一种基于算术编码的文本数据压缩算法

李 英[1]，崔艳鹏[2]，高新波[1]

(1. 西安电子科技大学电子工程学院 西安 710071；2. 西安电子科技大学网络行为研究中心 西安 710071)

【摘要】提出了一种基于算术编码的文本数据压缩算法，将扫描产生的偏移量、匹配数据长度等全局优化问题转化为局部优化问题，并从Glomb编码思路出发，推导出一种参数选择算法；对LZ77算法进行修正，提出一种预测编码方法，获得预测参数。对预测参数、偏移量、数据匹配长度、保留文本数据使用MQ算术编码器进行编码，针对不同类型数据，设计出不同的编码算法和相应的上下文算法。对算法进行仿真，并与Winzip、WinRar压缩效率进行比较，结果表明对纯文本数据、Word文档数据、C语言程序代码，图像数据等，该压缩算法优于Winzip；在纯文本数据、Word文档数据、C语言程序代码压缩方面与WinRar相当或者略好，但在图像压缩方面的性能与WinRar相比略有不足。

关 键 词 算术编码； 参数优化； 预测编码； 文本数据压缩

中图分类号 TP391.1 **文献标志码** A doi:10.3969/j.issn.1001-0548.2016.06.009

A Novel Algorithm for Text Data Compression Based on Arithmetic Codec

LI Ying [1], CUI Yan-peng[2], and GAO Xin-bo[1]

(1.School of Electronic Engineering, Xidian University Xi'an 710071; 2. Institute for Internet Behavior, Xidian University Xi'an 710071)

Abstract A novel algorithm for text data compression is proposed based on arithmetic codec. The global parameters optimization is converted into the local parameter optimization, then Glomb code principle is used to solve the local optimization, and a parameter choice method is derived. The LZ77 scanning algorithm is improved in which a prediction code is proposed, and the prediction data is preserved. The parameters such as prediction data, offset, match data length and preserved text data are loaded into MQ codec in which the data can be compressed. To improve the compression efficiency, the corresponding compression algorithms and the context design algorithm are proposed. The proposed algorithm for text data compression is simulated and compared with Winzip and WinRAR. The results show that our compression algorithm has an advantage in compression effect over the Winzip for the data such as texts, word documents, C language program codes and images. Compared with WinRar, our algorithm achieved almost the same compression results for texts, word documents, C language program codes except images.

Key words arithmetic code; parameters optimization; predict code; text data compression

随着计算机技术和网络技术的发展，各种类型的数据层出不穷，海量的数据需要传输和存储。为了减少数据传输和存储的代价需要对数据进行压缩。根据不同的数据种类及重建质量要求，压缩算法也各不相同。比如语音压缩、图像压缩[1-6]等，根据重建质量的不同要求，可以进行限失真压缩或者无失真压缩。

由于文本数据必须进行精确重建，只能进行无失真压缩。目前文本压缩算法众多，许多算法是针对各种不同类型的应用[7-11]；广泛使用的文本压缩工具是WinRar和Winzip，这两种压缩算法涉及知识产权保护，详细编码过程未见公布，可能采用了预测编码、游程长度编码、LZ算法或者LZW等改进算法[7-11]。而这些算法主要突出于文本搜索算法，其中LZW搜索算法需要建立码书[9-10]，使用码书可以提高搜索速度，对于长串匹配数据而言，还可以有效减少LZ算法中的偏移量，利于提高编码效率。而这两种文本压缩算法使用的熵编码以及其他细节未见文献报道。

为了设计自主知识产权的文本压缩方法，本文拟采用原始LZ77算法，整个过程并不需要建立码表，目的主要在于尝试使用算术编码对文本数据扫描参数进行压缩，并为后续进一步研究奠定基础。

本文主要工作如下：提出了一种局部参数优化

收稿日期：2015－08－24；修回日期：2016－03－23

基金项目：国家自然科学基金(61571354)

作者简介：李英(1976－)，女，博士生，高级工程师，主要从事图像处理及编码、移动通信技术等方面的研究.

