

huffman编码与算数编码 matlab编程实现及比较实验

原创

[iamsongyu](#) 于 2018-09-11 15:42:42 发布 5312 收藏 30

分类专栏: [编程](#) 文章标签: [matlab](#) [无损编码](#) [实验报告](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/iamsongyu/article/details/82626957>

版权



[编程](#) 专栏收录该内容

13 篇文章 0 订阅

订阅专栏

前一段时间有一个课程, 需要自己实现两种无算编码方式, 并比较其效率, 这确实是难倒我了, 在网上找了几个但是实在看不懂别人的程序, 只能硬着头皮自己写, 大概的效果还好吧, 需要的人可以借鉴一下

完整的代码和文件, 还有实验报告我提供在百度云上, 请到最后的位置下载, 如果可以的话大家多多支持啊~

第一个是huffman编码 (matlab代码)

```
function [code,efficent,codetable,time] = Huffmanencode(text,type)
% HUFFMANENCODE 此处显示有关此函数的摘要
% 此处显示详细说明
% text为输入文本 type文本类型 按照实验要求1标识英文 0标识汉语
% code编码符号 efficent压缩率 codetable码表 time耗时
% Huffman编码
t1=clock;

Hztar=[' '];
Hznum=[0];
%读取文本
target=text;
%strrep(english,' ','x') ;
for i=1:length(target);
    %target(i)
    if target(i)==' '
        Hznum(1)=Hznum(1)+1;
    elseif strfind(Hztar,target(i))>0
        position=strfind(Hztar,target(i));
        Hznum(position)=Hznum(position)+1;
    else
        Hztar=[Hztar,target(i)];
        Hznum=[Hznum;1];
    end
end

%符号数
numstar=length(Hznum);
%编码表
codetable=cell(2,numstar);
%保存好对应的符号与个数
oritable=cell(2,numstar);
for i=1:numstar
    codetable{1,i}=Hztar(i);
```

```

codetable{2,1}='';
oritable{1,1}=Hztar(i);
oritable{2,1}=Hznum(i);
end
%根据是否只剩下最后两个判断是否完成编码
while (length(find(Hznum==100000))<=numstar-2)
    %编码还有符号 两个
    select=[0,0];
    num=[0,0]; %频数
    posi=-1; %记录位置
    temp=find(Hznum==min(Hznum)); %找第一个最小
    select(1)=temp(1);
    numz(1)=Hznum(temp(1));
    Hznum(temp(1))=100000; %大幅度提升频数 不让参与下次分配
    temp=find(Hznum==min(Hznum)); %找第二个
    select(2)=temp(1);
    numz(2)=Hznum(temp(1));
    Hznum(temp(1))=100000; %大幅度提升频数 不让参与下次分配
    Hznum(select(1))=numz(2)+numz(1); %将两个合成一个节点 保存在左侧子树
    %组合成为一个二叉树
    %进行倒序编码 第一个编码0 第二个为1
    for p=1:2
        str=oritable{1,select(p)};
        for j=1:length(str);
            %找出每个位置 增加前面的树节点编码
            for k=1:numstar
                if (codetable{1,k}==str(j));
                    posi=k;
                    break;
                end
            end
            %位置遍历完毕
            codetable{2,posi}=[numzstr(p-1),codetable{2,posi}]; %扩充编码
        end
    end
    %组合新的节点符号与频数
    oritable{1,select(1)}=[oritable{1,select(1)},oritable{1,select(2)}];
    oritable{2,select(1)}=Hznum(select(1));
end

%编码原文
code='';
a=[];
posi=0;
for h=1:length(target)
    %a=[a,find(Hztar==english(h))];
    posi=find(Hztar==target(h));
    code=[code,codetable{2,posi}];
    l1=length(codetable(1,:));
end
t4=clock;
%计算效率 编码后长度算上码表的
nowlength=length(code); %编码长度
if type==1
    origainlength=7*length(target);
    % nowlength=nowlength+l1*7+sqrt(length(target))*length(codetable(1,:));%算上面表里的字符和编码
else
    origainlength=16*length(target);
    % nowlength=nowlength+l1*16+sqrt(length(target))*length(codetable(1,:));
end
efficent=nowlength/origainlength

```

```
time=etime(t4,t1);
end
```

```
function [decode,time] = Huffmandecode(codetable,code)
%HUFFMANDECODE 此处显示有关此函数的摘要
% 此处显示详细说明
% codetable编码表 code编码信息序列
%译码
t1=clock;
decode='';
unfinish=1;
while(unfinish)
    flag=0;
    %循环取更长的码 直到可以翻译
    for o=1:length(code)
        str=code(1:o);
        %寻找是否存在译码
        for p=1:length(codetable)
            %找到则译码
            if strcmp(codetable{2,p},str)
                decode=[decode,codetable{1,p}];
                flag=1;
                break;
            end
        end
        end
        if flag==1; %这次的找到了 跳出for进行下一个
            if length(code)-o==0 %判断是否完全以译码
                unfinish=0;
            else %截断后继续 %第一次截错了 code=code(o+1:length(code)-o); 这是长度 不是下标
                code=code(o+1:length(code));
            end
            break;
        end
        end %找到了一个 继续下一个
    end
t2=clock;
time=etime(t2,t1);
end
```

