文章编号:2095-7386(2016)02-0079-04
DOI:10.3969/j.issn.2095-7386.2016.02.014

基于哈夫曼编码的选择算法

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.鄂钢驰久钢板弹簧有限责任公司,湖北 鄂州 436000)

摘 要:针对同一哈夫曼树有多种不同哈夫曼编码的问题,本文提出一种哈夫曼编码的选择算法。算法以哈夫曼编码的多样性为基础,在哈夫曼树的非叶子节点处提供编码方式0或1,由所有非叶子节点的编码方式组成一个二进制序列,最后根据该二进制序列进行节点的哈夫曼编码。鉴于哈夫曼编码的递归子结构,本文设计了一种不同于传统哈夫曼编码的回溯算法。算例仿真表明,一方面同一事件有时可以构造不同的哈夫曼树,另一方面同一哈夫曼树根据编码方式的不同可以得到不同的哈夫曼编码结果。

关键词:哈夫曼树;哈夫曼编码;选择算法;回溯算法;递归子结构
中图分类号:TP 391 **文献标识码**:A

Selection algorithm based on huffman coding

WANG Fang-xiu[1],*LIU Chun-hong*[2]
(1. School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan, 430023, China;
2. Ezhou Iron and Steel Plate Spring Co., Ltd. Ezhou, 436000, China)

Abstract: Aiming at the same Huffman tree having a variety of different Huffman coding, this paper proposes a Huffman code selection algorithm. Based on diversity of Huffman coding, the algorithm provides 0 or 1 as coding method for every non leaf node of the huffman tree. A binary sequence is constructed by the composition of all non leaf node coding method, finally Huffman coding is obtained according to the binary sequence. In view of the fact that the recursive substructures of Huffman coding, this paper designs a backtracking algorithm that is different from the traditional Huffman coding. Simulation results show, on the one hand, sometimes the same event can contruct different Huffman tree. on the other hand according to the different coding methods different results of Huffman coding can obtained from the same Huffman tree.

Key words: Huffman tree; Huffman code; Selection algorithm; Backtracking algorithm; Recursive sub structure

1 引言

作为一种无损压缩编码,哈夫曼编码比其他编码具有更高的压缩效率[1]。常见的哈夫曼编码,是先建立哈夫曼树,然后由哈夫曼树得到哈夫曼编码[2-4]。按照现有的哈夫曼编码算法[5,6],只能得到两种编码结果中的一种。然而,根据哈夫曼树的结

收稿日期:2015-12-31.
作者简介:王防修(1973-),男,副教授,硕士,E-mail:wfx323@126.com.
基金项目:国家自然科学基金资助项目(61179032)

构和所采用的编码方式,应该存在更多的编码结果。因此,对一个具体事件的哈夫曼树而言,究竟可以得到多少不同的编码结果,该问题的研究尚未见之于文献。此外,实现哈夫曼编码不同结果的任意选择,也是需要解决的问题。针对这两个问题,本文一方面需要从理论上证明由同一哈夫曼树可以得到不同的编码结果,另一方面需要设计为得到不同哈夫曼编码结果的选择算法。

2 哈夫曼编码不唯一

对同一事件进行哈夫曼编码,虽然编码的平均码长是唯一的,但哈夫曼编码的结果不唯一[1]。具体原因表现在以下两个方面:(1)同一事件的哈夫曼树可能不唯一;(2)同一哈夫曼树的编码方式不唯一。

定理 如果编码事件包含 n 个权重,则该事件对应的哈夫曼树有 2^{n-1} 种不同的哈夫曼编码结果。

证明 因为编码事件有 n 个权重,故相应的哈夫曼树总共有 $2n-1$ 个节点。除了 n 个叶子节点没有孩子节点外,其他 $n-1$ 个非叶子节点均有两个孩子节点。对每个非叶子节点而言,如果对指向左孩子的分支编码为 0,由前缀码[1]可知指向右孩子的分支编码必须为 1;相反,如果指向左孩子的分支编码为 1,则指向右孩子的分支编码必须为 0。也就是说,从每个非叶子节点出发,都有两种编码方式。由于有 $n-1$ 个非叶子节点而这些节点又是相互独立的,故由排列组合可知,总共有 2^{n-1} 种不同的编码结果。

总之,对同一事件编码,不同哈夫曼树的哈夫曼编码结果不同,而同一哈夫曼树按不同的编码方式也可以得到不同的哈夫曼编码结果。如果同一事件可以构造 m 棵不同的哈夫曼树,由每棵哈夫曼树有 2^{n-1} 种不同编码结果,则总共就有 $2^{n-1}m$ 种不同编码结果。

3 哈夫曼编码的选择算法原理

一个事件的哈夫曼编码不仅与建立的哈夫曼树有关,而且与采用的编码方式有关。为方便起见,一般常采用以下两种编码方式:(1)所有非叶子节点指向左孩子的分支统一编码为 0,指向右孩子的分支统一编码为 1;(2)所有非叶子节点指向左孩子的分支统一编码为 1,而指向右孩子的分支统一编码为 0。

