因此对于n个输入的门，将有2^n个可能的输入状态。 通常情况下，逻辑电路中的组件是确定性的，但并非总是如此。

- 概率图：具有单比特输入和单比特输出的过程可以通过与2个可能的输入比特模式和2个可能的输出模式相关的概率来建模。 例如，图7.3（下一页）显示了一个具有两个输入（四个比特模式）和一个输出的门。 这样一个门的例子是 AND门，其概率模型显示在图7.4中。概率图在第7.2节中进一步讨论。

- 信息图：显示过程之间明确的信息流动的图表是有用的。 为了处理带有噪声或损失的过程，可以显示与其相关的信息。 信息图在第7.6节中进一步讨论。

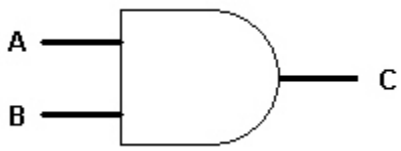


图 7.2：一个 AND门的电路图

7.2 概率图

图7.5显示了具有 n个输入和 m个输出的过程的概率模型，其中 n和 m是整数。 n个输入状态是互斥的， m个输出状态也是互斥的。 如果这个过程由逻辑门实现，输入至少需要$\log_2(n)$个线，但不需要与 n一样多的线。

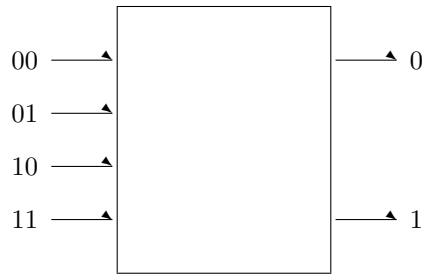


图7.3：一个双输入单输出门的概率模型。

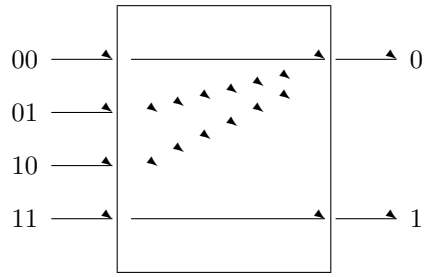


图7.4：一个与门的概率模型

这个过程模型在概念上简单而通用。它适用于具有少量比特的过程。 它在第6章中用于二进制信道。

不幸的是，当输入比特的数量中等或较大时，概率模型变得笨拙。
原因是输入和输出以互斥事件集的形式表示。 如果事件描述了五根导线上的信号，每根导线可以携带高电压或低电压，表示布尔值1或0，那么就会有32个可能的事件。 与表示物理变量的五个输入相比，绘制逻辑门要容易得多，而不是具有32个输入状态的概率过程。 当过程表示具有大量原子的物理系统的状态演变时，可能的输入状态数量的“指数爆炸”问题变得更加严重。 例如，一个摩尔气体中的分子数是阿伏伽德罗常数 $N_A = 6.02 \times 10^{23}$。 如果每个原子只有一个关联的布尔变量，就会有$2^{N_A}$个状态，远远超过宇宙中的粒子数。 甚至没有时间列出所有粒子，更不用说进行任何计算：自大爆炸以来的微秒数小于5×10^{23}。 尽管存在这个限制，概率图模型在概念上是有用的。

让我们回顾一下通信中的基本概念，在第6章中介绍的情况下，使用这样的图表。

我们假设过程的每个可能的输入状态都可以导致一个或多个输出状态。对于每个

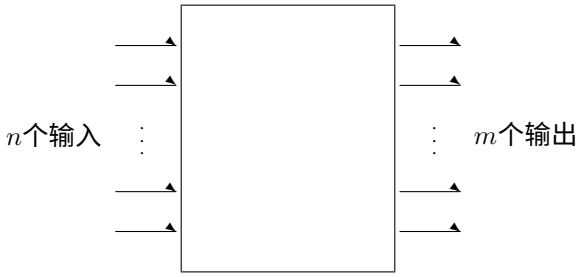


图7.5：概率模型

输入 i表示这个输入导致输出 j为 c_{ji}的概率。 这些转移概率c_{ji}可以被看作是一个表格或矩阵，列数与输入状态数相同，行数与输出状态数相同。 我们将使用 i作为输入状态的索引， j作为输出状态的索引，并将与选择输入 i相关的事件表示为 A_i，与输出 j相关的事件表示为 B_j。转移概率是过程的属性，不依赖于过程的输入。

转移概率介于0和1之间，对于每个 i，它们在输出索引 j上的总和为1，因为对于每个可能的输入事件，恰好发生一个输出事件。 如果输入状态的数量与输出状态的数量相同，则 c_{ji}是一个方阵；否则，它的列数多于行数或反之亦然。

$$0 \leq c_{ji} \leq 1 \tag{7.1}$$

$$1 = \sum_j c_{ji} \tag{7.2}$$

这个描述具有很大的普遍性。 它适用于确定性过程（尽管可能不是最方便的——通常更简单的是考虑给出每个输入的输出的真值表）。对于这样的过程，矩阵的每一列都包含一个元素为1，而其他所有元素都为0。 它还适用于非确定性信道（即带有噪声的信道）。 它适用于源编码器和解码器，压缩器和扩展器，以及信道编码器和解码器。 它适用于逻辑门和执行任意无记忆计算的设备（有时称为“组合逻辑”，与可能涉及先前状态的“时序逻辑”相区别）。 它甚至适用于物理系统从一个状态转移到下一个状态的转换。 如果输出状态的数量大于输入状态的数量（例如信道编码器）或小于输入状态的数量（例如信道解码器），则它适用。

如果一个过程的输入由随机事件 A_i决定，并且具有概率分布 $p(A_i)$，那么可以计算出各种其他概率。 在给定输入的条件下，条件输出概率为

$$p(B_j \mid A_i) = c_{ji} \tag{7.3}$$

每个输出的无条件概率 $p(B_j)$为

$$p(B_j) = \sum_i c_{ji} p(A_i) \tag{7.4}$$

最后，可以使用贝叶斯定理找到每个输入与每个输出的联合概率 $p(A_i, B_j)$和反向条件概率 $p(A_i \mid B_j)$：

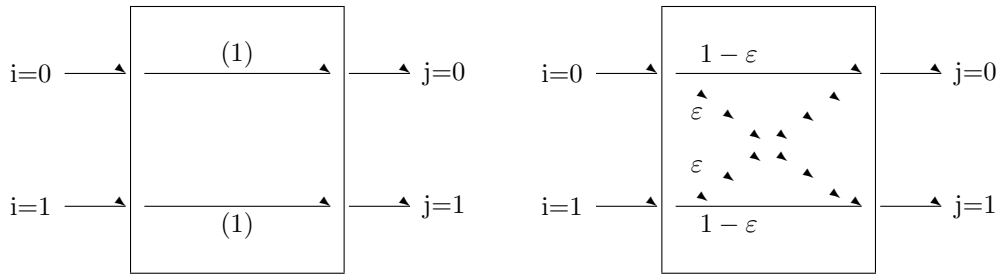
$$\begin{aligned} p(A_i, B_j) &= p(B_j)p(A_i \mid B_j) && (7.5)\\ &= p(A_i)p(B_j \mid A_i) && (7.6)\\ &= p(A_i)c_{ji} && (7.7) \end{aligned}$$

7.2.1 示例：与门

与门是确定性的（没有噪声），但是有损失，因为仅凭输出无法推断输入。过渡矩阵为

$$\begin{bmatrix} c_{0(00)} & c_{0(01)} & c_{0(10)} & c_{0(11)} \\ c_{1(00)} & c_{1(01)} & c_{1(10)} & c_{1(11)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.8}$$

该门的概率模型如图7.4所示。



(a) 无错误 (b) 对称错误

图7.6：无错误二进制信道和对称二进制信道的概率模型

7.2.2 例子：二进制信道

二进制信道可以很好地用概率模型来描述。 它的属性，其中许多在第6章中已经讨论过，如下所述。

首先考虑一个无噪声的二进制信道，当输入值为0或1时，它会将该值准确地传输到输出。 这是一个非常简单的离散无记忆过程的例子。 我们用一个具有两个输入和两个输出的概率模型来表示这个信道。 为了表示输入在输出处被准确复制的事实，图7.6(a)中展示了盒子的内部结构，由两条路径组成，分别从每个输入到相应的输出，并且每条路径都标有概率(1)。 该信道的转移矩阵为

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.9}$$

这个过程的输入信息 I是1比特，如果两个值是等可能的，或者如果 $p(A_0) = p(A_1)$ 输入信息是

$$I = p(A_0)\log_2\frac{1}{p(A_0)} + p(A_1)\log_2\frac{1}{p(A_1)} \tag{7.10}$$

输出信息 J有一个类似的公式，使用输出概率 $p(B_0)$ 和 $p(B_1)$。 由于在这种情况下输入和输出是相同的，当观察到输出时，总是可以推断出输入。 输出的信息量 J与输入的信息量 I相同： $J = I$。 这个无噪声信道对于其预期目的是有效的，即允许接收器在输出端推断出输入值。

接下来，让我们假设这个信道偶尔会出错。 因此，如果输入为1，则输出不总是1，而是通过“比特错误概率” ε翻转为“错误”值0，因此只有概率$1-\varepsilon$才是“正确”的。 同样，对于输入为0的情况，错误的概率是 ε。 然后，转移矩阵为

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{bmatrix} \tag{7.11}$$

这个具有随机行为的模型有时被称为对称二进制信道（SBC），在两个方向上的错误（从0到1和从1到0）同样可能发生。 该信道的概率图如图7.6(b)所示，每个输入有两条路径离开，每个输出有两条路径汇聚。

显然，SBC中的错误引入了一些不确定性，超过了输入信号中的不确定性。直观地说，我们可以说噪声已经被添加进来，因此输出部分由期望信号组成，部分由噪声组成。 或者我们可以说在信道中丢失了一部分信息。 这两种效应都发生了，但是正如我们将看到的，它们并不总是相关的；

过程可能会引入噪声但没有损失，或者反之亦然。 在第7.3节中，我们将计算因噪声或损失而丢失或获得的信息量，以比特为单位。

信息的丢失是因为当观察到输出时，不再能够确定输入信号是什么。 损失在图7.6(b)中显示为两条或多条路径汇聚到同一个输出。 噪声是因为输出不是由输入精确确定的。

噪声在图7.6(b)中显示为两条或多条路径从同一个输入分离出来。 然而，尽管有噪声和损失，一些信息仍然可以从输入传输到输出（即，观察输出可以让人们对输入做出一些推断）。

现在我们回到一个一般的离散无记忆非确定性有损过程的模型，并推导出噪声、损失和信息传输（称为“互信息”）的公式。 然后我们将回到对称二进制信道并解释这些公式。

7.3 信息、损失和噪声

对于一般的离散无记忆过程，可以定义有用的信息量度，包括输入处呈现的信息量和传输到输出处的信息量。 我们假设过程状态由随机事件 A_i以概率分布 $p(A_i)$表示。 输入处的信息量 I等同于该源的熵。 （我们选择使用字母 I表示输入信息，不是因为它代表“输入”或“信息”，而是因为它代表遍历输入概率分布的索引 i。输出信息将用 J表示，原因类似。）

$$I = \sum_i p(A_i) \log_2 \sum \frac{1}{p(A_i)} \sum \tag{7.12}$$

这是我们对输入的不确定性量度，如果我们不知道输入是什么，或者在源选择之前。

输出也可以使用类似的公式表示。 输出信息 J也可以用输入概率分布和信道转移矩阵表示：

$$\begin{aligned} J &= \sum_j p(B_j) \log_2 \sum \frac{1}{p(B_j)} \sum \\ &= \sum_j \left(\sum_i c_{ji} p(A_i) \left(\log_2 \sum \frac{1}{\sum_i c_{ji} p(A_i)} \sum \right. \right. \end{aligned} \tag{7.13}$$

请注意，这个输出的信息度量指的是输出状态的身份，而不是输入状态。 它代表了我们在发现输出状态之前对输出状态的不确定性。如果我们的目标是确定输入状态，那么 J并不是我们想要的。 相反，我们应该询问我们对输入状态的知识的不确定性。 这可以从输出的角度来表达，通过询问在给定一个特定的输出状态下输入状态的不确定性，然后对这些状态进行平均。 对于每个 j，这种不确定性可以用类似上面的公式来表示，但使用相反的条件概率 $p(A_i \mid B_j)$

$$\sum_i p(A_i \mid B_j) \log_2 \sum \frac{1}{p(A_i \mid B_j)} \sum \tag{7.14}$$

然后，在学习输出后，您对输入的平均不确定性是通过计算输出概率分布的平均值来找到的，即通过乘以 $p(B_j)$并对 j求和。

$$\begin{aligned} L &= \sum_j p(B_j) \sum_i p(A_i \mid B_j) \log_2 \sum \frac{1}{p(A_i \mid B_j)} \sum \\ &= \sum_{ij} p(A_i, B_j) \log_2 \sum \frac{1}{p(A_i \mid B_j)} \sum \end{aligned} \tag{7.15}$$

请注意，第二个公式使用了联合概率分布 $p(A_i, B_j)$。 我们用 L表示这个平均不确定性，并称之为“损失”。这个术语是合适的，因为它是通过检查输出状态无法确定的输入信息的数量；在这个意义上，它在从输入到输出的转换中“丢失”了。 在特殊情况下，如果过程允许唯一地识别每个可能的输出状态的输入状态，该过程是“无损失”的，正如您所期望的那样，$L = 0$。

在第6章中证明了 $L \leq I$，或者换句话说，学习输出后的不确定性小于（或可能等于）之前的不确定性。 这个结果是使用吉布斯不等式证明的。

我们在被告知输出状态之前对输入状态所了解的信息量是我们的不确定性，即 I，减去被告知之后的不确定性，即 L。 我们刚刚证明了这个量不能为负，因为 $L \leq I$。 正如第6章所做的那样，我们将我们所学到的量表示为 $M = I - L$，并称之为输入和输出之间的“互信息”。 这是一个重要的量，因为它是通过过程传递的信息量。

为了总结这些信息量之间的关系：

$$I = \sum_i p(A_i) \log_2 \sum \frac{1}{p(A_i)} \sum \tag{7.16}$$

$$L = \sum_j p(B_j) \sum_i p(A_i \mid B_j) \log_2 \sum \frac{1}{p(A_i \mid B_j)} \sum \tag{7.17}$$

$$M = I - L \tag{7.18}$$

$$0 \leq M \leq I \tag{7.19}$$

$$0 \leq L \leq I \tag{7.20}$$

具有可以由多个输入产生的输出的过程会有损失。 这些过程也可能是非确定性的，即一个输入状态可能导致多个输出状态。 具有损失且也是非确定性的对称二进制信道是一个例子。

然而，有一些过程是有损耗但是确定性的。 一个例子是 AND逻辑门，它有四个互斥的输入00 01 10 11和两个输出0和1。 其中三个输入导致输出为0。 这个门有损耗但是完全确定性，因为每个输入状态都导致一个确定的输出状态。 有损耗意味着 AND门不可逆。

有一种类似于 L的量来描述一个非确定性过程，无论它是否有损耗。
非确定性过程的输出包含无法从输入中预测的变化，就像音频系统中的噪音一样。 我们将过程的噪音 N定义为在所有输入状态上，给定输入状态的情况下输出的不确定性的平均值。 它与损耗的定义非常相似，但是输入和输出的角色被颠倒了。因此

$$\begin{aligned} N &= \sum_i p(A_i) \sum_j p(B_j \mid A_i) \log_2 \sum \frac{1}{p(B_j \mid A_i)} \sum \\ &= \sum_i p(A_i) \sum_j c_{ji} \log_2 \sum \frac{1}{c_{ji}} \sum \end{aligned} \tag{7.21}$$

与上述损失类似的步骤显示出类似的结果。 可能不明显，但可以很容易地证明，互信息 M在噪声方面起到与损失相同的作用。
与上述损失类似，与其他信息度量相关的公式如下，其中互信息 M相同：

$$J = \sum_i p(B_j) \log_2 \frac{1}{p(B_j)} \tag{7.22}$$

$$N = \sum_i p(A_i) \sum_j c_{ji} \log_2 \frac{1}{c_{ji}} \tag{7.23}$$

$$M = J - N \tag{7.24}$$

$$0 \leq M \leq J \tag{7.25}$$

$$0 \leq N \leq J \tag{7.26}$$

根据这些结果可以得出结论

$$J - I = N - L \tag{7.27}$$

7.3.1 示例：对称二进制信道

对于具有误码率 ε的SBC，即使两个输入概率 $p(A_0)$和 $p(A_1)$不相等，也可以计算这些公式。如果它们恰好相等（每个为0.5），那么SBC的各种信息度量以比特为单位特别简单：

$$\text{我} = 1\text{比特} \tag{7.28}$$

$$J = 1\text{比特} \tag{7.29}$$

$$L = N = \varepsilon \log_2 \frac{1}{\varepsilon} + (1-\varepsilon) \log_2 \frac{1}{1-\varepsilon} \tag{7.30}$$

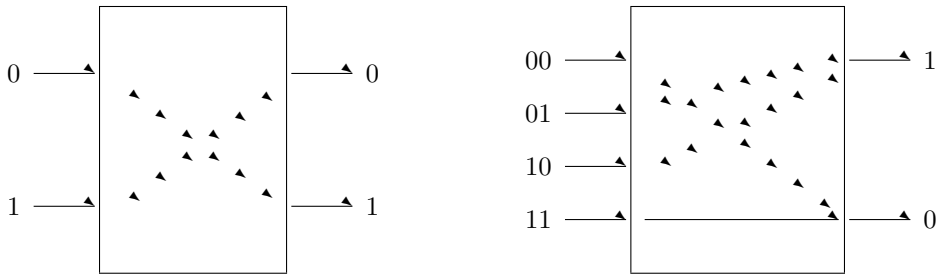
$$M = 1 - \varepsilon \log_2 \frac{1}{\varepsilon} - (1-\varepsilon) \log_2 \frac{1}{1-\varepsilon} \tag{7.31}$$

信道中的错误破坏了一些信息，从输出端来看，它们阻止了观察者确定输入是什么。因此，它们只允许通过信道到输出的信息量为 $M = I - L$。

7.4 确定性示例

这个概率模型适用于具有互斥输入和输出的任何系统，无论转换是否随机。 如果所有的转换概率 c_{ji}都等于0或1，则该过程是确定性的。

确定性过程的一个简单例子是NOT门，它实现了布尔取反。 如果输入为1，则输出为0，反之亦然。 输入和输出信息相同， $I = J$，并且没有噪声或损失： $N = L = 0$。 通过门传递的信息为 $M = I$。 参见图7.7(a)。一个稍微复杂的确定性过程是异或（ XOR）门。 这是一个两个输入变量的布尔函数，因此有四种可能的输入值。 当门被表示为



(a) 非　　门　　(b)异或门

图7.7：确定性门的概率模型

在电路图中，有两根输入线代表两个输入。 当门被表示为一个使用概率图的离散过程，如图7.7(b)，有四个互斥的输入和两个互斥的输出。 如果四个输入的概率都是0.25，那么$I = 2$比特，两个输出的概率都是0.5，所以$J = 1$比特。因此有1比特的损失，互信息为1比特。损失是由于两个不同的输入产生相同的输出；例如，如果观察到输出1，输入可以是01或10。 输出中没有引入噪声，因为每个转换参数都是0或1，即没有具有多个转换路径的输入。

其他更复杂的逻辑函数可以用类似的方式表示。 然而，对于具有物理输入的逻辑函数，如果n大于3或4，则概率图表会变得笨拙，因为输入的数量是2的n次方。

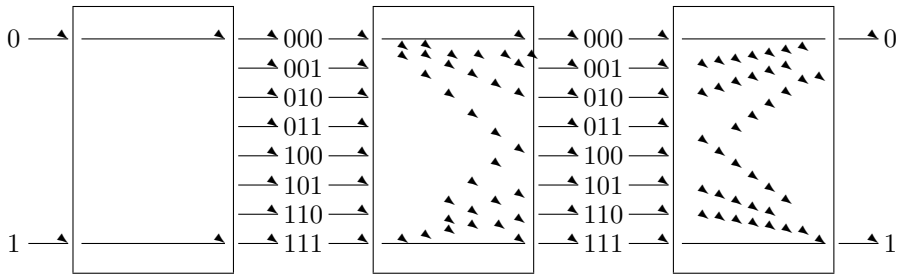
7.4.1纠错示例

汉明码编码器和解码器可以用这种形式表示为离散过程。 考虑$(3, 1, 3)$码，也称为三重冗余。 编码器有一个1位输入（2个值）和一个3位输出（8个值）。 输入1直接连接到输出111，输入0连接到输出000。

其他六个输出没有连接，因此概率为0。参见图7.8(a)。编码器具有N=0，L=0和M=I=J。 请注意，尽管使用三个物理位来表示输出信息，但输出信息并不是三位，因为有意的冗余。

三重冗余编码器的输出意图通过一个通道传递，每个3位块中可能发生单个位错误。 这个有噪声的通道可以被建模为一个非确定性过程，具有8个输入和8个输出，图7.8(b)。每个8个输入都与相应输出连接，且与Hamming距离为1的其他三个值分开的低概率连接。 例如，输入000只与输出000（高概率）以及001、010和100（每个低概率）连接。 这个通道引入了噪声，因为每个输入都有多条路径。 一般来说，当输入任意位模式时，也会有损失。 然而，当从图7.8(a)的编码器驱动时，损失为0位，因为只有八个位模式中的两个具有非零概率。 噪声通道的输入信息为1位，输出信息由于添加了噪声而大于1位。 这个例子说明了噪声和损失的值都取决于通道的物理特性和输入信号的概率。

解码器用于恢复最初放入编码器的信号，如图7.8(c)所示。转换参数很简单-每个输入只连接到一个输出。 解码器具有损失（因为多个路径汇聚到每个输出）但没有噪声（因为每个输入只到一个输出）。



(a) 编码器 (b) 信道 (c) 解码器

图7.8：三重冗余错误校正

7.5 容量

在这些笔记的第6章中，定义了信道容量。 这个概念可以推广到其他过程。

将最大速率W定义为过程的输入状态可以在输出中被检测到的最大速率。 然后，信息通过过程的速率可以达到$W\,M$的大小。 然而，这个乘积取决于输入概率分布$p(A_i)$，因此不是过程本身的属性，而是取决于它的使用方式。 通过观察M如何随不同的输入概率分布变化，可以找到过程容量的更好定义。 选择任何输入概率分布的最大互信息，并将其称为$M\max$。 然后，过程容量C被定义为

$$C = W\,M_{max} \tag{7.32}$$

很容易看出，M_{max}不能任意大，因为$M \leq I$且$I \leq \log_2 n$其中n是不同输入状态的数量。

在对称二进制信道的例子中，很容易证明最大化M的概率分布是每个两个输入状态都有相等概率的分布。

7.6 信息图

信息图是明确显示信息传递量的一个或多个过程的表示。 它是表示输入、输出、互信息、噪声和损失的有用方式。 信息图在抽象层面上，并不显示导致这些信息度量的详细概率。

已经证明所有五个信息度量，I、J、L、N和M都是非负的。 L和N不一定相同，尽管对于输入0和1的概率相等的对称二进制信道来说它们是相同的。 可能存在有损失但没有噪声的过程（例如XOR门），或者有噪声但没有损失的过程（例如三重冗余的噪声信道）。

方便地将信息视为通过这个过程传输的物理量，就像物质在生产线上被加工一样。 正在生产的材料进入制造区域，由于错误或其他原因，部分材料会丢失，可能会添加一些污染物（如噪声），输出量是输入量减去损失再加上噪声。 有用的产品是输入量减去损失，或者是输出量减去噪声。 通过这个范例，可以展示离散无记忆过程中信息的流动，如图7.9所示。

一个有趣的问题出现了。 概率取决于你当前的知识状态，一个观察者的知识可能与另一个观察者的知识不同。 这意味着损失、噪声和传输的信息都取决于观察者。 重要的工程量如噪声和损失是否取决于你是谁和你知道什么，这样可以吗？ 如果你碰巧对输入有所了解

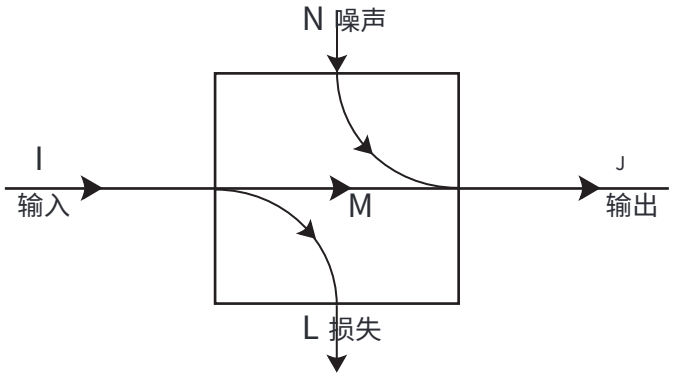


图7.9：离散无记忆过程中的信息流

如果你的同事不这样做，你的非确定性过程设计是否可以不同，并利用你的知识？ 这个问题值得思考；有时候，如果你的知识是正确的，它可以在简化设计方面非常有价值，但有时候，最好是根据输入概率的最坏情况假设进行设计，这样即使输入不符合你假设的概率，你的设计仍然能够工作。

信息图通常不用于通信系统。 通常不需要考虑噪声源或丢失信息的情况。 然而，在噪声和丢失不可能发生的领域，这样的图表是有用的。 一个例子是可逆计算，这是一种在原理上可以反向运行整个过程的计算方式。 另一个例子是量子通信，其中不能丢弃信息而不影响环境。

7.6.1 符号

不同的作者使用不同的符号来表示我们这里称为 I, J, L, N和 M的量。 在他的原始论文中，香农将输入概率分布表示为 x，输出分布表示为 y。 输入信息 I被表示为 $H(x)$，输出信息 J被表示为 $H(y)$。 损失 L（香农称之为“不确定性”）被表示为 $H_y(x)$，噪声 N被表示为 $H_x(y)$。 互信息 M被表示为 R。 香农用“熵”一词来表示信息，大多数作者都遵循了他的做法。

