

Homework7

学生: 华园 (202000120027)

时间: 2022.3.23

Problem 1.

Solution. 首先介绍我的编程思路之后再出示代码实现 (内含注释):

(1) *parity-check*矩阵的生成, 我利用Python的一个包pyldpc中的生成H的方法, 进行了直接的生成。

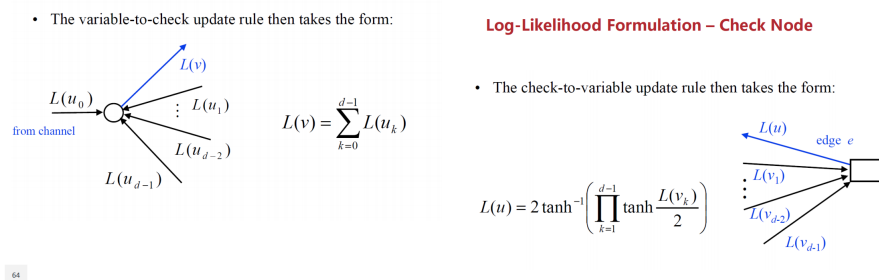
(2) 00.....000000BSC信道之后接收到的信息我们用一个数组R表示, 如何生成R, 我写了一个按照概率生成随机数的函数进行实现, 从而我们获得了接受到的信息, 每次接收到的具有随机性。

(3) 对于先验概率 $L(V)$, 我通过遍历R中元素, 根据元素是1还是0进行判断, 并修改其数值为 $\log(\frac{1-p}{p})$ 和 $\log(\frac{p}{1-p})$, 从而获得了先验概率的一维数组。

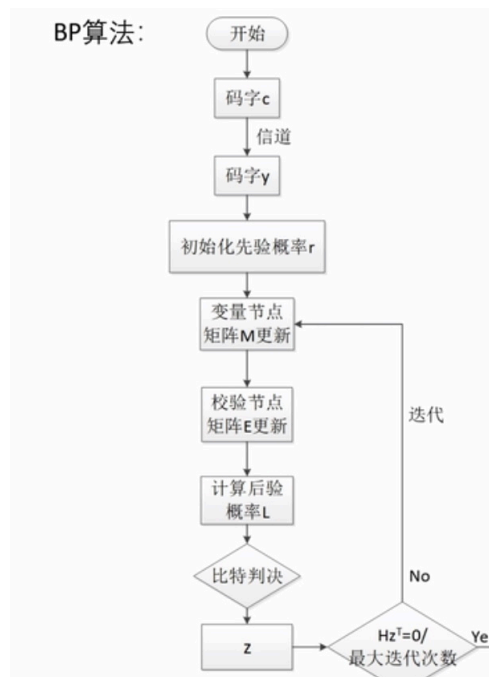
(4) 解码部分可分解为两部分: *variable-nodes*向*check-nodes*传递信息和*check-nodes*向*variable-nodes*传递信息。针对这部分内容, 我构造了M矩阵和E矩阵, 其中M矩阵代表*variable-nodes*传递的信息, 而E矩阵代表*check-nodes*传递的信息, 这两个矩阵均与H矩阵同型, 迭代的过程就是对这两个矩阵进行更新的过程。对M进行更新的过程, 要对E矩阵进行列求和, 并且减去对应位置的数值。而对E矩阵更新的过程就是对M矩阵进行一定的行操作。

(5) 每进行一次迭代, 根据后验概率进行判断, 获得解码的码字, 如果不满足 $H \cdot C^T = 0$, 则进行迭代, 如果满足, 直接跳出, 若在规定的最大迭代次数内没有正确解码, 则视为无法解码。

(6) 输入不同的 p 值进行测试, 绘图, 基本思路可以由下图表示:



整体编程思路可用下图总结: (摘自自B站网课)



下面给出代码：（1）生成随机数方法以及整体代码：（目录中附了代码，学长学姐看不清可以直接看代码）

```

import numpy as np
from pyldpc import make_ldpc, encode, decode, get_message
import math
import random as random

def p_random(arr1, arr2):
    assert len(arr1) == len(arr2), "Length does not match."
    assert sum(arr2) == 1., "Total rate is not 1."

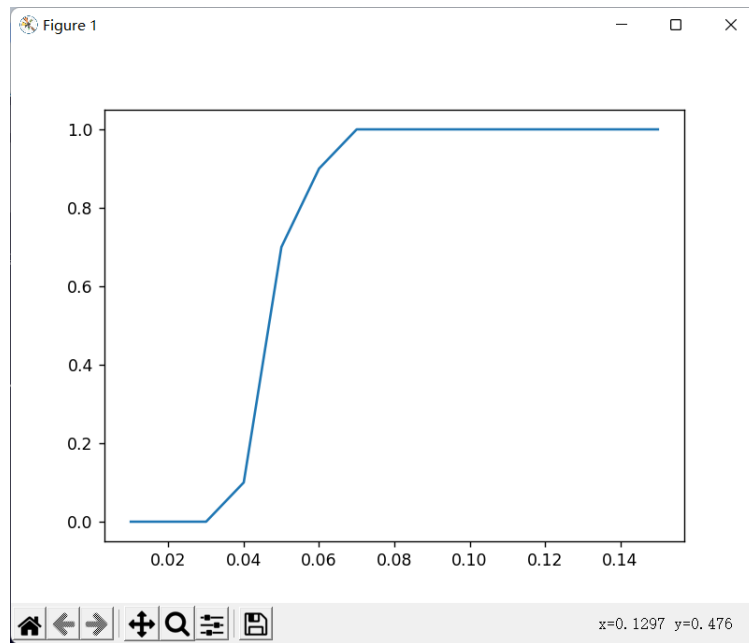
    sup_list = [len(str(i).split("."))[-1]] for i in arr2
    top = 10 ** max(sup_list)
    new_rate = [int(i*top) for i in arr2]
    rate_arr = []
    for i in range(1, len(new_rate)+1):
        rate_arr.append(sum(new_rate[:i]))
    rand = random.randint(1, top)
    data = None
    for i in range(len(rate_arr)):
        if rand <= rate_arr[i]:
            data = arr1[i]
            break
    return data
  