算法，从备选参数中选择合理的局部优化参数；对LZ77算法进行改进，增加了预测编码，并记录预测标记。预测编码标记为0和1组成的序列，使用已经编码的前面数据产生上下文，并使用算术编码器进行编码，能够有效提高编码效率。选择出的扫描参数，即偏移量、匹配数据长度和保留文本数据，都使用算术编码器进行编码，并根据各类数据的特点使用不同方法产生效应上下文。

1 改进文本数据压缩算法

1.1 基于Glomb编码的优化参数选取

LZ77文本压缩算法是将当前位置开始的、未知长度的文本数据与已经扫描过的文本数据进行匹配，查找到合理的匹配文本数据，记录下相应的偏移量、匹配数据长度、首次出现的文本数据等信息。

假设当前位置为i，从第一个匹配文本数据开始，搜索到的所有偏移量、匹配数据长度分别为Δ_{ij}、l_{ij}，其中j=1,2,…, N_i， N_i表示搜索到的匹配文本数据的数量。按照某种准则从N_i个备选参数集合(Δ_{ij}, l_{ij})来选择其中一组参数作为当前搜索的结果，从而使得整个文本数据压缩后的码流最短。

选择好的参数(Δ_i, l_i)来自于备选参数集合(Δ_{ij}, l_{ij})，表示为：

$$(\Delta_i, l_i)=\mathrm{T}(\Delta_{ij}, l_{ij}) \tag{1}$$

式中，T(·) 表示从参数集合中选择一种最佳参数。为了实现文本数据压缩，参数选择后应该经过熵编码器进行编码，从而得到文本压缩的码流，并从各种可能中选择最优参数。显然这种全局优化选择方法在工程中几乎无法实现。

除了上述备选集合参数外，还涉及到另外一个问题，就是首次出现的文本数据问题。如果文本压缩是全文搜索，且不论压缩效率如何，对备选参数集合中的参数进行压缩，那么首次出现的文本数据的数量实际是非常有限的，例如将文本数据按照8 bit划分，首次出现的文本数据的数量不大于2^8=256。更多情况下，文本数据压缩需要设置一个窗口，数据匹配是在窗口内进行的，即使在窗口内能够找到备选参数，为了提高压缩效率，也需要对备选参数进行选择，如果所有的备选参数都不能满足压缩效率的要求，则应该将当前文本数据视为首次出现文本数据。为了不至于产生误解，将这类文本数据称之为保留文本数据。

从备选参数集合中选择压缩编码所需参数，实际是一个全局优化问题。即：

$$L_{\min}=\min\left\{\sum_i [L(\Delta_i)+L(l_i)]+\sum_j L(\mathrm{pre}_j)\right\} \tag{2}$$

式中，$L(\Delta_i)$、$L(l_i)$分别表示被选择好的参数压缩所产生的比特数，如果没有从备选参数集合中选择出合适的参数，那么就将当前文本数据设为保留文本数据pre_j；$L(\mathrm{pre}_j)$表示保留字的压缩输出比特数。

对于该优化问题，最简单的思路就是考虑所有可能选择，从而得到最短压缩码流长度$L_{\min}$。但文本数据的统计特性十分复杂，且压缩编码的效率与参数选择过程密切相关，全局优化难以实现。考虑局部优化替代全局优化，从而便于工程实现。构造决策函数：

$$c_i = g\left(\frac{L(\Delta_{ij})+L(l_{ij})}{l_{ij}}, L(\mathrm{pre}_i)\right) \tag{3}$$

如果$c_i=0$，表示备选参数集合中没有参数被选中，当前文本数据作为保留文本数据；$c_i=1$，表示备选参数集合中有参数被选中，并将最优参数作为编码参数输出。

根据上述分析，可以得出文本数据压缩结构如图1所示。

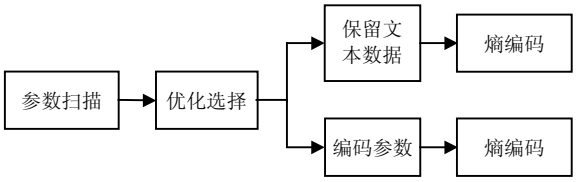

图1 文本压缩结构框图

上述局部优化看起来合理，但实际上无法实现。因为实际编码并没有进行，无法知道参数的编码比特数，无论是偏移量、长度还是保留文本数据。为了解决上述问题，本文采用一种较为简洁算法解决上述局部优化问题，具体如下。

首先假设偏移量、长度都是采用Glomb编码，对应的参数为k_1、k_2，根据Glomb编码可知，编码输出码长分别为：

$$L(\Delta_{ij})=\frac{\Delta_{ij}}{2^{k_1}}+k_1+1 \tag{4a}$$

$$L(l_{ij})=\frac{l_{ij}}{2^{k_2}}+k_2+1 \tag{4b}$$

同时选择调节参数α_1，决策函数为：

$$c_i=\begin{cases}0 & L(\Delta_{ij})+L(l_{ij})<\alpha_1 l_{ij} L(\mathrm{pre}_i)\\ 1 & \text{其他}\end{cases} \tag{5}$$

