

## 拓展资源 7.2 实验指导



### 7.2.1 哈夫曼编码

#### 1. 实验内容

已知某符号集中各符号出现的概率，对符号进行哈夫曼编码。

#### 2. 实验原理

(1) 首先将信源符号按出现的概率大小顺序排列，并将两处概率最小的进行相加，形成一组新的概率，其次再将这组新的概率按大小顺序排列，如此重复直到只有两个概率为止。

(2) 分配码字，码字分配从最后一步开始反向进行，对最后两个概率一个赋予“0”码字，另一个赋予“1”码字。

#### 3. 实验方法及程序

设输入图像的灰度级 $\{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8\}$ 出现的概率分别为 0.40, 0.18, 0.10, 0.10, 0.07, 0.06, 0.05, 0.04，进行哈夫曼编码。

```
function [h,l]=huffman(p)
if (length(find(p<0))~=0)
    error('Not a prob,negative component');
end
if (abs(sum(p)-1)>10e-10)
    error('Not a prob.vector,component do not add to 1')
end
n=length(p);
q=p;
m=zeros(n-1,n);
for i=1:n-1
    [q,l]=sort(q);
    m(i,:)=[l(1:n-i+1),zeros(1,i-1)];
    q=[q(1)+q(2),q(3:n),1];
end
for i=1:n-1
    c(i,:)=blanks(n*n);
end
c(n-1,n)='0';
c(n-1,2*n)='1';
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)));
c(n-i,n)='0';
c(n-i,n+1:2*n-1)=c(n-i,1:n-1);
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));
end
end

for i=1:n
    h(i,1:n)=c(1,n*(find(m(1,:)==i)-1)+1:find(m(1,:)==i)*n);
    ll(i)=length(find(abs(h(i,:))~=32));
end
l=sum(p.*ll);
实验结果:
p=[0.40 0.18 0.10 0.10 0.07 0.06 0.05 0.04]
p =
    0.4000    0.1800    0.1000    0.1000    0.0700    0.0600
0.0500    0.0400
>> [h,l]=huffman(p)
h =
    0
   110
  1111
   100
  1011
  1010
  11101
  11100
l =2.6100
    
```

#### 4. 实验分析

计算平均码字长度为

$$L = \sum_{K=1}^M l_K P_K = 0.40 \times 1 + 0.18 \times 3 + 0.10 \times 3 + 0.10 \times 4 + 0.07 \times 4 + 0.06 \times 4 + 0.05 \times 5 + 0.04 \times 5 = 2.61$$

实验结果的平均码字长度与计算结果一致。

#### 5. 思考题

- (1) 对参考程序给出功能注释。
- (2) 分析编码结果和书中例题 7.3 不一致的原因。



## 7.2.2 算术编码

### 1. 实验内容

选一组信源符号，已知各符号的概率，对序列进行算术编码，然后再解码。

### 2. 实验原理

算术编码不是将单个信源符号映射成一个码字，而是把整个信源表示为  $0\sim 1$  的一个区间，其长度等于该序列的概率，再在该区间内选择一个代表性的小数，转化为二进制作为实际的编码输出。解码是编码的逆过程。

### 3. 实验方法及程序

以  $a, b, c, d$  分别代表 00, 01, 10, 11，假设信源符号为  $X=\{a, b, c, d\}$ ，其中各符号的概率为  $P(X)=\{0.1, 0.4, 0.2, 0.3\}$ 。对消息序列为 *cadacdb*，进行算术编码。参考程序如下。

编码子程序：

```
function arcode=arenc(symbol,pr,seqin)
%算术编码
%输出：码串
%输入：symbol：字符行向量
%pr：字符出现的概率
%seqin：待编码字符串
high_range=[]
for k=1:length(pr)
    high_range=[high_range sum(pr(1:k))]
low_range=[0 high_range(1:length(pr)-1)]
sidx=zeros(size(seqin))
for i=1:length(seqin)
    sidx(i)=find(symbol==seqin(i))
end

low=0
high=1
for i=1:length(seqin),
    range=high-low
    high=low+range*high_range(sidx(i))
    low=low+range*low_range(sidx(i))
end
arcode=low
```

解码子程序:

```
function symseq=ardec(symbol,pr,codeword,symlen)
%给定字符概率的算术编码
%输出: symse: 字符串
%输出: symbol: 由字符组成的行向量
%pr: 字符出现概率
%codeword: 码字
%symlen: 待解码字符串长度
format long
high_range=[]
for k=1:length(pr)
    high_range=[high_range sum(pr(1:k)) ]
end
low_range=[0 high_range(1:length(pr)-1)]
prmin=min(pr)
symseq=[]
for i=1:symlen,
    idx=max(find(low_range<=codeword));
    codeword=codeword-low_range(idx)
    if abs(codeword-pr(idx))<0.01*prmin
        idx=idx+1
        codeword=0
    end
    symseq=[symseq symbol(idx)]
    codeword=codeword/pr(idx)
    if abs(codeword)<0.01*prmin
        i=symlen+1;
    end
end
end
```

调用程序:

```
clear all
format long
symbol=['abcd']
pr=[0.1 0.4 0.2 0.3]
seqin=['cadacdb']
codeword=arenc(symbol,pr,seqin)
seqout=ardec(symbol,pr,codeword,7)
```

#### 4. 实验结果与分析

pr =

```
0.100000000000000000    0.400000000000000000    0.200000000000000000
0.300000000000000000
codeword =
0.5143876000000000
seqout =
cadacdb
```

编码的结果为 0.5143876000000000，解码的结果为 *cadacdb*。

## 5. 思考题

- (1) 注释参考程序的功能。
- (2) 该实验可以应用到哪些实际问题中？