04104-106通信原理II第二讲

March 6, 2007

1 互信息

信息量是不确定性，就是你不可能猜出来的那些东西。对于随机变量X，平均信息量是$H(X)$。

对于两个随机变量，已经知道X后，对于Y我们仍然有一些猜不出来的地方。平均猜不出的程度就是条件熵$H(Y|X)$，它必然小于等于$H(Y)$。也就是说，如果不知道X的具体内容，猜Y猜不出的程度会更大一些。

换言之，已知X使我们对Y多少知道了一些，这个程度是$H(Y) - H(Y|X)$, 它叫互信息（mutal information,MI），记为I(X;Y)。因此，互信息就是已知X后，我们能依据X而猜出的关于Y的信息。

注意记号，我们有时候把自信息$I[P(X = x_i)]$写成$I(X)$, 如同我们会把概率$P(X = x_i)$写成$P(X)$一样。因此需要注意自信息和互信息的符号差别，$I(X,Y)$是自信息，它是$I[P(X = x, Y = y)]$的简写。互信息是$I(X;Y)$。这里一个是分号“;”，一个是逗号“,”。

有如下关系：

$$\begin{aligned} I(X;Y) = I(Y;X) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned} \tag{1}$$

证明：若A、B独立，则$P(A,B) = P(A)P(B)$，$H(A,B) = H(A) + H(B)$。由于$P(X,Y) = P(X)P(Y|X) = P(Y)P(X|Y)$，因此，$Y$和$X|Y$是独立的，$X$和$Y|X$是独立的。因此$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$，代入上式即可得证。#

式(1)表明，已知X猜Y能猜出多少，已知Y猜X也能猜出同样多。课本301页的图7.4.1以文式图（Venn diagram）的方式示出了有关量值的关系。两个圆分别表示$H(X), H(Y)$，是我们对单独的X、Y而言所不知道的东西；X、Y之间有一些共同的东西，这就是互信息$I(X;Y)$。如果知道了X，那么Y中与X共同的这部分信息就知道了，余下的就是条件熵$H(Y|X)$。就整体而言，我们不确定的程度不是$H(X) + H(Y)$，因为这两个不确定中有共同的部分，所以总的不确定程度是$H(X,Y) = H(X) + H(Y) - I(X;Y)$。

2 信源的表达方式、信源的效率

同样的信息可以有不同的表达方式，比如母亲对孩子说“看我不揍你”，其含义是“我要揍你”；某个小伙子A 听B说了某句话后，愤怒地说“你再说一遍试试”，意思是禁止B再这样说。再比如，传送掷筛子的结果{6 6 4 1 2}时，如果改变为{000 000 111 101 110}并无不妥，只要收发都约定好：$000 \to 6, 111 \to 4, \cdots$,即可。

通信中，要把信源的输出$\{X_1, X_2, \cdots, X_L\}$传递到信宿，我们也有多种表达方式可选。当然要选最有效率的方式。因此，实际传送的可能是另外一个序列$\{Y_1, Y_2, \cdots, Y_M\}$。比方说Y是二进制符号，则传送它需要M个bit。

若序列$\{X_i\}$是n进制，则$\{X_1, X_2, \cdots, X_L\}$有$n^L$种不同结果。顺序编号为$0, 1, \cdots, n^L-1$，再转为二进制，则需要$\log_2 n^L = L\log_2 n$ bit。若序列$\{Y_i\}$是m进制，则$\{Y_1, Y_2, \cdots, Y_M\}$有$m^M$种不同结果，顺序编号为$0, 1, \cdots, m^M-1$，再转为二进制，则需要$M\log_2 m$ bit。只要$M\log_2 m < L\log_2 n$，那么传送（或存储）序列$\{Y_k\}$显然更有效率。这就是信源编码的基本思想。

乍一看，这个思想是有问题的：如果$n^L > m^M$，那么序列$\{Y_i\}$的全部结果只能表示序列$\{X_i\}$的一部分结果。正如：原序列是6次掷筛子的结果，总共有$6^6 = 46656$种不同的结果，而我们打算用7张不同花色的扑克牌来表达原序列，后者只能表达$4^7 = 16384$种结果。

奥妙就在于熵这个概念。Shannon说，别看序列$\{X_1, \cdots, X_L\}$有n^L种不同的结果，但因为符号不等概、序列也可能前后相关，所以它的全部信息量只有$H(X_1, X_2, \cdots, X_L) < \sum_{i=1}^{L} H(X_i) = LH(X) < L\log_2 n$，其中$H(X)$是单个符号的熵（假设平稳，则每个符号的熵都一样大）。设$H(X_1, X_2, \cdots, X_L)$的数值为a *bit*，那么用a个YES/No问题就能确定出具体$\{X_1, \cdots, X_L\}$的结果，因此实际只需要传a个bit(a个YES/NO问题的答案)，它要比$L\log_2 n$小得多。

这里的原因在于“典型”“非典型”序列这样的概念。Shannon说，如果$L \to \infty$，那么所有$\{X_i\}$的结果可分为两类，“典型序列”和“非典型序列”，根本不用考虑非典型序列，而典型序列只有$2^{H(X_1, \cdots, X_L)}$个。