如果认为保留字采用等长编码，即$L(\mathrm{pre}_i)$是恒定值，令$\alpha_2=\alpha_1 L(\mathrm{pre}_i)$，则式(5)退化为：

$$c_i=\begin{cases}0 & L(\Delta_{ij})+L(l_{ij}))<\alpha_2 l_{ij}\\ 1 & \text{其他}\end{cases} \tag{6}$$

将式(4a)、式(4b)代入式(6)，可以得到：

$$c_i=\begin{cases}0 & \Delta_{ij}/2^{k_1}+k_1+k_2+1<\alpha l_{ij}\\ 1 & \text{其他}\end{cases} \tag{7}$$

式中，$\alpha=\alpha_2-1/2^{k_2}$。在实际工程中通过调节以实现参数优化。

对于满足上述条件的参数，以长度最大且偏移量最小为准则，作为最终选择参数(Δ_i, l_i)；如果参数集合中所有参数满足式(7)，则令偏移量$\Delta_i=0$，输出保留文本数据；否则输出偏移量Δ_i和匹配数据长度l_i。

1.2　预测编码

在文本数据压缩中，偏移量Δ_i、匹配数据长度l_i的分布都比较复杂。偏移量Δ_i中经常会出现许多幅值很大的数据，其分布动态范围很大；为了记录保留文本数据的位置信息，偏移量为0表示该数据为保留文本数据，译码时一旦遇到0，就可以从已经译码出的保留文本数据中取出数据，从而实现数据重建。位置信息的引入，又增加了数据分布的复杂性，也相应增加了编码复杂度。与此同时，l_i的幅值尽管相对比较小，但是数据分布也是非平稳的，无法使用数学分解工具对其进行分解，从而提高压缩效率。

考虑到上述因素，为了提高编码效率，可以进行预测编码。文本数据往往具有一定的规律，每隔一定间隔，有些数据就会重复出现，这类情况可以加以利用。

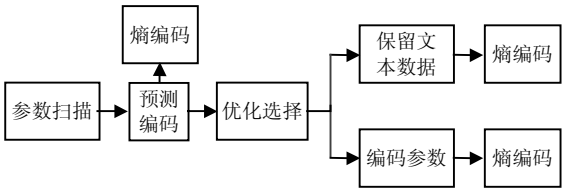

图2　改进算法结构框图

设定一个间隔gap，如果文本数据$x(i)$满足$x(i-\text{gap})=x(i-\text{gap}\times 2)$，则进入预测编码模式，预测函数如式(8)，保留下来的预测参数可以送往熵编码器进行编码。

$$\text{predict}=\begin{cases}1 & x(i)=x(i-p)\\ 0 & \text{其他}\end{cases} \tag{8}$$

2　参数的算术编码实现

MQ算术编码是改进的Q算法，属于自适应的二进制算术编码方法[4-5]，能够有效实现高效数据压缩，该编码器在图像压缩中已得到广泛应用。MQ编码器是通过自适应状态跳转，从而使最终编码输出的字节数量能够尽可能小。为了提高编码效率，MQ算术编码提供了数据分类算法，数据分类采用上下文(CX)表示，编码输入为二进制判决(D)，如图3所示。

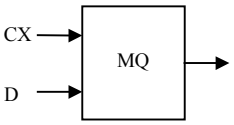

图3　MQ算术编码器

MQ编码器效果与上下文设计密切相关。在文本数据压缩中使用MQ算术编码器，需要研究上下文、判决的形成问题。根据上文可知，文本数据扫描过程中会产生偏移量、匹配数据长度、保留文本数据、预测参数等。这些数据都可以使用算术编码进行熵编码。

预测参数是二进制，不涉及判决产生问题，只需要进行简单的数据分类即可。本文中，上下文为前一个预测值。偏移量、匹配数据长度、保留文本数据的编码都是采用比特平面编码方法，其判决形成与图像压缩中的方法相同，而上下文的形成和编码过程则需要根据各类数据的特点进行研究。偏移量、匹配数据长度的编码分为两个步骤：零编码和细化编码。零编码对当前比特平面编码之前还不重要的系数进行比特平面编码；而细化编码则是对在当前比特平面编码之前已经重要的系数进行编码。