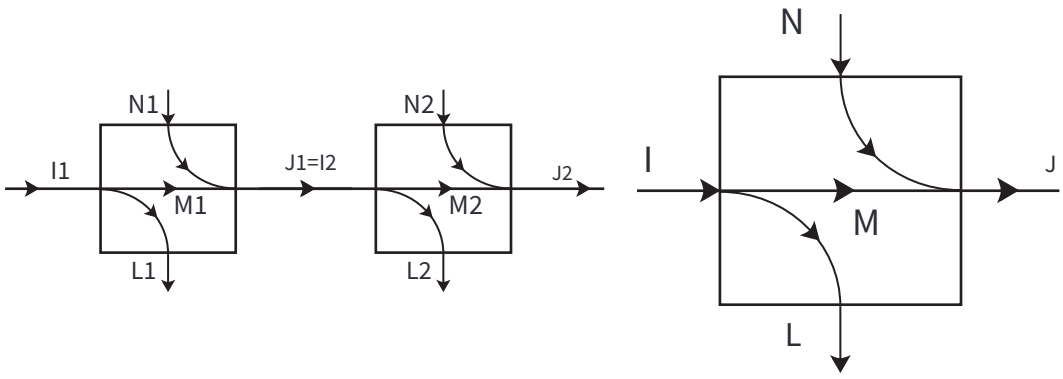
通常，信息量被表示为 I, H或 S，通常作为概率分布或“集合”的函数。在物理学中，熵通常表示为 S。

另一种常见的表示法是使用 A表示输入概率分布或集合，B表示输出概率分布。 如果需要联合表示与A和B相关的信息（而不是条件性的），可以表示为 $I(A, B)$ 或 $I(AB)$。

7.7级联过程

考虑两个级联的过程。 这个术语指的是将一个过程的输出作为另一个过程的输入。 如果“内部”状态被隐藏，那么这两个级联的过程可以被建模为一个更大的过程。 我们已经看到离散无记忆过程的特征是 I、 J、 L、 N和 M的值。 图7.10(a)显示了一对级联的过程，每个过程都有自己的参数。 当然，第二个过程的参数取决于它遇到的输入概率，这些概率由第一个过程的转移概率（和输入概率）确定。

但是这两个过程的级联本身是一个离散无记忆过程，因此应该有它自己的五个参数，如图7.10(b)所示。整体模型的参数可以计算得到



(a)这两个过程 (b)等效单一过程

图7.10: 两个离散无记忆过程的级联

有两种方式之一。 首先，可以从连接在一起的两个模型的转移概率中找到整体过程的转移概率；实际上，转移概率矩阵只是过程1和过程2的转移概率矩阵的矩阵乘积。 所有的参数都可以从这个矩阵和输入概率计算得到。

另一种方法是寻找整体过程的 $I,\ J,\ L,\ N$和 M 的公式，以组成过程的相应量为基础。 对于输入和输出量来说，这是微不足道的：$I = I_1$和 $J = J_2$。 然而，对于 L和 N来说，这更加困难。 尽管 L和 N通常不能从 $L_1,\ L_2,\ N_1$和 N_2准确地找到，但是可以找到它们的上下界。 这些上下界对于理解级联的运行非常有用。

可以很容易地证明，由于 $I = I_1$， $J_1 = I_2$，且 $J = J_2$，

$$L - N = (L_1 + L_2) - (N_1 + N_2) \tag{7.33}$$

然后可以直接（尽管可能有点乏味）证明整个过程的损耗 L并不总是等于两个组件的损耗之和 $L_1 + L_2$，而是

$$0 \leq L_1 \leq L \leq L_1 + L_2 \tag{7.34}$$

因此损耗有上下界。此外，

$$L_1 + L_2 - N_1 \leq L \leq L_1 + L_2 \tag{7.35}$$

因此，如果第一个过程是无噪声的，那么 L恰好等于 $L_1 + L_2$。

类似的公式也适用于 N与 $N_1 + N_2$的关系：

$$0 \leq N_2 \leq N \leq N_1 + N_2 \tag{7.36}$$

$$N_1 + N_2 - L_2 \leq N \leq N_1 + N_2 \tag{7.37}$$

类似的级联互信息公式可以从这些结果中得出：

$$M_1 - L_2 \leq M \leq M_1 \leq I \tag{7.38}$$
$$M_1 - L_2 \leq M \leq M_1 + N_1 - L_2 \tag{7.39}$$
$$M_2 - N_1 \leq M \leq M_2 \leq J \tag{7.40}$$
$$M_2 - N_1 \leq M \leq M_2 + L_2 - N_1 \tag{7.41}$$

其他关于 M 的公式可以通过将方程7.19应用于第一个过程和级联，以及将方程7.24应用于第二个过程和级联来轻松推导出来：

$$\begin{aligned} M &= M_1 + L_1 - L \\ &= M_1 + N_1 + N_2 - N - L_2 \\ &= M_2 + N_2 - N \\ &= M_2 + L_2 + L_1 - L - N_1 \end{aligned} \tag{7.42}$$

这里每种情况下的第二个公式都来自于方程7.33的使用。

注意， M不能超过 M_1或 M $_2$这与 M的解释一致，即通过级联的信息必须能够通过第一个过程和第二个过程。

作为一个特例，如果第二个过程是无损的， L $_2 = 0$，那么 $M = M_1$。 在这种情况下，第二个过程不会降低互信息低于第一个过程的互信息。 同样，如果第一个过程是无噪声的，那么 $N_1 = 0$， $M = M_2$。

级联的信道容量 C同样不会超过第一个过程或第二个过程的信道容量： C $\leq$ 与信道容量相关的其他结果不是上述公式的平凡结果，因为C根据定义是所有可能输入概率分布中的最大 M值——最大化 M_1的分布可能不会导致最大化 M_2的第二个过程的输入概率分布。

第8章

推理

在第7章中，过程模型被引入为一种解释信息在离散、有限、无记忆的过程中的流动的方式，这些过程可能是非确定性的和有损失的。 虽然该模型是受到许多通信系统工作方式的启发，但它更加通用。给出了输入信息 I、损失 L、互信息 M、噪声 N和输出信息 J的公式。 这些量都是以比特为单位衡量的，

尽管在选择许多符号的设置中，它们可以乘以符号选择的速率，然后以每秒比特的形式表示。 信息流如图8.1所示。所有这些量都取决于输入概率分布P(A_i)。

如果输入概率已知，并且观察到特定的输出结果，就可以推断导致该结果的输入事件。 有时可以确定输入事件，但更常见的是通过改变初始输入概率来进行推断。 这通常是通信系统的工作方式——观察输出并推断出“最有可能”的输入事件。 在这种情况下，推断有时被称为估计。 这是第8.1节的主题。另一方面，如果输入概率未知，这种方法就不起作用。 我们需要一种获取初始概率分布的方法。 基于信息分析的方法在第8.2节和本笔记的后续章节中进行了讨论。 这就是最大熵原理。

8.1 估计

通常需要在仅观察到输出事件时确定输入事件。 这是通信系统的情况，其目标是推断源发射的符号，以便在输出处重新生成。 这也适用于存储系统的情况，其目标是无误地重新创建原始的位模式。

原则上，如果已知输入概率分布 $p(A_i)$和条件输出概率（在输入事件条件下） $p(B_j \mid A_i) = c_{ji}$，则这种估计是直接的。 这些“前向”条件概率 c_{ji}形成一个矩阵，行数与输出事件数量相同，列数与输入事件数量相同。 它们是过程的属性，不依赖于输入概率 $p(A_i)$。

每个输出事件B_j的无条件概率 $p(\ B_j)$是

$$p(B_j) = \sum_i c_{ji} p(A_i) \tag{8.1}$$

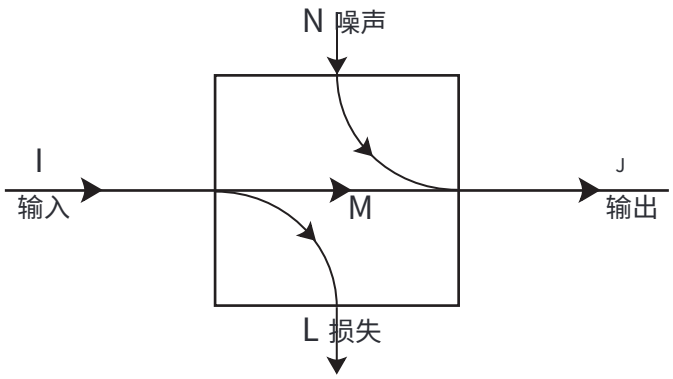


图8.1：离散无记忆过程中的信息流

每个输入与每个输出的联合概率 $p(A_i, B_j)$和反向条件概率
$p(A_i \mid B_j)$可以使用贝叶斯定理来计算：

$$\begin{aligned} p(A_i, B_j) &= p(B_j)p(A_i \mid B_j) \\ &= p(A_i)p(B_j \mid A_i) \\ &= p(A_i)c_{ji} \end{aligned} \tag{8.2}$$

现在假设已经观察到了特定的输出事件 B_j。 “导致”这个输出的输入事件只能估计出一个概率分布，覆盖了所有可能的输入事件。
对于每个输入事件 A_i，它是输入的概率就是特定输出事件Bj的反向条件概率
$p(A_i \mid B_j)$，可以使用方程8.2来表示

$$p(A_i \mid B_j) = \frac{p(\blacksquare i)c_{ji}}{p(B_j)} \tag{8.3}$$

如果过程没有损失($L = 0$)，那么对于每个 j，输入事件 A_i中的一个具有非零概率，因此其概率 $p(A_i \mid B_j$) 为1。 在更一般的情况下，存在非零损失，估计就是对一组输入概率进行细化，使其与已知的输出一致。 请注意，这种方法仅在已知原始输入概率分布的情况下有效。 它只是在新的知识（即观察到的输出）的光线下细化了该分布。

人们可能认为新的输入概率分布的不确定性比原始分布小。这总是正确的吗?

概率分布的不确定性当然是其熵，如前所定义。 在输出事件未知之前，对于输入事件的不确定性是

$$U_{\text{之前}} = \sum_i p(A_i) \log_2 \sum \frac{1}{p(A_i)} \sum \tag{8.4}$$

在某个特定的输出事件已知之后，剩余的不确定性是

$$U_{\text{之后}}(B_j) = \sum_i p(A_i \mid B_j) \log_2 \sum \frac{1}{p(A_i \mid B_j)} \sum \tag{8.5}$$

那么问题是，是否 $U_{\text{after}}(B_j) \leq U_{\text{before}}$. 答案通常是肯定的，但并不总是如此。然而，很容易证明剩余不确定性的平均值（在所有输出状态上）小于原始不确定性：

$$\sum_j p(B_j) U_{\text{after}}(B_j) \leq U_{\text{before}} \tag{8.6}$$

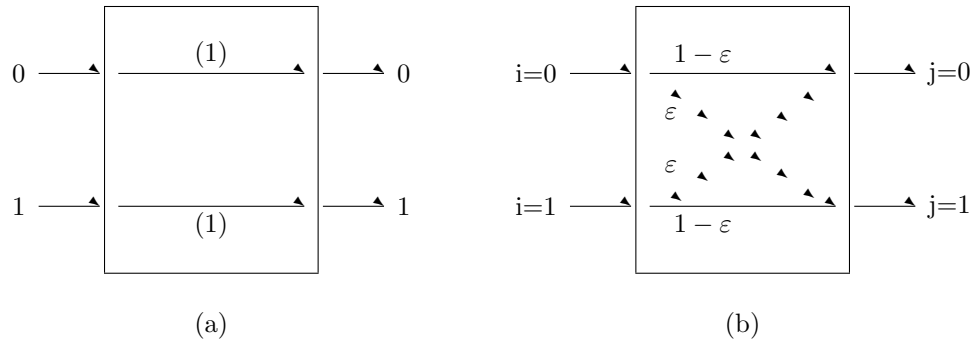


图8.2： （a）无噪声的二进制信道（b）有错误的对称二进制信道

简言之，这个陈述说的是，平均而言，我们对输入状态的不确定性不会因为了解输出状态而增加。 换句话说，平均而言，这种推断技术帮助我们更好地估计输入状态。

以下两个例子将在后续章节中继续讨论，包括下一章关于最大熵原理的对称二进制信道和Berger's Burgers。

8.1.1 对称二进制信道

图8.2(a)中显示的无噪声、无损耗的二进制信道是一个具有两个输入值（称为0和1）、两个相应命名的输出值和一个过渡矩阵 c_{ji}的过程，保证输出等于输入：

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{8.7}$$

该信道没有损耗和噪声，互信息、输入信息和输出信息都相同。

对称二进制信道（图8.2(b)）类似，但偶尔会出错。 因此，如果输入为1，则输出不总是1，而是以“位错误概率” ε翻转为“错误”值0，并且只有概率为$1-\varepsilon$才是“正确”的。 同样，对于输入为0，错误的概率是 ε。

然后过渡矩阵为

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{bmatrix} \tag{8.8}$$

该信道在从0到1和从1到0的错误方向上是对称的，错误的概率是相等的。

由于损失，与输出事件 B_0相关联的输入事件无法确定其确定性。 尽管如此，上述公式仍然可以使用。 在两个输入概率相等的重要情况下（因此每个概率都等于0.5），输出为0意味着输入事件 A_0的概率为$1-\varepsilon$，输入事件 $A1$的概率为 ε。 因此，如果在设计用于低误差通信的信道中， ε很小，那么合理推断产生输出事件 B_0的输入事件是事件 A_0。

8.1.2 非对称二进制信道

非对称二进制信道是指输入0和1的错误概率不同的信道，即 $c_{01} = c_{10}$。 我们通过基于亨廷顿病的医学测试的极端案例来说明非对称二进制信道。

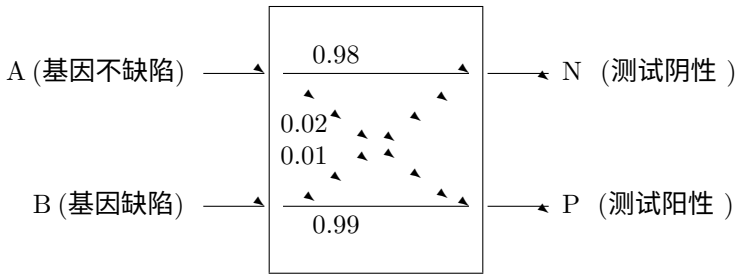


图8.3：亨廷顿病测试

亨廷顿病是一种罕见的、进行性的、遗传性的脑部疾病，目前没有已知的治愈方法。它是以乔治·亨廷顿博士（1850-1915）命名的，他是一位长岛的医生，在1872年发表了一篇描述。 它是由一个缺陷基因引起的，该基因在1983年被鉴定出来。 可能最著名的患有该疾病的人是词曲创作家伍迪·格思里。

携带有缺陷基因的人的生物子女有50%的机会继承该缺陷基因。 对于整个人群来说，携带有缺陷基因的概率要低得多。 根据美国亨廷顿病协会（http://www.hdsa.org）的数据，“超过25万名美国人患有亨廷顿病或有遗传病风险。” 这约占人口的1/1000，因此对于一个随机选择的、家族史未知的人来说，估计携带受影响基因的概率为1/2000。

携带有缺陷基因的人最终都会患上这种疾病，除非他们死于其他原因
先。 症状通常在中年人中出现，即40或50岁的人，可能在他们
已经组建了一个家庭并可能将缺陷基因传给下一代之后。
虽然这种疾病不会致命，但处于晚期的人通常会死于其并发症。 直到最近，患有这种疾病家族史的人面临着不确定的生活，不知道他们是否携带有缺陷基因，也不知道如何管理他们的个人和职业生活。

1993年开发出了一种可以检测是否携带有缺陷基因的测试。 不幸的是，这个测试并不完美；存在误报阳性（报告你有缺陷基因，实际上并没有）和误报阴性（报告你的基因没有缺陷，实际上是有缺陷）的概率。 为了我们的目的，我们将假设这个测试只给出是/否的答案，误报阳性的概率为2%，误报阴性的概率为1%。 （实际测试实际上更好——它还估计缺陷的严重程度，这与症状开始的年龄相关。）

如果你参加测试并了解结果，你当然想推断你最终是否会患上这种疾病。 上述开发的技术可以提供帮助。

让我们将这个测试建模为一个离散无记忆过程，输入为 A（无缺陷基因）和 B（有缺陷基因），输出为 P（阳性）和 N（阴性）。 如图8.3所示的过程不是对称二进制信道，因为两个错误概率不相等。

首先，考虑将该测试应用于有家族史的人，其中 $p(A) = p(B) = 0.5$。然后，如果测试结果为阴性，那个人患有该缺陷的概率为$1/99 = 0.0101$，不患有该缺陷的概率为$98/99 = 0.9899$。另一方面，如果测试结果为阳性，那个人携带有缺陷基因的概率为$99/101 = 0.9802$，不携带有缺陷基因的概率为$2/101 = 0.0198$。该测试非常有效，因为两个输出以高概率表示不同的输入。

一个有趣的问题是，这个测试的存在引发了一个问题，但我们的数学模型没有解决，那就是一个有家族史的人会选择接受测试，还是宁愿不知道未来会发生什么。 这个测试的开发是由一个包括古思瑞的遗孀和由一位名叫米尔顿·韦克斯勒（1908-2007）的人领导的团队资助的，他对自己的女儿们很担心，因为他的妻子和她的兄弟都患有这种疾病。 韦克斯勒的女儿们，她们的情况启发了这个测试的开发，决定不接受测试。

接下来，考虑将这个测试应用于一个家族史未知的人，使得 $p(A)$ =0.9995 和 $p(B) = 0.0005$。 然后，如果测试结果为阴性，那个人携带有缺陷基因的概率 $p(B \mid N)$ 是

$$\frac{0.0005 \times 0.01}{0.0005 \times 0.01 + 0.9995 \times 0.98} = 0.000005105 \tag{8.9}$$

而携带正常基因的人的概率 $p(A \mid N)$ 是

$$\frac{0.9995 \times 0.98)}{0.0005 \times 0.01 + 0.9995 \times 0.98} = 0.999994895 \tag{8.10}$$

另一方面，如果测试结果为阳性，那么携带缺陷基因的人的概率 $p(B \mid P)$ 是

$$\frac{0.0005 \times 0.99}{0.0005 \times 0.99 + 0.9995 \times 0.02} = 0.02416 \tag{8.11}$$

而没有缺陷的概率 $p(A \mid P)$ 是

$$\frac{0.9995 \times 0.02}{0.0005 \times 0.99 + 0.9995 \times 0.02} = 0.97584 \tag{8.12}$$

从测试结果来看，似乎无法区分两种可能的输入，因为无论测试结果如何，人们携带正常基因的概率都是压倒性的。 换句话说，如果你得到一个阳性的测试结果，更有可能是由于测试错误而不是缺陷基因引起的。 似乎没有对没有家族史的人进行测试的有用目的。 (当然可以进行重复测试以减少假阳性率。)

信息分析清楚地显示了这两种情况之间的差异。 首先，回想一下概率是主观的，即依赖于观察者。 执行测试的实验技术人员可能不知道是否有家族史，因此无法从结果中推断出任何信息。 只有了解家族史的人才能做出有用的推断。 其次，计算两种情况下的信息流是有益的。 请记住，所有五个信息度量（I， L， M， N和 J）都依赖于输入概率。 对这两种情况进行直接计算，得到表8.1中的信息量（以比特为单位）（请注意，如果没有已知的家族史， N比 M大得多）。

	$p(A)$	$p(B)$	I	L	M	N	J
家族史	0.5	0.5	1.00000	0.11119	0.88881	0.11112	0.99993
未知的家族史	0.9995	0.0005	0.00620	0.00346	0.00274	0.14141	0.14416

表8.1：亨廷顿病测试过程模型特征

显然，这个测试通过减少疾病家族史的不确定性来传达关于患者状况的信息。 另一方面，如果没有家族史，几乎没有什么信息可以传达，因为最初的不确定性很小。

8.1.3 Berger的汉堡

一位曾经参加过6.050J/2.110J课程的学生开了一家快餐店，并以该课程的优秀本科助教的名字命名。 在Berger的汉堡店，餐点是用最先进的高科技设备准备的，使用可逆计算进行控制。 为了减少熵的产生，没有加热台，而是利用丢弃信息产生的熵来保持食物温暖。

由于计算中丢弃信息的速率是不可预测的，食物并不总是保持温暖。 对于不同的菜单项，存在一定的概率使餐点“COD”（送餐时冷）。

三个原始菜单项是套餐1、2和3。 套餐1（汉堡）售价1美元，含有1000卡路里，有0.5的概率会冷。 套餐2（鸡肉）售价2美元，含有600卡路里，有0.2的概率会冷。 套餐3（鱼）售价3美元，含有400卡路里，有0.1的概率会冷。

项目	主菜	成本	卡路里	到达热的概率	到达冷的概率
					到达冷的概率
套餐1	汉堡	1.00美元	1000	0.5	0.5
套餐2	鸡肉	2.00美元	600	0.8	0.2
套餐3	鱼	$3.00	400	0.9	0.1

表8.2：伯格的汉堡

关于伯格的汉堡可以提出几个推理问题。所有这些问题都需要对公众的购买习惯进行初始假设，即对每种餐的概率进行估计 $p(B), p(C)$和 $p(F)$。 然后，当得知另一个事实，例如某个顾客的餐到达时是冷的，这些概率可以被修正，从而得到对所点餐的更好估计。

假设你和朋友们来到伯格汉堡店并点了你们的餐点。 假设金钱充裕，你和朋友们点任何一种三种餐点的可能性都相等。 还假设你没有听到朋友们点了什么，也没有看到他们付了多少钱。还假设你不知道朋友们的口味偏好，而且餐点都是相同包装，所以你无法通过外观判断别人点了什么。

在餐点送达之前，你对朋友们点了什么一无所知，可能会假设每种餐点的概率都是$1/3$。 你可以估计每份餐点的平均支付金额（$2.00），平均卡路里含量（667卡路里），以及任意订单为货到付款的概率（0.267）。

现在假设你的朋友艾丽斯说她的餐点很冷。 在知道这个信息的情况下，她点汉堡的概率是多少？（0.625）鸡肉？（0.25）鱼肉？（0.125）她支付的餐费的期望值是多少？（$1.50）她的卡路里摄入量的期望值是多少？（825卡路里）

接下来假设你的朋友鲍勃说他为她感到遗憾，并给她一些他的热食。
直接应用上述公式可以确定他点了什么，
以及预期的卡路里和成本。

8.1.4 推理策略

通常，仅计算各种可能输入事件的概率是不够的。 系统的正确运行可能需要明确选择一个输入事件。 对于没有损失的过程，可以准确地进行选择。 然而，对于有损失的过程，必须使用一些策略将概率转换为单一选择。

一种简单的策略是“最大似然”，即在输出事件已知后选择具有最高概率的输入事件。 对于许多应用，特别是具有小误差的通信，这是一个好策略。 当两个输入概率相等时，它适用于对称二进制信道。然而，有时它根本不起作用。例如，如果用于没有家族史的亨廷顿病测试，无论测试结果如何，这种策略都不会说这个人有缺陷基因。

推理在许多感兴趣的领域中都很重要，例如机器学习、自然语言处理和其他人工智能领域。 一个当前研究兴趣的开放问题是哪种推理策略最适合特定目的。

8.2 最大熵原理：简单形式

在上一节中，我们讨论了一种估计过程的输入概率的技术，假设已知输出事件。 这种技术依赖于贝叶斯定理的使用，只有在过程是无损的情况下才有效（在这种情况下，输入可以被确定）或者假设了初始输入概率分布（在这种情况下，它会被修正以考虑已知的输出）。

最大熵原理是一种可以更普遍地估计输入概率的技术。 结果是一个与已知约束一致的概率分布，以一个或多个数量的平均值或期望值来表示，但在其他方面尽可能无偏（这里的“偏见”一词不是指统计学上的技术意义，而是指抑制公正判断的日常意义上的偏好）。 首先，这个原理是针对一个约束和三个输入事件的简单情况进行描述的，在这种情况下，可以进行解析地进行技术实施。 然后，在第9章中更一般地进行描述。

这个原则在许多领域都有应用，但最初是受统计物理学的启发，试图将物理系统的宏观可测量性质与原子或分子水平的描述联系起来。 它可以用信息论的观点来接近物理系统，因为概率分布可以通过避免观察者拥有比实际可用信息更多的假设来推导出来。 信息论，特别是以概率分布的定义为基础的信息定义，提供了一个量化的无知（或不确定性或熵）的度量，可以在数学上最大化，以找到最好地避免不必要假设的概率分布。

这种统计物理学的方法是由爱德温·T·杰恩斯（1922-1998）开创的，他是华盛顿大学和之前的斯坦福大学的教授。这个开创性的出版物是

- E. T. 杰恩斯，“信息论与统计力学”，物理评论，卷106，第4期，pp. 620-630；1957年5月15日。

  (http://bayes.wustl.edu/etj/articles/theory.1.pdf)

Jaynes的其他相关参考资料包括：

- 这篇论文的延续，E.T.Jaynes，“信息论与统计力学。 II，”物理评论，第108卷，第2期，第171-190页；1957年10月15日。
  (http://bayes.wustl.edu/etj/articles/theory.1.pdf)
- 一篇综述论文，包括一个关于估计不公平骰子概率的例子，E.T.Jaynes，“信息论与统计力学”，在”统计物理学”中的第181-218页，Brandies夏季学院1962年，W.A.Benjamin，Inc.，纽约，纽约；1963年。
  (http://bayes.wustl.edu/etj/articles/brandeis.pdf)
- 这种方法的个人历史，Edwin T. Jaynes，“我们在最大熵问题上的立场在哪里？”，在”最大熵形式主义”中的第15-118页，Raphael D. Levine和Myron Tribus，编辑，MIT出版社，马萨诸塞州剑桥市；1979年。
  (http://bayes.wustl.edu/etj/articles/stand.on.entropy.pdf)

将最大不确定性作为热力学方法的哲学思想在这里进行了讨论

- M. Tribus的《热力学和热力学》第3章，D. Van Nostrand Co, Inc., Princeton, NJ; 1961.

