# 《信息论基础》编程作业2

## Coding task – Tunstall Encoder/Decoder

### 要求

* Based on Prof. Jim Massey’s notes on Tunstall codes, implement (in whichever programming language you ﬁnd comfortable with) a binary Tunstall codec (encoder/decoder) and test its performance using randomly generated source symbol strings.
* 算法Tunstall's Algorithm for Optimum D-ary Block Encoding with Block-Length N of a Proper Message Set for a K-ary DMS with Output Variable U .
  + **Step0**: Check to see that DN K. If not, abort because no such coding is possible. Otherwise, calculate the quotient q when DN − K is divided by K − 1.
  + **Step1**: Construct the Tunstall message set of size M = K + q × (K −1) for the DMS by beginning from the extended root and making q extensions of the most likely leaf at each step.
  + **Step2**: Assign a distinct D-ary codeword of length N to each message in the Tunstall message set.
* **仅考虑K=2, D=2 的情况，**N，p(P(1)) 可以任取.

### 源代码

源代码(使用Matlab)及详细注释参见附件。

* buildTunstallTree.m 该函数用来创建Tunstall树，要求***输入***根节点root，P(1)的概率p, 建树的循环步数q, ***输出***建好的树tree 以及平均编码长度averageMLength
* TunstallCoding.m 该脚本是可执行的脚本。主要包含以下几部分
  1. 生成随机01序列：seqLength=500，可修改
  2. 创建Tunstall树: 定长编码长度N 可以修改
  3. 创建词典Lexicon & 编码
  4. 解码
  5. 输出结果: 包括源序列，编码后的序列，字典，解码后的序列，平均编码长，E(L)/N

### 测试

1. P与平均编码长度的关系，**N=3**,D=2,K=2



1. P与平均编码长度的关系，**N=4**,D=2,K=2



可以看到，P离0.5越近，E(L)N → 1.E(L)关于p=0.5对称.

1. 一个**N=3,p=0.6**,K=2,D=2时的例子

Table 1 Lexicon(N=3,p=0.6)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Message | 00 | 110 | 101 | 100 | 011 | 010 | 1111 | 1110 |
| Codeword(BIN) | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| Codeword(DEC) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

**命令行输出：（seqLength=50）**

*源序列*

*oriStr =*

*10111101001111110101010100001100101110101011111110*

*编码后*

*codedBinary =*

*010111011110001010101011000001101111010100110*

*解码后*

*deSource =*

*101111010011111101010101000011001011101010111111*

*字典*

*lexicon =*

*'00' '110' '101' '100' '011' '010' '1111' '1110'*

1. 一个**N=4,p=0.6**,K=2,D=2时的例子

Table 2 Lexicon N=4,p=0.6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Message* | 1110 | 000 | 1101 | 1100 | 1011 | 1010 | 0111 | 0110 | 11111 | 11110 | 0011 | 0010 | 1001 | 1000 | 0101 | 0100 |
| *Codeword（DEC）* | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | *14* | *15* |
| *Codeword(BIN)* | *0000* | *0001* | *0010* | *0011* | *0100* | *0101* | *0110* | *0111* | *1000* | *1001* | *1010* | *1011* | *1100* | *1101* | *1110* | *1111* |

**命令行输出：（seqLength=1000）**

*源序列*

*oriStr =*

*0111010001101110101101110101111000111010111011011111100111110111001110001100101111000111001010001110111111101011010011011111111111101111010110001101110011001100111101101101111110111110110000101101101001101101110111001001000111011001101001101000010100110111110101001101111001001011110001101011110101110111110111101011001110011111100111101100011110111101111010101110111010101011101011100000111000110100110111111001111010100000110001111100111011101110100111010010101101110000101111000100010111111111011011110111111011101010001110001101111010000011010010101110011110101111011011011011101100110111100001110011001001111011100011100001100110001000110001001001110100100100000000011011100110110011011011011101101001010101111110001011101100001001111111110011100100000110101011111100111101110100100011001010111001000100010110111101011100011011110110101010001011010111000000011101100111011101110110010111101111000111001011110101110111000011001100011111110001001101110010111110101110011100000101010100011101101001*

*编码后*

*codedBinary =*

*01101111011100000100011011100000101001010000001010001010000000000110000100110100001101101011110100001000001001111100010010001000011001010011011100000111011101100100011110000100000000111011001001010111001000100011110000010000001100101010111110111100010000000101011110011111010000110111010000100110011000100000010010101100100010100010110110011001100101010000000001010100010100000001011000010010101001100000011001011101101000011000101001000100010101101111010100100011101110011011101110001001001000001000010001011101000010100110010100010111110011100011100101000010010001110000001100100000000100000111111110010000101011010111011110111010000111001010010111111101000100010010001100101100010001110000001010110101100011010100010000011111100010010110101100010111010110001100000000001100000100110101000011111111111001000010011000010010000000101110000101001110001100011010010010100100010001001011100110011010110001100101000000000001001100110110100110110111000011101001010011000011000101010101101001001111*

*解码后*

*deSource =*

*011101000110111010110111010111100011101011101101111110011111011100111000110010111100011100101000111011111110101101001101111111111110111101011000110111001100110011110110110111111011111011000010110110100110110111011100100100011101100110100110100001010011011111010100110111100100101111000110101111010111011111011110101100111001111110011110110001111011110111101010111011101010101110101110000011100011010011011111100111101010000011000111110011101110111010011101001010110111000010111100010001011111111101101111011111101110101000111000110111101000001101001010111001111010111101101101101110110011011110000111001100100111101110001110000110011000100011000100100111010010010000000001101110011011001101101101110110100101010111111000101110110000100111111111001110010000011010101111110011110111010010001100101011100100010001011011110101110001101111011010101000101101011100000001110110011101110111011001011110111100011100101111010111011100001100110001111111000100110111001011111010111001110000010101010001110110100*

*字典*

*lexicon =*

*Columns 1 through 7*

*'1110' '000' '1101' '1100' '1011' '1010' '0111'*

*Columns 8 through 14*

*'0110' '11111' '11110' '0011' '0010' '1001' '1000'*

*Columns 15 through 16*

*'0101' '0100'*