****



**本 科 生 课 程 报 告**

**课程名称:信息论导论**

**学 院：生命**

**专 业：生物医学工程**

**姓 名：**

**学 号：**

**教 师:任鹏**

实验一 信息测度

一、实验背景

本实验通过计算给定的信源的熵，加深对信源及其扩展信源的熵的概念的理解。

二、实验环境

　　windows,MATLAB

三、实验原理

信源输出的各消息的自信息量的数学期望为信源的信息熵，表达式如下



信源熵是信源的统计平均不确定性的描述，是概率函数的函数。



四、实验内容

有两个二元随机变量X和Y，它们的联合概率分布函数如下表：

|  |  |  |
| --- | --- | --- |
| ***Y X*** | 0 | 1 |
| 0 | 1/8 | 3/8 |
| 1 | 3/8 | 1/8 |

同时定义另一随机变量Z=X\*Y，试求：

a、熵H(X),H(Z),H(X,Z)和H(X,Y,Z);

b、条件熵H(X|Y),H(X|Z),H(Y|X,Z);

c、互信息I(X;Y),I(X;Z),I(X;Y|Z);

五、实验结果

h = 1

hy = 1

hxz = 1.4056

hxyz = 1.8113

hz = 0.5436

h1 = 0.8113

h2 = 0.8621

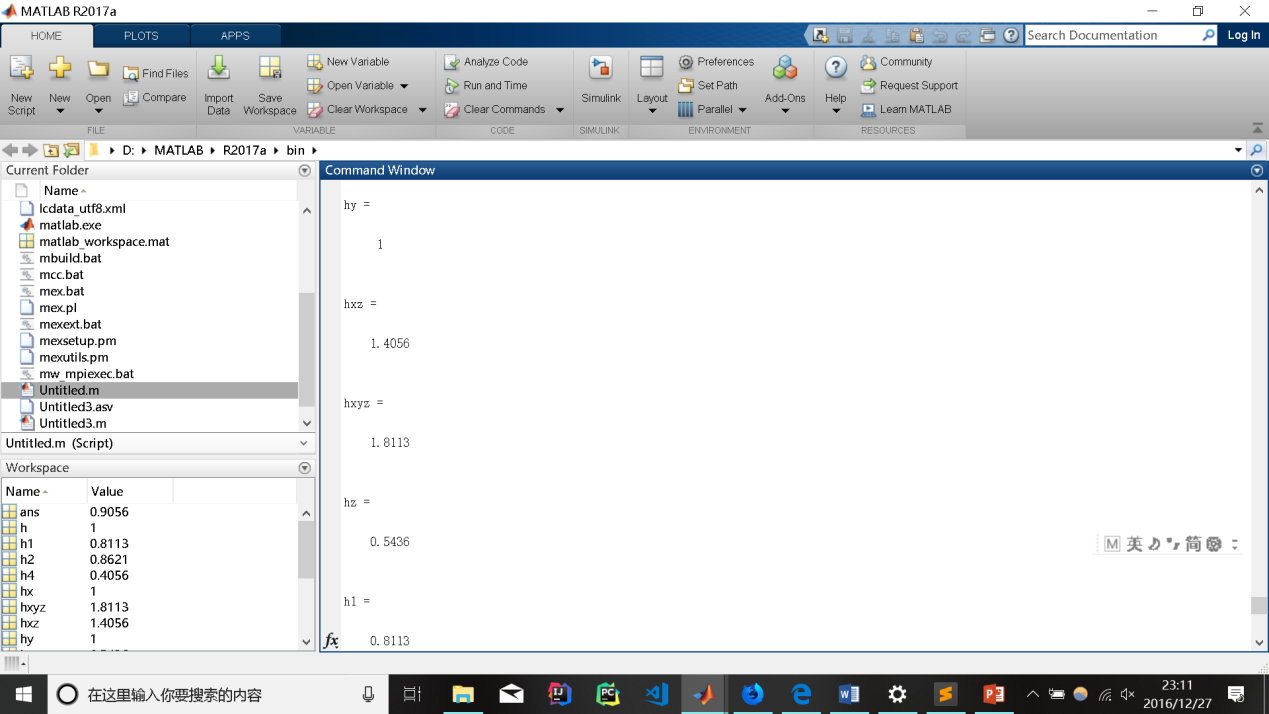
h3 = 1.3113

h4 = 0.4056

ix = 0.1887

ixz = 0.1379

i1 = 0.4564



六、实验结果分析

所得结果与手算结果相同。

Matlab代码：

p=[1/8 3/8,3/8,1/8]

