04104-106通信原理II第七讲

April 10, 2007

1 Preliminaries

1.1 关于位置编号的约定

位置编号是一个容易让初学者感到混乱的问题。

编码器的输出是n个比特，比如(1000111)。对于不同的输入信息，编码结果也不同。为了分析研究，我们自然会想到用代数的方法，引入变量符号来记述这7个比特值，比如(u, v, w, a, b, c, x)。也可以用一个单字母符号配上不同的下标：$(b_a, b_b, b_u, b_x, b_y, b_w, b_z)$。用数字做下标当然更为便利，此时也有许多标记方法，如：$(b_1, b_2, \cdots, b_7)$, $(b_6, b_5, \cdots, b_0)$, $(b_7, b_6, \cdots, b_1)$ 等等。由于这些下标仅仅是一个label，所以写成$(b_3, b_2, b_1, b_6, b_4, b_5, b_0)$ 也无不可。

总之，如何对比特位置进行编号只是我们自己如何描述它的问题，与被研究的问题无关。任何人都可以按照自己的偏好来标记。不同的文献对此也没有共识。本课约定，在线性分组码中，默认的编号规则是从左到右对应位置编号从大到小，最右的那个比特编为0。这种规则也就是MSB(Most Sigmificant Bit，最高有效位)在左。如果你更习惯其他编号的话，也完全可以。不过在考试中，如果不使用默认编号的话，必须加以注明，以避免误解。

实际当中，这n个比特发送的时间次序有可能是从左到右、从右到左或者其他。如果信道没有差异的话，任何次序都是无关紧要的。

1.2 Galois Field

我们日常所用的加减乘除都是二元运算的一个特例，也即二元函数。对于集合Ω，如果任取两个元素$a, b \in \Omega$，将(a, b)对应到某个$c \in \Omega$，即$c = f(a, b)$，就是定义了一种二元运算。我们可以随意设计一个记号来标记这种运算，比如$a \boxplus b = c$，也可以给它起一个名字，比如叫“*法”。我们平常所用的算术中，$1 + 2 = 3$是把$(1, 2)$对应给了3，$+$是我们为了标记这个二元运算所设计的符号，我们还将其名之为“加法”。

实数以及加减乘除这些算术规则构成了我们非常熟悉的算术系统。对于任意集合Ω，同样也可以建立一个算术系统，只须定义出相应的二元运算规则即可。

设有集合Ω，在这个范围内定义了o_A，o_B这样两种二元运算。不妨将其命名为“加法”和“乘法”，并分别使用记号$+$和$\times$，约定$a \times b$可以简写成ab。这样做不光是为了复用这些符号和术语（以降低记号成本），更因为我们所熟悉的算术只是一个特例。

对于集合Ω中的元素，可以采用任何一种自认为方便的方式来标记，比如$a, b, \cdots$。如果Ω是有限的，或者可数的，也可以用ω_i这样的方式来标记其元素，其中i是整数。当然也可以直接用整数来标记，即3这个符号代表Ω中的某一个元素，4则是另一个符号。如同用学号来标记学生。这里的3,4都只是些符

号，除非对运算、排序等进行了定义，否则它们不具有我们平常所说的“数”的意义。

根据本课的需要，我们需要简要学习一种叫做域（field）的数学系统，它有如下规则（其中的某些规则可以用其它规则导出）

1. 存在唯一一个特定的元素$a \in \Omega$，使得对于任何$b \in \Omega$，有$a + b = b$。我们用“0”这个符号来标记这个特殊的元素。

2. 存在唯一一个特定的元素$x \in \Omega$，使得对于任何$b \in \Omega$，有$x \times b = b$。我们用“1”这个符号来标记这个特殊的元素。

3. 对于任意的$a \in \Omega$，存在唯一一个元素$b \in \Omega$，使得$a + b = 0$，称此b为a的负数。如果必要的话，我们用记号$-a$来表示它。

