**霍夫曼编码算法的研究**

张书琪

东南大学信息工程与科学学院，江苏省南京市，211100

**摘 要：**霍夫曼在1952年提出一种构造最优码的方法，该方法完全依据字符出现概率来构造编码长度，称之为霍夫曼编码（Huffman Code）。这是一种以霍夫曼树——即最优二叉树为核心的编码方式，也是可变长编码（VLC）的一种，多用在数据压缩，可以得到平均最短码长。在计算机信息处理中，霍夫曼编码是一种熵编码法。熵编码法是指使用一张特殊的编码表对各种信号进行编码。在这张编码表中，各个信息按概率大小排序、建立，出现概率大的信息所得编码长度短，出现概率小的信息所得编码长度长，这使得编码后的编码字符串的平均长度最短，从而达到无损压缩数据的目的。论文介绍了霍夫曼编码在matlab下的算法，算法根据用户输入的各信息概率，计算得出相对应的霍夫曼编码、总体平均码长以及编码效率。

**关键词：**霍夫曼编码；霍夫曼树；数据压缩

**Huffman Coding Algorithm Research**

Zhang Shuqi

(Southeast University, Nanjing, Jiangsu Province, 211100)

**Abstract:** Huffman proposed an optimal code construction method in 1952, which constructs the encoding length completely based on the probability of occurrence of characters, which is called Huffman Code. This code is based Huffman tree - the optimal binary tree as the core of the encoding, but also a variable length coding (VLC), used in data compression, can get the average shortest code length. Hoffman coding is an entropy coding method in computer information processing. Entropy coding refers to the use of a special code table to encode various signals. In this encoding table, all information is sorted and established according to the probability. The encoding length of information with high probability is short, while that of information with low probability is long, which makes the average length of encoded string the shortest, thus achieving the purpose of lossless compression of data. This paper introduces the algorithm of Huffman coding in matlab. The algorithm calculates the corresponding Huffman coding, the overall average code length and the coding efficiency according to the probability of each information input by users.

**Keyword:** Huffman Coding; Huffman Tree; Data Compression

在网络技术蓬勃发展的现代社会，互联网中的存储流动的信息数据之庞大犹如浩瀚汪洋，数据压缩显示出其非常的重要性，也是计算机科学中经久不衰的课题，一个好的数据压缩方法往往能明显减少数据的存储空间，大大提高存储媒体的访问速度。[1]霍夫曼编码作为最优码之一，能有效降低平均编码长度，是一种十分简单有效的编码方式。它完全按照数据出现概率为序，决定各数据的编码长度，出现概率小则编码长度长，反之则编码长度短。本文将从编码原理，最优性证明和算法探究三个部分解释霍夫曼编码。

**1 编码原理及步骤简述**

**1.1 例**

以下述为例：有8个符号的信源X，概率P表示各符号出现概率。

P(X1)=0.12 , P(X2)=0.17 ,

P(X3)=0.24 , P(X4)=0.08 ,

P(X5)=0.03 , P(X6)=0.21 ,

P(X7)=0.09 , P(X8)=0.06 .

进行霍夫曼编码，探究此编码方式。

1.将各信息符号依照概率大小排序；

P(X5)=0.03 , P(X8)=0.06 ,

P(X4)=0.08 , P(X7)=0.09 ,

P(X1)=0.12 , P(X2)=0.17 ,

P(X6)=0.21 , P(X3)=0.24 .

