编码设计中的子集划分问题

摘要

在本数学建模中，我们分析了 BER 最小值的上下界，并用二分和随机结合的方法给出了一个分组算法。

关键词：二分, 随机

1 问题重述

通信系统在当今社会中扮演了十分重要的角色。其中，信息的传递是传输信息串实现的，而每一个信息串由若干个比特（0 或 1）组成。显然，由于实际环境中的噪音，信息传递不可能完全准确。在本题中，我们只考虑一种较为简单的噪音：比特是通过二元对称信道传输的。在二元对称信道中，发送一个比特，接收到的比特有概率 p 与原来不同。假定 $p \in (0,1)$ 是一个常数，且每一个比特的发送和接收是独立的。

设 $V = \{0,1\}^n$ 是含 n 个比特的信息全体。给定 $k < n$，划分 V 为 $m = 2^k$ 个集合 $V_1, V_2, \ldots, V_m$, 即 V_i 两两不交，且 $\bigcup_{i=1}^m V_i = V$。对每个 V_i，选取一个 x_i 作为其代表。以后，我们仅发送这些选定的代表。若发送 x_i，接收到的信息为 y，则解码为 y 所在集合 V_j 的代表 x_j。记 e_i 为“错误解码”的概率，即 $e_i = \mathrm{Prob}\,(x_j \neq x_i)$。

定义 $\mathrm{BER} = \max_{1\le i\le m} e_i$。我们需解决以下两个问题：

1. 对给定的 $r = k/n$，设计 $V_1, \ldots, V_m$ 及代表 $x_1, \ldots, x_m$，使得 BER 尽可能小。
2. 设计一套算法，对输入的 n 和 k 能够给出相应的 $V_1, \ldots, V_m$ 和 $x_1, \ldots, x_m$, 使得 BER 尽可能小。以 $k = 24$, $n = 32$, $p = 0.1$ 为例进行分析。

2 问题分析

本题中两个问题的主要目标都是使 BER 的值尽可能小，因此我们先做整体上的分析。

首先我们假设已经设计好 $V_1, \ldots, V_m$，并设 $V_i = \{v_{i1}, \ldots, v_{ia_i}\}$, 这里 $a_i = \#V_i$ 是 V_i 的元素个数，满足 $\sum_{i=1}^m a_i = 2^n$。我们的目的是选择合适的 $x_i \in V_i$，这也即要求发送 x_i，接收到的信号仍在 V_i 内的概率最大。为此，我们可以计算发送出 v_{ij} 而接收到 v_{il} 的概率 p^i_{jl}，这里上标 i 表示考虑的集合是 V_i。显然，$p^i_{jl} = p^i_{lj}$。这样，发送 v_{ij} 后接收到的信号仍在 V_i 的概率为 $P_{ij} = \sum_{l=1}^{a_i} p^i_{jl}$。取 x_i 为使得 P_{ij} 最大的 v_{ij} 即可。于是 $\mathrm{Prob}\,(x_j \neq x_i) = e_i = 1 - \max_{1\le j\le a_i} P_{ij}$，则 $\mathrm{BER} = \max_{1\le i\le m} e_i$。

从以上分析可以看出，一旦设计好 $V_1, \ldots, V_m$，则可以设计出合适的 $x_1, \ldots, x_m$。现在需要考虑如何设计 $V_1, \ldots, V_m$。

3 模型的建立与求解

3.1 问题一的求解

我们将给出 BER 最小值的一个估计。

$\boxed{\text{引理}}$ 代数式 $\binom{n}{q} p^q (1-p)^{n-q}$ 的最大值在 $q = \lceil (n+1)\,p \rceil - 1$ 时取得。