```

```
python学习 - 算术运算符.py
python学习 - unit3 - 算术运算符.py

项目
├── python学习
│   ├── oop
│   ├── practise
│   ├── unit1
│   ├── unit2
│   └── unit3
│       ├── data1.csv
│       ├── input函数.py
│       ├── 位运算符.py
│       ├── 布尔运算符.py
│       ├── 比较运算符.py
│       ├── 算术运算符.py
│       └── 赋值运算符.py
├── unit4
├── unit5
│   ├── break语句-for循环.py
│   ├── break语句-while循环.py
│   ├── continue语句.py
│   ├── else语句-for.py
│   ├── else语句-while.py
│   ├── for-in循环.py
│   ├── for循环练习题.py
│   ├── range函数的使用.py
│   ├── while循环.py
│   └── 二重循环中的continue与break.py
├── unit6
├── unit7
├── unit8
├── unit9
├── unit12
├── venv
├── 作业
│   ├── gift.py
│   ├── 华园.py
│   ├── 开方操作.py
│   ├── 第二次作业-华园-20200120027.py
│   ├── 素数判断.py
│   ├── 计算器.py
│   ├── 阿松大.py
├── 数学建模
│   └── 建模规划篇
│       └── 线性规划.py
└── 外部库
    └── 临时文件和控制台

24 for ab in range(A):
25     n = 1024
26     d_v = 3
27     d_c = 8
28     p = 0.006
29     H, G = make_ldpc(n, d_v, d_c, systematic=True, sparse=True) #生成H和G
30     k = H.shape[0]
31     r = np.zeros([2], dtype=float)
32     P = np.zeros([2], dtype=float)
33     R = np.zeros([n], dtype=float)
34     r[1] = 1
35     P[0] = 1 - p
36     P[1] = p
37     for i in range(n):
38         R[i] = p_random(r, P)
39     for i in range(n):
40         if R[i] == 0:
41             R[i] = math.log((1 - p) / (p), 2) #求取信道传输的r
42         else:
43             R[i] = math.log(p / (1 - p), 2)
44
45     """-----解码过程-----"""
46     M = np.zeros([k, n], dtype=float)
47     E = np.zeros([k, n], dtype=float)
48     """-----迭代过程-----"""
49     for i in range(50): #最大迭代次数50次
50         for j in range(n):
51             for a in range(k): #循环经过每个元素
52                 if H[a, j] != 0: #如果遇到非0元素
53                     X = E.sum(axis=0) #先对E矩阵按列求和，获得一个一维矩阵
54                     M[a, j] = X[j] - E[a, j] + R[j] #对元素进行更新，H=E对应列求和-E+信道输入
55
56         for a in range(k):
57             Y = 1
58             for b in range(n):
59                 if M[a, b] != 0:
60                     Y = Y * math.tanh((M[a, b]) / 2) #对M矩阵进行处理
61             for j in range(n): #循环经过每个元素
62                 if M[a, j] != 0:
63                     E[a, j] = 2 * math.atanh(Y / math.tanh(M[a, j] / 2)) #更新结果
64
65             Y = E.sum(axis=0) #将M矩阵按列求和并进行译码
66
67         for b in range(n): #如果大于0，则译为0，小于0，则译为1
68             Y[b] = Y[b] + R[b]
69             if Y[b] > 0:
70                 Y[b] = 0
71             else:
72                 Y[b] = 1
73         N = 0
74
75         for b in range(n): #对比原始码字和译码码字，找出不同的个数
76             if Y[b] != 0:
77                 N = N + 1
78         if N == 0:
79             print("迭代次数:", i)
80             break
81
82     print(N)
83     if N > 0:
84         F = F + 1
85     F = float(F)
86     print(F/A)
87
88 for ab in range(A) : if N>0
```

(2) 图像:



(3)最大迭代数设置为: 20; 生成校验矩阵和生成矩阵的时间约为0.5s,针对每个 p 进行1000次解码, 并且最大迭代数字为20, 由于python解1000个码字还是比较吃力地, 因此相应耗时比较长, cpu时间大约分为两类, (1) 当可解码的时候 ($p \leq 0.04$), 每个 p 时长大约为38分钟.(2)完全不可解码后($p \geq 0.05$),每个 p 时长平均4.16个小时左右