x=[0.5 0.5]

y=[0.5 0.5]

z=[1/8 7/8]

xz=[1/2 3/8 1/8]

h = -sum(x.\*log2(x))%H(x)

hy=-sum(y.\*log2(y)) %H(Y)

hxz=-sum(xz.\*log2(xz)) %H(XZ)

hxyz= -sum(p.\*log2(p)) %H(XYZ)

hz= -sum(z.\*log2(z))

h1=hxyz-hy %H(X|Y)

h2=hxz-hz %H(X|Z)

h3=(1/8)\*log2(1/4)+3/8\*log2(3/4)+(3/8)\*log2(3/8)+(1/8)\*log2(1/8) %H(Y|X,Z)

h4=hxyz-hxz

ix=h-h1 %I(X;Y)

ixz=h-h2 %I(X;Z)

i1=h2-h4 %I(X;Y|Z)

实验二 无失真信源编码

一、实验背景介绍

　　香农第一定理：

设离散无记忆信源为



熵为H(S)，其N次扩展信源为



熵为H(SN)。码符号集X=（x1,x2,…,xr）。先对信源进行编码，总可以找到一种编码方法，构成惟一可以码，使S中每个信源符号所需的平均码长满足：



当N时 

是平均码长  是对应的码字长度

二、实验内容

在给定离散无记忆信源

=

**S**

**P**

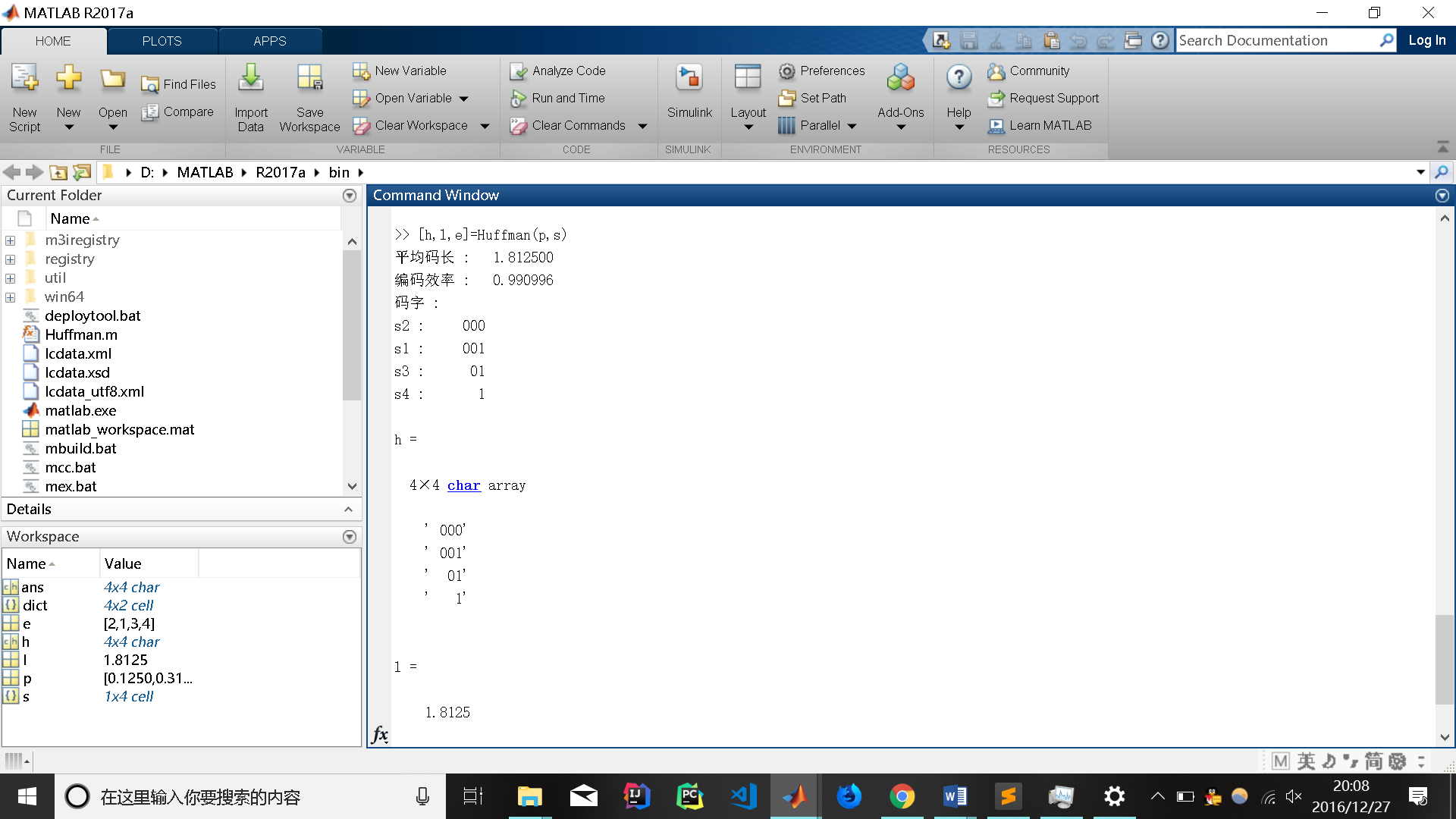
s1 s2 s3 s4

1/8 5/16 7/16 1/8

条件下，实现二进制霍夫曼编码，求最后得到的码字并算出编码效率。

三、实验结果

编写函数计算码字、码长和效率并保存为Huffman.m

在命令行中输入p并调用Huffman函数，结果如下图

输出结果：

**>> [h,l,e]=Huffman(p,s)**

**平均码长 : 1.812500**

**编码效率 : 0.990996**

**码字 :**

**s1 : 000**

**s2 : 001**

**s3 : 01**

**s4 : 1**

**h =**

**4×4 char array**

**' 000'**

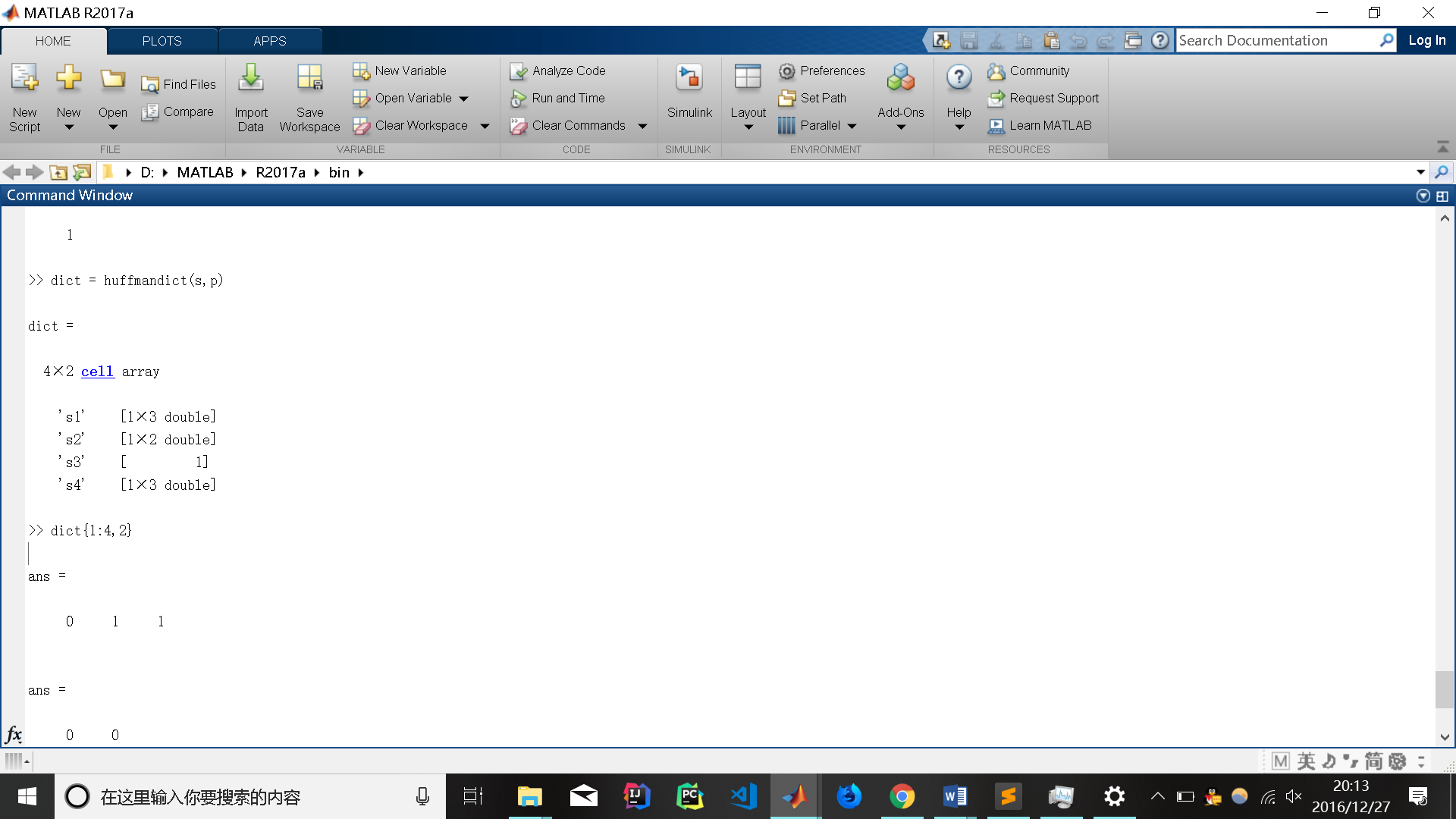