故由以上两种编码方式只能得到两种编码结果,而无法得到哈夫曼编码的其他任何一种结果。如果想得到其他哈夫曼编码结果,则必须改变分支的编码方式,也就是必须对非叶子节点分支的编码方式能够任意选择,即从哈夫曼树的根节点出发,对树中的每个非叶节点分支的编码方式不是统一规定而是逐个规定。具体做法是:在哈夫曼树的遍历过程中,每遇到一个非叶子节点,则为其提供一个 0 或 1 的整数。如果该整数是 0,则表示当前非叶子节点指向左孩子的分支编码为 0,指向右孩子的分编码为 1;否则,如果提供的整数是 1,则当前非叶子节点指向左孩子的分支编码为 1,指向右孩子的分编码为 0。总之,上述选择编码方式要求对每个非叶子节点的分支编码方式都要依次规定。

如果待编码事件有 n 个权重,则由其构造的哈夫曼树有 $n-1$ 个非叶子节点,每一个非叶子节点有两种编码方式,故总共有 2^{n-1} 种编码方式。如果为每个非叶子节点提供 0 或 1 的编码方式,则总共需要提供 $n-1$ 个 0 或 1 的整数,即每种哈夫曼编码对应一个 $n-1$ 位的二进制数。由于 $n-1$ 位的二进制数能表示 2^{n-1} 种不同的二进制数,故一个 $n-1$ 位的二进制数对应一种哈夫曼编码结果。不难发现,常见的哈夫曼编码对应的二进制数是 $\underbrace{00\cdots0}_{n-1个0}$ 或 $\underbrace{11\cdots1}_{n-1个1}$。因此,要得到不同于这两种结果的哈夫曼编码,只需要提供一个不同的 $n-1$ 位二进制数即可。

总之,一个 $n-1$ 位二进制数对应一种哈夫曼编码结果,不同的 $n-1$ 位二进制数对应不同的哈夫曼编码结果。只要产生不同的 $n-1$ 位二进制数,就可以得到不同的哈夫曼编码结果。

4 哈夫曼编码的选择算法实现

要实现某一事件的哈夫曼编码,必须首先建立哈夫曼树,然后根据建立的哈夫曼树进行哈夫曼编码。

4.1 哈夫曼树建立

为了说明编码事件的哈夫曼树有可能不唯一,不妨将哈夫曼树的构造过程描述如下:

(1) 设 $w_i(i=1,2,\cdots,n)$ 是给定的 n 个权重,由这些权重构成 n 棵二叉树的对应集合为 $T=\{t_i \mid i=1,2,\cdots,n\}$,其中 t_i 的权重为 $t_i.weight=w_i$,而 t_i 的左右孩子满足 $t_i.lchild=t_i.rchild=\wedge$。

(2) 从 T 中选取权重最小的两棵二叉树 t_l 和 t_r,即 t_l 和 t_r 分别满足

$$t_r.weight = \min\{t.weight \mid t \in T\}, \quad (1)$$

$$t_r.weight = \min\{t.weight \mid t \in T - \{t_l\}\} \quad (2)$$

(3) 构造一棵根节点为 s 而左右孩分别为 t_l 和 t_r 的新的二叉树,其中 s 是两个孩子的全重之和,即

$$s.lchild = t_l, s.rchild = t_r \quad (3)$$

$$s.weight = t_l.weight + t_r.weight \quad (4)$$

(4) 从集合 T 中删除节点 t_l 和节点 t_r,向集合 T 中增加节点 s,即

$$T = T - \{t_l, t_r\}, \quad (5)$$

和 $T = T + \{s\}$。 (6)

(5)如果 $T \neq \{s\}$,则转步骤(2);否则,构造过程结束,二叉树 s 就是一棵哈夫曼树。

从上述算法中可以发现,新生成二叉树的根节点的权重有可能与原集合 T 中的某棵二叉树的根节点的权重相等,如果存在这种情况,并且在选择最小权重的节点时只能是其中之一,那到底是选择前者还是选择后者呢。如果选择前者,则得到的哈夫曼编码的码长变化较小。反之,如果选择后者,则码长变化较大。之所以会出现这种情况,是由于构造的二叉树不同所致。

4.2　哈夫曼编码的选择算法

有了哈夫曼树后,依据该哈夫曼树就可以进行哈夫曼编码了。现有的哈夫曼编码算法很多,不外乎递归或非递归算法。此处,不妨设计一个尚未见之于文献的回溯算法。

设 $b_i(i = 1,2,\cdots,n)$ 是存储权重 w_i 的二进制编码,$S = s_1 s_2 \cdots s_{n-1}$ 是一个 $n-1$ 位的二进制数。如果用递归函数 $f(t)$ 实现回溯编码,则 $f(t)$ 表示由 t 的双亲到 t 的编码过程。若用 c 保存递归过程的中间编码,则由当前编码 c 到 t 的编码必须满足:

如果 t 是深度优先搜索的第 i 个非叶子节点,则 t 的分支编码方式由 s_i 决定。如果 $s_i = 0$,则 t 到左孩子的编码为 $c = 2c + 0$,而 t 到右孩子的编码为 $c = 2c + 1$。相反,如果 $s_i = 1$,则 t 到左孩子的编码为 $c = 2c + 1$,而 t 到右孩子的编码为 $c = 2c + 0$。