证明 记 $f(q) = \binom{n}{q} p^q (1-p)^{n-q}$, 则

$$\frac{f(q+1)}{f(q)} = \frac{p\,(n-q)}{(1-p)\,(q+1)}$$

由此可知 $q \le (n+1)p-1$ 时 $f(q)$ 单调增加，否则 $f(q)$ 单调减少。 $\square$

记 $q_0 = \lceil (n+1)p \rceil - 1$, $p_0 = \binom{n}{q_0} p^{q_0}(1-p)^{n-q_0}$。因为任何 p_{jl}^i 不超过 p_0, 于是我们可以给出 BER 的最小值的一个下界。

$$
\begin{aligned}
\mathrm{BER}_{\min} &= \max_{1\le i\le m} e_i \\
&\ge \frac{\sum_{i=1}^m e_i}{m} \\
&= \frac{m - \sum_{i=1}^m \sum_{l=1}^{a_i} p_{jl}^i}{m} \\
&\ge \frac{m - p_0 \sum_{i=1}^m a_i}{m} \\
&= \frac{m - p_0 \cdot 2^n}{m} \\
&= 1 - p_0 \cdot 2^{n-k}
\end{aligned}
$$

下面来估计 BER 的最小值的上界。我们取 V 的一个划分，使得诸 e_i 相等。取 t 是使得 $2\sum_{l=1}^t \binom{n}{l} \le 2^{n-k}$ 的最大整数，则显然 $t \le n/2$。通过估计各 $f(s)$ 的个数可得

$$
e_i \le 1 - \sum_{s=0}^{t} \binom{n}{s} \left(p^s (1-p)^{n-s} + p^{n-s}(1-p)^s\right)
$$

故

$$
\mathrm{BER}_{\min} \le e_i \le 1 - \sum_{s=0}^{t} \binom{n}{s} \left(p^s (1-p)^{n-s} + p^{n-s}(1-p)^s\right)
$$

3.2 问题二的求解

由第一问

$$
1 - p_0 \cdot 2^{n-k} \le \mathrm{BER}_{\min} \le 1 - \sum_{s=0}^{t} \binom{n}{s} \left(p^s (1-p)^{n-s} + p^{n-s}(1-p)^s\right)
$$

通常 $\mathrm{BER}_{\min}$ 的上下界相差不大，随机方法有很好的效果，算法如下：

1. 定义数池的初始值 $\mathrm{Pool} = V = \{0,1\}^n$
2. 在 $[l,r]$ 区间内二分 $\mathrm{BER}_{\min}$, 二分值记为 m
   (a) 从数池 Pool 中随机取出 2^{n-k} 个数形成一组 V_i
   (b) 计算这组 V_i 的 e_i
   (c) 如果 $e_i > m$, 则清空所有分好的组，从头重新随机分配。
   (d) 若随机分配了若干次，仍有 $e_i > m$, 则在 $[m,r]$ 继续二分；否则在 $[l,m]$ 继续二分

(e) 当 l, r 相近时 ($r - l < \mathrm{eps}$, 其中 eps 是一个小量), 则可退出二分

3. $\mathrm{BER}_{\min} = r$, V 的分组即为相应的随机分配结果。

Pool 是一个数据结构，为优化时间复杂度，其要满足：

1. $O(1)$ 插入、删除

2. $O(1)$ 返回其中的一个随机数值

实现方法是[1]，考虑一个由哈希表 H 和数组 A 组成的数据结构。哈希表的键是数据结构中的元素，值是它们在数组中的位置。

1. insert(value): 将值附加到数组中，并让 i 成为它在 A 中的索引。设置 H[value]=i。

2. remove(value)：我们将用 A 中的最后一个元素替换 A 中包含 value 的单元格。设 d 是数组 A 中的最后一个元素。设 i 为 H[value]，即要删除的值在数组中的索引。设置 A[i]=d, H[d]=i，将数组大小减一，并从 H 中删除 value(令 H[value]=0)。

3. getRandomElement()：让 r=random(A.size())。返回 A[r]。

由上，时空复杂度均为 $O(2^n)$, 时间复杂度中具体的常数大小需要由随机的次数决定。

参考文献

[1] r0u1i, *Data structure: insert, remove, contains, get random element, all at O(1)*, URL: https://stackoverflow.com/a/5684892/18156586 (2022/6/10).