**' 001'**

**' 01'**

**' 1'**

四、实验结果分析

使用matlab自带的huffmandict，huffmanenco，huffmandeco函数对结果进行验证：如下图所示：



>> dict = huffmandict(s,p)

dict =

4×2 cell array

's1' [1×3 double]

's2' [1×2 double]

's3' [ 1]

's4' [1×3 double]

>> dict{1:4,2}

ans =

0 1 1

ans =

0 0

ans =

1

ans =

0 1 0

得到的结果和自定义函数结果相同，平均码长 : 1.812500

编码效率 : 99.10%

五、Matlab代码

function [ h,l,e ] = Huffman( p ,s )

%p为离散无记忆信源的概率分布

% h为各个元素的码字

% e为输出的平均码长

if ~isempty(find(p<0, 1))

error('概率之和小于0，请输入正确数值')

end

if abs(sum(p)-1)>10e-10

error('概率之和大于1，请输入正确数值')

end

n=length(p);

p=sort(p);

q=p;

str=s;

m=zeros(n-1,n);

for i=1:n-1

[q,e]=sort(q);

m(i,:)=[e(1:n-i+1),zeros(1,i-1)]; %由数组l 构建一个矩阵，该矩阵表明概率合并时的顺序，用于后面的编码

q=[q(1)+q(2),q(3:n),1];

end

for i=1:n-1

c(i,1:n\*n)=blanks(n\*n); %c 矩阵用于进行huffman 编码

end

c(n-1,n)='1'; %由于a 矩阵的第n-1 行的前两个元素为进行huffman 编码加和运算时所得的最后两个概率