2.取出现概率最小的两个信息符号分别赋以0和1的码元；

3.将出现概率最低的两个概率相加，重新排序；

P(X4)=0.08 , P(X7)=0.09 ,

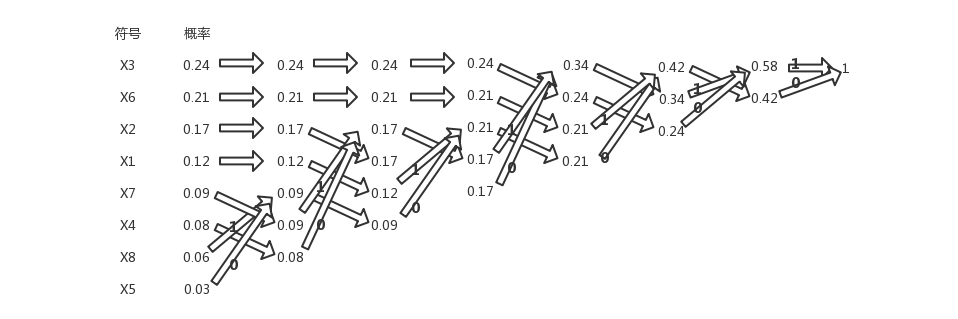
P(X58)=0.09 , P(X1)=0.12,

P(X2)=0.17 , P(X6)=0.21,

P(X3)=0.24.

4.重复步骤2.3.直至最后所加概率总和为1或全部概率叠加完毕；

5.从最后一级依次回溯读取码元；



编码结果：

X3 10

X6 00

X2 110

X1 011

X7 1111

X4 1110

X8 0101

X5 0100

**1.2计算例中数据**

1.编码平均码长：

L=2\*(0.24+0.21)+3\*(0.17+0.12)+4\*(0.09+0.08+0.06+0.03)=2.81（码元/符号）

2.信息熵：



=-(0.24log0.24+0.21log0.21+0.17log0.17+0.12log0.12+0.09log0.09+0.08log0.08+0.06log0.06+0.03log0.03)=1.91868（码元/符号）

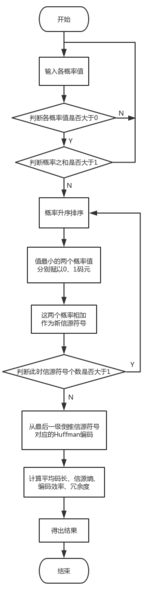
3.编码效率：

η==1.91868/2.81=0.6828

4.冗余度：

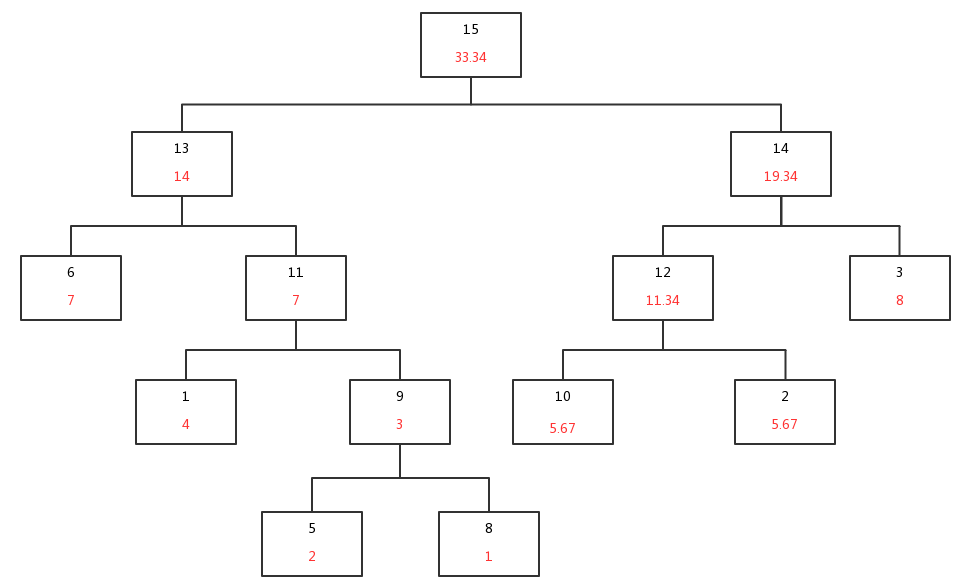
r=1-η=1-0.6828=0.3172

5.程序流程图



1. 霍夫曼树

给定n个权值作为n个[叶子结点](https://baike.baidu.com/item/%E5%8F%B6%E5%AD%90%E7%BB%93%E7%82%B9/3620239" \t "https://baike.baidu.com/item/%E5%93%88%E5%A4%AB%E6%9B%BC%E6%A0%91/_blank)，构造一棵二叉树，若该树的带权路径长度达到最小，称这样的二叉树为最优二叉树