零编码的上下文是根据前后数据的重要性产生，即：

$$\text{CX}_i=\text{sig}(i-1)\times 2+\text{sig}(i+1) \tag{9}$$

式中，CX_i表示当前数据的上下文；$\text{sig}(i-1)$表示前一个数据的重要性；$\text{sig}(i+1)$表示后一个数据的重要性。$\text{sig}(i)=0$表示该数据是不重要的，$\text{sig}(i)=1$表示该数据是重要的。如果数据在当前比特平面编码之前的判决都为0，而在当前比特平面的判决为1，则该数据在当前比特平面编码时由不重要变为重要。

而细化编码的上下文则是采用零编码没有使用的固定上下文，即$\text{CX}_i=4$。保留文本数据的编码只有一个步骤，而上下文则取当前比特编码之前的数据。

3　实验结果与分析

对所提出的文本压缩算法进行仿真和测试。采用4种文本数据对本算法性能进行测试，并与Winzip、WinRar压缩效率进行比较，具体结果如表1所示。其中Test1是Word文档，Test2是纯文字文档，Test3是C语言程序代码(JPEG-LS核心算法)，Test4是

Lena图像。

由表1结果可以看出，对这4类不同类型数据，本文算法压缩性能明显好于Winzip，而在Word文档或者纯文字文档压缩方面与WinRar相当或者略好；而在C语言程序代码压缩方面，本文算法也与WinRar相当或者略低，而对图像压缩进行压缩时，本文压缩效率与WinRar还有一定差距。

表1 算法比较 byte

文件名(字节数)	Winzip	WinRar	本文算法
Test1(333 824)	72 205	63 523	63 408
Test2(37 492)	36 844	36 173	35 166
Test3(13 525)	3 469	3 337	3 426
Test4(262 144)	222 707	168 938	183 726

与WinRar比较结果可以看出，本文算法对图像数据压缩没有取得好的效果，这是因为本文算法没有进一步使用数据之间的相关性进行编码，因此如何进一步利用相关性进行编码值得进一步研究。

为了考察参数α变化对压缩效率影响，选择参数$k_1=10$，$k_2=6$，使用Test1进行测试，具体结果如表2所示。从表2可以看出，当α大于一定值时对压缩效率影响非常有限。从式(7)可以看出，只有当偏移量很大时，参数选择才有意义，其目的是去除那些偏移量很大，而匹配字节长度较小的那些参数。而当α大于一定值时，选择参数的差异并不是太大，所以压缩效率变化较小。

表2 参数α变化对压缩效率影响 byte

α	压缩文件长度	α	压缩文件长度
4.0	65 987	6.50	63 591
5.0	64 635	7	63 574
5.5	63 648	8.0	63 664
6.0	63 632	9	63 606
6.20	63 408		

当α取值较小时，参数选择的变化就体现出来，一些偏移量很大而匹配字节长度有限的参数被选择，从而降低了编码效率，编码输出文件长度增加较大，从而影响总体编码效率。

为了考察Glomb参数选择对压缩效率的影响，取$\alpha=6.2$，$k_2=6$，改变参数k_1，使用Test1进行测试，结果如表3所示。

表3 参数k_1变化对压缩效率影响 byte

k_1	压缩文件长度	k_1	压缩文件长度
6	66 724	10	63 408
7	65 819	11	63 415
8	64 867	12	63 354
9	64 509	13	63 431

从表3可以看出，参数的变化对压缩效率有一定影响，参数小于10时，随着参数减小，压缩效率明显降低；而当参数大于等于10时，由于偏移量大而匹配数据长度小的参数被去除，压缩效率没有明显变化。

为了观察k_2变化对压缩效率影响，取$\alpha=6.2$，$k_1=10$，改变参数k_2，使用Test1进行测试，实验结果如表4所示。

表4 参数k_2变化对压缩效率影响 byte

k_2	压缩文件长度	k_2	压缩文件长度
3	63 612	7	63 617
4	63 625	8	63 671
5	63 618	9	64 867
6	63 408	10	64 557

从表4可以看出，随着k_2的变化，对压缩效率有一定影响，但是不是十分明显。从式(7)可看出，由于大偏移量受到k_1约束，k_2取值只是辅助参数选择的细节，且受到α变化的制约，因此对总体效率的影响不是很大，其取值大小与匹配字节长度小的参数选择产生一定影响。当其取值太大，会增加小匹配字节长度选取的门槛，所以对压缩效率影响较大；而取值较小时，其影响反而不是太大。

