文章编号:2095-7386(2016)01-0088-04
DOI:10.3969/j.issn.2095-7386.2016.01.019

一种哈夫曼编码的改进算法

王防修[1],刘春红[2]
(1.武汉轻工大学 数学与计算机学院,湖北 武汉 430023;2.九州通医药集团物流有限公司,湖北 武汉 430040)

摘 要:针对哈夫曼编码需要用到指针和结构体而导致使用受到限制的问题,提出一种不用指针和结构体也能进行哈夫曼编码的算法。算法以哈夫曼编码的编码原理为基础,先自底向上得到各个中间结点的双亲结点和孩子结点,然后自顶向下得到各个结点的二进制码字,最后得到的叶子结点的码字就是哈夫曼编码。由于所设计的哈夫曼编码算法只需要使用一维数组即可以实现,故对完成编码的计算机语言没有任何限制。算例仿真表明,使用三个一维数组即可实现任何事件的哈夫曼编码。

关键词:哈夫曼编码;中间结点;码字;叶子结点
中图分类号:TP 391 **文献标识码**:A

An Improved Algorithm of Huffman Encoding

WANG Fang-xiu[1], *LIU Chun-hong*[2]
(1. School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan 430023, China;
2. Jointown Phamaceutical Group Logistics Co., Ltd. Wuhan 430040, China)

Abstract: According to the application limitation of Huffman encoding due to the need of using the pointer and structure body, a Huffman encoding algorithm without the pointer and structure body was presented in this paper. The algorithm is based on the encoding principle of Huffman encoding. First, from the bottom up to the top it can get the parent node and child nodes of every intermediate node. Second, from the top down to the bottom it can get the binary code of each node. Finally, codewords of all the leaf nodes consist of the Huffman encoding. The Huffman encoding algorithm can be achieved only by using a one-dimensional in this paper, so the completion of the encoding doesn't depend on any computer language. The simulation results show that three one-dimension arrays can realize the Huffman encoding of any event.

Key words: Huffman coding; intermediate node; code word; Leaf node

1 引言

作为一种压缩率最高的无损压缩编码[1],哈夫曼编码的算法实现一直是人们关注的热点问题[2-5]。在哈夫曼编码的计算机实现过程中,普遍都需要使用指针和结构体构造哈夫曼树。然而,许多计算机高级语言并不支持指针和结构体,比如常用的basic语言。同样,许多网络语言也不支持指针和结构体,

收稿日期:2015-12-16.
作者简介:王防修(1973-),男,副教授,E-mail:wfx323@126.com.
基金项目:国家自然科学基金资助项目(61179032).

比如 vbscript 脚本语言。如果能够设计一个不需要指针和结构体的哈夫曼编码算法,则哈夫曼编码的使用可以得到进一步推广[6-8]。因此,本文研究并设计一种不需要指针和结构体也可实现哈夫曼编码的算法。

2　哈夫曼编码的原理

设 $x_i(i = 1,2,\cdots,n)$ 是信源 S 的 n 个事件,而 w_i 是 x_i 在信源 S 中出现的频率。由哈夫曼编码的原理,可以得到事件 x_i 对应的编码 b_i。为了方便计算机实现同时又不能使用指针和结构体,需要探索现有这些变量之间的联系。

2.1　中间结点权重的计算

根据哈夫曼编码原理,如果把 $w_i(i = 1,2,\cdots,n)$ 看作编码结点的权重,则已经具有哈夫曼编码的 n 个叶子结点权重。由于哈夫曼编码总共需要 $2n-1$ 个结点权重,故还需要计算 $n-1$ 个中间结点的权重 $w_i(i+1,i+2,\cdots,2n-1)$。对于第 j 个中间结点的权重 w_j 而言,其权重是其左右孩子结点的权重之和。因此,要求权重 w_j,必须先求出它的两个孩子权重。为方便起见,不妨设 w_j 的左孩子权重为 w_l 和右孩子权重为 w_r。

第 j 个中间结点的左孩子结点权重的选择算子如下:

$$\begin{cases} w_l = \min\{w_i \mid f_i = 0, i = 1,2,\cdots,j-1\}. \\ f_l = -1. \end{cases} \tag{1}$$

式(1)中的 w_l 表示从所有未被选择的权重中选择最小的权重,其中 $f_i = 0$ 表示权重 w_i 没有被选。而式中的 $f_l = -1$ 表示权重 w_l 将作为第 j 个中间结点的左孩子权重。