在使用最大熵原理之前，需要建立问题领域。对于涉及物理系统的情况，这意味着需要确定系统可能存在的各种状态，并了解约束条件中涉及的所有参数。 例如，假设已知与每个状态相关的能量、电荷和其他量。 通常需要使用量子力学来完成这个任务。 在这一步骤中，并不假设系统处于特定的状态中（或者经常

表达的是，实际上哪个状态是“占据”的); 实际上我们假设我们不知道也无法确定这一点，因此我们处理的是每个状态被占据的概率。因此，我们使用概率作为应对我们不完全了解的手段。自然而然，我们希望避免无意中假设我们拥有更多的知识，而最大熵原理就是做到这一点的技术。在应用于非物理系统时，各种事件（可能的结果）必须被确定，并且与每个事件相关联的各种数值属性也必须被确定。在这些笔记中，我们将推导出最大熵原理的一个简单形式，并将其应用于第8.1.3节中设置的餐厅示例。

8.2.1 Berger的汉堡

通过一个例子介绍最大熵原理。 这个例子在第8.1.3节中有描述。快餐店Berger的汉堡提供三种餐：汉堡、鸡肉和鱼肉。 每种餐的价格、卡路里含量和送餐冷却的概率如表8.2所示。

8.2.2 概率

这个例子被定义为三种餐之一的选择构成了一个结果。 如果我们不知道这个结果，我们仍然可能有一些知识，我们使用概率来表达这个知识。 问题是如何分配与我们可能拥有的任何信息一致的概率。

在Berger的汉堡的情况下，有三个概率，为简单起见，我们用B)、C)和F)表示这三种餐。 概率分布A_i)具有每个概率在0和1之间或等于0和1的属性，并且由于输入事件是互斥且穷尽的，所有概率的总和为1：

$$\begin{aligned} 1 &= \sum_i p(A_i) \\ &= p(B) + p(C) + p(F) \end{aligned} \tag{8.13}$$

如果任何一个概率等于1，那么其他所有概率都为0，我们就能准确地知道系统所处的状态；换句话说，我们没有不确定性，也不需要使用概率。

8.2.3 熵

更一般地说，我们对所选择的餐点或所占据的状态的不确定性是以我们所不知道的信息的量来量化的。这是

$$\begin{aligned} S &= \sum_i p(A_i) \log_2 \frac{1}{p(A_i)} \\ &= p(B) \log_2 \frac{1}{p(B)} + p(C) \log_2 \frac{1}{p(C)} + p(F) \log_2 \frac{1}{p(F)} \end{aligned} \tag{8.14}$$

在这里，信息的度量单位是比特，因为我们使用以2为底的对数。

在物理系统的背景下，这种不确定性被称为熵。 在通信系统中，关于要传输的实际消息是哪个的不确定性也被称为源的熵。 请注意，一般情况下，熵因为是以概率为基础而依赖于观察者。 一个人对系统的了解可能与另一个人不同，因此对熵的数值计算也会不同。 最大熵原理用于发现导致这种不确定性最大值的概率分布，从而确保没有假设任何信息。 得到的概率分布不依赖于观察者。

8.2.4 约束

熵公式的一个特性是，当所有概率相等时，它达到最大值（我们假设可能状态的数量是有限的）。 这个特性可以很容易地通过使用吉布斯不等式来证明。 如果我们对系统没有额外的信息，那么这样的结果似乎是合理的。
然而，如果我们有额外的信息，那么我们应该能够找到一个概率分布，它在某种意义上比较不确定性更小。

为了简单起见，我们只考虑一个这样的约束，即我们知道某个量的期望值（最大熵原理可以处理多个约束，但数学过程和公式会变得更加复杂）。 所讨论的量是一个每个系统状态都有自己数量的属性，并且通过考虑这些状态的概率来计算期望值。 因此，如果有一个属性，每个状态都有一个值 $g(A_i)$，并且我们知道实际值 G，那么我们应该只考虑那些期望值等于 G的概率分布。

$$G = \sum_i p(A_i)g(A_i) \tag{8.15}$$

请注意，如果 G小于最小的 $g(A_i)$或大于最大的$g(A_i)$，则无法满足此约束。

对于我们的$\mathrm{Berger's\ Burgers}$示例，假设我们被告知一顿饭的平均价格是$\$1.75$，并且我们想要估计各种餐点的单独概率，而不做任何其他假设。
那么我们的约束条件将是

$$\$1.75 = \$1.00p(B) + \$2.00p(C) + \$3.00p(F) \tag{8.16}$$

请注意，概率是无量纲的，因此约束的期望值和各个值必须用相同的单位表示，本例中为美元。

8.2.5 最大熵，解析形式

在这里，我们演示了最大熵原理的简单情况，其中有一个约束和三个变量。 可以通过解析的方式进行所有步骤。

假设你已经被卡尼沃尔公司聘请，该公司是伯格汉堡的母公司，以分析他们的全球销售情况。 你访问了世界各地的伯格汉堡餐厅，并确定平均每个人支付1.75美元的餐费。 （作为卡尼沃尔公司对全球一致性的承诺，每个餐厅的餐费在将当地货币转换为美元后完全相同。）

当你回来后，你的主管们询问顾客点餐三种套餐的概率。 换句话说，他们想知道B的概率$\mathsf{P}(B)$，C的概率$\mathsf{P}(C)$，和F的概率$\mathsf{P}(F)$。 你惊恐地意识到你没有保留原始数据，也没有时间重复你的旅行。 你必须根据你所知道的两件事情，尽力估计B的概率$\mathsf{P}(\mathsf{B})$，C的概率$\mathsf{P}(\mathsf{C})$，和F的概率$\mathsf{P}(\mathsf{F})$。

$$1 = p(B) + p(C) + p(F) \tag{8.17}$$

$$\$1.75 = \$1.00p(B) + \$2.00p(C) + \$3.00p(F) \tag{8.18}$$

由于你有三个未知数和只有两个方程，没有足够的信息来解出未知数。

你应该怎么办？ 有一系列的概率值与你所知的一致。 然而，这些会给你留下不同程度的不确定性 S

$$S = p(B)\ \log_2 \frac{1}{p(B)} + p(C)\ \log_2 \frac{1}{p(C)} + p(F)\ \log_2 \frac{1}{p(F)} \tag{8.19}$$

如果你选择一个 S很小的值，那么你就是在假设你不知道的东西。 例如，如果你的平均值是$2.00而不是$1.75，你可以假设每个人都买了鸡肉餐来满足你的两个约束条件。那么你的不确定性将为0比特。 或者你可以假设一半的订单是汉堡，一半是鱼，那么不确定性将为1比特。这些假设都不是特别合适，因为它们超出了你所知道的范围。 你如何找到那个概率分布，它不需要除了你已经知道的信息之外的任何进一步假设?

最大熵原理是基于一个合理的假设，即在满足约束条件的情况下，应选择使剩余不确定性最大化（即最大熵）的概率分布。 这样，你在计算中没有引入任何额外的假设。

对于三个概率和两个约束条件的简单情况，可以通过解析方法轻松实现。 通过处理这两个约束条件，可以用第三个未知概率来表示其中的两个概率。 对于我们的情况，我们可以将上述第8.17式乘以1.00并从第8.18式中减去，以消除p（B）。
然后，我们可以将第一个式子乘以2.00并从第二个式子中减去，从而消除p（C）：

$$p\ (C)\ = 0.75 - 2p\ (F) \qquad p\ (B\ )\ = 0.25 + p\ (F) \tag{8.21}$$

接下来，可以确定概率的可能值范围。 由于这三个值都在0和1之间，因此可以从这些结果中得出结论

$$0 \leq p(F) \leq 0.375 \tag{8.22}$$
$$0 \leq p(C) \leq 0.75 \tag{8.23}$$
$$0.25 \leq p(B) \leq 0.625 \tag{8.24}$$

接下来，可以将这些表达式代入熵的公式中，以便用单个概率表示。因此

$$S = (0.25 + p(F))\ \log_2 \sum \frac{1}{0.25 + p(F)} \sum + (0.75 - 2p(F))\ \log_2 \sum \frac{1}{0.75 - 2p(F)} \sum + p(F)\ \log_2 \sum \frac{1}{p(F)} \sum \tag{8.25}$$

现在可以使用几种技术来找到使得 S最大的F的值为 $p(F)$。 在这种情况下，最大值出现在 $p(\ \mathsf{F}) = 0.216$，因此 $p(B) = 0.466$， $p(C) = 0.318$，且 S= 1.517 位。
在估计输入概率分布之后，可以估计该分布上的任何平均值。
例如，在这种情况下，可以计算平均卡路里数（为743.2卡路里），或者餐点被冷藏的概率（31.8%）。

8.2.6 总结

让我们回顾一下我们所做的事情。 我们已经用未知的概率分布来表达我们的约束条件。 其中一个约束条件是概率的总和为1。 另一个约束条件涉及某个量的平均值，这里是成本。 我们使用这些约束条件来消除两个变量。 然后，我们用剩下的变量来表示熵。 最后，我们找到了使熵最大的剩余变量的值。 结果是一个与约束条件一致但具有最大不确定性的概率分布。 因此，我们没有无意中引入任何不需要的假设到概率估计中。

这种技术要求在开始时就了解系统的模型；唯一不知道的是概率分布。 如本节所述，当未知数较少且比约束多一个时，可以进行解析推导。 对于更复杂的情况，需要采用更一般的方法。 这是第9章的主题。

第9章

最大熵原理

第8.2节介绍了一种估计过程输入概率的技术，这些概率是无偏的，但与已知约束一致，这些约束以一个或多个数量的平均值或期望值的形式表达。 这种技术，即最大熵原理，是在那里针对一个约束和三个输入事件的简单情况下开发的，在这种情况下，可以进行解析推导。 在这里，将描述更一般的情况。

9.1问题设置

在使用最大熵原理之前，需要设置问题域。在涉及物理系统的情况下，这意味着需要确定系统可能存在的各种状态，并了解与约束相关的所有参数。 例如，假定已知与每个量子态相关的能量、电荷和其他数量。 在此步骤中，并不假设系统实际上处于哪个特定状态（哪个状态是“占据的”）。 实际上，我们假设我们永远无法确定这一点，因此我们处理的是每个状态被占据的概率。 在应用于非物理系统时，需要列举各种可能的事件，并确定每个事件的属性，特别是与每个约束相关的值。 在本章中，我们将将这一一般数学推导应用于两个示例，一个是商业模型，另一个是物理系统模型（都非常简单和粗糙）。

9.1.1 Berger的汉堡

这个例子在第8章中用于处理推理和最大熵原理的解析形式。一家快餐店提供三种餐：汉堡、鸡肉和鱼。 现在我们假设菜单已经扩展，包括一份美食低脂豆腐餐。表9.1列出了每种餐的价格、卡路里含量和送餐冷却的概率。

9.1.2 磁偶极模型

一组磁偶极子（将它们视为微小的磁铁）受到外加磁场 H的作用，因此系统的能量取决于它们的方向和外加磁场。 为了简单起见，我们的系统只包含一个这样的偶极子，它不时能够与两个环境中的任何一个交换信息和能量，这两个环境是更大的偶极子集合。

项目		主菜	成本 卡路里	概率 到达冷的概率	到达热的概率
餐	1	汉堡 1.00美元	1000	0.5	0.5
餐	2	鸡肉 2.00美元	600	0.8	0.2
餐	3	鱼 $3.00	400	0.9	0.1
餐	4	豆腐 $8.00	200	0.6	0.4

表9.1：伯格的汉堡

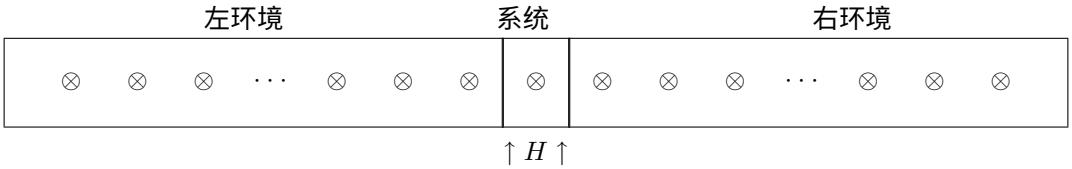


图9.1：偶极矩示例
（每个偶极矩可以是向上或向下）

系统和其两个环境中的偶极矩都可以是“向上”或“向下”。系统只有一个偶极矩，因此它只有两个状态，对应于该偶极矩的两个状态，“向上”和“向下”（如果系统有n个偶极矩，则它将有$2n$个状态）。 每个偶极矩的能量与施加的场成正比，并且取决于其方向；系统的能量是系统中所有偶极矩的能量之和，在我们的例子中只有一个这样的偶极矩。

状态	对齐	能量
上	向上	$-m_dH$
下	向下	m_dH

表9.2：磁偶极矩

常数 m_d以每特斯拉焦耳为单位表示，其值取决于特定偶极的物理性质。 例如，偶极可能是电子自旋，此时 $m_d = 2\mu_B\mu_0$其中 $\mu_0 = 4\pi \times 10^{-7}$亨利每米（在有理化的MKS单位中）是自由空间的磁导率， $\mu_B = \hbar e/2m_e = 9.272 \times 10^{-24}$特斯拉焦耳是玻尔磁子， $\hbar = h/2\pi$， $h = 6.626 \times 10^{-34}$焦耳秒是普朗克常数， $e = 1.602 \times 10^{-19}$库仑是电子的电荷大小，而 $m_e = 9.109 \times 10^{-31}$千克是电子的静止质量。

在图9.1中，系统显示在两个环境之间，并且环境和系统之间有障碍物（用垂直线表示），防止相互作用（稍后我们将移除障碍物以允许相互作用）。 系统和环境中的偶极子用符号 $\otimes$表示，可以是自旋向上或自旋向下。 所示的磁场仅作用于系统，而不作用于环境。

只有一个偶极子的模型的优点是足够简单，可以轻松进行计算。 当然，这样的模型过于简单，不能指望得到数值精确的结果。 一个更现实的模型将需要如此多的偶极子和状态，以至于无法对集合进行实际计算。 例如，化学元素的一个摩尔在日常标准下是一个小量，但它包含阿伏伽德罗常数 $N_A = 6.02252 \times 10^{23}$个原子，以及相应数量的电子自旋；可能的状态数量将是2的该幂。 通过注意到地球上的原子数量不超过2^{170}个，可知这个数字有多大，而可见宇宙中大约有2^{265}个原子；这两个数字都远小于该模型中的状态数量。 即使我们采用一个更小的样本进行计算，比如

进行200次旋转，并希望用计算机表示每个状态的概率（每个状态仅使用8位），我们仍然需要比地球上的原子数量更多的字节内存。 显然，计算如此多的状态是不可能的，因此这些笔记中描述的技术无法详细实施。
尽管如此，我们仍然能够得出某些结论和一般关系。

9.2 概率

虽然问题已经设定，但我们不知道系统实际处于哪个状态。为了表达我们尽管存在这种无知或不确定性，我们假设每个可能的状态 A_i都有一定的占用概率 $p(A_i)$，其中 i是在可能的状态范围内的索引。 概率分布 $p(A_i)$具有以下特性：每个概率值介于0和1之间（可能等于0或1），并且（由于输入事件是互斥且穷尽的）所有概率的总和为1：

$$1 = \sum_i p(A_i) \tag{9.1}$$

正如之前提到的，由于他们的不同知识，两个观察者可能使用不同的概率分布。 换句话说，概率以及所有基于概率的量都是主观的，或者依赖于观察者。 下面的推导可以适用于任何观察者。

9.3 熵

我们对所占据状态的信息不确定性以数量化的方式来表达，这些信息是我们不知道的。

$$S = \sum_i p(A_i) \log_2 \frac{1}{p(A_i)} \tag{9.2}$$

信息以比特为单位进行测量，这是因为在方程式9.2中使用了以2为底的对数。在处理具有大量状态和因此熵为非常大比特数的真实物理系统时，方便地将上述求和乘以玻尔兹曼常数$k_B = 1.381 \times 10^{-23}$焦耳/开尔文，并且使用自然对数而不是以2为底的对数。

然后 S将以每开尔文焦耳的形式表示：

$$S = k_B \sum_i p(A_i) \ln \frac{1}{p(A_i)} \tag{9.3}$$

在物理系统和通信系统的背景下，不确定性被称为熵。 请注意，由于熵是以概率的形式表示的，它也取决于观察者，因此对于系统有不同了解的两个人会计算出不同的熵值。

9.4 约束条件

当所有概率相等时，熵达到最大值（我们假设可能的状态数是有限的），熵的结果值是状态数的对数，可能还有一个像 k_B这样的比例因子。 如果我们对系统没有额外的信息，那么这样的结果似乎是合理的。然而，如果我们有额外的约束条件的信息，那么假设概率相等可能与这些约束条件不一致。 我们的目标是找到具有最大不确定性的概率分布，从而尽可能地无偏。

为了简单起见，我们在这里只考虑一个这样的约束。 我们假设我们知道某个量的期望值（最大熵原理可以处理多个约束，但数学上的程序和公式更加复杂）。我们所关心的量是每个系统状态都有自己的数量，并且期望值是通过对每个状态对应的值进行平均，考虑到这些状态的概率来计算的。因此，如果有一个量G，每个状态都有一个值g（A i），那么我们只想考虑那些期望值是已知值G的概率分布。

程序和公式更加复杂）。我们所关心的量是每个系统状态都有自己的数量，并且期望值是通过对每个状态对应的值进行平均，考虑到这些状态的概率来计算的。因此，如果有一个量 G，每个状态都有一个值 g（A_i），那么我们只想考虑那些期望值是已知值 $\widetilde{G}$的概率分布。

$$\widetilde{G} = \sum_i p(A_i)g(A_i) \tag{9.4}$$

当然，如果 $\widetilde{G}$小于最小值 g（A_i）或大于最大值g（A_i），则无法满足此约束。

9.4.1 示例

对于我们的Berger's Burgers的例子，假设我们被告知一顿饭的平均价格是2.50美元，并且我们想要在不做任何其他假设的情况下估计各种餐点的概率。
那么我们的约束条件将是

$$2.50 \;=\; 1.00\mathsf{P}(B)+2.00\mathsf{P}(C)+3.00\mathsf{P}(F)+8.00\mathsf{P}(T) \tag{9.5}$$

对于我们的磁偶极子的例子，假设状态 U和 D的能量分别用 $E(i)$表示，其中 i可以是 U或 D，并且假设能量的期望值已知为某个值$\widetilde{\mathsf{E}}$。 所有这些能量都用焦耳表示。然后

$$\widetilde{E} = \mathsf{E}(U)\mathsf{P}(U) + \mathsf{E}(D)\mathsf{P}(D) \tag{9.6}$$

能量U和D的值取决于外加磁场 H。 这个参数在推导过程中将起到重要作用。 如果在这里使用表9.2中的公式i，

$$\widetilde{E} = m_d H[p(D) - p(U)] \tag{9.7}$$

9.5 最大熵，解析形式

最大熵原理的基础是，在估计概率分布时，应选择使剩余不确定性最大（即最大熵）且与约束条件一致的分布。 这样，您在计算中没有引入任何额外的假设或偏见。

这个原理在第8章中用于三个概率和一个约束条件的简单情况。 熵可以通过解析方法最大化。 利用约束条件和概率之和为1的事实，我们用第三个概率表示了另外两个未知概率。

接下来，通过每个概率值都介于0和1之间这一事实，确定了可能的概率值范围。 然后，将这些表达式代入熵的公式 S中，以便用单个概率表示。 然后可以使用几种技术之一来找到使 S最大的概率值。

这种分析技术不适用于具有超过三个可能状态和仅有一个约束条件的情况。 这种方法只有在约束条件可以用单个变量表示熵时才实用。 如果有四个未知数和两个方程，熵将保留为两个变量的函数，而不是一个变量。需要在平面上搜索其最大值。 也许这看起来可行，但如果有五个未知数呢？ （或者十个？）需要在三维（或八维）空间中进行搜索，这更加困难。

下一节将介绍一种不同的方法，适用于单个约束和多个概率。

9.6 最大熵，单个约束

假设某个与各个事件A_i相关联的值 $g(A_i)$ 的平均值已知，称之为 $\widetilde{G}$（这是约束条件）。 因此，有两个方程，一个来自于约束条件，另一个来自于概率之和等于1的事实：

$$1 = \sum_i p(A_i) \tag{9.8}$$

$$\widetilde{G} = \sum_i p(A_i) g(A_i) \tag{9.9}$$

其中 $\widetilde{G}$不能小于最小的 $g(A_i)$ 或大于最大的 $g(A_i)$。

与这个概率分布相关的熵是

$$S = \sum_i p(A_i) \log_2 \frac{1}{p(A_i)} \tag{9.10}$$

以比特表示。 在下面的推导中，将使用这个熵的公式。 它在具有少量状态的示例中效果很好。 在本笔记的后面章节中，我们将开始使用更常见的物理系统熵的表达方式，以开尔文每焦耳表示。

$$S = k_B \sum_i p(A_i) \ln \frac{1}{p(A_i)} \tag{9.11}$$

9.6.1 双变量

有时候，通过观察一个更一般的问题，可以澄清原问题。 在这种情况下，我们不再关注 G的特定值，而是关注所有可能的 G值，即 $g(A_i)$的最小值和最大值之间的范围。 因此， G成为一个变量而不是一个已知值（已知值将继续用 $\widetilde{G}$表示）。 然后，我们将引入一个新的双变量 β，并将所有感兴趣的量，包括 G，用 β表示。然后，原问题就简化为找到与已知的期望值 $\widetilde{G}$相对应的 β的值，即找到使 $G(\beta) = \widetilde{G}$的 β的值。这个新变量β被称为拉格朗日乘子，以法国数学家约瑟夫-路易・拉格朗日（1736-1813）的名字命名[1]。 拉格朗日使用这样的变量发展了一种通用技术，用于进行约束最大化，而我们当前的问题是一个非常简单的情况。我们不会使用拉格朗日乘子的数学技术——它比我们需要的更强大和更复杂。

这是我们将要做的替代方案。 我们将从答案开始，其他人使用拉格朗日乘数法推导出来的答案，并证明它是正确的。 也就是说，我们将给出一个概率分布的公式 $p(A_i)$以及 β和 $g(A_i)$参数，并证明从该分布计算出的熵$S(\beta)$至少与具有相同期望值的任何概率分布的熵一样大，即$G(\beta)$。 因此，使用 β自动地最大化了熵。 然后，我们将展示如何找到 β的值，从而间接地找到所有感兴趣的量，对于感兴趣的特定值 $\widetilde{G}$（这是可能的，因为 $G(\beta)$是 β的单调函数，因此可以使用零点查找技术计算其逆函数）。

9.6.2 概率公式

我们想要的概率分布 $p(A_i)$已经由他人推导出来。 它是双变量 β的函数：

$$p(A_i) = 2^{-\alpha} 2^{-\beta g(A_i)} \tag{9.12}$$

[1]在 http://www-groups.dcs.st-andrews.ac.uk/~history/Biographies/Lagrange.html上查看拉格朗日的传记

这意味着

$$\log_2 \frac{1}{p(A_i)} = \alpha + \beta g(A_i) \tag{9.13}$$

其中 α是 β的这个函数的方便缩写[2]：

$$\alpha = \log_2 \left(\sum_i 2^{-\beta g(A_i)} \right) \tag{9.14}$$

请注意，方程9.12中的 α的这个公式确保了方程9.8中的 $p(A_i)$的总和为1。

如果 β已知，函数 α和概率 $p(A_i)$可以找到，并且如果需要，可以计算熵 S和约束变量 G。 实际上，如果需要熵 S，可以直接计算，而不需要评估概率$p(A_i)$——这在处理几十个或更多概率时非常有帮助。 这个快捷方式是通过将方程9.13乘以概率 $p(A_i)$，并对 i求和得到的。 左边是熵 S，右边简化了，因为 α和 β与 i无关。结果是

$$S = \alpha + \beta G \tag{9.15}$$

其中 S、 α和 G都是 β的函数。

9.6.3 最大熵

很容易证明，从这个概率分布计算得到的熵至少与导致相同期望值 G的任何概率分布的熵一样大。

回想一下吉布斯不等式，方程 6.4，它将在这里被重写，其中 $p(A_i)$和 $p'(A_i)$互换位置（无论哪种方式都有效）：

$$\sum_i p'(A_i) \log_2 \frac{1}{p'(A_i)} \leq \sum_i p'(A_i) \log_2 \frac{1}{p(A_i)} \tag{9.16}$$

其中 $p'(A_i)$是任意概率分布， $p(A_i)$是任意其他概率分布。 当且仅当两个概率分布相同时，不等式成立。

吉布斯不等式可以用来证明方程9.12的概率分布具有最大熵。 假设存在另一个概率分布 $p'(A_i)$，它导致期望值为G'和熵为 S'，即

$$1 = \sum_i p'(A_i) \tag{9.17}$$

$$G' = \sum_i p'(A_i) g(A_i) \tag{9.18}$$

$$S' = \sum_i p'(A_i) \log_2 \frac{1}{p'(A_i)} \tag{9.19}$$

然后很容易证明，对于任意的 β值，如果 $G' = G(\beta)$，那么 $S' \leq S(\beta)$：

[2]函数 $\alpha(\beta)$ 与统计物理学的配分函数 $Z(\beta)$ 相关： $Z = 2^{\alpha}$或 $\alpha = \log_2 Z$。

$$
\begin{aligned}
S' &= \sum_i p'(A_i)\,\log_2 \frac{1}{p'(A_i)} \\
&\le \sum_i p'(A_i)\,\log_2 \frac{1}{p(A_i)} \\
&= \sum_i p'(A_i)[\alpha + \beta g(A_i)] \\
&= \alpha + \beta G' \\
&= S(\beta) + \beta[G' - G(\beta)] \qquad (9.20)
\end{aligned}
$$

在这里使用了方程9.16、9.13、9.17、9.18和9.15。 因此，与任何导致约束变量取相同值的备选概率分布相关的熵都不能超过使用 β的分布的熵。

9.6.4 评估对偶变量

到目前为止，我们认为对偶变量 β是一个独立变量。 如果我们从一个已知的值 $\widetilde{G}$开始，我们希望将 G 作为一个独立变量，并计算 β的值。换句话说，我们需要反转函数 $G(\beta)$，或者找到满足方程9.9的 β的值。

这个任务并不简单；事实上，与最大熵原理相关的大部分计算困难都在这一步。 如果有适度数量的状态和除了涉及概率之和的方程之外只有一个约束条件，这一步并不困难，正如我们将看到的那样。如果有更多的约束条件，这一步将变得越来越复杂，如果有大量的状态，计算将无法完成。 对于更现实的物理系统模型，这个求和是不可能计算的，尽管除了A_i的概率之外的一般关系仍然成立。

要找到 β，从方程9.12开始计算 $p(A_i)$，将其乘以 $g(A_i)$和2^{α}，并对概率求和。 左边变成 $G(\beta)2^{\alpha}$，因为 α和 $G(\beta)$都不依赖于 i。 我们已经有了一个关于 β的表达式（方程9.14），所以左边变成 $\sum_i G(\beta)2^{-\beta g(A_i)}$。右边变成 $\sum_i g(A_i)2^{-\beta g(A_i)}$。因此，

$$0 = \sum_i [g(A_i) - G(\beta)]2^{-\beta g(A_i)} \qquad (9.21)$$

如果这个方程乘以 $2^{\beta G(\beta)}$，结果就是

$$0 = f(\beta) \qquad (9.22)$$

其中函数 $f(\beta)$ 是

$$f(\beta) = \sum_i [g(A_i) - G(\beta)]2^{-\beta[g(A_i) - G(\beta)]} \qquad (9.23)$$