例如：考虑1万个掷硬币的结果。所有可能结果有2^{10000}种。如果正反面出现机会不等，比如反面出现机会是2/3，如果你设计的系统压根不打算传送某些结果，比如正面占90%的结果，那么你的系统基本上不会有什么风险。因为正面出现的机会少，所以1万个硬币有9000个正面这种情况虽然有可能，但比起其他“典型”结果来说，可能性可以忽略。信息论对这种风险有具体的估算办法，如课本上的例7.6.2。我们对此不作特别要求，只是再次重申一下信息论的观点：

传输（或存储）序列$\{X_1, X_2, \cdots, X_L\}$（$L$非常大）所需要的比特数最少是$H(X_1, X_2, \cdots, X_L)$，折合到每个符号仅为:

$$H_\infty(X) \triangleq \lim_{L\to\infty} \frac{H(X_1, X_2, \cdots, X_L)}{L} \tag{2}$$

对于序列$\{X_i\}$，传送X_1的信息量是$H(X_1) = H(X)$，是单符号的熵；再传送X_2时，只需传送新增的不确定性$H(X_2|X_1)$。依此类推，传送第L个符号X_L时，只需传送$H(X_L|X_{L-1}, \cdots, X_1)$。当L非常大时，以后每出现的一个符号新增的信息量$\lim_{L\to\infty} H(X_L|X_{L-1}, \cdots, X_1)$并不大。这个增量自然就是序列的总信息量平均到每个符号的结果：

$$\lim_{L\to\infty} H(X_L|X_{L-1},\cdots,X_1) = \lim_{L\to\infty}\frac{H(X_1,X_2,\cdots,X_L)}{L} \triangleq H_\infty(X) \qquad (3)$$

这个结果不仅符合直觉，也是可以证明的。首先

$$\begin{aligned}
\lim_{L\to\infty} \quad & \frac{H(X_1,X_2,\cdots,X_L)}{L} = \lim_{L\to\infty}\frac{H(X_1)+H(X_2|X_1)\cdots H(X_L|X_{L-1},\cdots,X_1)}{L} \\
= \quad & \lim_{L\to\infty}\frac{H(X_L)+H(X_L|X_{L-1}),\cdots,H(X_L|X_{L-1},\cdots,X_1)}{L} \\
\geq \quad & \lim_{L\to\infty}\frac{H(X_L|X_{L-1},\cdots,X_1)+H(X_L|X_{L-1},\cdots,X_1),\cdots,H(X_L|X_{L-1},\cdots,X_1)}{L} \\
= \quad & \lim_{L\to\infty}\frac{1}{L}H(X_L|X_{L-1},\cdots,X_1)
\end{aligned}$$

即$H_\infty(X) \geq \lim_{L\to\infty}\frac{1}{L}H(X_L|X_{L-1},\cdots,X_1)$。上面用到了平稳性和增加条件将使熵变小这两个性质。其次

$$\begin{aligned}
& \frac{H(X_1,X_2,\cdots,X_L)}{L} = \frac{H(X_1)+H(X_2|X_1)\cdots H(X_L|X_{L-1}\cdots X_1)}{L} \\
= \quad & \frac{H(X_1)+\cdots+H(X_M|X_{M-1}\cdots X_1)+H(X_{M+1}|X_M\cdots X_1)+\cdots+H(X_L|X_{L-1}\cdots X_1)}{L} \\
\leq \quad & \frac{H(X_1)\cdots H(X_M|X_{M-1}\cdots X_1)+H(X_{M+1}|X_M\cdots X_2)+\cdots+H(X_L|X_{L-1}\cdots X_{L-M+1})}{L} \\
= \quad & \frac{H(X_1)+\cdots+H(X_{M-1}|X_{M-2}\cdots X_1)+(L-M+1)H(X_M|X_{M-1}\cdots X_1)}{L}
\end{aligned}$$

先固定M,令$L\to\infty$，再令$M\to\infty$，上式结果就是$H_\infty(X) \leq \lim_{L\to\infty}\frac{1}{L}H(X_L|X_{L-1},\cdots,X_1)$。因此必然有$H_\infty(X) = \lim_{L\to\infty}\frac{1}{L}H(X_L|X_{L-1},\cdots,X_1)$

综上所述，对于信源$\{X_i\}$，如果不采用任何压缩技术，平均每个符号需要$\log_2 n \triangleq H_0(X)$比特。如果采用高效的压缩技术，最好情况下有望每个符号平均只需要$H_\infty(X)$比特。压缩比极限可达到$\frac{H_\infty(X)}{H_0(X)}$。这个比值也叫信源的效率$\eta$，相应定义信源的冗余度为

$$R = 1-\eta = \frac{H_0(X)-H_\infty(X)}{H_0(X)} = 1-\frac{H_\infty(X)}{H_0(X)}$$