4. 对于任意的$a \in \Omega$，存在唯一一个元素$c \in \Omega$，使得$a + c = 0$，称此c为a的逆。如果必要的话，我们用记号a^{-1}来表示它。

5. 所定义的加法和乘法满足交换律：即对于$a, b \in \Omega$，$a + b = b + a$，$ab = ba$；

6. 所定义的加法和乘法满足结合律：即对于$a, b, c \in \Omega$，$(a + b) + c = a + (b + c)$，$(ab)c = a(bc)$；

7. 所定义的乘法对加法有分布律：即对于$a, b, c \in \Omega$，$(a + b)c = ac + bc$；

若Ω是实数$\mathbb{R}$，那么我们日常使用的加减乘除四则运算构成了实数域。如果Ω只有有限个元素，即$|\Omega| = q < \infty$，这样的域就是伽罗华域，记为$\mathrm{GF}(q)$。

对于$\mathrm{GF}(2)$，$\Omega = 0, 1$，加法定义为逻辑异或，乘法定义为逻辑与。

1.3 线性代数

N长实数向量$\mathbf{x} = (x_1, x_2, \cdots, x_N)$的每个元素都属于实数集合$\mathbb{R}$。$\mathbf{x}$属于实数的扩展域$\mathbb{R}^N$。两个向量的加法是逐元素相加。向量$\mathbf{x} \in \mathbb{R}^N$ 和标量$\alpha \in \mathbb{R}$的乘积定义为$\mathbf{x}$的每个元素乘以α：$\alpha\mathbf{x} = (\alpha x_1, \alpha x_2, \cdots, \alpha x_N)$。两个向量$\mathbf{x}_1, \mathbf{x}_2$ 的线性组合是$\alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2$，其中$\alpha_1, \alpha_2 \in \mathbb{R}$。

现在考虑n长的二进制向量$\mathbf{b} = b_{n-1}, b_{n-2}, \cdots, b_0$，其元素属于$\mathrm{GF}(2)$，$\mathbf{b}$属于扩展二元域$\mathrm{GF}(2^n)$。标量$a \in \mathrm{GF}(2)$和向量$\mathbf{b} \in \mathrm{GF}(2^n)$的积定义为$\mathbf{b}$的每个元素都乘上$a$：若$a = 0$, 则$a\mathbf{b} = (0, 0, \cdots, 0)$，若$a = 1$，则$a\mathbf{b} = \mathbf{b}$。两个向量$\mathbf{b}_1, \mathbf{b}_2$ 的线性组合是$a_1\mathbf{b}_1 + a_2\mathbf{b}_2$，其中$a_1, a_2 \in \mathrm{GF}(2)$。

将k个向量$\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k \in \mathrm{GF}(2^n)$做线性组合，得到向量$\mathbf{c} = u_{k-1}\mathbf{g}_1 + u_{k-2}\mathbf{g}_2 + \cdots + u_0\mathbf{g}_N$。对于不同的组合系数$u_{k-1}, u_{k-2}, \cdots, u_0$，组合结果有可能不同。将所有可能的组合结果装入一个集合C，则$C \subset \mathrm{GF}(2^n)$。称$C$是由$\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k$张成的，称$\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k$是$C$的一组基。$C$至多有$2^k$个元素，且必然包括全零向量$(0, 0, \cdots, 0)$。如果$\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k \in \mathrm{GF}(2^n)$线性不相关，则$C$中的元素个数是$2^k$个。线性不相关的意思是说，$\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k$中的任何一个都不可能是其他向量的线性组合。对于我们所考虑的二进制情形，两个向量线性不相关=两个向量不相同，k个向量线性不相关=任何一个都不是其它若干个的和。