方程9.22是要根据特定的 $G(\beta)$ 值来解决的基本方程，例如$\widetilde{G}$. 函数 $f(\beta)$ 取决于问题的模型（即各种 $g(A_i)$），以及 $\widetilde{G}$，仅此而已。它不明确依赖于 α或概率 $p(A_i)$。

我们如何知道是否存在某个 β值使得 $f(\beta) = 0$？ 首先，注意到由于 $\widetilde{G}$位于最小和最大 $g(A_i)$之间，至少存在一个 i使得（$g(A_i) - \widetilde{G}$）为正，至少存在一个使其为负。 很容易证明 $f(\beta)$ 是 β的单调函数，即如果$\beta_2 > \beta_1$，则 $f(\beta_2) < f(\beta_1)$。 对于较大的正值 β在总和中，占主导地位的项是具有最小值的项（$g(A_i)$），因此f为负。 同样，对于较大的负值β，f为正。 因此，它必须为一个且仅为一个值为零的值。 β （这种推理依赖于$f(\beta)$是一个连续函数的事实。）

9.6.5 示例

对于Berger's Burgers的例子，假设你被告知平均餐价为$2.50，并且你想估计概率$p(B)$，$p(C)$，$p(F)$和$p(T)$。以下是你所知道的：

$$1 = p(B) + p(C) + p(F) + p(T) \tag{9.24}$$

$$0 = \$1.00p(B) + \$2.00p(C) + \$3.00p(F) + \$8.00p(T) \text{ - } \$2.50 \tag{9.25}$$

$$S = p(B)\log_2\frac{1}{p(B)} + p(C)\log_2\frac{1}{p(C)} + p(F)\log_2\frac{1}{p(F)} + p(T)\log_2\frac{1}{p(T)} \tag{9.26}$$

熵在受限制条件下最大

$$p(B) = 2^{-\alpha}2^{-\beta\$1.00} \tag{9.27}$$
$$p(C) = 2^{-\alpha}2^{-\beta\$2.00} \tag{9.28}$$
$$p(F) = 2^{-\alpha}2^{-\beta\$3.00} \tag{9.29}$$
$$p(T) = 2^{-\alpha}2^{-\beta\$8.00} \tag{9.30}$$

其中

$$\alpha = \log_2(2^{-\beta\$1.00} + 2^{-\beta\$2.00} + 2^{-\beta\$3.00} + 2^{-\beta\$8.00}) \tag{9.31}$$

且 β是使得 $f(\beta) = 0$ 的值，其中

$$f(\beta) = \$0.50 \times 2^{-\$0.50\beta} + \$5.50 \times 2^{-\$5.50\beta} - \$1.50 \times 2^{\$1.50\beta} - \$0.50 \times 2^{\$0.50\beta} \tag{9.32}$$

一点

试错（或使用零点查找程序）给出 $\beta = 0.2586$ 比特/美元， $\alpha = 1.2371$ 比特，$p(B) = 0.3546$， $p(C) = 0.2964$， $p(F) = 0.2478$， $p(T) = 0.1011$，且 $S = 1.8835$ 比特。 熵比使用固定长度编码来编码四种可能餐点之一的订单所需的2比特要小。 这是因为对平均价格的了解会在一定程度上减少我们的不确定性。 如果对订单有更多的信息了解，那么结合该信息的概率分布将具有更低的熵。

对于磁偶极子的例子，我们将导出过程中磁场 H设置为某个未指定的值。 所有结果都取决于 H以及 E。

$$1 = p(U) + p(D) \tag{9.33}$$

$$\begin{aligned}\widetilde{E} &= e(U)p(U) + e(D)p(D) \\ &= m_d H[p(U) - p(D)]\end{aligned} \tag{9.34}$$

$$S = p(U)\log_2\frac{1}{p(A)} + p(D)\log_2\frac{1}{p(D)} \tag{9.35}$$

熵是最大的，对于能量 $\widetilde{E}$和磁场 H，如果

$$p(U) = 2^{-\alpha}2^{-\beta m_d H} \tag{9.36}$$
$$p(D) = 2^{-\alpha}2^{\beta m_d H} \tag{9.37}$$

其中

$$\alpha = \log_2(2^{-\beta m_d H} + 2^{\beta m_d H}) \tag{9.38}$$

并且 β是使 $f(\beta) = 0$ 的值，其中

$$f(\beta) = (m_d H - \widetilde{E})2^{-\beta(m_d H - \widetilde{E})} - (m_d H + \widetilde{E})2^{\beta(m_d H + \widetilde{E})} \tag{9.39}$$

请注意，这个只有一个偶极子的例子，因此只有两个状态，实际上不需要最大熵原理，因为有两个未知数的两个方程， $p(U)$ 和 $p(D)$（你可以用代数解方程9.39来求 β）。如果有两个偶极子，就会有四个状态，代数就不足够了。如果有更多的可能状态，计算β的方法将变得不切实际或者至少非常困难。因此，在这些笔记的第11章中，我们问，即使我们实际上不能使用状态求和来计算它们的数值，我们对各种数量能告诉我们什么。

第10章

物理系统

到目前为止，我们通过处理抽象的概念，如信息，忽略了物理系统的大部分方面。 虽然我们假设存储或传输的每个比特都体现在某个物理对象中，但我们专注于抽象的比特，并忽略了物理定律所造成的任何限制。 这是信息时代的基本口号。

过去并不总是这样，未来也不会如此。 在过去的几个世纪中，信息的物质表现形式因其巨大的成本而非常重要。 为了保存或传递信息，必须编写书籍，甚至将文字刻在石头上。 例如，想象一下在中世纪创作一本手稿的过程。 页面被费力地复制和插图。

今天，人们对这些作品的艺术性和文化重要性赞赏有加，部分原因是它们的制作成本非常高昂。社会只能处理它认为最重要的信息，而精美艺术品的成本与其他生产成本相比并不高。

多年来的进步提高了信息存储和传输的效率——想想印刷机、电报、电话、无线电、电视、数字信号处理、半导体和光纤。 这些技术导致了诸如计算机和数据网络之类的复杂系统，并塑造了娱乐业等公司用于创建和分发信息密集型产品的方法。 随着处理和分发数据的成本降低，考虑到该成本与创建、维护和使用信息的成本相比较小的情况是相关的。 正是在这个领域，信息论、比特、编码以及计算机科学的所有抽象思想占主导地位。 现代社会的各个领域都在应对日益增长的信息量。 在信息处理的经济学变化下，知识产权、版权、专利和商业秘密的基本概念正在被重新思考。 欢迎来到信息时代。

将信息模型与其物理体现分离当然是对现实的一种近似。 随着我们将微电子系统变得越来越复杂，使用越来越小的组件，我们将需要面对的根本限制不是我们制造小结构的能力，而是物理定律。 所有物理系统都受到量子力学的控制。

量子力学通常被认为只对小结构（比如原子大小）很重要。 尽管在那个长度尺度上是不可避免的，但它也统治着日常物体。 在处理物理系统中的信息处理时，考虑到既有少量信息位的非常小系统，也有大量信息的大系统是相关的。

到目前为止，我们所使用的关键思想需要在量子力学重要的范围内重新解释。

•数字抽象由可以恢复带有小扰动的数据的设备实现

•在面对不确定性时，使用概率来表达我们的知识

•最大熵原理作为一种无偏估计概率的技术

10.1 量子力学的性质

量子力学很奇怪。 似乎没有办法让它看起来不奇怪。 它的许多预测与我们通常遇到的物体大小的日常经验不符。

量子力学对于非常优秀的物理学家来说也是神秘的。 其方程和技术的基本哲学和解释是有争议的。

量子力学很难使用。需要相对高级的数学技能。 基本方程虽然是线性的，但是是一个偏微分方程，除了在很少的简单情况下无法解析求解。 通常需要数值解。

像其他物理理论一样，量子力学在建模和数学方面都需要技巧和判断力。 一般在研究生或高级本科水平之前不会深入教授。
量子力学有不同的形式。 它有许多替代的表述。 一般来说，这些表述在预测实验结果方面是等效的，但在学习或特定目的上并不同样容易。

鉴于这些缺点，为什么量子力学很重要？因为它有效。 它是唯一一个在如此广泛的情况下有效的基本物理理论。 它的预测一次又一次地通过实验证实。 它适用于日常大小的物体，以及天文物体（尽管通常对它们不是必需的）。 它适用于原子大小的物体，电磁波和亚原子物体。 有一个与特殊相对论理论兼容的版本。 目前唯一处理不好的物理现象是重力；量子力学尚未扩展为与广义相对论理论兼容。

在这些笔记中，我们无法深入讨论量子力学。 为了研究物理系统中的信息处理，我们只需要了解一些必备的一般特征。 特别地，我们需要一个物理系统的模型，其中有许多可能的状态，每个状态都有自己的概率成为系统实际所处的状态（即“占据”状态）。 这些状态都有与之相关的物理属性，其中能量就是其中之一。 量子力学证明了这个模型的合理性。

我们将在两种情况下使用这个模型。 第一种情况（如下所示）是具有许多状态的情况，目标是理解与这些状态的占用相关的信息如何影响能量的流动。
第二种情况（在这些笔记的后面章节中）是具有非常少状态的情况，信息是通过这些状态的占用来表示，目标是理解量子力学所提供的信息处理的限制和机会。

下面的两个章节，第10.2节“量子力学简介”和第10.3节“稳态”，可以被准备接受状态模型而不需要解释的读者跳过。 他们可以直接跳到第10.4节“多状态模型”。其他读者可以从接下来的两个章节中了解到量子考虑如何导致该模型，并在此过程中可能会发现量子力学的某些方面不再神秘。

10.2 量子力学导论

也许关于一个物体的第一个问题是：“它在哪里？”在日常经验中，可以通过测量仪器的质量来精确回答这个问题。
然而，在非常小的物体领域，存在一些基本限制，必须使用量子力学来回答这个问题。

量子力学的核心是能量。 由于质量和能量的等价性（记住爱因斯坦的著名公式 $E = mc^2$其中 c是光速，2.998×10^8米每秒），量子力学也涉及具有质量的粒子。 由于光子的能量与其频率之间的关系（$E = hf$其中 h是普朗克常数，6.626×10^{-34}焦耳秒），量子力学涉及光子。

根据量子力学，问题“它在哪里”无法确定地回答。 我们如何处理不确定性？ 通过分配概率。 由于空间的连续性以及空间被认为是无限的（至少在忽略广义相对论的情况下），这变得稍微复杂一些，但思想与有限事件的概率相同。 概率密度是非负的，并且在整个空间上积分为1（这类似于互斥且穷尽的所有事件的概率之和为1）。

因此，在量子力学中，一个物体被表示为随时间演变的“概率波团”。
它如何演变？ 基本方程不是以概率密度的形式书写的，而是以另一个关于空间和时间的函数的形式书写，通过这个函数可以找到概率密度。
考虑概率密度的平方根作为空间和时间的函数。 然后，为了增加一些普遍性，让平方根可以是正数或负数-当你将其平方得到概率密度时，任何一个都可以。接下来，为了更加普遍，允许这个平方根在复平面上具有任意相位，使其具有实部和虚部。 我们不再称其为平方根，而是称之为“波函数” ψ（r，t），它是一个关于空间 r和时间 t的复值函数。 概率密度则是波函数的模的平方。

$$|\psi(r,\,t)|^2 = \psi(r,\,t)\psi^*(r,\,t) \tag{10.1}$$

其中星号 *表示复共轭。
在之前处理概率时，我们从未用更原始的东西来表示它们。为什么现在需要呢？ 因为量子力学的基本方程涉及 $\psi(r,\,t)$而不是概率密度。为什么呢？别问。这只是量子力学的许多奇特特性之一。

量子力学的基本方程是由奥地利物理学家埃尔温·薛定谔（1887-1961）于1926年发表的薛定谔方程。[1]

$$i\hbar\frac{\partial\psi(r,\,t)}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\psi(r,\,t) + V(r)\psi(r,\,t) \tag{10.2}$$

其中 i是虚数单位的平方根， m是物体的质量， $V(r)$是势能函数，而 $\hbar = h/2\pi$= 1.054×10^{-34}焦耳秒。
请注意，这个方程包含了空间和时间的偏导数。 对时间的导数是一阶的，而空间的导数是二阶的。 拉普拉斯算子 ∇^2的定义为

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \tag{10.3}$$

其中 x， y和 z是三个空间维度。
这个方程10.2经常被解释为将其两边乘以 $\psi^*(r,\,t)$并在空间上积分。 然后左边被认定为总能量，右边被认定为动能和势能的和（假设波函数被归一化，使得空间积分$|\psi(r,\,t)|^2$等于1，这是以概率密度解释所要求的性质）。 除此之外，将 $i\hbar\partial/\partial t$称为能量算符是方便的。 它是数学意义上的算符（作用于一个函数并产生另一个函数的东西），并且具有正确的能量单位。 量子力学常常以类似的算符形式表述。

薛定谔方程看似简单。 从数学上讲，它是一个关于 $\psi(r,\,t)$ 的线性方程，如果ψ_1和 ψ_2都是解，那么它们的线性组合也是解。

$$\psi_{总} = \alpha_1\psi_1 + \alpha_2\psi_2 \tag{10.4}$$

[1]在 http://www-groups.dcs.st-andrews.ac.uk/~history/Biographies/Schrodinger.html上查看薛定谔的传记

其中 α_1和 α_2是复常数（如果线性组合要得到有效的概率分布，则 α_1和 α_2的值必须使得 $|\psi$总$|^2$在整个空间上的积分为1）。 然而，除了最简单的 $V(r)$ 的情况外，该方程尚未以闭式解求解。

严格来说，薛定谔方程只有在所描述的物体是整个宇宙且 $V(r) = 0$的情况下才是正确的，但这个方程非常复杂，因此是无用的。 然而，在宇宙被认为是由两部分组成的情况下，它通常被用作近似值——一个小的部分（物体）的波函数正在被计算，而宇宙的其余部分（“环境”）对物体的影响被假设为 $V(r)$项表示。请注意，物体可以是一个光子、一个电子，或者两个或更多的粒子，即它不一定对应于日常概念中的一个单独粒子。

一个物体可能与其环境相互作用。 当然，如果物体改变了其环境（例如对物体的某个属性进行测量），那么环境也会改变物体。 因此，在测量之后，物体通常会有一个不同的波函数，并且关于物体的一些信息可能不再可访问。 量子力学的一个特点是，这个新的波函数与环境的变化是一致的；这个特点是薛定谔方程的结果还是量子力学的一个独立方面尚不清楚。

10.3 稳态

尽管对于给定的 $V(r)$项，薛定谔方程可能无法以闭合形式求解，但在不详细了解解的情况下，可以对其解的性质进行很多推断。 这是通过将$\psi(r, t)$表示为一系列称为定态的函数之和来实现的。

定态是薛定谔方程的解的一种特殊形式，即空间函数乘以时间函数的乘积。 可以很容易地从薛定谔方程中证明定态的最一般形式可以是

$$\psi(r, t) = \phi(r)e^{-iEt/\hbar} \tag{10.5}$$

对于某个实常数 E（实数，否则 $\psi(r, t)$会在非常大或非常小的时间下无限增长），其中 $\phi(r)$满足不涉及时间的方程

$$E\phi(\text{半径}) = -\frac{\hbar^2}{2\text{质量}}\nabla^2\phi(\text{半径}) + \text{势能 (半径)}\phi(\text{半径}) \tag{10.6}$$

并且在所有空间上的积分 $|\phi(\text{半径})|^2$为1。 将 $\psi(\text{半径, 时间})$的依赖关系分离开来的技术有时被称为“变量分离”。

对于 $\phi(\text{半径})$ 的非零解不能获得所有的 E值。 可能存在一些范围，在这些范围内任何 E值都可以，而在其他范围内只有特定的离散 E值才会导致非零波函数。 一般来说，对应于离散 E值的解在远离时变得很小（即在“无限远处消失”），因此它们在空间中是局部化的，尽管它们的“概率波团”可能在几个地方具有较大的值，因此可以被认为代表两个或更多的粒子。

这些解被称为“稳态”，因为波函数的大小（因此也是概率密度）在时间上不变；它只是空间的一个函数。

对于这些稳态， E有一个有趣的解释。 如果我们将方程10.6的每一边都乘以 $\phi^*(r)$并在空间上积分，我们会发现（就像在前一节中一样） E是右边两项的和，被解释为物体的动能和势能。 因此， E是与该解相关的总能量。

当然，一般情况下，具有这个势能 $V(r)$的薛定谔方程的解不是稳态，即不具有方程10.5的特殊形式。但请记住，薛定谔方程的任何线性组合也是一个解。 我们可以将这些稳态作为构建更一般解的基石。

我们最感兴趣的是在空间中局部化的稳态，因此允许的 E
的值是离散的，尽管可能有很多（甚至可能是可数无限多个）。如果我们让 j

成为稳态的索引，那么可以定义结果的波函数 ψ_j（ r，t），使得它们在每个平方的幅度的空间积分为1，而且在所有空间积分时，任意一个与另一个的复共轭的乘积为零，即“归一化”和“正交”。 然后，我们将解释为该状态相关联的能量的 E的值，用 e_j表示。

然后，薛定谔方程的一般解可以写成稳态的线性组合

$$\psi(r,\, t) = \sum_j a_j \phi_j(r) e^{-ie_j t/\hbar} \tag{10.7}$$

其中 a_j被称为扩展系数，可以是复数。 如果波函数 $\psi(r,\, t)$ 被归一化
那么很容易证明

$$1 = \sum_j |a_j|^2 \tag{10.8}$$

并且与函数相关的能量可以用 e_j表示

$$\sum_j e_j |a_j|^2 \tag{10.9}$$

从这些关系中我们可以观察到 $|a_j|^2$的行为类似于事件的概率分布其中包括各种状态的占据情况，并且这个分布可以用来计算物体的平均能量。

我们对量子力学的简要探讨的结论是为了证明下一节中给出的多态模型。 那些愿意接受这个模型而不需要任何解释的读者已经跳过了过去的两节，现在重新加入我们。

10.4 多状态模型

我们对物理对象的模型，通过前两节中对量子力学的简要讨论得到了合理的解释。 该物体具有一个波函数 ψ，原则上可以描述其随时间的行为。 这个波函数可能很难或者不可能计算，并且当物体与其环境发生相互作用时，它可能会以不可预测的方式发生变化。

该物体具有有限（或者可能是可数无限）数量的“稳定态”，这些稳定态更容易计算（尽管对于复杂的物体来说，找到它们可能仍然是不可能的）。 每个稳定态都有自己的波函数 ψ_j，其中 j是稳定态的索引。 如果实际的波函数是这些稳定态之一（即该状态被“占据”），那么物体将无限期地保持在该状态中（或者直到它与其环境发生相互作用）。 每个稳定态都有自己的能量 E_j，并且可能有自己的其他感兴趣的物理量的值。

物体的波函数可以表示为静态状态的线性组合，形式为

$$\psi = \sum_j a_j \psi_j \tag{10.10}$$

其中 a_j是称为展开系数的复数。 如果物体占据其中一个静态状态，则除其中一个外，所有 a_j 都为0。 不失一般性，可以定义展开系数，使其模的平方之和为1：

$$1 = \sum_j |a_j|^2 \tag{10.11}$$

测量物体的属性，如能量，涉及与物体的环境的相互作用，并且环境会发生变化（即使只是为了记录答案）。这是

量子力学的一个结果是，如果物体处于其稳定状态之一并且测量其能量，则测量结果只是该状态的能量，并且状态不会改变（即测量不会改变扩展系数，除了一个系数外，其他系数都为0）。另一方面，如果物体不处于稳定状态之一，则测量结果是其中一个稳定状态的能量，并且物体立即进入该稳定状态。因此，每次测量后，物体都处于一个稳定状态。是哪个状态呢？状态 j被选择的概率是 $|a_j|^2$。因此，通过实验测量得到的能量的期望值是

$$\sum_j e_j |a_j|^2 \tag{10.12}$$

其中 e_j是与静止状态 j相关联的能量。量子力学中的测量不像测量日常物体那样，人们认为能量或其他物理性质可以以任意精度测量，并且这样的测量不会干扰物体。量子测量的性质是量子力学中必须接受的那些方面之一，即使它可能不符合日常生活中形成的直觉。

10.4.1 能量系统

存储、传输或转换能量的物体必须具有可能的状态。这样的物体通常可能由大量（例如阿伏伽德罗常数 $N_A = 6.02 \times 10^{23}$）的相似或相同粒子组成，因此具有大量的静止状态。在这种情况下，薛定谔方程无法求解。为了将能量转移给环境或从环境中获取能量，必须经常与环境发生相互作用。无法知道系统是否处于静止状态，即使知道，与环境的不可预测的相互作用也会迅速使这种知识变得无关紧要。

对于这样的系统，最多可以处理各个稳定态的占据概率 p_j

$$p_j = |a_j|^2 \tag{10.13}$$

能量的期望值 E将会是

$$E = \sum_j e_j p_j \tag{10.14}$$

这个模型的设置非常适合使用最大熵原理来估计占据概率分布 p_j。这个主题将在这些笔记的第11章中进行探讨。

10.4.2 信息系统

旨在进行信息存储、传输或处理的对象应该避免与环境的不可预测交互中固有的错误。似乎最简单的这样的对象需要两个状态来处理信息。一个比特的信息可以与占据的状态相关联。更复杂的对象，具有超过两个状态，可以表示多个比特的信息。

量子信息系统，包括计算机和通信系统，将是这些笔记的第13章的主题。

第11章

能量

在这些笔记的第9章中，我们介绍了最大熵原理作为一种与约束一致的概率分布估计技术。

一个可以通过解析方法求解的简单情况是存在三个概率、一个平均值约束以及概率之和为一的情况。因此，有两个方程和三个未知数，可以直接用其中一个未知数表示熵，消除其他未知数，并找到最大值。 如果存在四个概率和两个平均值约束，这种方法也适用，此时方程的数量比未知数少一个。

另一个特殊情况是存在许多概率但只有一个平均约束。 尽管熵不能用一个概率表示，但第9章的解决方案在可以计算求和的情况下是实用的。

在将最大熵原理应用于物理系统时，可能的状态数量通常非常大，因此解析和数值解都不实用。然而，最大熵原理仍然有用，因为它导致不同量之间的关系。 在本章中，我们将研究这些系统的一般特征。

因为我们现在对物理系统感兴趣，所以我们将熵表示为每开尔文焦耳，而不是比特，并使用自然对数而不是以2为底的对数。

11.1 磁偶极模型

下面的大部分结果适用于由量子力学隐含的物理系统的一般多态模型；请参阅第10章。 然而，一个重要的方面是能量对外部参数的依赖。 例如，对于磁偶极，外部参数是磁场 H。 这里简要回顾一下磁偶极，以便在下面用作示例。

这个模型在第9.1.2节中介绍。图11.1显示了一个具有两个偶极子和两个与系统进行交互的环境的系统。（当然，任何实际系统都会有比两个偶极子多得多，但重要的思想可以用只有两个偶极子来说明。）偶极子受到外加磁场 H的作用，因此系统的能量取决于偶极子的取向和外加场。 系统中的每个偶极子以及其两个环境中的偶极子都可以是“上”或“下”，因此系统有四种可能的状态，“上-上”，“上-下”，“下-上”和“下-下”。每个偶极子的能量如果向下则为 $m_d H$，如果向上则为 $-m_d H$，而每个状态的能量是两个偶极子能量的总和。

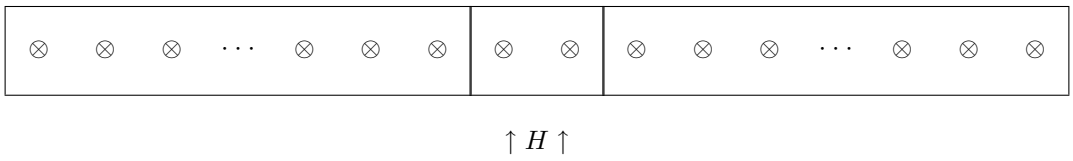


图11.1：偶极矩示例。 每个偶极矩可以是向上或向下的。

11.2 物理系统的最大熵原理

根据量子力学激发的多态模型（见本笔记的第10章），系统有限（或可数无限）数量的量子态。 我们将使用 i作为索引来表示这些态。 这些态具有能量 E_i，并且可能还具有其他物理属性。 在列举和描述这些态之后，可以作为单独的步骤使用最大熵原理来估计每个态被占据的可能性。

我们用事件 A i表示态$_i$的占据情况。 态 i的占据概率为 $p(A_i)$。
为简单起见，我们将这个概率 $p(A_i)$写作 p_i。 我们使用最大熵原理来估计与平均能量 E为已知量 $\widetilde{E}$一致的概率分布 p_i。因此

$$\widetilde{E} = \sum_i p_i E_i \tag{11.1}$$

$$1 = \sum_i p_i \tag{11.2}$$

熵是

$$S = k_B \sum_i p_i \ln \frac{1}{p_i} \tag{11.3}$$

其中 $k_B = 1.38 \times 10^{-23}$焦耳每开尔文，被称为玻尔兹曼常数。
在第9章中，我们提出了在满足类似于方程11.2的约束条件下最大化熵的概率分布，即方程9.12。该公式是在熵以比特为单位的情况下给出的；对于以焦耳每开尔文为单位的物理系统，该公式相同，只是使用了 e 而不是2：

$$p_i = e^{-\alpha} e^{-\beta E_i} \tag{11.4}$$

所以

$$\ln \frac{1}{p_i} = \alpha + \beta E_i \tag{11.5}$$

概率的总和必须为1，因此

$$\alpha = \ln \left(\sum_i e^{-\beta E_i} \right. \tag{11.6}$$

根据最大熵原理，目标是找到给定期望能量 E的各种量。 然而，在除了最简单的情况下，通常更容易进行相反的计算。 也就是说，更容易将 β作为独立变量，计算 α，然后找到 p_i，然后计算熵 S和能量 E。

11.2.1 一般性质

因为 β起着核心作用，直观地了解它可能假设的不同值如何影响事物是有帮助的。

首先，如果 $\beta = 0$，所有概率都是相等的。这只能在状态数有限的情况下发生。

其次，如果$\beta > 0$，则能量较低的状态具有更高的被占据的概率。 同样，如果$\beta < 0$，则能量较高的状态具有更高的被占据的概率。 由于对能量的指数依赖性，除非 $|\beta|$很小，否则具有很高被占据概率的状态只有能量接近最小可能值的状态（β为正）或能量接近最大可能值的状态（β为负）。

