信息论第四次小测

中国科学技术大学《信息论 A》006125.01 班助教组

2024 年 4 月 15 日

第 1题

求证：

$$(1-xy)^n \leq 1-x+e^{-ny},\ \forall\ 0 \leq x,y \leq 1 \tag{1}$$

证明: 记$f(x,y)=(1-xy)^n$，对$n\geq 2$，有：

$$\frac{\partial^2 f}{\partial x^2}=n(n-1)(1-xy)^{n-2}y^2 \geq 0 \tag{2}$$

，对于$n=1$，不难验证$\frac{\partial^2 f}{\partial x^2}=0$。从而$f(x,y)$是关于$x$的凸(convex)函数，由Jensen不等式：

$$\begin{aligned}(1-xy)^n &\leq (1-x)f(0,y)+xf(1,y) &(3)\\ &=(1-x)(1-0)^n+x(1-y)^n &(4)\\ &\leq 1-x+x(e^{-y})^n \leq 1-x+e^{-ny} &(5)\end{aligned}$$

对于$n\geq 2, y>0$，式子是严格凸的，从而第一个等号只能在两个端点取到。而第二处不等式$1-y\leq e^{-y}$取等又要求$y=0$，紧接着第三处取等要求$x=1$，从而当$(x,y)=(1,0)$取到等号。

我们特别关心取等条件是$(x,y)=(1,0)$，也是因为他在我们4.4.4节的取等中确实发挥了作用：我们在讲义的推导中确实令$x\to 1$，并允许$y\to 0$，然后对于任一组(x,y)，令$n\to\infty$，最后使得$0\leq(1-xy)^n\leq 1-x+e^{-ny}\to 0$，从而$(1-xy)^n\to 0$，完成了可达性的证明。 □

第 2题

我们在*4.4.2*里所提出的编码规则是最优的吗？你觉得可以怎样改进获得更小的平均失真？

答: 可能不是，我们4.4.2中提出的编码规则如下：对于每一个信源，倘若能够在码本中寻找到一个下标w满足：

$$d(\underline{s}, \mathcal{C}(w)) \le D + \epsilon, \tag{6}$$

$$I(S;\hat{S}) - \epsilon \le i(\underline{s}; \mathcal{C}(w)) \le I(S;\hat{S}) + \epsilon, \tag{7}$$

详细表达附在下：

$$d(\underline{s}, \mathcal{C}(w)) = \frac{1}{n}\sum_{i=1}^{n} d\left(s_i, \mathcal{C}_i(w)\right) \tag{8}$$

$$i(\underline{s}; \mathcal{C}(w)) = \frac{1}{n}\log_2 \frac{P_{\underline{S},\underline{\hat{S}}}(\underline{s}, \mathcal{C}(w))}{P_{\underline{S}}(\underline{s}) P_{\underline{\hat{S}}}(\mathcal{C}(w))} \tag{9}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\log_2 \frac{P_{S,\hat{S}}\left(s_i, \mathcal{C}_i(w)\right)}{P_S\left(s_i\right) P_{\hat{S}}\left(\mathcal{C}_i(w)\right)}. \tag{10}$$

若存在多个满足，那我们就选取一个w即可，如果没有满足，我们就选第一个。（为什么是第一个？因为他总得有一个吧，他可以是任意的！我们在4.4.2的后续证明中会令这个概率趋于0，表示这种无法有效编码的情况不太会发生。）

这种方式怎么会是最优的呢？要明白，我们编码的目的就是在码本$\mathcal{C}$给定之后，找到一组从信源到码字的映射，使得失真尽量小即可。我们提出几个点：

1. 我们在课上多次提到，第二个令人感到困惑的条件(叫做信息密度的约束)是在推导中需要的。倘若我们去掉这个条件，想一下，他会使得我们的失真约束更松，满足条件的w更好找到（毕竟条件变少了），那么是不是更容易将这个信源编为一个满足失真的码字？
2. 第一个约束，我们对所有小于D的失真一视同仁，然而事实上，当然是越小越好了。而条件只要求小于D即可，这是因为在长度n趋于无穷长的时候，绝大多数的信源会被编码到D左右的失真。（不好描述，几何意义是高维球壳占绝大多数体积）但如果我们想获得更小的失真，对于每一个信源，我们要求找到一个下标w对应的码字使得失真最小，而不仅仅是小于等于D，这也是一种改良。
3. 此外，当没有满足的，我们都选择第一个，这也是一个过于暴力的方法，证明中敢这么做也是因为没有满足两个条件的信源出现的概率被证明趋于零。而实际编码中，我们也希望就算找不到一个优良的估计，我们也去寻求一个使得失真尽量小的码字。