2 线性分组码

分组码的编码器的输入是k个比特$\mathbf{u}=(u_{k-1},u_{k-2},\cdots,u_0)$，输出是$n$个比特$\mathbf{c}=(c_{n-1},c_{n-2},\cdots,c_0)$。从数学角度看，编码器的功能就是一个函数$\mathbf{c}=f(\mathbf{u})$。如果这个函数是线性函数，这样的编码器就是“线性分组码”的编码器。所谓线性函数是指，对于任意的$\mathbf{u}_1,\mathbf{u}_2$，总有$f(\mathbf{u}_1+\mathbf{u}_2)=f(\mathbf{u}_1)+f(\mathbf{u}_2)$，即和的编码结果等于编码结果之和。也可以等价地说成是“$\mathbf{c}$中的任何一个比特都是$(u_{k-1},u_{k-2},\cdots,u_0)$中某些比特之和”，或者“$\mathbf{c}$是$\mathbf{u}$的线性变换”。因此，编码关系可以写成矩阵形式：

$$\mathbf{c}=\mathbf{u}G \tag{1}$$

其中的G是一个$k\times n$的矩阵，其元素属于GF(2)。给定G便给定了函数f，便给定了编码器。此矩阵称作该编码器的“生成矩阵”：$\mathbf{c}$是通过G产生的。

记G的k个行为向量$\mathbf{g}_1,\mathbf{g}_2,\cdots,\mathbf{g}_k$，那么(1)可以写成$\mathbf{c}=u_{k-1}\mathbf{g}_1+u_{k-2}\mathbf{g}_2+\cdots+u_0\mathbf{g}_k$，即$\mathbf{c}$是$G$的某些行的和。比如，对于(7,4)线性分组码，$G$有4行，每行是一个长为7的行向量，信息$\mathbf{u}=(1100)$的编码结果是$G$的第一行和第二行之和。

对于正常的编码，G的各行是线性不相关的，这一点保证不同的信息分组$\mathbf{u}$的编码结果不同。由于$\mathbf{u}$ 有2^k种不同，因此$\mathbf{c}$也有2^k种不同，其中包括n长的全零向量$\mathbf{0}=(0,0,\cdots,0)G$。记$C$为所有$\mathbf{c}$的集合，则$C$是$G$的行所张成的线性空间。很明显$G$的各行一定在$C$中。

从C中任意选出k个线性不相关，都可以张成C，因此，用C中任意k个线性不相关的行作为编码器的生成矩阵，编码结果的集合都是相同的（但信息到码字的对应关系不同）。若不同编码器所生成的码字集合相同，就说它们是等价的。由于序号的编号次序并不重要，因此若两个编码器的结果的差别只是次序交换的话，它们也是等价的。

可对照第6章信号星座的概念来理解：8PSK有8个点，每个点对应3个比特。无论比特和点之间怎么对应（比如格雷映射或者自然映射），它们都是8PSK，其符号错误率是一样的。在线性分组码中，如果集合C相同，译码后码字的错误率就是相同的。

如果原始信息$\mathbf{u}$能够原样出现在$\mathbf{c}$中，这样的编码叫“系统码”，直接出现在$\mathbf{c}$中原始信息位叫“系统位”，其余叫“校验位”。由于次序并不重要，所以我们默认假设系统码的系统位出现在左边（高位）。即编码结果一定可以写成$(\mathbf{u},\mathbf{p})$的形式，其中$\mathbf{p}=(p_{r-1},p_{r-2},\cdots,p_0)$是校验位，校验比特的个数是$r=n-k$。从$\mathbf{u}G=(\mathbf{u},\mathbf{p})$可知，系统码的$G$一定具有$(I,Q)$的形式，其中$I$是单位阵。

任意给定一个G，如果它不是系统码的生成矩阵，一般可以通过初等行变换将其变换为系统码形式，如果这样不行的话，再配合列交换总是可以的。具体操作的示例见课本391页。如果是只通过初等行变换得到的，那么结果是唯一的。

线性分组码编码器的实现是一个矩阵乘法，见课本395页的图9.2.1和图9.2.2。

线性分组码的最小码距是非全零码字中最小的码重。对于任意的(n,k)分组码，得到d_{min}需要对所有2^k个码字进行两两比较，需要比较$(2^k-1)!$次。对于线性分组码，只需要判断2^k-1个非全零码字的码重就能得知d_{min}。