**2 霍夫曼编码最优性证明**

**2.1相关概念简述**

1、路径长度

在树中一个结点到另一个结点所经历的分支构成路径上的分支数目成为路径长度。

2、树的路径长度

从树根到树中的每一个结点的路径长度之和。

3、树的带权路径长度（Weighted Path Length of Tree, WPL）

**2.2证明**

结点的带权路径长度：结点到树根之间的路径长度与该结点上权的乘积。

树的带权路径长度（WPL）：树中所有叶结点的带权路径长度之和

n表示叶结点的数目

分别表示叶结点的权值和根到结点之间的路径长度

WPL=

其中，假设是是其中最小的两个权值，则这两个数对应的结点是兄弟结点，且这两结点在二叉树中的深度不小于其他任何一个叶结点的深度。

证明：由霍夫曼树的构建过程可知，所在结点是兄弟节点。假设这两个结点的深度不是最深的，则存在一些结点（假设为结点X，其父结点为U），其深度大于所在结点的深度。则所在结点的父结点（V）的权重应大于结点X权重，否则构建霍夫曼树时，应选择V而不是X作为U的子结点。但这是不可能的，因为假设时权重最小的两个元素，矛盾，问题得证。定理1：哈夫曼树是最优的。

证明：用归纳法证明。

基本情况: 当 n=2 时, 哈夫曼树具有最小权重外部路径（EPW），因为树 仅有二种可能，有二个叶结点的二种哈夫曼树下的EPW是相同的。假设: 设有哈夫曼树有 个叶子时，定理成立。推导: 令T为有n （n>=2）个叶子的哈夫曼树。不失 一般性，设 w1 <= w2 <=... <=wn。令是w1 与w2的父结点。由引理1知,

在T中，不存在叶结点，其深度大于叶结点w1 与w2的深度。若存在深度大于w1, w2深度的结点，我们可以通过将之与w1, w2交换，由此得到更小的WPL。按如下方式得到到二叉树T'：以结点V'替换结点V, 其中V'的权重是w1+w2，则T'是相应于{w1+w2,w3,...,wn}的一棵哈夫曼树。根据归纳假设，T'具有最小权重外部路径，T是最优的（EPW最小）。在T'的结点V'上添加叶结点w1,

w2，可得T，则T是具有最小权重外部路径的哈夫曼树。由此，我们由数学照片纳法证明了定理1. 【3】

**3 霍夫曼编码的特点**

能载荷一定的信息量，且码字的平均长度最短，可分离的变长的码字集合称为最佳变长码。为此必须将概率大的信息符号编以短的码字，概率小的符号编以长的码字，使得平均码字最短，为最佳编码方法。霍夫曼码是用概率匹配的方法进行信源编码。

**3.1最佳编码之一**

1.霍夫曼的编码方法保证了概率大的符号对应于短码，概率小的符号度对应于长码，充分利用了短码；

2.霍夫曼是一种即时码：由于代表信源符号的节点都是终端节点，因此其编码不可能是其他终端节点对应的编码的前缀，即霍夫曼编码所得的码字为即时码。

3.霍夫曼编码是一种分组码，各个信源符号都被映射成一组固定次序的码符号。

4.霍夫曼编码是一种唯一可解的码，任何符号序列只能以一种方式译码。

注：它可以单个信源符号编码或用L较小的信源序列编码。但是应当注，要达到很高的效率仍然需要按长序列计算，这样才能使平均码字长度降低。编码过程中，当缩减信源的概率分布重新排列时，应使合并得到的概率和尽量处于高的位置，这样可使合并的元素重复编码次数减少，使短码得到充分利用。

**3.2霍夫曼编码的非唯一性**

霍夫曼编码方法得到的码并非是唯一的，造成并非唯一的原因是：（1）首先，每次对信源缩减时，赋予信源最后两个概率最小的符号，用0和1可以任意的，所以可以得到不同的霍夫曼码，但不会影响码字的长度。（2）对信源进行缩减时两个概率最小的符号合并后的概率与其他信源符号的概率相同时，这两者在缩减信源中进行概率排序，其位置放置次序是可以任意的，故会得到不同的霍夫曼码。此时将影响码字的长度，一般将合并的概率放在上面，这样可以获得较小的码方差。[2]