c(n-1,2\*n)='0';

for i=2:n-1

c(n-i,1:n-1) = c(n-i+1,n\*(find(m(n-i+1,:)==1))-(n-2):n\*(find(m(n-i+1,:)==1))); %矩阵c 的第n-i 的第一个元素的n-1 的字符赋值为对应于a 矩阵中第n-i+1 行中值为1 的位置在c 矩阵中的编码值

c(n-i,n)='0';

c(n-i,n+1:2\*n-1) = c(n-i,1:n-1); %矩阵c 的第n-i 的第二个元素的n-1 的字符与第n-i 行的第一个元素的前n-1 个符号相同，因为其根节点相同

c(n-i,2\*n)='1';

for j=1:i-1

c(n-i,(j+1)\*n+1:(j+2)\*n) = c(n-i+1,n\*(find(m(n-i+1,:)==j+1)-1)+1:n\*find(m(n-i+1,:)==j+1));

%矩阵c 中第n-i 行第j+1 列的值等于对应于a 矩阵中第n-i+1 行中值为j+1 的前面一个元素的位置在c 矩阵中的编码值

end

end

for i=1:n

h(i,1:n)=c(1,n\*(find(m(1,:)==i)-1)+1:find(m(1,:)==i)\*n); %用h表示最后的huffman 编码

len(i)=length(find(abs(h(i,:))~=32)); %计算每一个编码的长度

end

l=sum(p.\*len); %计算平均码长e

hx=sum(-p.\*log2(p));%信源熵

e=hx/l;%计算编码效率

fprintf('平均码长 : \t %f',l);

fprintf('\n');

fprintf('编码效率 : \t %f\n',e);

fprintf('码字 : \n');

for i = 1:n

fprintf('%s : ', str{1,e(i)});

fprintf('\t');

for j=1:length(p)

fprintf('%s',h(i,j));

end

fprintf('\n');

end