第 j 个中间结点的右孩子结点权重的选择算子如下:

$$\begin{cases} w_r = \min\{w_i \mid f_i = 0, i = 1,2,\cdots,j-1\}. \\ f_r = 1. \end{cases} \tag{2}$$

式(2)中的 $f_r = 1$ 表示权重 w_r 将作为第 j 个中间结点的右孩子权重。

通过式(1)和式(2)可以求出第 j 个中间结点的左孩子结点权重 w_l 和右孩子结点权重 w_r,则第 j 个中间结点的权重计算如下:

$$\begin{cases} w_j = w_l + w_r, \\ f_j = 0. \end{cases} \tag{3}$$

式(3)中 $f_j = 0$ 表示权重 w_j 可以进一步作为其它尚未求出权重的中间结点的子权重。

当 w_l 和 w_r 分别被选择为第 j 个中间结点的左右孩子权重后,然后通过式(3)求出第 j 个中间结点的权重 w_j 后,为了方便接下来的哈夫曼编码,需要更新 w_l 和 w_r 的值,更新过程如下:

$$w_l = w_r = j. \tag{4}$$

式(4)中的 w_l 和 w_r 被用来保存第 l 个结点和第 r 个结点的双亲结点位置 j。

当计算出所有中间结点的权重之后,除了第 $2n-1$ 个结点的权重 w_{2n-1} 存储所有非根结点的权重之和外,其他权重变量 w_i 保存的都是其双亲结点的位置。同样,只有 $f_{2n-1} = 0$,其它的标志变量 f_i 不是-1 就是 1。总之,接下来变量 w_i 和 f_i 将作为求码字 b_i 的依据。

2.2　求每个结点的码字

设 $b_i(i = 1,2,\cdots,2n-1)$ 是第 i 个结点的码字,则需要先求出其双亲结点的码字,然后才能求出孩子结点的码字。

首先,根结点不需要编码,故做如下的初始化工作:

$$b_{2n-1} = w_{2n-1} = 0. \tag{5}$$

式(5)中 $b_{2n-1} = 0$ 是方便后继结点的编码而设置的,而 $w_{2n-1} = 0$ 表示第 $2n-1$ 个结点的码字长度为 0。

接下来是依次求码字 b_i,而每个码字只有先求出其双亲的码字,然后才能求出它自身的码字,故求解 b_i 的顺序是下标递减,即 $i = 2n-2,2n-3,\cdots,1$。

当求第 i 个码字 b_i 时,需要根据 f_i 的值来决定编码过程。

如果 $f_i = -1$,则 b_i 是双亲结点的左孩子码字,故

$$b_i = 2b_{w_i} + 0. \tag{6}$$

如果 $f_i = 1$,则 b_i 是双亲结点的右孩子码字,故

$$b_i = 2b_{w_i} + 1. \tag{7}$$

在式(6)和式(7)中,由于 w_i 表示第 i 个结点的双亲位置,故第 i 个结点的编码就是在其双亲码字的末位补上一个二进制位,其中式(6)表示末位补 0,而式(7)表示末位补 1。

当求出码字 b_i 后,w_i 已经完成其编码任务。也就是说,此后不再需要用它来保存第 i 个结点的双亲位置。所以,可以用 w_i 保存码字 b_i 的码长,其计算过程如下:

$$w_i = w_{w_i} + 1. \quad (8)$$

显然,式(8)体现了孩子结点的码长比其双亲结点的码长多1。

3 哈夫曼编码的算法实现

从上面的编码原理可知,要想求出某一信源的哈夫曼编码,必须先求出所有中间结点的权重。当计算出所有中间结点的权重后,再由结点之间的关系得到哈夫曼编码。因此,编码算法实现过程分为两个阶段。

3.1 求中间结点的权重

对于一个包含 n 个事件的信源而言,它总共有 $2n-1$ 个权重,其中 n 个权重是 n 个事件在信源中的频率,而其他 $n-1$ 个权重需要计算得到。具体的求解过程如下:

(1) 初始化。为了从权重序列中选择最小权重,需要设置各权重的初始标志位为0,即令 $f_i = 0(i = 1,2,\cdots,2n-1)$。

(2)令 $j = n+1$,则 j 表示第一个需要求权重的中间结点的位置。

(3) 由式(1)求出第 j 个中间结点的左孩子权重 w_l。

(4) 由式(2)求出第 j 个中间结点的右孩子权重 w_r。

(5)由式(3)求出第 j 个中间结点的权重 w_j。

(6)由式(4)保存左孩子和右孩子结点的双亲位置。

(7)令 $j = j+1$。如果 $j \le 2n-1$,则转步(2)重复进行;否则,求中间结点权重的过程结束。

从上述过程可以看出,求 w_j 是一个递增的过程,即 $j = n+1, n+2, \cdots 2n-1$。

3.2 求各结点的码字

当所有中间结点的权重计算完成后,除 w_{2n-1} 保存的是所有结点的权重之和外,其它结点的权重变量 $w_i(i = 1,2,\cdots,2n-2)$ 保存的不再是自身权重,而是第 i 个结点的双亲位置。同样,除了 $f_{2n-1} = 0$ 外,其它标志变量 $f_i(i = 1,2,\cdots,2n-2)$ 不是1就是-1。因此,信源的哈夫曼编码的过程如下:

(1)令 $b_{2n-1} = w_{2n-1} = 0$ 和 $i = 2n-2$。

(2)如果 $f_i = -1$,那么 $b_i = 2b_{w_i}$;否则,$b_i = 2b_{w_i} + 1$。

(3)令 $w_i = w_{w_i} + 1$,表示第 i 个结点的码长是其双亲的码长加1。

(4)令 $i = i-1$。如果 $i \ge 1$,则转步(2)重复进行;否则,编码过程结束。

从上述编码过程可以看出,$w_{2n-1} = 0$ 表明根结点的码字 b_{2n-1} 是长度为0的空码,而 b_i 是长度为 w_i 的码字,其中 $i = 1,2,\cdots,2n-2$。在这里,只有 $b_i(i = 1,2,\cdots,n)$ 是前缀码。因此,$b_i(i = 1,2,\cdots,n)$ 就是所求的哈夫曼编码。

4 算法仿真

算例1 假设有7个信源符号,其频率分布为 $\{20,19,18,17,15,10,1\}$,要求用本文算法对其进行哈夫曼编码。

解 首先,叶子结点的权重 w_i 和选择标志 f_i 的初始化,即

$$w_1 = 20,\ w_2 = 19,\ w_3 = 18,\ w_4 = 17,$$
$$w_5 = 15,\ w_6 = 10,\ w_7 = 1.$$
$$f_i = 0,\ i = 1,2,\cdots,7.$$

其次,通过算法3.1求第 i 个中间结点的权重 w_i,其中 $i = 8,\cdots,13$,其过程如下:

(1)求第8个结点的权重 w_8 及其左右孩子结点选择:

$$w_7 = \min\{w_1,w_2,w_3,w_4,w_5,w_6,w_7\}$$

$f_7 = -1$。所以 w_7 是左孩子结点权重。因为 $w_6 = \min\{w_1,w_2,w_3,w_4,w_5,w_6\}$,$f_6 = 1$.
所以 w_7 是右孩子结点权重。故

$$w_8 = w_7 + w_6 = 11,\ f_8 = 0,\ w_6 = w_7 = 8.$$

(2)求第9个结点的权重 w_9 及其左右孩子结点选择:

$$w_8 = \min\{w_1,w_2,w_3,w_4,w_5,w_8\},\ f_8 = -1.$$
$$w_5 = \min\{w_1,w_2,w_3,w_4,w_5\},\ f_5 = 1.$$

所以 $w_9 = w_5 + w_8 = 26$ 和 $f_9 = 0$,$w_5 = w_8 = 9$.

(3)求第10个结点的权重 w_{10} 及其左右孩子结点选择:

$$w_4 = \min\{w_1,w_2,w_3,w_4,w_9\},\ f_4 = -1,$$
$$w_3 = \min\{w_1,w_2,w_3,w_9\},\ f_3 = 1.$$

所以 $w_{10} = w_3 + w_4 = 35$,$f_{10} = 0$,$w_3 = w_4 = 10$.

(4)求第11个结点的权重 w_{11} 及其左右孩子结点选择:

$$w_2 = \min\{w_1,w_2,w_9,w_{10}\},\ f_2 = -1.$$
$$w_1 = \min\{w_1,w_9,w_{10}\}\ f_1 = 1.$$

所以有 $w_{11} = w_1 + w_2 = 39$,$f_{11} = 0$. $w_1 = w_2 = 11.$

(5)求第12个结点的权重 w_{12} 及其左右孩子结点选择:

$$w_9 = \min\{w_9,w_{10},w_{11}\},\ f_9 = -1.$$
$$w_{10} = \min\{w_{10},w_{11}\},\ f_{10} = 1.$$

$w_{12} = w_9 + w_{10} = 61$, $w_9 = w_{10} = 12$.