由于有 n 个权重,故有 n 个编码,只有叶子节点的编码需要保存。如果 i 和 j 的初始值为 $i = j = 1$,则递归函数 $f(t)$ 的回溯过程如下:

(1) 如果 t 是叶子节点,则 c 当前存储的是权重 $t.weight$ 的编码,故令 $b_i = c$ 和 $i = i + 1$。然后返回递归上一层。

(2) 如果 $s_j = 0$,则 t 的左分支编码为:$ct = c$,$c = 2c + 0$ 和 $f(t.lchild)$;而 t 的右分支编码为:$c = ct$,$c = 2c + 1$ 和 $f(t.rchild)$。

(3)如果 $s_j = 1$,则 t 的左分支编码为:$ct = c$,$c = 2c + 1$ 和 $f(t.lchild)$;而 t 的右分支编码为:$c = ct$,$c = 2c + 0$ 和 $f(t.rchild)$。

(4) 令 $j = j + 1$,转步骤(1)。

从以上算法可以看出,它实质是一个深度优先搜索算法,搜索的结果完全取决于相应的哈夫曼树和编码方式 $S = s_1 s_2 \cdots s_{n-1}$。

5　算法仿真

本算法使用 VC6.0 作为仿真工具,在 CPU 为 3.2GHz 和内存为 1.86GB 的个人台式电脑上完成仿真。

算例 已知编码事件 $x_i(i = 1,2,\cdots,5)$ 的权重 $w_i(i = 1,2,\cdots,5)$ 分别为 8,4,4,2,2,求该事件的哈夫曼编码。

解 建立哈夫曼树。建法一:合并后的新权重总是比其他等权重要后选择,则建立的哈夫曼树见图 5.1 所示。

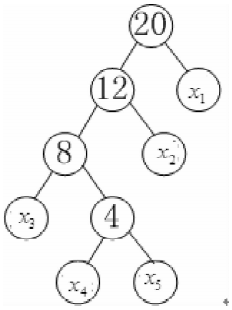

图 5.1　哈夫曼树(建法一)

建法二:合并后的新权重总是比其他等权重要先选择,则建立的哈夫曼树见图 5.2 所示。

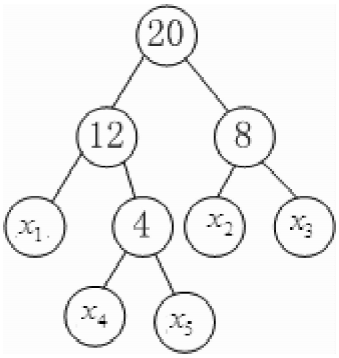

图 5.2　哈夫曼树(建法二)

由于建立的哈夫曼树有 4 个非叶子节点,故每棵哈夫曼树有 16 种编码结果。对图 5.1 所示的哈夫曼树,如果编码方式选择 1001 或 1010,则编码结果如表 5.1 所示。

表 5.1 建法一的两种编码

符号	权重	1001 编码	1010 编码
x_1	8	0	0
x_2	4	11	11
x_3	4	100	101
x_4	2	1011	1000
x_5	2	1010	1001

与表 5.1 相比,通过传统算法得到的哈夫曼编码的结果如表 5.2 所示。

表 5.2 建法一的传统编码

符号	权重	0000 编码	1111 编码
x_1	8	1	0
x_2	4	01	10
x_3	4	000	111
x_4	2	0010	1101
x_5	2	0011	1100

对图 5.2 所示的哈夫曼树,如果编码方式选择 1101 或 0010,则编码结果如表 5.3 所示。

表 5.3 建法二的两种编码

符号	权重	1101 编码	0010 编码
x_1	8	11	00
x_2	4	00	11
x_3	4	01	10
x_4	2	101	010
x_5	2	100	011

与表 5.3 相比,通过传统算法得到的哈夫曼编码的结果如表 5.4 所示.

表 5.4 建法二的传统编码

符号	权重	0000 编码	1111 编码
x_1	8	00	11
x_2	4	10	01
x_3	4	11	00
x_4	2	010	101
x_5	2	011	100

从上述不同的编码结果可以发现,每种编码结果中的码字互为前缀码。虽然编码的平均码长都是 2.2,但由建法二所建哈夫曼树得到的哈夫曼编码的各码字的码长变化相对较小。

6 结束语

本文以哈夫曼树为基础,通过对常规哈夫曼编码算法的改进,提出了一种可以得到更多哈夫曼编码结果的选择算法。仿真结果表明:对于一个有 n 个符号的信源而言,其哈夫曼编码的结果可以通过一个长度为 $n-1$ 的二进制数选择,证明了算法的有效性和哈夫曼编码结果的多样性。

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(02):35-37.

[5] 阚君满. 基于改进哈夫曼编码的全文索引结构压缩算法[J]. 吉林大学学报, 2011,29(5):473-476.

[6] 邓宏贵,郭晟伟,李志坚. 基于哈夫曼编码的矢量量化图像压缩算法[J]. 计算机工程,2010,36(4):218-222.