第三，我们可以将上述方程乘以 $ln(1/p_i)$，并对$_i$求和，得到

$$S = k_B(\alpha + \beta E) \tag{11.7}$$

即使无法用 E的形式找到 β或计算多个值的 p_i，该方程仍然有效且有用。

第四，在第11.2.2节中，我们将研究能量 E的微小变化 dE，并探究其他变量的变化。 这种一阶关系或“微分形式”在解释公式时提供了直观的帮助。

第五，在第11.2.3节中，我们将以磁偶极系统及其外部参数 H为例，考虑能量对外部参数的依赖。

上述关键方程为方便起见列在此处

$$1 = \sum_i p_i \tag{11.8}$$

$$E = \sum_i p_i E_i \tag{11.9}$$

$$S = k_B \sum_i p_i \ln \frac{1}{p_i} \tag{11.10}$$

$$p_i = e^{-\alpha} e^{-\beta E_i} \tag{11.11}$$

$$\alpha = \ln\left(\sum_i e^{-\beta E_i}\right) = \frac{S}{k_B} - \beta E \tag{11.12}$$

11.2.2微分形式

现在假设 E_i不依赖于外部参数，并且 E通过一个小量 dE变化。 我们将根据上述方程计算其他量的变化，仅保留一阶变化（即忽略类似于（dE）^{2}这样的项，对于足够小的 dE来说，它们非常小）

$$0 = \sum_i dp_i \tag{11.13}$$

$$dE = \sum_i E_i \, dp_i \tag{11.14}$$

$$
\begin{aligned}
dS &= k_B \sum_i \ln \frac{1}{p_i} \, dp_i + k_B \sum_i p_i \, d\left[\ln \frac{1}{p_i} \right] \\
&= k_B \sum_i \ln \frac{1}{p_i} \, dp_i - k_B \sum_i \frac{p_i}{p_i} \, dp_i \\
&= k_B \sum_i (\alpha + \beta E_i) \, dp_i \\
&= k_B \beta \, dE \qquad (11.15) \\
d\alpha &= \frac{1}{k_B} \, dS - \beta \, dE - E \, d\beta \\
&= -E \, d\beta \qquad (11.16) \\
dp_i &= p_i(-d\alpha - E_i \, d\beta) \\
&= -p_i(E_i - E) \, d\beta \qquad (11.17)
\end{aligned}
$$

从中可以很容易地证明

$$
dE = -\left(\sum_i p_i (E_i - E)^2 \right) d\beta \qquad (11.18)
$$

$$
dS = -k_B \beta \left(\sum_i p_i (E_i - E)^2 \right) d\beta \qquad (11.19)
$$

这些方程可以用多种方式使用。 注意，所有一阶变化都表示为关于 $d\beta$的函数，因此自然地将 β视为独立变量。 但这并非必需；无论哪种变化引起其他变化，这些方程仍然有效。

通过这些方程获得的洞察力的一个例子是，注意到将 dE和 $d\beta$相关的公式，即方程11.18，意味着如果 E增加，则 β减小，反之亦然。

11.2.3 带有外部参数的微分形式

现在我们想要将这些微分形式扩展到约束量取决于外部参数的情况。 在我们的磁偶极子示例中，每个状态的能量都取决于外部施加的磁场 H。 每个 E_i可以写成 $E_i(H)$ 的形式以强调这种依赖关系。 因此，约束可以写成显示这种依赖关系的形式：

$$
E = \sum_i p_i E_i(H) \qquad (11.20)
$$

然后所有的量 $(p_i, \alpha, \beta,$和 $S)$ 都可以被认为是依赖于 E和 H的。 在我们的磁偶极模型中，能量 $E_i(H)$ 恰好与 H成正比，比例常数取决于 i但不取决于 H。 在其他模型中，对于其他物理系统， E可能以不同的方式依赖于 H或其他参数。

考虑一下如果 E和 H都稍微变化一点，分别变化了 dE和 dH，与用来计算 $p_i, \alpha, \beta,$和 S的值相比。这些量中将会有小的变化 $dp_i, d\alpha, d\beta,$和 dS，这些变化可以用小的变化 dE和 dH来表示。 由于 dE引起的变化已经计算过了。 由于 dH的变化通过与每个状态相关联的能量变化 $dE_i(H)$ 进入（类似下面几个公式可以推导出由任何外部参数引起的变化，不仅仅是磁场）

$$0 = \sum_i dp_i \tag{11.21}$$

$$dE = \sum_i E_i(H)\, dp_i + \sum_i p_i\, dE_i(H) \tag{11.22}$$

$$dS = k_B\beta\, dE - k_B\beta \sum_i p_i\, dE_i(H) \tag{11.23}$$

$$d\alpha = -E\, d\beta - \beta \sum_i p_i\, dE_i(H) \tag{11.24}$$

$$dp_i = -p_i(E_i(H) - E)\, d\beta - p_i\beta\, dE_i(H) + p_i\beta \sum_j p_j\, dE_j(H) \tag{11.25}$$

$$dE = -\left[\sum_i p_i(E_i(H) - E)^2\right[d\beta + \sum_i p_i(1 - \beta(E_i(H) - E))\, dE_i(H) \tag{11.26}$$

$$dS = -k_B\beta\left[\sum_i p_i(E_i(H) - E)^2\right[d\beta - k_B\beta^2 \sum_i p_i(E_i(H) - E)\, dE_i(H) \tag{11.27}$$

对于在此考虑的特定磁偶极模型，涉及 $dE_i(H)$的项可以通过注意到每个状态的能量 $E_i(H)$与参数 H成正比来简化

$$dE_i(H) = \frac{\tilde{E_i(H)}}{\tilde{H}}\, \tilde{dH} \tag{11.28}$$

$$\sum_i p_i dE_i(H) = \frac{E}{H}\, dH \tag{11.29}$$

因此这些公式简化为

$$0 = \sum_i dp_i \tag{11.30}$$

$$dE = \sum_i E_i(H)\, \tilde{dp_i} + \frac{\tilde{E}}{\tilde{H}}\, \tilde{dH} \tag{11.31}$$

$$dS = k_B\beta\, dE - \frac{k_B\beta E}{\tilde{H}}\, dH \tag{11.32}$$

$$d\alpha = -E\, d\beta - \frac{\beta E}{H}\, dH \tag{11.33}$$

$$dp_i = -p_i(E_i(H) - E)(d\beta + \frac{\tilde{\beta}}{H}\, \tilde{dH}) \tag{11.34}$$

$$dE = -\left[\sum_i p_i(E_i(H) - E)^2\right[(d\beta + \frac{\tilde{\beta}}{H}\, \tilde{dH}) + \frac{\tilde{E}}{H}\, \tilde{dH} \tag{11.35}$$

$$dS = -k_B\beta\left[\sum_i p_i(E_i(H) - E)^2\right[(d\beta + \frac{\tilde{\beta}}{H}\, \tilde{dH}) \tag{11.36}$$

这些公式可以用来描述变量之间的趋势。 例如，最后一个公式表明，β的百分之一变化产生的熵变与H的百分之一变化相同。

11.3 系统和环境

到目前为止，这些公式适用于系统，如果求和是在系统的状态上进行的，如果求和是在系统和环境的更大数量的状态上进行的，它们也适用于系统和环境。（图11.1的磁偶极模型甚至显示了一个能够与两个环境之一进行交互的系统，这是能量转换所需的特性。）下面是系统和其环境相互作用的一些结果。

11.3.1 分割模型

让我们将系统和其环境（暂时只考虑一个环境）建模为宇宙的一部分，它们各自具有自己的可能状态集，并且可以相互隔离或接触。 也就是说，系统在与其环境分离时具有可以描述的状态。 每个状态都有与之相关联的能量，可能还有其他物理性质。 这种描述与确定实际占据的状态是分开的，确定实际占据的状态是使用最大熵原理进行的。

我们还假设环境有自己的状态集，每个状态都有自己的能量和可能的其他物理属性。 再次强调，这种状态的描述与实际上被占据的状态无关。

我们对这两者之间的相互作用建立了一个模型（或者等价地说，我们对总组合如何被分割成系统和环境的模型），即组合的每个状态都由环境和系统中的一个状态组成。 因此，例如，如果系统有四个状态（就像我们简单的双偶极子模型一样），而环境有1000个状态，那么组合就会有4000个状态。 组合的每个状态都对应于系统的一个状态和环境的一个状态。

我们需要一个符号来保持事物的清晰。 我们将使用索引 i表示系统，使用索引 j表示环境。 然后，我们可以使用 i和 j来表示总组合的状态，形式为 i,j，就像联合概率的符号表示一样（因为它确实是联合概率）。 对总组合状态的求和就是对 i和 j的求和。

我们将假设系统和环境可以相互隔离（偶极子图示中的垂直线代表相互作用的屏障），然后在其他时候，两者可以相互作用。 无论它们是隔离的还是相互作用的，都不会影响状态或与状态相关的物理性质，尽管它可能会影响状态的占据概率。

11.3.2 相互作用模型

我们采用分区模型的原因是我们想要控制系统与其
环境之间的相互作用。 不同的物理系统会有不同的相互作用模式和不同的
隔离不同部分的机制。 这里描述了一种简单的磁偶极子相互作用模型，它们
排列在一行中。这只是一个示例。

假设容纳磁偶极子的装置允许相邻的偶极子相互影响。 这种影响可能导致一个偶极子从上向下或从下向上的变化。 自然地，如果一个偶极子影响其邻居，那么同时也会被其邻居影响。合理地假设，如果一个偶极子从上向下改变其状态，那么与其相互作用的邻居应该以相反的方向改变其状态。 结果是两个偶极子交换了它们的方向。 每个方向定向的偶极子的总数保持不变。

考虑两个相邻的偶极子交换它们的方向-左边的偶极子最终具有右边的起始方向，反之亦然。 只有几种不同的情况。

首先，如果两个偶极子具有相同的方向，什么都不会改变。 另一方面，如果两个偶极子具有不同的方向，效果将是方向的模式发生了变化-向上的方向已经移动到左边或右边。 尽管如此，这已经发生了

偶极子本身没有移动。 由于两种可能的排列所关联的能量不同，即使总能量没有变化，能量的位置也发生了微小的变化。

其次，如果两个偶极子都在系统中，或者两个都在环境中，那么能量可能在系统或环境中的位置发生了变化，但在它们之间并没有移动。

第三，如果两个偶极子的初始排列不同，并且它们位于系统和环境之间的边界的两侧，则能量从系统流向环境或者反之亦然。 这并不是因为偶极子移动了，而是因为它们的方向发生了变化。

作为这种相互作用的结果，传递给系统或者从系统传递出去的能量被称为热量。 下面给出了一个关于概率分布变化的热量公式。

有时候这种过程被称为“混合”，因为其效果类似于不同种类的粒子被混合在一起。 然而，在这个类比中，偶极子并没有移动；它们的方向模式或者微观能量发生了变化和混合。

让我们假设我们可以通过放置或者移除适当的屏障来阻止或者允许这个过程。 例如，通过物理上将系统远离其环境，可以抑制这个过程。 能量转换设备通常使用不同时间鼓励或者阻止混合的序列。

11.3.3广义量和强度量

这种分区模型导致了物理量可能具有的一个重要属性。 一些物理量被称为“广义量”，另一些被称为“强度量”。

无论系统是与环境隔离还是与之相互作用，无论系统的概率分布 $p_{s,i}$和环境的概率分布 $p_{e,j}$，以及组合的概率分布 $p_{t,i,j}$如何，系统状态和环境状态的能量相加形成相应总状态的能量（下标 s， e和 t表示系统、环境和总体）：

$$E_{t,i,j} = E_{s,i} + E_{e,j} \tag{11.37}$$

总状态 k的占据概率是两个相关状态 i和 j的概率的乘积：

$$p_{t,i,j} = p_{s,i}p_{e,j} \tag{11.38}$$

在这个背景下，很容易证明总能量的期望值是系统能量和环境能量的期望值之和：

$$\begin{aligned}
E_t &= \sum_{i,j} E_{t,i,j}p_{t,i,j} \\
&= \sum_{i,j} \left[E_{s,i} + E_{e,j}\right] p_{s,i}p_{e,j} \\
&= \sum_{i} \sum_{j} \left[E_{s,i} + E_{e,j}\right] p_{s,i}p_{e,j} \\
&= \sum_{i} p_{s,i} \sum_{j} E_{e,j}p_{e,j} + \sum_{j} p_{e,j} \sum_{i} E_{s,i}p_{s,i} \\
&= \sum_{j} E_{e,j}p_{e,j} + \sum_{i} E_{s,i}p_{s,i} \\
&= E_e + E_s
\end{aligned} \tag{11.39}$$

无论系统和环境是孤立的还是相互作用的，这个结果都成立。 它表明系统的能量和环境的能量相加得到总能量。这是一个结果

事实上，与每个总状态相关联的能量是与相应的系统和环境状态相关联的能量之和。

一个具有其总值为两个（或更多）部分的值之和的性质的量被称为 extensive量。 能量具有这个性质，正如刚刚证明的那样。 熵也是 extensive的。也就是说，

$$
\begin{aligned}
S_t &= \sum_{i,j} p_{t,i,j} \ln \frac{1}{p_{t,i,j}} \\
&= \sum_{i,j} p_{s,i} p_{e,j} \left[\ln \frac{1}{p_{s,i}} + \ln \frac{1}{p_{e,j}} \right] \\
&= \sum_{i} \left(\sum_{j} p_{s,i} p_{e,j} \left[\ln \frac{1}{p_{s,i}} + \ln \frac{1}{p_{e,j}} \right] \right) \\
&= \sum_{i} p_{s,i} \sum_{j} p_{e,j} \ln \frac{1}{p_{e,j}} + \sum_{j} p_{e,j} \sum_{i} p_{s,i} \ln \frac{1}{p_{s,i}} \\
&= \sum_{j} p_{e,j} \ln \frac{1}{p_{e,j}} + \sum_{i} p_{s,i} \ln \frac{1}{p_{s,i}} \\
&= S_e + S_s
\end{aligned}
\tag{11.40}
$$

再次，无论系统和环境是否隔离或相互作用，这个结果都成立。

并非所有感兴趣的量都是广延量。 特别地， α和 β不是。考虑 β。 这是一个例子，其中与系统、环境和总配置相关的值可能有关，也可能无关。 如果系统和环境是隔离的，即对每个系统和环境分别应用最大熵原理，则 β_s和 β_e之间没有关联的理由。

另一方面，如果系统和环境相互作用以交换能量，则系统和环境之间的能量分布可能未知，因此最大熵原理只能应用于组合，而不能分别应用于系统和环境。

然后，相同的 β值将在整个过程中适用。

当作为一个整体进行分析时，像 β这样在系统中处处相同的量被称为强度。

11.3.4 平衡

当系统和其环境在被隔离后允许接触时，分区模型会产生有趣的结果。 在热力学中，这个过程被称为总体配置达到平衡。

假设系统和环境已经被隔离，并且因此具有不同且无关的能量、熵和其他量的值。 然后假设它们被允许相互作用，使用一个总能量不变的相互作用模型。 由于混合，能量可以从系统流向环境，反之亦然，这种能量的流动被称为热量。 因此，占据概率将发生变化，尽管状态和属性的描述，包括它们的能量，不会改变。

我们已经发展了一般公式，用于描述概率、E、S、α和β的小变化，现在可以使用。如果系统的能量被假设有所变化（因为混合），那么这个事实可以纳入到最大熵原理对系统的新应用中，这将导致概率、E、S、α和β的修改。 特别是，我们之前看到$\mathrm{d}E$和$\mathrm{d}\beta$的符号是相反的，所以如果E上升，β就会下降，反之亦然。

很快，系统和环境之间的能量转移可能导致我们不知道每个部分的能量，只知道总能量（由于混合而不改变）。 在这种情况下，适合使用最大熵原理来处理总组合

考虑系统和环境一起。 这样做后，将会有一个新的单一值 β和一个新的总熵。 关于这些值可以说些什么？

首先，新熵是系统的新熵和环境的新熵之和，因为熵是一个广延量。 此外，旧的总熵（在交互开始时）是旧系统熵和旧环境熵的总和，原因相同。 然而，有趣的是新的总熵与旧的总熵的比较。

新熵是根据最大熵原理得出的概率分布来评估的，它是与总能量一致的最大值。 与相同总能量一致的任何其他概率分布都会导致较小（或可能相等）的熵。 一种这样的概率分布是混合之前的分布，即导致旧熵值的分布。

因此，由于系统和环境之间的相互作用，总熵增加（或至少保持不变）。 可能是系统的熵单独减少了，但如果是这样的话，环境的熵必须至少增加同样多。

系统和环境的能量发生了变化，结果 β_s和 β_e的值也发生了变化，方向相反。 它们的新值相同（每个都等于 β_t），因此这个新值位于两个旧值之间。

11.3.5 能量流动、功和热量

让我们回到磁偶极模型，如图11.1所示。

在本节中，我们将只考虑与两个环境中的一个进行交互。 在第12章中，我们将考虑同时使用两个环境，这将使得机器可以用作热机或制冷机。

首先考虑系统与环境隔离的情况，如图11.1所示（垂直线表示相互作用的障碍）。 系统处于某种状态，我们不一定知道是哪种状态，尽管可以从最大熵原理中获得概率分布 p_i。状态的改变通常需要非零能量，因为不同的状态具有不同的能量。 我们总是可以想象一个足够小的变化 dH in H，以至于磁场无法提供或吸收改变状态所需的能量。 然后，我们可以想象一系列这样的 H的变化，其中没有一个可以改变状态，但当它们一起构成足够大的 H的变化时，就会变得明显起来。

我们得出结论，改变孤立系统的 H并不能单独改变状态。 因此，概率分布P$_i$不变。 当然，通过改变 H的量 dH确实会改变能量，从而导致 $E_i(H)$的变化：

$$dE = \sum_i p_i \, dE_i(H) \tag{11.41}$$

这种变化是可逆的：如果场被改变回来，能量可以以电力、磁力或机械形式恢复（在这个模型中没有其他地方可以去）。 这种能量流，可以以电力、磁力或机械形式（或其他形式）恢复，被称为工作。 如果$dE > 0$，那么我们说工作是正的，即外部源对系统做了功；如果 $dE < 0$，那么我们说工作是负的，即系统对外部源做了功。 当然，在能量转换设备中，知道工作是正的还是负的很重要。 在许多情况下，简单地将机器倒转会改变工作的符号；这在其他形式的能量转移中并不总是成立，下面将讨论这一点。

当系统的一个或多个参数发生变化时，如果它不能与环境相互作用，那么对系统的改变被称为绝热变化。由于概率分布不会因此而改变，它们对系统的熵没有影响。 这是一个普遍原理：绝热变化不会改变概率分布，因此保持熵不变。

在上面给出的一般情况下，感兴趣的量的一阶变化是在 E和各种 E_i发生变化的情况下给出的。 如果变化是绝热的，那么 dE只由变化 dE_i引起，一般方程简化为

$$dp_i = 0 \tag{11.42}$$
$$dE = \sum_i p_i \, dE_i(H) \tag{11.43}$$
$$dS = 0 \tag{11.44}$$
$$d\alpha = -E \, d\beta - \beta \sum_i p_i \, dE_i(H) \tag{11.45}$$
$$0 = \left[\sum_i p_i (E_i(H) - E)^2 \right] \left[d\beta + \beta \sum_i p_i (E_i(H) - E) \, dE_i(H) \right] \tag{11.46}$$

如果，如我们的磁偶极模型所示，状态的能量与 H成正比，则这些绝热公式进一步简化为

$$dp_i = 0 \tag{11.47}$$
$$dE = \frac{E}{H} \, dH \tag{11.48}$$
$$dS = 0 \tag{11.49}$$
$$d\alpha = 0 \tag{11.50}$$
$$d\beta = -\frac{\beta}{H} \, dH \tag{11.51}$$

接下来，考虑系统不再是孤立的，而是与其环境相互作用。 交互模型允许热量在系统和环境之间流动，按照惯例，如果能量从环境流向系统，则热量为正，反之为负。能量可以同时通过热量和功进行传递。 工作由个体状态的能量变化 dE_i表示，热量由概率的变化 p_i表示。 因此，上述 dE的公式变为

$$dE = \sum_i E_i(H) \, dp_i + \sum_i p_i \, dE_i(H) \tag{11.52}$$

其中第一项是热量，第二项是功。

11.3.6 可逆能量流

我们在11.3.4节中看到，当系统与其环境相互作用时，总熵通常会增加。 在这种情况下，通过进一步混合无法将系统和环境恢复到它们之前的状态，因为这样的恢复将需要更低的总熵。 因此，混合通常是不可逆的。

总熵保持恒定的极限情况是，如果系统发生了变化，它可以恢复到其之前的状态。 很容易推导出在这种意义下这些变化是可逆的条件。

根据之前给出的公式，特别是方程式11.23，系统熵的变化与热能变化的部分成比例。因此，

$$dS_s = k_B\beta_s\, dE_s - k_B\beta_s \sum_i p_{s,i}\, dE_{s,i}(H) \tag{11.53}$$

$$= k_B\beta_s \left[dE_s - \sum_i p_{s,i}\, dE_{s,i}(H) \right[\tag{11.54}$$

$$= k_B\beta_s\, dq_s \tag{11.55}$$

其中 dq_s代表由于相互作用机制而进入系统的热量。

这个公式适用于系统，类似的公式适用于环境：

$$dS_e = k_B\beta_e\, dq_e \tag{11.56}$$

这两个热量除了符号不同外是相同的

$$dq_s = -dq_e \tag{11.57}$$

因此，总熵 $S_s + S_e$不变（即 $dS_s = -dS_e$）当且仅当系统和环境的两个 β的值相同：

$$\beta_s = \beta_e \tag{11.58}$$可逆变化（总熵不变）可以涉及功和热量，因此系统的能量和熵会发生变化，但系统和环境必须具有相同的β值。 否则，变化是不可逆的。此外，我们注意到在第11.3.4节中，系统和环境之间的相互作用导致了一个介于两个起始值 β_s和 β_e之间的新值 β，因此可逆变化不会改变 β。

早先给出的一阶变化公式可以通过简单地设置 $d\beta$= 0来考虑与环境的可逆相互作用

$$0 = \sum_i dp_i \tag{11.59}$$

$$dE = \sum_i E_i(H)\, dp_i + \sum_i p_i\, dE_i(H) \tag{11.60}$$

$$dS = k_B\beta\, dE - k_B\beta \sum_i p_i\, dE_i(H) \tag{11.61}$$

$$d\alpha = -\beta \sum_i p_i\, dE_i(H) \tag{11.62}$$

$$dp_i = -p_i\beta\, dE_i(H) + p_i\beta \sum_j p_j\, dE_j(H) \tag{11.63}$$

$$dE = \sum_i p_i(1 - \beta(E_i(H) - E))\, dE_i(H) \tag{11.64}$$

$$dS = -k_B\beta^2 \sum_i p_i(E_i(H) - E)\, dE_i(H) \tag{11.65}$$

与之前一样，在个体状态的能量与 H成正比的情况下，这些公式可以进一步简化

$$0 = \sum_i dp_i \tag{11.66}$$

$$dE = \sum_i E_i(H)\, dp_i + \underset{\sim}{\frac{E}{H}}\underset{\sim}{} dH \tag{11.67}$$

$$dS = \underset{\sim}{k_B\beta}\, dE - \underset{\sim}{k_B\beta}\ \underset{\sim}{\frac{E}{H}}\ dH \tag{11.68}$$

$$d\alpha = -\ \underset{\sim}{\frac{\beta E}{H}}\ dH \tag{11.69}$$

$$dp_i = -\ \underset{\sim}{\frac{p_i\beta}{H}}\ (E_i(H) - E)\, dH \tag{11.70}$$

$$dE = \ \underset{\sim}{\frac{E}{H}}\ dH - \ \underset{\sim}{\frac{\beta}{H}}\ \left[\sum_i p_i(E_i(H) - E)^2\right[dH \tag{11.71}$$

$$dS = -\ \underset{\sim}{\frac{k_B\beta^2}{H}}\ \left[\sum_i p_i(E_i(H) - E)^2\right[dH \tag{11.72}$$

这些公式将在本笔记的下一章中用于推导涉及热量的能量转换机器的效率约束

第12章

温度

在这些笔记的前几章中，我们介绍了最大熵原理作为一种估计与约束一致的概率分布的技术。

在第8章中，我们讨论了可以通过解析方法解决的简单情况，其中有三个概率、一个平均值约束条件，以及概率之和为一的事实。因此，有两个方程和三个未知数，可以直接用一个未知数表示熵，消除其他未知数，并找到最大值。 如果有四个概率和两个平均值约束条件，这种方法也适用，此时方程比未知数少一个。

在第9章中，我们讨论了一般情况，其中有许多概率但只有一个平均约束条件，因此熵无法用单个概率表示。 之前用拉格朗日乘数法推导出的结果已经给出。

在第11章中，我们研究了最大熵原理对量子力学驱动的多态模型的物理系统的影响，正如第10章所概述的那样。
我们发现双重变量 β起着核心作用。 它的值表示高能量或低能量态是否被占据（或具有更高的占据概率）。 通过它，可以计算出其他所有量，包括能量的期望值和熵。

在本章中，我们将进一步解释 β，并将其倒数定义为（在一个比例因子内）物质的温度。然后，我们将看到能量转换的效率存在约束，这些约束可以自然地用温度来表达。

12.1 温度刻度

热机是一种从环境中提取热量并产生工作的机器，通常以机械或电的形式。 正如我们将看到的，热机需要两个不同的环境才能正常运行。 下面的公式限制了能量转换的效率，以两个环境的不同 β值为基础。 我们将推导出这些限制。

然而，首先，处理 β的倒数而不是 β本身是有用的。 回想一下， β是一个强度属性：如果两个具有不同 β值的系统接触在一起，它们最终会有一个共同的 β值，介于原始两个值之间，并且总熵会增加。 对于$1/\beta$以及任何常数乘以$1/\beta$，同样适用。 （实际上，如果 β的两个值中一个是正的，另一个是负的，那么这个陈述是不正确的；在这种情况下，得到的 β值是中间的，但1/β的结果不是。）注意，通过使用第11章中的公式，1/β可以解释为能量的微小变化除以引起它的熵变化，与比例因子kB相差无几。让我们将“绝对温度”定