(6)求第 13 个结点的权重 w_{13} 及其左右孩子结点选择:

$w_{11} = \min\{w_{11}, w_{12}\}$, $f_{11} = -1$, $f_{12} = 1$.

$w_{13} = w_{11} + w_{12} = 100$, $w_{11} = w_{12} = 13$.

最后,运用算法 3.2 进行哈夫曼编码。

(1)令 $b_{13} = \Phi$ 。因为 $w_{11} = w_{12} = 13$,而 $f_{11} = -1$, $f_{12} = 1$,所以 $b_{11} = 0$, $b_{12} = 1$.

(2)因为 $w_9 = w_{10} = 12$,而 $f_9 = -1$, $f_{10} = 1$,所以 $b_9 = 10$, $b_{10} = 11$.

(3)因为 $w_5 = w_8 = 9$,而 $f_8 = -1$, $f_5 = 1$,所以 $b_5 = 101$, $b_8 = 100$.

(4)因为 $w_6 = w_7 = 8$,而 $f_7 = -1$, $f_6 = 1$,所以 $b_6 = 1001$, $b_7 = 1000$.

(5)因为 $w_3 = w_4 = 10$,而 $f_4 = -1$, $f_3 = 1$,所以 $b_3 = 111$, $b_4 = 110$.

(6)因为 $w_1 = w_2 = 11$,而 $f_2 = -1$, $f_1 = 1$,所以 $b_1 = 01$, $b_2 = 00$.

算例 2　假设有 4 个信源符号,其权重分布为{40,30,20,10},要求用本文算法对其进行哈夫曼编码。

解　首先,对结点的权重和选择标志的初始化如表 1 所示。

表 1　权重和标志的初始化

I	1	2	3	4	5	6	7
w_i	40	30	20	10	0	0	0
f_i	0	0	0	0	0	0	0

通过算法 3.1 得到如表 2 的哈夫曼编码的中间结果。

表 2　编码的中间结果

I	1	2	3	4	5	6	7
w_i	7	6	5	5	6	7	100
f_i	-1	-1	1	-1	1	1	0

通过算法 3.2 得到如表 3 的哈夫曼编码。

表 3　哈夫曼编码结果

I	1	2	3	4	5	6	7
w_i	1	2	3	3	2	1	0
b_i	0	10	111	110	11	1	

从表 3 可以得到对应的哈夫曼编码为{0,10,111,110}。

表 1 中的 w_i 表示第 i 个结点的权重,表 2 中的 w_i 表示第 i 个结点的双亲结点的位置,表 3 中的 w_i 表示码字 b_i 的码长。

5　结束语

笔者提出了一种不用指针和结构体的哈夫曼编码算法。该算法使用一维数组保存各结点的编码信息,在整个编码过程中不需要建立哈夫曼树。无论从算法设计还是算法仿真可以看出,整个算法都只使用一维数组,而数组是任何计算机语言都具有的功能。需要说明的是,前面所说到的双亲结点和孩子结点,只是为方便说明元素之间的关系而已,并不代表该编码需要建立哈夫曼树。根据笔者设计的哈夫曼编码算法,用任何计算机语言都可实现哈夫曼编码,而这对于哈夫曼编码的推广和应用具有重要意义。

参考文献:

[1] 叶芝慧,沈克勤. 信息论与编码[M]. 北京:电子工业出版社,2013.

[2] 王向鸿. 基于 Matlab 文本文件哈夫曼编解码仿真[J]. 现代电子技术,2013,36(20):31-32.

[3] 薛向阳. 基于哈夫曼编码的文本文件压缩分析与研究[J]. 科学技术与工程, 2010,10(23):5779-5781.

[4] 程佳佳,熊志斌. 哈夫曼算法在数据压缩中的应用[J]. 电脑编程技巧与维护,2013(2):35-37.

[5] 高健,陈耀. 分组无损图像压缩编码方法[J]. 计算机工程与设计, 2010,31(15):3447-3450.

[6] 李灵华,刘勇奎. Freeman 四方向链码压缩率提高的方法研究 [J]. 计算机工程与设计,2013,34(3):1132-1136.

[7] 王敏超,王敏莉,李秋生,等. 无损自适应分布式算术编码的研究及应用[J]. 计算机工程与设计,2011,32(10):3470-3475.

[8] 王防修,周康,同小军. 一种不用构造二叉树的哈夫曼编码[J]. 武汉工业学院学报,2012,31(2):52-54.