实际信源的特征往往是：符号不等概，符号之间有很强的相关性。因此它一定可以压缩，具体方法的压缩能力或许有区别，最大的压缩程度可由η或R衡量。计算$H_\infty(X)$需要用到无限维的概率分布（因为计算熵$H(X_1,\cdots,X_L)$或者条件熵$H(X_L|X_{L-1},\cdots,X_1)$都需要L维概率$P(X_1,X_2,\cdots,X_L)$）。这些概率可以用统计方法获得。表7.3.4是统计得到的H_∞。

上述最好的方法充分利用了信源序列的相关性。对于第L个符号，它充分扣除了过去所有符号在X_L上的贡献，只传送纯新的增量信息。如果我们只考虑A个符号，即对于第L个输出符号，只扣除前$A-1$个符号的贡献，那么效率就会低一些，平均每符号最少需要的比特数将是$H_A(X) \triangleq H(X_L|X_{L-1},\cdots,X_{L-A+1})$。

$H_2(X) = H(X_L|X_{L-1})$只考虑前一个符号的贡献；$H_1(X) \triangleq H(X)$只考虑单个符号，不考虑相关性，但考虑了不等概特性；$H_0(X)$则连不等概都不考虑，即不进行压缩。下面的不等式是显而易见的

$$0 \le H_\infty(X) \le \cdots \le H_2(X) \le H_1(X) \le H_0(X) = \log_2 n \tag{4}$$

3 信源编码

上面我们已经看到，信息论的基本意思是，对于信源我们只需要传送它的熵就可以了。信息论对此有更为严格和精确的表述。我们略讲一二。

3.1 等长无失真编码

对于不等概的独立序列信源，每符号最少可以压缩到$H_1(X)$（因为独立，所以没有相关性可利用，$H_\infty = H_1$）。一种做法是：将信源序列分为等长的组，每组长度比如是L。每组的可能结果有n^L种，扣除非典型序列，对剩下的进行编码（就是进行二进制整数编号）。虽着L趋于无限长，这种方法可以趋向极限的效率: 每符号所用的比特数仅为$H(X)$。

无失真的意思是，用编码结果$\{Y_i\}$能够完全复原出原序列$\{X_i\}$(不能复原出非典型序列结果，但这个概率可以忽略)。也思就是说$I(X;Y) = I(X)$。

3.2 变长无失真编码

如上的方法实际上是不现实的，经过一番计算我们发现，要使这种编码是保险的，分组长度必须非常长。保险的意思是，非典型序列的概率确实可以忽略。课本例7.6.2就是这种计算的一例，虽然如何计算不在我们的要求之内，但从这个例子的结果能看到分组长度到底需要多长。

一种简单，而且有效的方法是采用变长编码。其思想是，出现频率高的符号用短码，出现概率低的用长码。理想状态下，这样做效果非常不错。

变长编码的一个经典例子是Huffman编码。

3.3 限失真信源编码

前面谈论的是离散信源。对于连续信源，要想精确表达信源，熵是无穷大，这当然是做不到的。例如，对于均匀分布在$[0,1]$内的X, 如果按间隔0.1量化，结果有10种，熵是$1Det = \log_2 10bit$；如果量化间隔是0.01，熵就是$\lg 100 = 2Det$。可见，真要想无限精确地传送X的话，信源的熵将是无限大。

好在对许多应用来说，我们没有必要无限精确。比如一个听MP3的学生不可能抱怨说，它听到的某个歌的某个抽样值比原音的电压值$3.04V$低了$0.000001V$，人类的耳朵不可能有这么高的分辨率。因此，我们退而寻求这样的编码，它把原始信息X映射成Y，虽然$I(X;Y) < H(Y)$（就是说，我们得知Y并不能完全精确的复原X，否则就是$I(X;Y) = I(X)$），但由Y复原的结果和X相比，“失真”完全可以满意。失真的意思是$Y \neq X$，失真的程度D是一个数，我们用它来说X和Y不相等的程度。一般用均方失真（但也可以用其他定义）：$D = E[(Y-X)^2]$。

对于可允许的失真度D，存在许多将X映射为Y的方法，其中的互信息$I(X;Y)$就是该种方法所需要的传输量，也称为信息率（单位一般是每符号多少bit）。如果数学模型齐备的话，信息论能算出所有这些失真符合要求的方法最小的互信息，这个值定义为“率失真函数”，记为$R(D)$，它是给定可允许的失真为D的条件下，最少需要传送的信息量。它是一个函数，自变量是失真程度D，函数值是信息率R，所以叫率失真函数。

4 利用序列的相关性

实际信源都是有相关性的，如前所述，如果能够理性地扣除随机变量之间的关联，信源实际的熵要比看上去小得多。因此，实际应用中的压缩技术都在这方面花费了很大力气。

一种方法是预测编码，在发送样值X_L之前，它先预测一下这个值可能是多少，预测的结果当然不可能绝对准确(X_L毕竟会有新的信息)，我们再把预测误差$X_L - \hat{X_L}$传过去。因为误差毕竟比较小，所以传它的时候，量化比特就可以少一些。

另一种方法是变换编码。一个随机向量经过适当线性变换后，可以变得彼此不相关。这使我们能把相关信源改造成不相关信源。因为熵不大，所以变换后的表现是，某些值可能非常小，以致于忽略它影响不大。这样我们就能大大减少传输量。