**4 霍夫曼编码思路简析（伪代码）**

输入概率数组[A,B,C...],定义n为数组长度，即编码元素个数；

For i=1：n

If p(i)<0

输出“概率值不能小于0”

重新输入数据

end

If abs(sum(p))>1

输出“概率总和不能大于1”

重新输入数据

end

q=p

a=zeros(n-1,n)生成一个n-1行n列的数组

For i=1:n-1

[q,l]=sort(q) 概率升序排序

记录每行最小量概率叠加后的概率排列次序

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

for i=1:n-1

c(i,1:n\*n)=blanks(n\*n)

生成n-1行，n\*n列的矩阵c

每行看作n段，每段长度n，记录一个码字（每个码字长度不超过n）

c(n-1,n)='0'; %?给C矩阵的N-1行的第一个段赋值0。?

c(n-1,2\*n)='1';%第二个段赋值1。（这两个码字对应编码中最后相加为一的两个概率。）

for i=2:n-1 %?主要的程序，循环N-2次

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

c(n-i,n)='0';%根据之前的规则，在分支的第一个元素最后补0

c(n-i,n+1:2\*n-1)=c(n-i,1:n-1)

c(n-i,2\*n)='1'%根据之前的规则，在分支的第一个元素最后补1决定矩阵C从倒数第二行开始到第一行的每段的码字值。每一行值都从下一行值得到，找到在下一行码字中相加本行最小两个概率得到的概率的对应码字，本行两个最小概率对应码字分别为此码字最后加“0”，加“1”。

for j=1:i-1

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

end

end 完成huffman码字的分配

计算平均码长

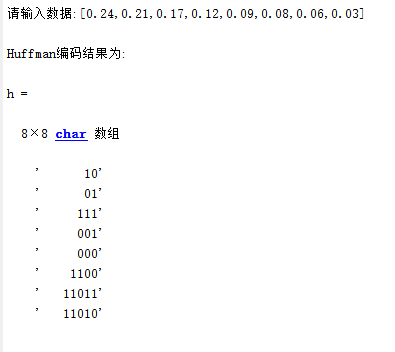
计算信源熵

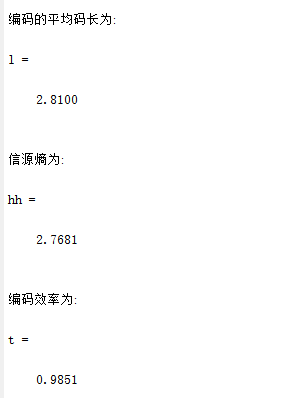
计算编码效率及冗余度

fprintf输出

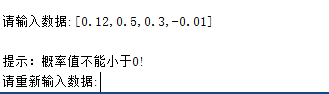
**5 上例算法运算结果分析**

上述例子中的数据运算结果如下图所示

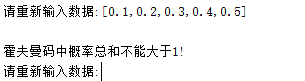




测试概率值小于0的情况：



测试概率值总和大于1的情况：



**总 结：**

霍夫曼编码是数据压缩领域中最著名的编码方式之一。它通过对象频率出现的不等性，构造最优编码，达到减小文件大小的目的。可以说霍夫曼编码是现在广泛应用的其他高效数据压缩算法（如算术编码，可预测编码）的基础。因此，学习研究霍夫曼编码的思想，对于深入学习数据结构以及数据压缩算法等学科的相关课题是十分有益的。霍夫曼编码的提出虽然已有一定年头，但其思路经典而不失其优越性，依旧非常值得学习借鉴。

[1]朱怀宏，吴楠，夏黎春.利用优化哈夫曼编码进行数据压缩的探索[A].文章编号：1005-3751(2002)05-0001-06

[2]网络博客:

<https://blog.csdn.net/BigLeo/article/details/41243779>

[3]Thomas Cover and Joy Thomas. Elements of Information Theory, Cambridge University Express[M]