$1/\beta$的结果值不是。）注意，通过使用第11章中的公式，$1/\beta$可以解释为能量的微小变化除以引起它的熵变化，与比例因子 k_B相差无几。让我们将“绝对温度”定义为

$$T = \frac{1}{k_B \beta} \tag{12.1}$$

其中 $k_B = 1.381 \times 10^{-23}$焦耳/开尔文是玻尔兹曼常数。 使用最大熵原理得到的概率分布，当用 T表示时，

$$p_i = e^{-\alpha} e^{-\beta E_i} \tag{12.2}$$

$$= e^{-\alpha} e^{-E_i/k_B T} \tag{12.3}$$

在温度的解释上， β与温度的日常特性是一致的，即两个处于相同温度的物体不会交换热量，如果两个处于不同温度的物体接触，一个物体会变热，另一个物体会变冷，使得它们的温度趋近于彼此。 在日常经验中，绝对温度是正值，对应的 β也是正值。

因为温度是一个更熟悉的概念，比起对偶变量或拉格朗日乘子，从现在开始我们将用温度来表达我们的结果。

绝对温度 T以开尔文（有时错误地称为开尔文度）为单位，以威廉·汤姆森（1824-1907）的名字命名，他在1848年提出了绝对温度尺度。[1]摄氏度尺度在世界大多数国家的一般公众中常用，与开尔文尺度有一个加法常数的差别，而华氏度尺度在美国常用，除了加法常数的差别外还有一个乘法因子的差别。 最后，为了完整列出尺度，威廉·兰金（1820-1872）提出了一个尺度，其0与开尔文尺度相同，但度的大小与华氏度尺度相同。

由于温度既用于科学目的（适用于开尔文温标），又用于日常经验，因此需要多个温度标度。 最初的标度是为了普通大众使用而设计的。 加布里埃尔·华氏（1686-1736）希望一个标度，使得欧洲最热和最冷的天气都在0和100之间。 他意识到大多数人最容易处理的数字范围就是这个范围。 安德斯·摄氏（1701-1744）于1742年决定，0到100之间的温度应该覆盖水为液体的范围。 在他最初的摄氏温标中，他将水的沸点表示为0度，冰点表示为100度。 两年后有人建议颠倒这些点。[2]结果在1948年以摄氏命名，并且现在在全世界范围内使用。

为了一般兴趣，表 12.1显示了四个标度上的一些感兴趣的温度，以及 β。

12.2 热机

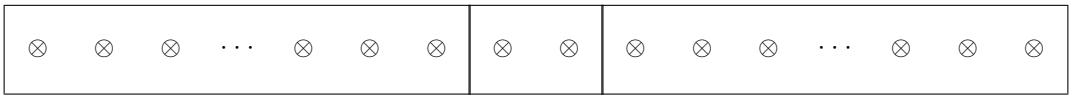
我们考虑的磁偶极系统如图12.1所示，其中有两个处于不同温度的环境，并且每个环境与系统的相互作用可以通过有无障碍来控制（如图中所示）。 尽管图12.1显示系统中有两个偶极子，但这里的分析适用于只有一个偶极子，或者有多个偶极子，只要系统中的偶极子数量远远少于任何一个环境中的偶极子数量即可。

现在让我们使用温度替换β，重新书写第11章的公式。 因此，方程11.8到11.12变为

[1] 汤姆森是苏格兰格拉斯哥大学的一位多产的科学家/工程师，对电磁学、热力学及其工业应用做出了重大贡献。 他发明了“麦克斯韦的恶魔”这个名字。1892年，他因在横跨大西洋的电缆工作中被封为拉格斯男爵。 凯尔文是流经该大学的河流的名字。[2] 根据一些记载，这个建议是由卡罗卢斯·林奈（1707-1778）提出的，他是乌普萨拉大学的同事，也是摄氏的叔叔的门徒。 林奈以发明植物和动物的科学记法而闻名，这种记法至今仍被植物学家和动物学家使用。

	K	◦摄氏度	◦华氏度	◦兰氏度	$k_BT = \frac{1}{\beta}$（焦耳）	β（焦耳$^{-1}$）
绝对零度	0	-273.15	-459.67	0	0	∞
外太空（近似值）	2.7	-270	-455	4.9	3.73×10^{-23}	2.68×10^{22}
液态氦沸点	4.22	-268.93	-452.07	7.6	5.83×10^{-23}	1.72×10^{22}
液态氮沸点	77.34	-195.81	-320.46	139.2	1.07×10^{-21}	9.36×10^{20}
水熔点	273.15	0.00	32.00	491.67	3.73×10^{-21}	2.65×10^{20}
室温（近似值）	290	17	62	520	4.00×10^{-21}	2.50×10^{20}
水沸点	373.15	100.00	212.00	671.67	5.15×10^{-21}	1.94×10^{20}

表12.1：各种感兴趣的温度
（沸点 = 沸点，熔点 = 熔点）



$\uparrow H \uparrow$

图12.1：偶极矩示例。
（每个偶极矩可以是向上或向下）

$$1 = \sum_i p_i \tag{12.4}$$

$$E = \sum_i p_i E_i \tag{12.5}$$

$$S = k_B \sum_i p_i \ln \frac{1}{p_i} \tag{12.6}$$

$$p_i = e^{-\alpha} e^{-E_i/k_BT} \tag{12.7}$$

$$\begin{aligned} \alpha &= \ln \left(\sum_i e^{-E_i/k_BT} \right) \\ &= \frac{S}{k_B} - \frac{E}{k_BT} \end{aligned} \tag{12.8}$$

从第11章的偶极模型的微分公式中，每个状态的能量与 H成正比，方程式11.30到11.36变为

$$0 = \sum_i dp_i \tag{12.9}$$

$$dE = \sum_i E_i(H)\, dp_i + \sum \frac{E}{H} \sum dH \tag{12.10}$$

$$T\, dS = dE - \sum \frac{E}{H} \sum dH \tag{12.11}$$

$$d\alpha = \sum \frac{E}{k_B T} \left[\left[\left[\frac{1}{T} \sum dT - \sum \frac{1}{H} \sum dH \right. \right. \right. \tag{12.12}$$

$$dp_i = p_i \left[\frac{E_i(H) - E}{k_B T} \left[\left[\left[\frac{1}{T} \sum dT - \sum \frac{1}{H} \sum dH \right. \right. \right. \right. \tag{12.13}$$

$$dE = \left[\sum_i p_i (E_i(H) - E)^2 \left[\sum \frac{1}{k_B T} \left[\left[\left[\frac{1}{T} \sum dT - \sum \frac{1}{H} \sum dH \right. \right. \right. \right. + \sum \frac{E}{H} \sum dH \tag{12.14}$$

$$T\, dS = \left[\sum_i p_i (E_i(H) - E)^2 \left[\sum \frac{1}{k_B T} \left[\left[\left[\frac{1}{T} \sum dT - \sum \frac{1}{H} \sum dH \right. \right. \right. \right. \tag{12.15}$$

能量的变化可以归因于工作 dw和热量 dq的影响

$$dw = \sum \frac{E}{H} \sum dH \tag{12.16}$$

$$\begin{aligned} dq &= \sum_i E_i(H)\, dp_i \\ &= T\, dS \end{aligned} \tag{12.17}$$

12.3 能量转换循环

如果系统与其环境以及外加磁场的相互作用都得到适当控制，该系统可以作为热机。 思路是使系统发生变化，以便可以描述其经历一系列状态并返回起始状态。

这代表一个循环，可以重复多次。 在一个循环中，热量与两个环境之间进行交换，工作则在系统与控制磁场的代理之间进行交换。 如果系统在一个循环中从环境中以热量的形式获得的能量比其向环境中放回的能量多，那么能量必须以工作的形式传递给控制磁场的代理。

热机的循环如图12.2所示。不失一般性，我们可以处理 H为正的情况。 假设左边的环境温度为 T_1，为正数，但小于右边环境的温度 T_2（设备工作必须满足这两个温度不同）。 该循环在对应于系统的 S和T的坐标轴形成的平面上显示，并形成一个矩形，角标记为 a， b， c和 d，边对应于值 S_1， S_2， T_1和 T_2。

由于温度被假设为正数，能级较低的占据概率较高。 因此，根据我们在这里定义的能量，能量 E为负数。 因此，随着场强度增加，能量变得更加负数，这意味着能量实际上从系统传递到磁装置。 将磁场视为增强，因为一个大的永久磁铁被物理上移向系统。 系统中的磁偶极子对该磁铁施加吸引力，使其向系统靠近，当磁铁移动时，该力可以

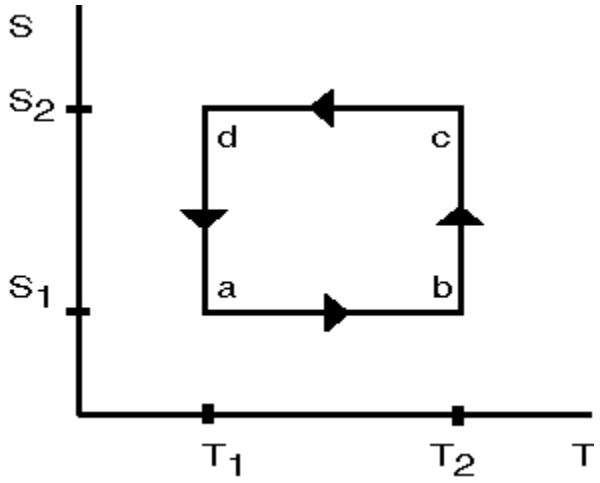


图12.2：温度循环

用于拉伸弹簧或克服重力提升重物，从而储存能量。 以这种形式进入系统（或离开系统）的能量，可以来自（或添加到）外部能量源，称为功（或负功）。

首先考虑循环的下部分，系统的温度在不改变熵的情况下从T_1增加到T_2。 没有熵变的操作称为绝热的。 根据上述方程$_{12.1}$5，通过增加H来增加T，同时不允许系统与其两个环境之一发生相互作用。 （换句话说，阻止系统中的偶极子与两个环境中的偶极子相互作用的屏障已经建立。）在这个过程中，系统的能量下降（变得更负），因此向产生磁场的外部装置提供了能量，对系统所做的功是负的。

接下来，考虑这个循环的右边部分，在此期间熵从 S_1增加到 S_2，在恒定温度 T_2下。 这个步骤，在恒定温度下，被称为等温的。 根据方程式12.15，这是通过在系统与右侧环境接触时减少 H来实现的，假设右侧环境的温度为 T_2。 （换句话说，图12.1中左侧的屏障保持不变，而右侧的屏障被撤销。）在这个过程中，能量的变化 E来自于高温环境中的热量流入以及外部磁装置的工作。 热量为$T_2(S_2 - S_1)$，而工作是正的，因为在这个过程中 H的减少将能量推向0。

接下来的两个过程与前两个类似，只是工作和热量的方向相反，即，热量为负，因为能量从系统流向低温环境。 在顶部过程中，系统与两个环境隔离，因此是绝热的。 在左侧的等温过程中，系统与低温环境相互作用。

在经历了这个循环之后，系统在能量、磁场和熵方面回到了起始状态。 这两个环境略有变化，但我们假设它们相对于存在的偶极子数量来说都要大得多，所以它们的变化不大。 净变化是高温环境熵的轻微损失和低温环境熵的相等增加。 因为它们处于不同的温度，当热量流动发生时转移的能量是不同的——它与温度成比例，因此离开高温环境的能量比进入低温环境的能量多。 差异是一种净负功，表现为磁装置中的能量。 因此，来自两个环境的热量被转化为功。 只有当两个环境处于不同的温度时，才会转化一定数量的能量。

表12.2总结了热机循环。

腿	开始	结束	类型	dS	dT	H	E	热量输入	功输入
底部	a	b	绝热	0	正	增加	减少	0	负
右	b	c	等温	正	0	减少	增加	正	正
顶部	c	d	绝热	0	负	减少	增加	0	正
左	d	a	等温	负	0	增加	减少	负	负
总计	a	a	完整循环	0	0	无变化	无变化	正	负

表12.2：能量循环

对于每个循环，高温环境损失的能量为 $T_2(S_2 - S_1)$，低温环境获得的能量为 $T_1(S_2 - S_1)$，因此净能量转化为差值$(T_2 - T_1)(S_2 - S_1)$。 热机尽可能将高温环境损失的热量转化为功是可取的。 这台机器的效率

$$\frac{\text{解决}}{\text{高温热进入}} = \frac{T_2 - T_1}{T_2} \tag{12.18}$$

这个比率被称为卡诺效率，以法国物理学家萨迪·尼古拉斯·莱昂纳德·卡诺（1796-1832）的名字命名。[3]他是第一个认识到热机不能具有完美的效率，并且这个效率极限（随后以他的名字命名）适用于所有类型的可逆热机。

上述操作是可逆的，即整个循环可以反向运行，结果是从低温环境向高温环境抽取热量。 这个动作在自然界中不会自发发生，事实上类似的分析表明，磁性装置必须向磁偶极子提供功，以使这种情况发生，从而使高温环境中放入的热量比低温环境中损失的热量更多。 以这种反向方式运行的热机被称为制冷机或热泵。

[3]有关传记，请访问 http://www-groups.dcs.st-andrews.ac.uk/~history/Mathematicians/Carnot_Sadi.html

第13章

量子信息

在这些笔记的第10章中，介绍了量子系统的多态模型。 然后，在第11章和第12章中，将该模型应用于能量转换系统。 现在将其应用于信息处理系统。

量子信息的科学和技术相对较新。量子位（称为量子比特）的概念首次以所需的形式在1995年提出。 关于量子信息仍然有许多未解答的问题（例如，量子版本的信道容量定理尚不确切）。 因此，该领域处于不稳定状态。 我们的知识中存在空白。

13.1量子信息存储

我们使用比特作为最简单的经典系统的数学模型来存储信息。 同样，我们需要一个模型来存储信息的最简单的量子系统。 它被称为“量子比特”。在最简单的情况下，量子比特可以被看作是一个具有两个状态的小物体，可以处于其中一个状态，并且可以通过测量仪器访问并揭示该状态。 然而，量子力学限制了可以用于将信息移动到系统中或从系统中移出的交互类型，并允许额外的信息存储和处理模式，这些模式在经典情况下是不存在的。

这些笔记的第9、11和12章中使用的量子位的一个例子是磁偶极。
其他一些潜在的技术重要性的例子是量子点（用于捕获电子的三维井）和光子（具有不同极化的光粒子）。

物理上处理量子位是困难的。 这就是为什么量子计算机目前还不可用的原因。 虽然创建量子位可能不难，但测量它们往往很困难，并且通常很难使它们不与宇宙的其他部分发生相互作用，从而不可预测地改变它们的状态。

假设我们的系统是一个单独的磁偶极。 该偶极可以是“上”或“下”，这些状态具有不同的能量。 系统仅由一个偶极组成，这使得系统很脆弱。

经典位不像这么脆弱的原因是它们使用更多的物质。例如，半导体存储器可以通过存在或不存在一千个电子来表示一个位。 如果一个电子缺失，其他电子仍然存在，测量仍然可以工作。 换句话说，存储数据的机制中存在大量的冗余。 冗余对于纠正错误是有效的。 出于类似的原因，可以在不改变其状态的情况下读取经典位，并且一个位可以控制两个或多个门的输入（换句话说，可以复制该位）。

然而，至少有三个原因使我们可能希望在没有这种大量冗余的情况下存储比特。
首先，这将更加高效。 可以在相同大小或成本的结构中存储或处理更多的比特。 半导体行业正在朝着这个方向迅速发展，在2015年之前，应该可以制造出使用如此少的原子的存储单元和门电路，以至于数据存储粒子数量的统计波动将成为一个问题。 其次，存储没有冗余的敏感信息将无法在不改变信息的情况下进行复制，因此可以安全地保护信息，或者至少知道其安全性是否受到破坏。 第三，量子力学的性质可能允许进行经典计算和通信无法实现的模式。

需要一个读写量子比特的模型。 我们的写入模型（有时称为“准备”比特）是将具有已知状态（“上”或“下”）的“探针”与系统的单个偶极子接触。 然后，系统和探针交换它们的状态。 系统最终具有探针之前的值，而探针最终具有系统之前的值。 如果之前的系统状态已知，则写入后探针的状态已知，可以再次使用探针。

如果不是这样，那么由于对其状态的不确定性，探针将无法重复使用。 因此，向一个具有未知数据的系统写入会增加对环境的不确定性。 这里的一般原则是，丢弃未知数据会增加熵。

读取量子位的模型并不简单。 我们假设测量仪器以某种方式与位相互作用，以确定其状态。 这种相互作用将系统强制进入其稳定状态之一，并且仪器的状态会以一种方式改变，该方式由系统最终进入的状态决定。如果系统已经处于其中一个稳定状态中，则选择该状态。

更一般地说，如果系统的波函数是稳定状态的线性组合，则选择其中一个状态的概率由扩展系数的平方大小决定。
我们现在提出三种量子位的模型，其行为越来越复杂。

13.2 模型1：微小的经典比特

量子位的最简单模型我们只会简要讨论。 它不够通用，无法适应量子信息的最有趣的属性。

这个模型类似于磁偶极模型，只有两种状态（上和下）是可能的。 每次测量都会将系统恢复到其两个值之一，因此小错误不会累积。 由于可以在不改变系统的情况下进行测量，因此可以复制一个比特。这个量子比特的模型基本上与经典比特相似，只是它的大小可能非常小，并且可能能够快速测量。

这个模型在能量转换系统中被证明是有用的。 它在这些笔记的第12章中使用过。

13.3 模型2：态的叠加（量子比特）

第二个模型利用了量子力学中的态可以用满足薛定谔方程的波函数来表示这一事实。 由于薛定谔方程是线性的，任何满足它的波函数的线性组合也满足它。因此，如果我们将逻辑值0与波函数 ψ_0关联起来，并将逻辑值1与波函数 ψ_1关联起来，那么任何形式的线性组合

$$\psi = \alpha\psi_0 + \beta\psi_1 \tag{13.1}$$

其中 α和 β是复数常数，满足 $\alpha^2 + \beta^2 = 1$，是系统的有效波函数。然后
测量返回值为0的概率为 $|\alpha|^2$，测量返回值为1的概率为 $|\beta|^2$。当进行测量时， α和 β的值会改变，使其中一个为1，另一个为0，与测量结果一致。

看起来，以这种方式定义的量子比特可以携带大量信息，因为 α和 β都可以取许多可能的值。 然而，测量只会返回0或1的事实以及这些系数会被测量破坏的事实意味着从单个量子比特中只能读取一位信息，无论最初在指定α和β时有多么小心。

事实上，这些系数会被测量破坏，并且只能从单个量子比特中读取一位信息，无论最初在指定 α和 β时有多么小心。

13.4 模型3：具有纠缠的多个量子比特

考虑一个具有四个状态的量子力学系统，而不是两个。假设我们可以对系统进行两种不同的测量，每种测量返回0或1。 自然地，我们用两个下标来表示稳定态，一个对应于第一次测量，另一个对应于第二次测量。 因此，一般的波函数形式为

$$\psi = \alpha_{00}\psi_{00} + \alpha_{01}\psi_{01} + \alpha_{10}\psi_{10} + \alpha_{11}\psi_{11} \tag{13.2}$$

其中复系数满足归一化条件

$$1 = \mid \alpha_{00} \mid^2 + \mid \alpha_{01} \mid^2 + \mid \alpha_{10} \mid^2 + \mid \alpha_{11} \mid^2 \tag{13.3}$$

你可以将这个模型看作是两个量子比特，一个对应于每个测量。 这些量子比特不是独立的，而是以某种方式纠缠在一起。然后自然而然地会问，如果其中一个被测量会发生什么。 例如，对第一个量子比特的测量将以概率$\mid \alpha_{00}\mid^2 + \mid \alpha_{01} \mid^2$返回0，并且如果是这样，波函数将坍缩为与这个测量值一致的稳定态。

$$\psi = \frac{\alpha_{00}\psi_{00} + \alpha_{01}\psi_{01}}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}} \tag{13.4}$$

（注意，结果波函数通过除以 $|\alpha_{00}|^2 + |\alpha_{01}|^2$进行了“重新归一化”）。

这个系统没有必要在一个地方物理上存在。 事实上，最有趣的例子之一涉及两个纠缠在这种方式下的量子比特，其中第一次测量在一个位置进行，第二次测量在另一个位置进行。 一个简单的情况是最初只有四个可能的静态态中的两个，所以 $\alpha_{01} = 0$， $\alpha_{10} = 0$。 这个系统具有一个非凡的特性，即通过一次测量，波函数会坍缩到两个可能的静态态之一，并且这个坍缩的结果可以被另一次测量检测到，可能是在远程位置。

可以定义几个作用于多个量子比特的有趣逻辑门。 它们具有可逆性的特性；这是量子力学系统的一般特性。

多比特的有趣应用之一是

- 比经典计算机更快地计算一些算法（包括整数因子分解）
- （用于重构量子态所需的）传送信息
- 密码系统
- 反向信息传输（在经典情况下不可能）
- 超密编码（如果之前发送了另一个比特，则可以在一个比特中传输两个经典比特）

这些应用在几本书和论文中有描述，包括以下三本：

- T. P. Spiller，“量子信息处理：密码学、计算和传送”，IEEE会议录，第84卷，第12期，页1719-1746；1996年12月。 尽管这篇文章已经几年了，但仍然是一篇很好的介绍。

- Michael A. Nielsen和Isaac L. Chuang，“量子计算与量子信息”，剑桥大学出版社，英国剑桥；2000年

- Hoi-Kwong Lo，Sandu Popescu和Tim Spiller，“量子计算与信息导论”，世界科学出版社，新加坡；1998年。 该书基于1996年11月至1997年4月在惠普实验室（英国布里斯托尔）举行的讲座系列。

13.5 详细信息：量子比特和应用

第13.5节到第13.11节是基于Luis P´erez-Breva于2005年5月4日撰写的笔记。

前几节已经介绍了关于波函数的量子信息的基本特征。 我们在第10章中首次介绍了波函数的概念，涉及到物理系统的背景。 波函数是一个有争议的对象，引发了关于量子力学物理解释的不同学派。 然而，它非常有用，可以用来推导出在给定时间内粒子位置的概率，并且它使我们能够在第10章中引入多态模型，作为薛定谔方程线性性质的直接结果。

在这些笔记的前几章中，我们介绍了比特作为研究经典信息科学的二进制（两态）量。 在量子信息科学中，我们也对两态系统感兴趣。 然而，与经典比特不同，量子比特还可以处于叠加态。 例如，我们可能对无限势阱的前两个能级的叠加态感兴趣。 处理两态系统（可能包括叠加态）动力学所需的数学工具是线性代数（我们在第2章中在离散余弦变换的背景下进行了回顾）。为了强调我们关注的是两态系统的动力学，而不是每个状态的动力学，最好抽象出波函数并引入一种新的表示法，即括号表示法。 在接下来的几节中，当我们介绍括号表示法时，我们会注意到与经典领域的第一个重要差异：无克隆定理和纠缠。 然后，我们概述了量子力学在通信（传送和密码学）、算法（Grover快速搜索，Deutsch-Josza）和信息科学（纠错码）方面的应用。

13.6 Qubit的括号表示法

括号表示法是由P. Dirac引入的量子力学中的表示法。 在这些笔记的背景下，括号表示法将为我们提供一种表示列向量、行向量、点积、矩阵和线性变换的新方法。 然而，括号表示法比这更一般化；它可以完全描述具有连续变量（如位置或动量）的波函数。[1]

13.6.1 Ket、Bra、Bracket和Operator

Ket、bra、bracket和operator是括号表示法的基本构件，它是量子力学系统中最常用的表示法。 它们可以被视为列向量、行向量、点积和矩阵。

$|\,Ket\rangle$

Ket只是由复数组成的列向量。它的表示形式为：

$$|\,k\rangle = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \overrightarrow{k}\,. \tag{13.5}$$

在 ket $|\,k\rangle$内部的符号 k是我们用来标识这个向量的标签。 两个 ket $|\,0\rangle$和 $|\,1\rangle$用于表示量子比特的两个逻辑状态，并具有标准的向量表示。

$$|\,0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |\,1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{13.6}$$

[1]对于对括号符号非常详细（和高级）的阐述感兴趣的读者可能会发现以下书籍的前几章很有用： “量子力学第一卷”（作者：科恩-塔努吉、伯纳德·迪尤和弗兰克·拉洛，*Wiley-Interscience*，*1996*年）。

从方程式 13.1中可以回忆起两个量子态的叠加 ψ_0和ψ_1是

$$\psi = \alpha\psi_0 + \beta\psi_1 \tag{13.7}$$

其中 α和 β是复数。 在括号符号表示法中，这个叠加 $\mid 0\rangle$和 $\mid 1\rangle$可以写成

$$\begin{aligned} \mid \psi\rangle &= \alpha \mid 0\rangle + \beta \mid 1\rangle && (13.8)\\ &= \alpha \begin{pmatrix}1\\0\end{pmatrix} + \beta \begin{pmatrix}0\\1\end{pmatrix} && (13.9)\\ &= \begin{pmatrix}\alpha\\\beta\end{pmatrix} && (13.10) \end{aligned}$$

$\rangle Bra \mid$

Bra是ket的共轭转置。 也就是说，对于给定的ket，相应的bra是一个行向量（ket的转置），其中的元素已经进行了复共轭。 例如，来自13.10的qubit有一个相应的bra $\rangle\psi \mid$，它是通过对方程式13.10进行共轭转置得到的。

$$\begin{aligned} (\mid \psi\rangle)^\dagger &= (\alpha \mid 0\rangle + \beta \mid 1\rangle)^\dagger && (13.11)\\ &= \alpha^* (\mid 0\rangle)^\dagger + \beta^* (\mid 1\rangle)^\dagger && (13.12)\\ &= \alpha^* \begin{pmatrix}1\\0\end{pmatrix}^\dagger + \beta^* \begin{pmatrix}0\\1\end{pmatrix}^\dagger && (13.13)\\ &= \alpha^* \begin{pmatrix}1 & 0\end{pmatrix} + \beta^* \begin{pmatrix}0 & 1\end{pmatrix} && (13.14)\\ &= \begin{pmatrix}\alpha^* & \beta^*\end{pmatrix} && (13.15) \end{aligned}$$

符号 †用于表示向量或矩阵的共轭转置操作。[2]星号 $(^*)$ 是复数的共轭记号： $(a + i\ b)^* = a - i\ b$，其中 a和 b是实数。

