INDIAN INSTITUTE OF INFORMATION TECHNOLOGY NAGPUR

Name: Uma

Roll Number: BT20ECE044

Semester: 6

Course: Coding Techniques

Lab Report

Shannon Fano coding

Theory:

Shannon Fano Algorithm is an entropy encoding technique for lossless data compression of multimedia. Named after Claude Shannon and Robert Fano, it assigns a code to each symbol based on their probabilities of occurrence. It is a variable-length encoding scheme, that is, the codes assigned to the symbols will be of varying lengths.

The steps of the algorithm are as follows:

1. Create a list of probabilities or frequency counts for the given set of symbols so that the relative frequency of occurrence of each symbol is known.
2. Sort the list of symbols in decreasing order of probability, the most probable ones to the left and the least probable ones to the right.
3. Split the list into two parts, with the total probability of both parts being as close to each other as possible.
4. Assign the value 0 to the left part and 1 to the right part.
5. Repeat steps 3 and 4 for each part until all the symbols are split into individual subgroups.

Code:

```
disp('Enter the probabilities:');
%ss=[0.25 0.125 0.5 0.125];
%ss=[0.25 0.125 0.0625 0.0625 0.0625 0.25 0.0625 0.125];
ss=[0.4 0.2 0.12 0.08 0.08 0.08 0.04];
%ss=[0.4 0.3 0.2 0.1]
%ss=[0.45 0.15 0.1 0.1 0.08 0.08 0.04]
%ss=[0.2 0.15 0.03 0.05 0.45 0.08 0.04]
%outputs = string of codewords,average codeword length
ss=ss./sum(ss); %if occurrences are inputted, probabilities are gained
ss=sort(ss,'descend');  %the probabilities are sorted in descending order
%siling=ceil(log2(1/ss(1))); %initial length is computed
siling=log2(1/ss(1)); %initial length is computed
siling=round(siling,1,'significant');
sf=0;
fano=0;
%initializations for Pk
n=1;Hx=0; %initializations for entropy H(X)
for i=1:length(ss)
  Hx=Hx+ ss(i)*log2(1/ss(i)); %solving for entropy
end
for k=1:length(ss)
info(k)=-(log2(ss(k))); %Information
end
for j=1:length(ss)-1
  fano=fano+ss(j);
  sf=[sf 0]+[zeros(1,j) fano]; %solving for Information for every codeword
  siling=[siling 0]+[zeros(1,j) ceil(log2(1/ss(j+1)))]; %solving for length
every codeword
end
for r=1:length(sf)
   esf=sf(r);
   for p=1:siling(r)
       esf=mod(esf,1)*2;
       h(p)=esf-mod(esf,1); %converting Pk into a binary number
   end
   hh(r)=h(1)*10^(siling(r)-1); %initializtion for making the binary a whole
number
   for t=2:siling(r)
       hh(r)=hh(r)+h(t)*10^(siling(r)-t);     %making the binary a whole
number
   end                                        %e.g. 0.1101 ==> 1101
end
c={'0','1'};
disp('Codeword');
for i=1:length(hh)
   u=1;                                       %converting the codes into a
string
  for t=siling(i):-1:1
```

```
      f=floor(hh(i)/10^(t-1));              %1001 ==>1 (getting the first
highest unit of a number)
      hh(i)=mod(hh(i),10^(t-1));            %1001 ==>001(eliminating the
first highest unit of a number)
      if f==1
          if u==1
               d=c{2};                      %conversion part (num(1001) to
str(1001))
          else
               d=[d,c{2}];
          end
      else
          if u==1
               d=c{1};
          else
               d=[d,c{1}];
          end
      end
      codex{i,:}={d};
      u=u+1;
  end
  disp([d]);
end
tao=siling(1)*ss(1); %initialization for codeword length
for u=1:length(ss)-1 %computing for codeword length
  tao=tao+siling(u+1)*ss(u+1);
end
T=tao/n; %computing for average codeword length
B=[flipud(rot90(ss)),flipud(rot90(siling)),flipud(rot90(info))];
disp(['Probability','   Length','   Information'])
disp(B)
disp(['Entropy H(X) = ',num2str(Hx),'bits/symbol'])
disp(['Average length,L = ',num2str(T),'bits/symbol'])
eff=((Hx/T)*100); %Coding efficiency
disp(['Efficiency=',num2str(eff),'%'])
redu=100-eff;   %Redundancy
disp(['Redundancy=',num2str(redu),'%'])
```

Output:

```
Codeword
0
011
1001
1011
1100
1110
11110
Probability   Length   Information
    0.4000    1.0000    1.3219
    0.2000    3.0000    2.3219
    0.1200    4.0000    3.0589
    0.0800    4.0000    3.6439
    0.0800    4.0000    3.6439
    0.0800    4.0000    3.6439
    0.0400    5.0000    4.6439

Entropy H(X) = 2.4205bits/symbol
Average length,L = 2.64bits/symbol
Efficiency=91.6858%
Redundancy=8.3142%
```