综合k_1、k_2变化对压缩效率影响，结果与式(7)说描述的含义是相符的，即：

1) k_1主要是限制偏移量大而匹配字节长度较小的参数，以提高编码效率；当其取值较小时，偏移量大，匹配字节长度较小的参数被选择，从而影响编码效率；而取值较大时，只有极少的参数被限制，对压缩效率的影响反而较小。因为匹配字节长度较大的参数，k_1变化对其没有约束。

2) k_2对偏移量变化没有限制作用，主要辅助设置匹配字节长度门槛。取值越大，更多长度较小的参数被去除，效率降低；而较小的取值，反而对结果影响不大。

4 结 论

本文提出了一种基于算术编码的文本数据压缩算法，与Winzip、WinRar算法相比，在对纯文本数据、Word文档数据、C语言程序代码进行压缩时，本文算法优于WinZip，与WinRar算法相当或略好，但在图像压缩方面的性能与WinRar相比略有不足。当然，本文算法还存在以下不足：一方面，简单使用LZ77扫描算法，从而导致偏移量数据较大，不利于后续数据压缩；另一方面，没有对相关性数据进

行进一步处理，对诸如图像数据这类关联性很强的数据压缩效率不足。针对上述不足，今后将考虑更好的扫描算法，以提高压缩效率；对数据的相关性进行检测，对相关性强的数据进行数据分解，进一步提高编码效率。

参 考 文 献

[1] TAUBMAN D. High Performance scalable image compression with EBCOT[J]. IEEE Trans on Image Processing, 2000, 9(7): 1158-1170.

[2] ISO/IEC. Image coding specification[EB/OL]. [2015-07-21]. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51609.

[3] DENG Jia-xian, DENG Hai-tao. An image joint compression-encryption algorithm based on adaptive arithmetic coding[J]. Chin Phys B, 2013, 22(9): 094202-1-094202-6.

[4] WU J Z, WANG Y J, DING L P, et al. Improving performance of network covert timing channel through Huffman coding[J]. Mathematical and Computer Modelling, 2012, 55(1-2): 69-79.

[5] 邓家先, 任玉莉. 基于改进零树编码的图像联合压缩加密算法[J]. 光子学报, 2013, 42(1): 121-126.
DENG Jia-xian, REN Yu-li. Image joint compression-encryption algorithm based on improved zero-tree coding[J]. Acta Photonica Sinica, 2013, 42(1): 121-126.

[6] 谢耀华, 汤晓安, 孙茂印, 等. 基于分类重排LZW的图像无损压缩算法[J]. 中国图象图形学报, 2010, 15(2): 236-241.
XIE Yao-hua, TANG Xiao-an, SUN Mao-yin, et al. A lossless image compression algorithm based on classification,re-ordering and LZW[J]. Journal of Image and Graphics, 2010, 15(2): 236-241.

[7] 王忠效. 汉语文本压缩研究及其应用[J]. 中文信息学报, 1997, 11(3): 57-64.
WANG Zhong-xiao. Research and application of chinese text compression[J]. Journal of Chinese Information Processing, 1997, 11(3): 57-64.

[8] 特日跟, 李雄飞, 李军. 基于整数数据的文档压缩编码方案[J]. 吉林大学学报, 2016, 46(1): 228-234.
TE Ri-gen, LI Xiong-fei, LI Jun. Document compression coding scheme based on integer data[J]. Journal of Jilin University, 2016, 46(1): 228-234.

[9] ZIV J, LEMPEL A. Compression of individual sequences via variable-rate coding[J]. IEEE Transactions on Information Theory, 1978, 24(5): 530-536.

[10] ZIV J, LEMPEL A. A universal algorithm for sequential data compression[J]. IEEE Transactions on Information Theory, 1977, 23(3): 337-343.

[11] 常为领, 方兴滨, 云晓春, 等. 一种支持ANSI编码的中文文本压缩算法[J]. 中文信息学报, 2010, 24(5): 96-105.
CHANG Wei-ling, FANG Xin-bin, YUN Xiao-chun, et al. A chinese text compression algorithm for ANSI coding[J]. Journal of Chinese Information Processing, 2010, 24(5): 96-105.

编 辑 税 红