$\rangle$布拉 $\mid$凯特$\rangle$

点乘是一个布拉（行向量） $\rangle q \mid$，与一个凯特（列向量） $\mid k\rangle$的乘积，它被称为内积，用 $\rangle q \mid k\rangle$表示，它符合线性代数的预期。

$$\rangle q \mid k\rangle = \begin{pmatrix}q_1^* & q_2^*\end{pmatrix} \times \begin{pmatrix}k_1\\k_2\end{pmatrix} = \sum_j q_j^* k_j \circ \tag{13.16}$$

请注意， $\rangle q \mid k\rangle$的结果是一个复数。

括号允许我们引入一个非常重要的kets属性。 假设kets始终是归一化的，这意味着一个ket与自身的点积等于1。 这意味着ket的列向量中至少有一个元素非零。 例如，任意qubit $(\mid \psi\rangle)$与自身的点积， $\rangle\psi \mid \psi\rangle = 1$，所以

$$\begin{aligned} \rangle\psi \mid \psi\rangle &= (\alpha^*\rangle 0 \mid + \beta^*\rangle 1 \mid) \cdot (\alpha \mid 0\rangle + \beta \mid 1\rangle)\\ &= \alpha^*\alpha\rangle 0 \mid 0\rangle + \beta^*\alpha\rangle 1 \mid 0\rangle \alpha^*\beta\rangle 0 \mid 1\rangle + \beta^*\beta\rangle 1 \mid 1\rangle\\ &= \alpha^*\alpha + \beta^*\beta\\ &= |\alpha|^2 + |\beta|^2 = 1 \end{aligned} \tag{13.17}$$

[2]这个操作有几个不同的名称，包括 “复共轭转置 ”和 “伴随 ”。

当我们将qubit引入为波函数的叠加时，这正是我们假设的结果。 在第10章中，我们看到波函数与其复共轭的乘积是一个概率分布，并且必须积分为1。这个要求与ket的归一化要求完全类似。[3]

点积可以用来计算量子比特处于可能状态之一的概率 $|0\rangle$和$|1\rangle$. 例如，如果我们想要计算测量结果为量子比特$|\psi\rangle$处于状态$|0\rangle$的概率，我们只需取$|0\rangle$和$|\psi\rangle$的点积并将结果平方

$$\begin{aligned}\Pr(|0\rangle) &= |\alpha\rangle 0|\psi\rangle|^2 \\ &= |\alpha 0\,|\,0\rangle + \beta 0\,|\,1\rangle|^2 \\ &= |\alpha_1 + \beta_0|^2 \\ &= |\alpha|^2\end{aligned} \tag{13.18}$$

算符

算符是将一个态矢$|k\rangle$转化为另一个态矢$|q\rangle$的对象. 算符用帽子表示: $\widehat{O}$. 根据我们对态矢的定义，算符就是一个矩阵,

$$\begin{aligned}O\rangle|^{卡} &= {}^{11}\begin{pmatrix}{}^{12} \\ o_{21}\ {}_{22}\end{pmatrix} \quad \begin{pmatrix} \times \begin{pmatrix}卡_1 \\ k_2\end{pmatrix}\end{pmatrix} \\ &= \begin{pmatrix}q_1 \\ q_2\end{pmatrix} \\ &= |\,q\rangle\end{aligned} \tag{13.19}$$

运算符对bra的作用方式类似

$$\rangle 卡\,|\,\widehat{O}^\dagger = \rangle q\,| \tag{13.20}$$

方程式13.19和13.20以及ket的归一化要求使我们能够推导出一个重要的性质。 将两个方程式相乘，我们得到

$$\rangle 卡\,|\,\widehat{O}^\dagger\widehat{O}\,|卡\rangle = \rangle q\,|\,q\rangle \tag{13.21}$$

$$\rangle 卡\,|\,\widehat{O}^\dagger\widehat{O}\,|卡\rangle = 1, \tag{13.22}$$

第二行是基于假设 $\widehat{O}$保持$k$ket的归一化，由于 $\rangle 卡\,|卡\rangle = 1$，它意味着 $\widehat{O}^\dagger\widehat{O} = \mathbb{I}$。 具有这种性质的运算符被称为酉算符，它们的逆等于它们的伴随。 所有的量子力学运算符必须是酉算符，否则，$k$et的概率分布的归一化将不会被变换保持。请注意，这与我们在第10章中要求时间演化是酉的推理完全相同。 从物理角度来看，酉性意味着通过 $\widehat{O}$定义的操作进行后再进行撤销，应该使我们得到与起始状态相同的结果（注意与可逆性的定义的相似性）。

如果我们知道输入和输出的凯特矢，有一种简单的方法来构造一个算符。 我们可以使用外积，即列向量与行向量的乘积（点积通常也称为内积或内部积，因此称为外积）。 我们可以使用凯特矢与布拉的外积来构造算符 $\widehat{O}$

$$\widehat{O}\,|\,k\rangle = (|\,q\rangle\rangle k\,|)\,|\,k\rangle = |\,q\rangle\rangle k\,|\,k\rangle = |\,q\rangle \tag{13.23}$$

[3] $\Psi^*\Psi dx$是一个点积可能不是立即显而易见的。 为了看到它是如此，离散化积分 $\int \Psi^*\Psi dx \to \sum_i \Psi_i^*\Psi_i$并与点积的定义进行比较。 你可能会争辩说，这样做我们已经将一个函数 Ψ转化为一个具有元素 Ψ_i的向量；但我们将一个 ket定义为一个向量，以将其与线性代数相关联。 如果 ket代表一个函数的话，那么点积的适当定义将是 $\langle\Phi\,|\,\Psi\langle = \int \Phi^*\Psi dx$。

请注意，如果kets未归一化为1，这是不可能的；换句话说，ket的归一化强制了以这种方式构建的算符是幺正的事实。

例如，要将处于状态 $\mid 0\rangle$的量子比特转换为上述定义的状态 $\mid \psi\rangle$，我们构造算符

$$\widehat{O}_2 = \alpha \mid 0\rangle\rangle 0 \mid + \beta \mid 1\rangle\rangle 0 \mid \tag{13.24}$$

我们可以验证该算符产生了预期的结果

$$\begin{aligned}\widehat{O}_2 \mid 0\rangle &= \alpha \mid 0\rangle\rangle 0 \mid 0\rangle + \beta \mid 1\rangle\rangle 0 \mid 0\rangle \\ &= \alpha \mid 0\rangle 1 + \beta \mid 1\rangle 1 \\ &= \alpha \mid 0\rangle + \beta \mid 1\rangle \\ &= \mid \psi\rangle\end{aligned} \tag{13.25}$$

我们刚刚进行了我们的第一次量子计算！

在量子力学书籍中，习惯上从算符（ $\widehat{O}$ 中省略帽子 $\rightarrow O$）以“简化符号。”在初级水平（以及高级水平），这种简化会导致算符和标量之间的混淆；在这些笔记中，我们将尽量避免这样做。

13.6.2 张量积—复合系统

到目前为止，我们介绍的符号表示单量子比特系统。 然而，有必要有一种符号表示法，允许我们描述复合系统，即多个量子比特的系统。 在集合论中，当我们有两个集合 A和 B并且我们想将它们视为一个整体时，会出现类似的情况。 在集合论中，为了形成集合集，我们使用笛卡尔积 $A \times B$来表示这两个集合的集合。 线性代数中的类比称为张量积，用符号 $\otimes$表示。 它同样适用于向量和矩阵（即ket和算符）。

从实际角度来看，张量积连接物理系统。 例如，一个两粒子系统可以表示为 |粒子 $1\rangle\otimes$ |粒子 $2\rangle$，粒子的电荷和自旋也可以通过张量积表示 | 电荷$\rangle\otimes$ |自旋$\rangle$。 如果我们有两个量子比特 $\mid \psi\rangle$和 $\mid \phi\rangle$，由这两个量子比特组成的系统可以表示为 $\mid \psi\rangle\otimes \mid \phi\rangle$。

尽管它们有着相似的目标，笛卡尔积和张量积在构建集合的元素方面有所不同。 笛卡尔积产生元组。 因此，如果 A和 B是两个数集，它们的笛卡尔积 $A \times B$是一对数(a, b)的集合，其中 a属于 A， b属于 B。 这是一个简单的连接。

张量积的元素是通过稍微不同的方式从组成部分获得的。例如，考虑两个 2×2矩阵

$$\mathbb{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \mathbb{B} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \tag{13.26}$$

张量积产生一个 4×4矩阵

$$\mathbb{A}\otimes\mathbb{B} = \begin{pmatrix} a\alpha & a\beta & b\alpha & b\beta \\ a\gamma & a\delta & b\gamma & b\delta \\ c\alpha & c\beta & d\alpha & d\beta \\ c\gamma & c\delta & d\gamma & d\delta \end{pmatrix} \tag{13.27}$$

虽然一开始可能不明显，但这种构建张量积的方式与我们进行矩阵乘法的方式是一致的。 作为一种运算，它具有一个非常有趣的特点，它从$2\times$ 2矩阵输出$4\times$ 4矩阵，但并不是所有的$4\times$ 4矩阵都可以通过这种方式生成（对于数学倾向的读者来说，张量积运算不是满射的），这就是张量积的这个特性。

这个产品将激发关于纠缠的讨论，可能是量子力学中最奇特的特征之一。

你应该花些时间适应张量积，并确保不要被我们在过去两节中介绍的各种不同的乘积搞混。

1. 点积（$\rangle k \mid q\rangle$）得到一个复数；
2. 外积（$\mid k\rangle\langle q \mid$）得到与 Ket 相同维度的方阵；
3. 张量积（$\mid k\rangle\otimes \mid q\rangle$）用于研究复合系统。 它得到一个维度等于两个Ket（或矩阵）的维度之和的向量（或矩阵）。

括号表示的张量积

正如我们之前提到的，两个量子比特 $\mid q_1\rangle$和 $\mid q_2\rangle$的张量积表示为 $\mid q_1\rangle\otimes \mid q_1\rangle$。
有时候符号被缩写，以下四种张量积的表示被等价地简化

$$\mid q_1\rangle\otimes \mid q_2\rangle \equiv\mid q_1\rangle \mid q_2\rangle \equiv\mid q_1, q_2\rangle \equiv\mid q_1 q_2\rangle \tag{13.28}$$

对于n个量子比特，通常缩写符号给每个量子比特一个指标 $\mid q\rangle$:

$$\mid q_1\rangle\otimes \mid q_2\rangle \otimes \ldots \otimes \mid q_n\rangle = \bigotimes_{j=1}^{n} \mid q_j\rangle \tag{13.29}$$

态矢量张量积的对偶是相应的态矢量对偶的张量积。 这意味着在缩写符号中，复共轭操作将态矢量变为对偶态矢量，但标签保持不变

$$\begin{aligned}(\mid q_1 q_2\rangle)^\dagger &= (\mid q_1\rangle\otimes \mid q_2\rangle)^\dagger \\ &= \rangle q_1 \mid \otimes\rangle q_2 \mid) \\ &= \rangle q_1 q_2 \mid).\end{aligned} \tag{13.30}$$

因此，两个复合系统的点积结果是按顺序取的各个点积的乘积

$$\begin{aligned}\rangle q_1 q_2 \mid w_1 w_2\rangle &= (\rangle q_1 \mid \otimes\rangle q_2 \mid)(\mid w_1\rangle\otimes \mid w_2\rangle) \\ &= \rangle q_1 \mid w_1\rangle \otimes \rangle q_2 \mid w_1\rangle\end{aligned} \tag{13.31}$$

在第二项中常常会产生困惑，在没有括号的情况下很容易混淆
by $\rangle q_2 \mid\mid w_1\rangle$ and interpret it as a $\rangle\mid\rangle$ (note the two vertical separators in the correct form), and then try to to take the dot products inside out, instead of taking them in parallel as it should be done.

13.6.3 纠缠态量子比特

我们之前介绍了纠缠的概念，它涉及到一个系统的波函数
allows two measurements to be made. 从张量积的性质可以得出这一结论
作为连接系统的一种方式。 考虑两个量子比特 $\mid \psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$和 $\mid \varphi\rangle = \begin{pmatrix}a\\ b\end{pmatrix}$，根据张量积的定义，

$$\mid \psi\rangle\otimes \mid \varphi\rangle = \begin{pmatrix}\alpha a\\ \alpha b\\ \beta a\\ \beta b\end{pmatrix} \tag{13.32}$$

如果我们在整体系统中操作（即忽略它由两个子系统组成），可以达到以下ket描述的状态并非不可想象

$$|\psi_{12}\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\1\\0\end{pmatrix} \tag{13.33}$$

事实证明，复合系统 $|\psi_{12}\rangle$ c不能表示为两个独立量子比特的张量积。 也就是说，直接在整体上操作，可以得到无法由两个孤立系统描述的状态。

为了看清为什么会这样，尝试将方程13.32和13.33等式化：ket的第一个元素要求 $\alpha a = 0$；这意味着 α或 a必须为零。 然而，如果 $\alpha = 0$，第二个元素不能为1，同样地，如果 $a = 0$，第三个元素将必须为零而不是一。因此，没有 α、β、a和b的组合能够使我们将方程13.33描述的系统写成像方程13.32描述的张量积那样。我们得出结论

$$|\psi_{12}\rangle = |\psi\rangle \otimes |\varphi\rangle. \tag{13.34}$$

我们在第11章的“混合”环境中已经遇到过类似的情况。 在那里，我们注意到，无法再用子系统来明确定义强度变量，而且如果混合两个子系统的过程不可逆，最终系统的熵将大于各个熵的总和，因此不再有意义试图用原始组成部分来表示复合系统。 这两种情况是不同的，但肯定存在类比的空间。

每当这种情况在量子层面上出现时，我们称两个量子比特（或任何两个系统）是纠缠的。 这个词是从德语单词“Verschr¨ankung”翻译而来，通常也被翻译为“交错”。薛定谔创造了这个词来描述以下情况：

“对于一个总系统的最大知识不一定包括对其所有部分的完全知识，即使这些部分完全分离，并且此刻彼此之间没有任何相互影响。”.[4]

与我们在混合过程中看到的最显著的区别在于，正如Schr¨odinger在上面的段落中指出的那样，即使在将它们分开之后，这些部分的“交错”仍然存在，并且其中一个部分的测量将会对另一个部分的结果产生影响。 这就是爱因斯坦所称之为“遥远的神秘行动”。

我们可以通过两个张量积的叠加来说明纠缠对测量的影响，通过重写方程式13.33 $\begin{pmatrix}0\end{pmatrix}$

$$|\psi_{12}\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\\0\end{pmatrix} \tag{13.35}$$

$$= \frac{1}{\sqrt{2}}\left[\begin{pmatrix}0\\0\\1\\0\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\end{pmatrix}\right] \tag{13.36}$$

$$= \frac{1}{\sqrt{2}}\left[|\,0\rangle_1\,|\,1\rangle_2 + |\,1\rangle_1\,|\,0\rangle_2\right]$$

$$= \frac{1}{\sqrt{2}}\left[|\,0_1 1_2\rangle + |\,1_1 0_2\rangle\right], \tag{13.37}$$

4摘自 “Die gegenwartige Situation in der Quantenmechanik,” Erwin Schr¨odinger, Naturwissenschaftern. 23 : pp. 807-812; 823-823, 844-849. (1935). 英文翻译：John D. Trimmer, Proceedings of the American Philosophical Society, 124, 323-38 (1980).

我们添加了下标以区分两个量子比特。 这也是一个很好的例子，可以开始欣赏括号表示法简化表达式的能力。

如果我们测量第一个量子比特，我们将得到 $|\ 0\rangle_1$或 $|\ 1\rangle_1$。 要计算结果为 $|\ 0\rangle_1$的概率，我们将纠缠态与 $|\ 0_1, ?_2\rangle = |\ 0\rangle_1$的点积。

$$\rangle 0_1, ?_2 \mid \psi_{12}\rangle = \rangle 0_1, ?_2 \mid \cdot \frac{1}{\sqrt{2}} [|\ 0\rangle_1 \mid 1\rangle_2 + |\ 1\rangle_1 \mid 0\rangle_2] \tag{13.38}$$

$$= \frac{1}{\sqrt{2}} (\rangle 0 \mid 0\rangle_1 \rangle ? \mid 1\rangle_2 + \rangle 0 \mid 1\rangle_1 \rangle ? \mid 0\rangle_2) \tag{13.39}$$

$$= \frac{1}{\sqrt{2}} \rangle ? \mid 1\rangle_2 \tag{13.40}$$

这个结果表明，如果第二个系统在同一时间坍缩到该状态，结果将是$0\rangle_1$的概率为$1/2$。

$|\ 1\rangle_2$ （请注意，如果问号代表0的话，那么）

（请注意，如果问号代表0，那么概率将等于零）。因此，对第一个系统的测量会对第二个系统的值产生影响，即使这些系统相距很远。

为了欣赏纠缠的可怕性，值得考虑在一个更平凡的环境中思考它。

想象一下，你与一个同事有着如此紧密的联系，以至于每当他打哈欠时，你也会系统性地打哈欠。 你们共同的朋友不会注意到这一点，因为这是一件正常的事情，我们知道当有人打哈欠时，周围的人往往也会打哈欠。 然而，如果你的同事去了欧洲，而你留在美国，每隔一段时间你都会被迫打哈欠，恰好在你的朋友打哈欠的时候，你肯定会吓到他们。事实上，为了吓到你的朋友，他们需要大洋两岸的裁判员记录事件并匹配时间标签。 问题会出现，你和你的同事是否可以利用你们的打哈欠联系进行即时通信，因为似乎需要一个裁判员。 这个卡通例子当然不是量子力学的，但它说明了纠缠的魅力和恐惧，这些魅力和恐惧同时吸引了我们这个时代一些最伟大的思想家。 同时，它也引入了量子通信的一个警告：需要进行经典交流来验证通信是否存在（裁判员）。

当我们讨论传送时，你将有机会在真正的量子力学环境中欣赏到这一点。

13.7 无克隆定理

在经典信息中，最自然的操作之一是复制比特，这在我们的计算机中经常发生。 量子逻辑在这个层面上已经与经典逻辑不同。量子比特不能被复制，或者通常说：量子比特不能被克隆。

有几个直观的论证可以帮助我们理解为什么会这样。请记住，在第10章中，我们强调测量会改变被测量的系统；如果系统处于叠加态，测量的结果将是叠加态的其中一个状态。 而且，叠加态被破坏了。 直观地说，如果克隆需要进行测量，那么我们将无法得知关于初始叠加态的任何信息，因此不可能有两个克隆体。 此外，叠加态本身也会被测量破坏。 这意味着一个可行的克隆设备不能使用测量。

假设我们有这样一台设备，并且它在不需要测量的情况下运行。 量子力学的基础之一是海森堡引入的不确定性原理。该原理表明，某些物理变量不能同时以任意精度进行测量。 以位置和动量为例；如果用给定精度Δx测量粒子的位置，则其动量的测量精度受到限制：$\Delta p > \hbar/2\Delta x$。 假设我们拥有所假设的克隆机器，可以在一个克隆体中以任意精度测量动量，在另一个克隆体中以任意精度测量位置，这可能违反了海森堡的原理。

这些论证本身并不能证明克隆的不可能性，但表明这个问题绝非琐碎。

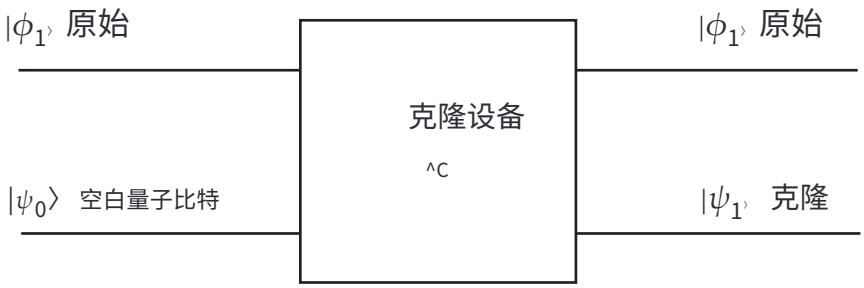


图13.1：建议的克隆设备

为了证明克隆是不可能的，让我们假设克隆是可能的，并且我们可以建立一个像图13.1中的那样的“机器”。克隆设备将一个量子比特$|\phi_1\rangle$的信息复制到另一个“空白”量子比特，结果是一个与$|\phi_1\rangle$相同的量子比特$|\psi_1\rangle$，而原始的$|\phi_1\rangle$保持不变。 根据我们在括号表示法概述中所看到的，这样的机器是一个算符（我们将其称为 $\widehat{C}$），因为它将两个量子比特转换为另外两个量子比特；作为一个算符， $\widehat{C}$必须是幺正的。因此我们定义 $\widehat{C}$

$$|\text{原始}\rangle\otimes|\text{空白}\rangle \xrightarrow{\widehat{C}} |\text{原始}\rangle\otimes|\text{克隆}\rangle \tag{13.41}$$

我们现在准备克隆两个任意的量子比特 $|\phi_1\rangle$和 $|\phi_2\rangle$ 分开。

$$\widehat{C}\,|\phi_1\rangle\,|\text{空白}\rangle = |\phi_1\rangle\,|\psi_1\rangle \tag{13.42}$$

$$\widehat{C}\,|\phi_2\rangle\,|\text{空白}\rangle = |\phi_2\rangle\,|\psi_2\rangle \tag{13.43}$$

$$\tag{13.44}$$

在这里我们理解 $|\phi_1\rangle = |\psi_1\rangle$和 $|\phi_2\rangle = |\psi_2\rangle$，并且我们给它们不同的名称以区分原始和副本。

由于克隆机是酉的，它保持点积不变，因此我们可以比较克隆前后的点积。

$$\langle\phi_2|\langle\text{空白}||\phi_1\rangle|\text{空白}\rangle = \langle\phi_2|\langle\psi_2||\phi_1\rangle|\psi_1\rangle \tag{13.45}$$

回顾张量积的点乘规则，张量积中的每个元素都与相同位置的共轭元素相乘，因此

$$\langle\phi_2|\phi_1\rangle\langle\text{空白}|\text{空白}\rangle = \langle\phi_2|\phi_1\rangle\langle\psi_2|\psi_1\rangle \tag{13.46}$$

要求态矢量归一化的条件是 $\langle$空白 $|$空白$\rangle = 1$. 上述方程只有在两种情况下才成立：

- $\langle\phi_2|\phi_1\rangle = 0$，这意味着 $|\phi_1\rangle$和 $|\phi_2\rangle$是正交的。这意味着我们可以克隆从一组正交态中随机选择的态。 这等价于说我们可以克隆 $|0\rangle$和$|1\rangle$，这我们早就知道，因为我们经常在经典情况下这样做。

- $\langle\psi_2|\psi_1\rangle = 1$，这意味着 $\psi_2 = \psi_1$，也就是说，每次操作得到的克隆是相同的。如果两个原始物体是不同的，正如我们所假设的那样，这个结果表明克隆体与原始物体是独立的，这对于克隆体来说是非常奇怪的属性！。

这个证明表明无法实现量子比特的完美克隆。 我们当然可以存储测量结果（这是另一种表述第一种情况的方式），但我们无法克隆叠加态。

13.8 量子比特的表示

无克隆定理使我们预期量子逻辑与经典逻辑相比会发生变化。
为了充分理解这些差异，我们必须找出一个与经典比特不同的量子比特表示方法，使我们能够描绘叠加态。 我们将介绍两种这样的表示方法，一种是以球面上的点表示量子比特比特，另一种是以电路中的线表示量子比特，类似于我们在第1章中探索的逻辑电路。

13.8.1 布洛赫球中的量子比特

考虑一个处于任意叠加态的量子比特

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle. \tag{13.47}$$

如果 α和 β是实数，方程13.17将定义一个半径为1的圆，我们将把量子比特描绘为圆的边界上的一个点。 然而， α和 β是复数，所以我们需要更加努力地推导出类似的直观理解。

每个复数都可以用相位和幅度表示，所以我们可以将 α和 β重写为：

$$\alpha = Ae^{ia} \qquad \beta = Be^{ib} \tag{13.48}$$

从态矢的归一化（方程13.17）可以推导出

$$\begin{aligned} 1 &= |\alpha|^2 + |\beta|^2 \\ &= A^2 + B^2, \end{aligned} \tag{13.49}$$

现在这是一个以原点为中心的圆的方程，所以 A和 B都可以用角度[5] $\theta/2$来重新表示。

$$A = \cos\frac{\theta}{2} \qquad B = \sin\frac{\theta}{2}\ . \tag{13.50}$$

让我们将这个结果引入原始叠加的方程13.47中:

$$|\psi\rangle = \cos\frac{\theta}{2}e^{ia}\,|0\rangle + \sin\frac{\theta}{2}e^{ib}\,|1\rangle. \tag{13.51}$$

我们还可以做一件事，将 e^{ia}提取为公因子

$$\begin{aligned} |\psi\rangle &= e^{ia}\left(\cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}e^{i(b-a)}\,|1\rangle\right) \\ &= e^{ia}\left(\cos\frac{\theta}{2}\,|0\rangle + \sin\frac{\theta}{2}e^{i\varphi}\,|1\rangle\right) \end{aligned} \tag{13.52}$$

我们将$\varphi = b - a$重新命名. 如果我们忽略全局相位因子(e^{ia})，那么两个角度 θ和 φ定义了一个单位球上的一个点。 这个球被称为布洛赫球，并且在图13.2中显示。它表面上的每个点代表了一种可能的态叠加$|0\rangle$和$|1\rangle$. 例如，考虑处于状态$|\eta\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$的量子比特，我们可以将这个状态与方程 13.52进行比较，并得出结论 $\theta/2 = \pi/4$， $\varphi = 0$，因此量子比特$|\eta\rangle$由与 x轴平行的矢量表示在布洛赫球上。

[5]将角度选择为 $\theta/2$ 而不是 θ对于我们来说是一个技术细节，对于使用电子自旋作为量子比特是合适的，如果我们使用光子的偏振，则适当的选择将是 θ。 这与费米子和玻色子有关，你可能听说过，但在这里对我们来说是无关紧要的。