第二个选择的是算术编码

```
function [code,numcode,efficent,codetable,time] = Arithmeticncode(text,type)
%ARITHMETICNCODE 算数编码函数
% text为输入文本 type文本类型 按照实验要求1标识英文 0标识汉语
% code编码符号 numcode每个编码符号标识的符号个数 efficent压缩率 codetable码表 time耗时
t1=clock;
Hztar=[' '];
Hznum=[0];
%读取文本
target=text;
%strrep(english,' ','>') ;
for i=1:length(target);
    %target(i)
    if target(i)==' '
        Hznum(1)=Hznum(1)+1;
    elseif strfind(Hztar,target(i))>0
```

```

        position=strfind(Hztar,target(i));
        Hznum(position)=Hznum(position)+1;
    else
        Hztar=[Hztar,target(i)];
        Hznum=[Hznum,1];
    end
end
%符号数
numstar=length(Hznum);
%保存好对应的符号与个数
all=sum(Hznum);
orbitable=cell(3,numstar+1);
for i=1:numstar %计算每个符号的起始左端点概率 和间隔长度
    orbitable{1,i}=Hztar(i);
    if i==1
        orbitable{2,i}=0;
    else
        orbitable{2,i}=sum(Hznum(1:i-1))/all;
    end
    orbitable{3,i}=Hznum(i)/all;
end
orbitable{1,numstar+1}='&'; %扩充最后一个用于解码
orbitable{2,numstar+1}=10;
orbitable{3,numstar+1}=1;

code=[];
numcode=[];
unfinish=1;
pos=1; %记录编码位置
l=length(target); %记录总长度
while(unfinish)
    num=0; %记录该数字编码的符号数目
    a=[0,1];
    sub=a(2)-a(1);
    for p=pos:l %从上一个编码结束的位置继续编码
        findstr=find(Hztar==target(p)); %找字符位置
        Pra=orbitable{2,findstr}; %获得字符左端点概率
        Len=orbitable{3,findstr}; %获取间隔长度
        a(1)=sub*Pra+a(1);
        a(2)=sub*Len+a(1);
        sub=a(2)-a(1);
        num=num+1;
        if(p+1>l) %如果所有符号编码完成就结束 保留最后一轮记录
            unfinish=0;
            sub=0.000000000001; %强制保存最后一个编码
        end
        if(sub<=0.000000000001) %如果i编码到一定界限就完成这轮
            log=sum(a)/2;
            code=[code,log];
            numcode=[numcode,num];
            pos=pos+1; %计算下一轮编码初始位置
            break;
        end
    end
end %一个编码循环
end
codetable=orbitable;
t2=clock;
time=etime(t2,t1);

```

```

codeLength=15*length(code); %编码长度
%方法同 Huffman 计算码表的长度
cl=length(orbitable(1,:));
if type==1 %计算压缩率 1英语
    oril=3*length(target);
    % codeLength=codeLength+cl*(5+3); % (每个编码符号的字符 概率)
else
    oril=5*length(target);
    % codeLength=codeLength+cl*(15+5);
end
efficent=codeLength/oril %压缩率
end

```

```

function [decode,time] = Arithmeticdecode(code,numcode,orbitable)
%ARITHMETICDECODE 算数解码
% code编码符号集合 numcode每个编码标识的符号数 orbitable码表
% decode解码数据 time解码时间
%解码循环
t1=clock;
decode='';
for ne=1:length(code)
    a=[0,1]; %每次解码重新置默认
    sub=a(2)-a(1);
    for i=1:numcode(ne) %需要解码的个数 根据编码给出的信息解码对应的个数
        for p=2:length(orbitable(1,:)) %因为找第一个比它大的 所以可以直接从第二个开始
            Pra=orbitable{2,p}; %获得字符左端点概率
            % Len=orbitable{3,p}; %获取间隔长度
            newpoint=sub*Pra+a(1);
            if code(ne)<newpoint %比较子区间所在位置 !!注意p-1才是正确的区间
                decode=[decode orbitable{1,p-1}]; %解码一个字符
                Pra=orbitable{2,p-1};
                Len=orbitable{3,p-1}; %获取间隔长度
                a(1)=sub*Pra+a(1); %计算新的区间
                a(2)=sub*Len+a(1);
                sub=a(2)-a(1);
                break;
            end
        end
    end
end
end
t2=clock;
time=etime(t2,t1);
end

```

对于两个最后的实验比较，代码如下

```

% clc
% clear
%-----设置英文汉语编码选项-----
type=0; %1英文 0汉语
if type==1
    load('english','english');
    MSG=english;
else
    load('chinese','chinese');
    MSG=chinese;
end
a=zeros(4,101);
% for ci=1:100
% Huffman 编码解码
[code,efficenth,codetable,timeen]=Huffmanencode(MSG,type); %编码
[DMSG,timed]=Huffmandecode(codetable,code); %解码
% a(1,ci+1)=timeen;
% a(2,ci+1)=timed;
%算数编码解码
[code,numcode,efficenta,codetable,timearen]=Arithmeticencode(MSG,type); %编码
[decode,timearde]=Arithmeticdecode(code,numcode,codetable);
% a(2,ci+1)=timearen;
% a(4,ci+1)=timearde;
% %性能计算函数
% end
finish=1; %是否解码正确
for k=1:length(decode)
    if DMSG(k)~=decode(k)
        finish=0;ss=k;
        break;
    end
end
if finish
%length(DMSG)==length(decode)&&length(decode)==length(MSG)
disp('-----编解码完全匹配-----');
disp(strcat('霍夫曼编解码时间为',num2str(timeen),'s_____',num2str(timed),'s    压缩率为',num2str(efficenth)));
disp(strcat('算数编解码时间为',num2str(timearen),'s_____',num2str(timearde),'s    压缩率为',num2str(efficenta)));
end
end

```

这个过程中，真的是考验我的逻辑思维啊，我想说太难了，最后的结果也还好，可以完美的运行。完整的代码和文件，还有实验报告我提供在百度云上

链接: <https://pan.baidu.com/s/13bYZobHWIW2UhZy5Xg-rOQ> 密码: huzf

test.m就是测试的文件，大家可以跑一下，我这水平还很差。。不要介意