所以，我们将条件更改为：

$$w^* = \arg\min_{w} d(\underline{s}, \mathcal{C}(w)) \tag{11}$$

这反而更符合大家的直觉，我们上面的那些条件，是出于祖师爷敏锐的嗅觉：他知道在$n \to \infty$时，采用上面那些复杂、可能失真更大的条件，仍然可以构造出可达的编码方式。 $\square$

第 3题

基于二，计算对于*Bernoulli(1/2) DMS* $\mathcal{S}$, *Hamming distortion*下面，当$n=4$, 码本为$\mathcal{C}=\{[0,0,0,0],[1,1,1,1]\}$时的平均失真。

分析: DMS:Discrete Memoryless Source 离散无记忆信源。我希望能让大家听懂，我们接着这个例子讲上面的事情。如果采用上面的编码方式：

$$d(\underline{s},\mathcal{C}(w))=\frac{1}{n}\sum_{i=1}^{n}d\left(s_i,\mathcal{C}_i(w)\right) \tag{12}$$

$$i(\underline{s};\mathcal{C}(w))=\frac{1}{n}\log_2\frac{P_{\underline{S},\underline{\hat{S}}}(\underline{s},\mathcal{C}(w))}{P_{\underline{S}}(\underline{s})P_{\underline{\hat{S}}}(\mathcal{C}(w))} \tag{13}$$

$$=\frac{1}{n}\sum_{i=1}^{n}\log_2\frac{P_{S,\hat{S}}\left(s_i,\mathcal{C}_i(w)\right)}{P_S\left(s_i\right)P_{\hat{S}}\left(\mathcal{C}_i(w)\right)}. \tag{14}$$

第一个式子就是当我们随便发了一个信源，比如$[0,0,0,1]$，在$\mathcal{C}=\{[0,0,0,0],[1,1,1,1]\}$中找一个码字来代表他。倘若我们设置$D=2$，那么只能编码为$[0,0,0,0]$；设置$D=3$，按照上述准则，我们可以任选一个——但事实上我们更应当果断地选择使得失真更小的码字。

第二个式子在这里发挥不了作用，因为我们没有解率失真问题，无法得到$P_{s,\hat{s}}$。但在我们实际编码中，为什么要考虑这个，对不对？我们只是想让失真最小。

解：按照准则$w^*=\arg\min_w d(\underline{s},\mathcal{C}(w))$，我们给出的信源编码策略如下表，平均失真计算如下：

$$D=\mathbf{E}[d(\underline{S},\underline{\hat{S}})]=\mathbf{E}\left[\frac{1}{n}\sum_{i=1}^{n}d(S_i,\hat{S}_i)\right] \tag{15}$$

$$=\frac{1}{4}\cdot\frac{1}{16}[2\cdot 0+8\cdot 1+6\cdot 2]=\frac{5}{16}. \tag{16}$$

□

注 1. 对于题目中给出的对于码长为$n=4$，码率为$R=\frac{\log_2 2}{n}=\frac{1}{4}$的率失真码，我们构造出的最小失真为$\frac{5}{16}\approx 0.32$。从而$(R,D)=(0.25,0.32)$是一对可达率失真对。

而*Bernoulli(1/2) DMS* 信源在*Hamming*失真度量下的率失真函数由讲义*(4.26)*给出：

$$R(D)=h_2(1/2)-h_2(D)\ \textit{if}\ 0\le D\le 1/2,\ \textit{and}\ 0\ \textit{otherwise.}$$

从而对于$R=1/4$，我们可以解出$R(0.2145)=1/4$，即对于$n\to\infty$时，我们理论上可以构造出一种压缩方式，使得$(R,D)=(0.25,0.2145)$是可达的。

这一章公式很难，证明很难，但是我希望大家可以体会到这一章在做什么事情，每一步在做什么，又是如何用信息论的语言去描述的。我们考察的计算往往很简单，正如第三题，只是需要一点理解，但正是这理解上的一点困难，导致很多同学做不出来。

Source	Probability	Codeword	Distortion
0000	1/16	0000	0
0001	1/16	0000	1
0010	1/16	0000	1
0011	1/16	0000 or 1111	2
0100	1/16	0000	1
0101	1/16	0000 or 1111	2
0110	1/16	0000 or 1111	2
0111	1/16	1111	1
1000	1/16	0000	1
1001	1/16	0000 or 1111	2
1010	1/16	0000 or 1111	2
1011	1/16	1111	1
1100	1/16	0000 or 1111	2
1101	1/16	1111	1
1110	1/16	1111	1
1111	1/16	1111	0

表 1: 0000 or 1111这种写法不规范，我们当然应该直接决定一个，这里只是为了和另一类失真1的做区分，具体的计算方法，大家自己去验证，记得从定义出发。