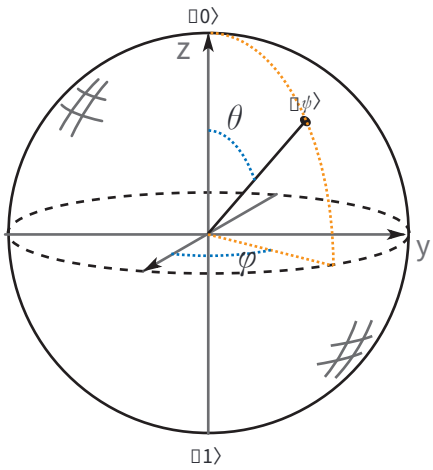


图13.2：量子比特的几何表示：布洛赫球

当我们引入算符时，我们说它们可以转换量子比特的叠加态，同时保持其归一化。 在布洛赫球的背景下，这意味着算符可以将点在单位球上移动，即定义轨迹。

回到方程式13.52，我们仍然需要考虑我们之前忽略的全局相位因子 e^{ia}的影响。 这个因子似乎意味着布洛赫球上的点可以绕自身旋转一个角度a。 然而，由于我们最终关心的是每个状态的概率（因为我们测量的是状态而不是叠加态），我们应该看看这个因子在取点积的平方时会发生什么。 例如，让我们来看一下从方程式13.52测量量子比特得到 $|1\rangle$作为答案的概率。

$$
\begin{aligned}
|\rangle 1 \mid \psi\rangle|^2 &= \left|\rangle 1 \mid \cdot e^{ia}\left(\cos\frac{\theta}{2} \mid 0\right\rangle + sin\frac{\theta}{2}e^{i\varphi} \mid 1\rangle\right)\right|^2 \\
&= \left|e^{ia}\right|^2 \times \left|\cos\frac{\theta}{2}\rangle 1 \mid 0\rangle + sin\frac{\theta}{2}e^{i\varphi}\rangle 1 \mid 1\rangle\right|^2 \\
&= 1 \times \left|0 + sin\frac{\theta}{2}e^{i\varphi} \times 1\right|^2 \\
&= \left|sin\frac{\theta}{2}e^{i\varphi}\right|^2 .
\end{aligned} \tag{13.53}
$$

我们看到全局相位因子平方为一，因此在概率计算中不起作用。 经常有人认为在计算概率时全局相位因子会消失，因此不可测量。

13.8.2 量子比特和对称性

布洛赫球将量子比特上的每个操作描绘为球上的轨迹。 然而，球上的任何轨迹都可以通过一系列绕三个轴的旋转来表示。 因此，解释量子比特上的操作的一种方法是研究布洛赫球轴上的旋转。

这与对称性的研究密切相关。 托马斯·玻尔是第一个建议使用对称性来解释量子力学的人。

如果在应用相应的对称操作（例如旋转）之后，物体看起来没有改变，我们就说该物体具有某种对称性；然后我们说该物体对该对称操作是不变的。 一般来说，对称操作包括旋转、反射和反演；以及不变的

意味着物体的起始位置和结束位置无法区分。 例如，图13.3显示了一个

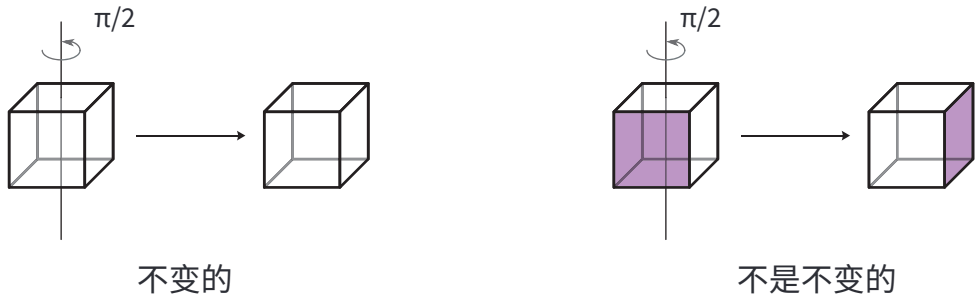


图 13.3：不变性的概念。 对称操作：绕垂直轴旋转 $\pi/2$。

立方体，对绕其任何一个面中心轴的 $\pi/2$旋转是不变的。 为了区分起始位置和结束位置，你需要给立方体的一个面涂上颜色，就像图13.3右侧的图片那样。然后立方体对绕 $\pi/2$的旋转就不再是不变的。 然后我们可以说图13.3左侧的立方体的对称群中包含了绕图中所示轴的 $\pi/2$旋转群，以及其他一些群。

物理学家使用对称性来通过研究最能描述对象如何变换以及其不变性的操作来表征对象。 然后，在布洛赫球中表示量子比特特别有用，因为它告诉我们要关注空间旋转的群。 在本节的其余部分中，我们将调和两种观点，即我们将运算符视为矩阵的观点和布洛赫球的对称性。

我们已经看到量子比特上的运算符是$2\times\ 2$的酉矩阵，为了获得空间旋转的群，我们还需要额外的技术要求，即矩阵的行列式为$+1$（而不是-1）。 这个群有一个名字，叫做$\mathrm{SU}(2)$[6]。 我们可以通过组合以下四个矩阵来构建所有的$\mathrm{SU}(2)$矩阵：

$$\mathbb{I} \stackrel{def}{=} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \sigma_x \stackrel{def}{=} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y \stackrel{def}{=} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z \stackrel{def}{=} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{13.54}$$

这些矩阵被称为保利矩阵，以纪念沃尔夫冈·保利。请注意，严格来说，它们不属于$\mathrm{SU}(2)$，为了数学严谨，我们需要将它们每个都乘以 i，即虚数（要验证这一点，计算它们在乘以 i前后的行列式）。

保利矩阵对任意量子比特的作用

捕捉保利矩阵背后的直觉最好的方法是将它们应用于任意叠加态的量子比特

$$|\,\psi\rangle = \alpha\,|\,0\rangle + \beta\,|\,1\rangle = \cos\frac{\theta}{2}\,|\,0\rangle + \sin\frac{\theta}{2}e^{i\varphi}\,|\,1\rangle \tag{13.55}$$

[6]群通常用字母命名，如O、U、SU、SO等。每个字母都有特定的含义。$\mathrm{SU}(2)$代表维度为2的特殊（S）幺正（U）矩阵群。 特殊意味着矩阵的行列式为$+1$，而幺正在这里与之前讨论的算符中的含义相同。

并解释结果

$$\sigma_x \mid 0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \rightarrow \text{绕x轴旋转 } \pi \tag{13.56}$$

$$\sigma_y \mid 0\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{pmatrix} = i \begin{pmatrix} -\beta \\ \alpha \end{pmatrix} \rightarrow \text{绕y轴旋转 } \pi \tag{13.57}$$

$$\sigma_z \mid 0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} \rightarrow \text{绕z轴旋转 } \pi \tag{13.58}$$

图 13.4说明了在布洛赫球上对一个处于任意叠加态的量子比特进行 σ_y操作的过程。

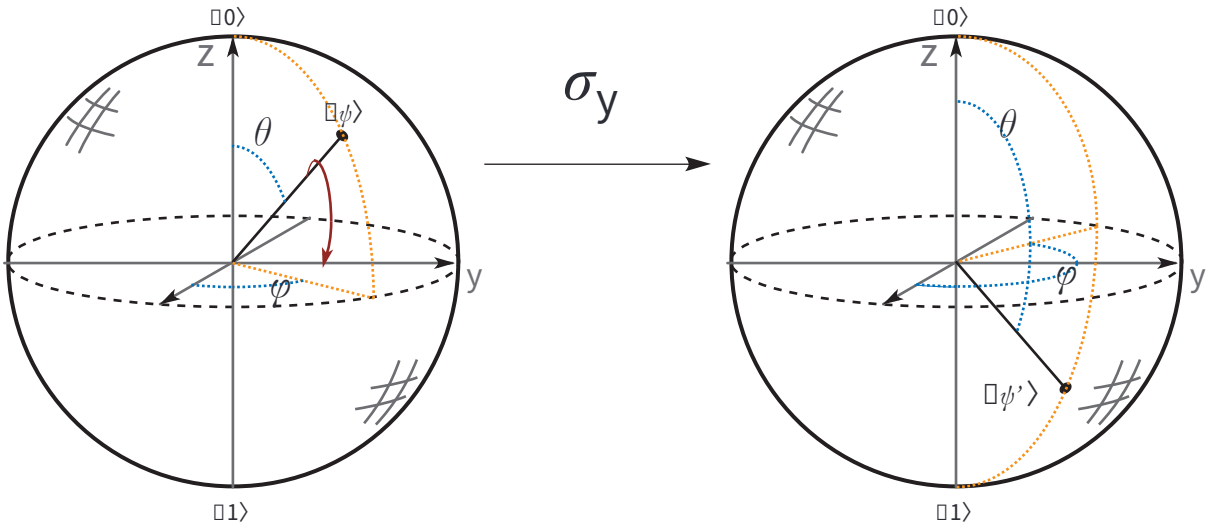


图13.4：在布洛赫球上对 σ_y进行的操作

因此，泡利矩阵是绕布洛赫球的每个轴旋转 π的操作（这解释了我们给它们的名称）。 然而，为了完全探索布洛赫球的表面，我们需要能够定义任意旋转（不仅仅是 π的倍数）。 为此，我们使用了将泡利矩阵指数化的巧妙技巧。回想一下欧拉公式将指数函数与正弦和余弦相关联，

$$e^{ix} = \cos x + i \sin x. \tag{13.59}$$

当 x是实数时，欧拉公式适用。 但我们对于获得类似的结果对于泡利矩阵感兴趣。 我们可以通过将 x替换为 $\frac{\theta}{2}\sigma_x$来证明泡利矩阵的欧拉公式的等价性，并且

将指数函数展开为泰勒级数（注意 $\sigma_x\sigma_x = \mathbb{I}$）

$$e^{i\sigma_x\theta/2} = 1 + i\frac{\theta}{2}\sigma_x - \frac{1}{2}\left(\frac{\theta}{2}\right)^2\mathbb{I} - i\frac{1}{3}\left(\frac{\theta}{2}\right)^3\sigma_x + \frac{1}{4}\left(\frac{\theta}{2}\right)^4\mathbb{I} + \cdots \tag{13.60}$$

$$= \left(1 - \frac{1}{2}\left(\frac{\theta}{2}\right)^2 + \frac{1}{4}\left(\frac{\theta}{2}\right)^4 + \cdots\right)\mathbb{I} + i\left(0 + \left(\frac{\theta}{2}\right) - \frac{1}{3}\left(\frac{\theta}{2}\right)^3 + \cdots\right)\sigma_x \tag{13.61}$$

$$= \cos\frac{\theta}{2}\mathbb{I} + i\sin\frac{\theta}{2}\,\sigma_x. \tag{13.62}$$

这个结果向我们展示了如何围绕 x轴进行任意角度 θ的旋转，结果算符通常被称为 $R_x(\theta) = e^{i\sigma_x\theta/2}$。 R_y和 R_z的情况完全类似。

总结一下，我们已经展示了如何将任何量子比特表示为布洛赫球中的一个点，并且我们已经学会了如何在任意三个轴上进行任意旋转来导航布洛赫球。 由此可见，我们得到了一个关于对称操作的群的表达式，这使我们能够写出作用在单个量子比特上的任意算符的形式。

13.8.3 量子门

在这些笔记的第一章中，我们探索了所有可能的具有一个和两个输入参数的函数，然后挑选出了最有用的布尔函数非、与、非与、非或、或、异或。 然后，我们将其与一个我们称之为门的象形图相联系，并回顾了构建逻辑电路进行计算的机制。

在前一节中，我们对量子比特做了同样的事情，我们表征了可能改变单个量子比特值的所有算符：我们定义了Pauli矩阵，并解释了如何进行任意旋转。
通过与经典情况类比，泡利矩阵和任意旋转是量子电路的门。
在更高级的量子计算论著中，您会希望证明关于量子门的各种结果，例如进行任何量子计算所需的最小门集以及从1比特到n比特的推广的各种结果。 在这里，我们将限制自己总结量子代数的主要细节，其符号表示以及其属性。

基本量子门

表13.1列出了5个基本量子门。它们的符号表示要简单得多。

泡利X	$X =$	$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} \equiv \sigma_x$	等同于执行NOT或位翻转
泡利Y	$Y =$	$\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix} \equiv \sigma_y$	
泡利Z	$Z =$	$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} \equiv \sigma_z$	改变内部相位
哈达玛	$H =$	$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$	
相位	$S =$	$\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$	

表13.1：基本量子门。

如图13.5所示，它们比它们的经典对应物更复杂。
表13.2列举了来自表13.1的一些基本量子门的属性。这些属性是我们在

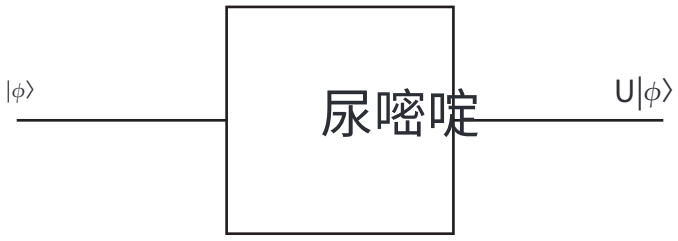


图13.5：通用量子门。 其中U是表示该门的通用酉矩阵的名称。

第1章。 这些和其他更高级的规则有助于简化量子电路，就像德摩根定律有助于简化经典电路一样。

$$H = \frac{1}{\sqrt{2}}(X+Z) \qquad HXH = Z$$
$$XYX = -Y \qquad HYH = -Y$$
$$XZX = -Z \qquad HZH = X$$
$$XR_y(\theta)X = R_y(-\theta) \qquad XR_z(\theta)X = R_y(-\theta)$$

表13.2：单量子比特门的一些主要属性。

双量子比特门。控制门

关于多量子比特门的第一件事是，这些操作符是幺正和方阵的，所以与经典门不同，量子门的输入和输出总是相同数量的。 另一种说法是，所有量子门都是自然可逆的，这是可以预期的，因为操作符是幺正的。

最重要的双量子比特门是控制门。 在控制门中，第一个输入比特是控制比特，具有与经典控制位相同的含义。如果它处于$|1\rangle$状态，它将触发作用于第二个比特的门，否则，它将不触发该门，第二个比特将保持不变。 图13.6显示了一个通用示例，该示例中的门被命名为C-U。

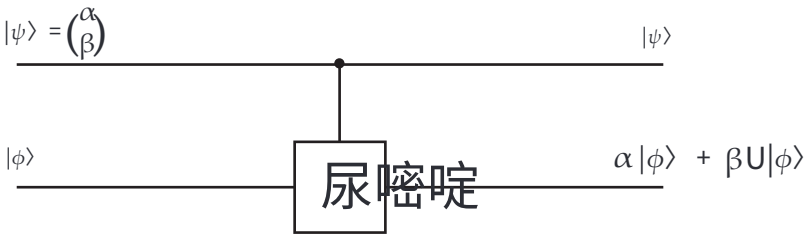


图13.6：通用量子控制门（C-U）。 其中U是表示该门的通用酉矩阵的名称。

C-X（也称为C-NOT）和C-Z（也称为C-Phase）是与我们稍后描述的算法非常相关的两个控制门。CNOT门的流行度使其获得了自己的符号，如图13.7所示。

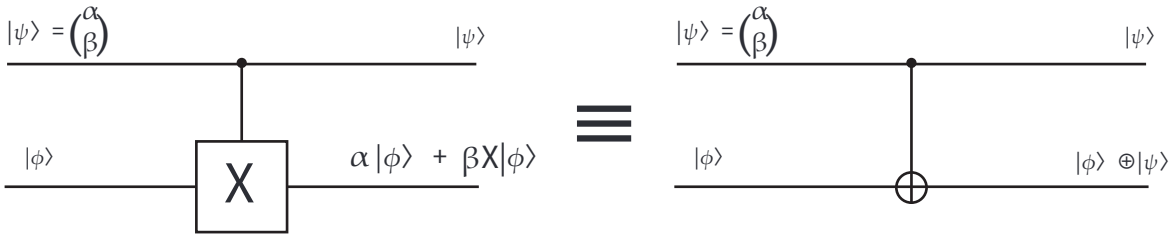


图13.7：CNOT门。

最后，值得回顾一下C-Z门的矩阵表示

$$C-Z=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&|1&0|\\0&0&|0&-1|\end{pmatrix} \tag{13.63}$$

我们强调右下方的方块是Z矩阵。

13.9量子通信

量子力学应用于信息理论的两个最引人注目的应用来自通信领域。 第一个是量子计算机可以在几乎没有时间的情况下破解经典密码。 第二个是量子比特可以进行传送。

在本节中，我们将回顾传送和量子密码学（解决密码破解问题的方法）背后的原理。 正如我们即将看到的，纠缠在这两个应用中起着关键作用。通信中最著名的两个角色：爱丽丝和鲍勃，将在传送中扮演主要角色。
对于量子密码学，夏娃将扮演一个支持性的角色。

13.9.1 传送 - 爱丽丝和鲍勃的故事

当爱丽丝和鲍勃第一次见面时，他们纠缠了一对量子比特 $|\phi_{AB}\rangle$，这是21世纪的传统，人们不再握手。

$$|\phi_{AB}\rangle=\frac{1}{\sqrt{2}}(|0_A\rangle\otimes|0_B\rangle+|1_A\rangle\otimes|1_B\rangle) \tag{13.64}$$

他们度过了一段美好的时光，但是放学后，生活将他们带上了不同的道路。 然而，他们每个人都保留了纠缠对的一部分（在21世纪这是非法的）。现在，这对对看起来像

$$|\phi_{AB}\rangle=\frac{1}{\sqrt{2}}\left(|0_A\rangle\overset{远}{\otimes}|0_B\rangle+|1_A\rangle\overset{远}{\otimes}|1_B\rangle\right(\tag{13.65}$$

爱丽丝现在决定向鲍勃坦白她对他的爱。然而，她害怕被拒绝，她听说量子比特可以“瞬间传送”，所以她决定最好的做法是用量子比特发送一封爱信 $|\psi_L\rangle=\alpha|0_L\rangle+\beta|1_L\rangle$（爱情就在态中）。

为此，爱丽丝小心地将她曾与鲍勃纠缠的比特与她的爱-态-信 $|\psi_L\rangle$放在一个复合系统中。 完整的三比特系统可以使用张量积表示

$$|\phi_A\psi_L\phi_B\rangle=\frac{1}{\sqrt{2}}\left(|0_A\rangle\otimes(\alpha|0_L\rangle+\beta|1_L\rangle)\overset{远}{\otimes}|0_B\rangle+|1_A\rangle\otimes(\alpha|0_L\rangle+\beta|1_L\rangle)\overset{远}{\otimes}|1_B\rangle\right(\tag{13.66}$$

请注意，交叉乘积中的顺序不重要。 只有在乘以两个复合系统时才重要，在这种情况下，我们必须确保每个子系统在我们乘以的每个ket中出现在相同的位置。

爱丽丝现在有一个双量子比特系统，而远在千里之外的鲍勃有一个与爱丽丝的两个量子比特之一纠缠在一起的量子比特。接下来，她拿出了贝尔分析仪1001（见图13.8），这是鲍勃送给她的礼物，她一直珍视着，并将其用于她的两个量子比特。 贝尔分析仪的操作如下，从开始

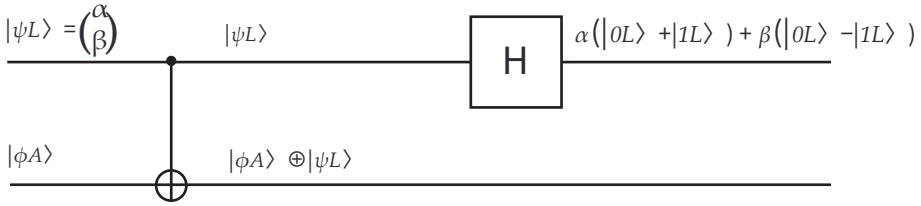


图13.8：贝尔分析仪。

从初始状态

$$|\ \phi_A\psi_L\phi_B\rangle = \frac{1}{\sqrt{2}}\Big[|\ 0_A\rangle \otimes (\alpha\ |\ 0_L\rangle + \beta\ |\ 1_L\rangle) \overset{远}{\otimes} |\ 0_B\rangle + |\ 1_A\rangle \otimes (\alpha\ |\ 0_L\rangle + \beta\ |\ 1_L\rangle) \overset{远}{\otimes} |\ 1_B\rangle\Big[$$

CNOT 门将爱丽丝的量子比特与爱情-凯特-字母耦合，间接地，因为爱丽丝的量子比特与鲍勃的量子比特纠缠在一起。 如果 ψ_L等于 $|\ 0\rangle$，爱丽丝的另一个量子比特将不会被修改（这是下面方程的第一行）。 相反，如果它是 $|\ 1\rangle$，它将被模2加到爱丽丝的另一个量子比特上，换句话说，爱丽丝的另一个量子比特将被翻转（这是下面第二行发生的情况）。 由于 ψ_L本身是一个叠加态，最终发生的是量子比特的一部分保持不变，振幅为 α，而另一部分被翻转，振幅为 β。 因此，在实践中，CNOT 门将叠加态转移到爱丽丝的量子比特上。

$$\begin{aligned}
&= \frac{1}{\sqrt{2}}\alpha\Big(|\ 0_A\rangle \overset{远}{\otimes} |\ 0_B\rangle + |\ 1_A\rangle \overset{远}{\otimes} |\ 1_B\rangle\Big(\otimes |\ 0_L\rangle \\
&+ \frac{1}{\sqrt{2}}\beta\Big(|\ 1_A\rangle \overset{远}{\otimes} |\ 0_B\rangle + |\ 0_A\rangle \overset{远}{\otimes} |\ 1_B\rangle\Big(\otimes |\ 1_L\rangle
\end{aligned}$$

在这一点上，爱丽丝和鲍勃的量子比特都具有最初在爱的态矢中的信息。 Hadamard门根据爱的态矢产生一个新的叠加态，如下所示

$$\begin{aligned}
&= \frac{1}{\sqrt{2}}\alpha\Big(|\ 0_A\rangle \overset{远}{\otimes} |\ 0_B\rangle + |\ 1_A\rangle \overset{远}{\otimes} |\ 1_B\rangle\Big(\otimes \frac{1}{\sqrt{2}}(|\ 0_L\rangle + |\ 1_L\rangle) \\
&+ \frac{1}{\sqrt{2}}\beta\Big(|\ 1_A\rangle \overset{远}{\otimes} |\ 0_B\rangle + |\ 0_A\rangle \overset{远}{\otimes} |\ 1_B\rangle\Big(\otimes \frac{1}{\sqrt{2}}(|\ 0_L\rangle - |\ 1_L\rangle)
\end{aligned}$$

在这一点上，原始的叠加态情书中关于叠加的信息不再在爱丽丝的手中。 然而，要欣赏到这一点，我们需要对交叉乘积进行一些操作和重新排序。 上面的表达式可以分为两个方面，爱丽丝拥有的和鲍勃拥有的，然后它分解成四个具有更清晰解释的项。

$$
\begin{aligned}
&= \frac{1}{2} \mid 0_A\rangle \otimes \mid 0_L\rangle \overset{\text{远}}{\otimes} (\alpha \mid 0_B\rangle + \beta \mid 1_B\rangle) \\
&\quad + \frac{1}{2} \mid 0_A\rangle \otimes \mid 1_L\rangle \overset{\text{远}}{\otimes} (\alpha \mid 0_B\rangle - \beta \mid 1_B\rangle) \\
&\quad + \frac{1}{2} \mid 1_A\rangle \otimes \mid 0_L\rangle \overset{\text{远}}{\otimes} (\alpha \mid 1_B\rangle + \beta \mid 0_B\rangle) \\
&\quad + \frac{1}{2} \mid 1_A\rangle \otimes \mid 1_L\rangle \overset{\text{远}}{\otimes} (\alpha \mid 1_B\rangle - \beta \mid 0_B\rangle)
\end{aligned} \tag{13.67}
$$

我们所做的操作毫无疑问，原始叠加态的所有信息都在鲍勃的一侧。 然而，信息以相位变化（符号）和位翻转产生的所有可能错误的叠加形式到达鲍勃！

爱丽丝现在意识到她无法避免与鲍勃谈论她的爱叠加态信件。 鲍勃拥有这些信息，但要从爱叠加态信件中恢复出精确的叠加态，他需要知道如何解密，只有爱丽丝能告诉他。下一步是无故障传送的关键。

- 爱丽丝测量她的两个量子比特，她将得到以下之一 $\mid 0_A0_L\rangle, \mid 0_A1_L\rangle, \mid 1_A0_L\rangle,$ 或 $\mid 1_A1_L\rangle$具有相等的概率。
- 当爱丽丝进行测量时，鲍勃的量子比特会取四种可能的叠加态之一。 因此，她的测量结果可以帮助鲍勃解码他的比特。
- 如果她测量的是$\mid 0_A0_L\rangle$，她会告诉鲍勃不要对他的量子比特进行任何操作。 如果她测量的是$\mid 0_A1_L\rangle$，鲍勃将需要修正相位（可以使用Z门进行修正）。 如果她测量的是$\mid 1_A0_L\rangle$，消息中的状态已经翻转，鲍勃需要使用位翻转（也称为非，也称为X）门来翻转它们。 最后，如果她测量的是$\mid 1_A1_L\rangle$，鲍勃将需要同时修正相位和位翻转。

总共，爱丽丝告诉鲍勃遵循4种可能性之一，所以她需要向他传达2个经典比特，
然后鲍勃就能够阅读这封爱的凯特信。

现在鲍勃最好准备好理解这封爱的信，因为爱丽丝不再拥有它！注意量子传送如何瞬间传输关于叠加的信息，但爱丽丝需要测量她的系统并告诉鲍勃结果，以便他解开他的比特。从某种意义上说，在传送之后，鲍勃根据爱丽丝测量的综合症进行错误校正；爱丽丝的第一个比特位表示位翻转错误，第二个比特位表示相位错误。 当我们谈论量子纠错时，我们将看到这是量子通信中出现的两种错误类型。

重要的是要认识到爱丽丝只能以经典方式传达综合症给鲍勃，因此，瞬间传送是不可能的，鲍勃将无法在比“远”/“光速”更短的时间内阅读这封爱的信。 还记得打哈欠连接的例子吗？在那里观察者也必须交换数据以确保通信已经发生。

量子传送已经作为无克隆定理的验证为我们服务（爱丽丝不再拥有情书的副本，就像“蜗牛”邮件一样），并帮助我们引入了错误和错误纠正的概念。 我们还看到即时通信是不可能的，因为需要经典通信来纠正错误。

13.9.2 量子密码学

本节仍在建设中。抱歉。

13.10 量子算法

本节仍在建设中。抱歉。

13.10.1 Deutsch Josza

本节仍在建设中。抱歉。

13.10.2 Grover

本节仍在建设中。抱歉。

13.11 量子信息科学

本节仍在建设中。抱歉。

量子纠错码

本节仍在建设中。抱歉。

6.050J / 2.110J 信息与熵

2008年春季