doi:10.11918/j.issn.0367-6234.2016.05.006

SLWE 概率估计方法在区间编码中的应用研究

陈 浩, 宿腾野, 滑 艺, 刘 东

(哈尔滨工业大学 电子与信息工程学院,150001 哈尔滨)

摘 要: SLWE 概率估计方法具有较强的适应非平稳数据能力,为拓展其在熵编码中的应用,更有效地编码非平稳数据,设计在区间编码上的应用方案. 首先针对概率估计模型替换时 SLWE 估计出的概率如何映射到区间的问题,不进行概率更新的计算,而是基于 SLWE 思想直接更新各字符所占区间大小,再根据区间编码中总区间上下界计算方法调整总区间. 既结合 SLWE 应对非平稳数据的优势,又避免概率运算. 同时,针对更新各字符所占的整型数据区间后字符所占区间大小可能小于 1 导致编码字符丢失的问题,采用设定每种字符最小区间作为阈值的控制方法. 对非平稳数据编码的实验结果表明,基于 SLWE 的区间编码比基于加窗法等传统概率估计方法的压缩率要高出 1%~5%.

关键词: 熵编码; 非平稳数据; 随机学习弱估计; 概率估计; 区间编码

中图分类号: TN911.73 **文献标志码**: A **文章编号**: 0367-6234(2016)05-0043-08

The application for probability estimation of SLWE on range coder

CHEN Hao, SU Tengye, HUA Yi, LIU Dong

(School of Electronics and Information Engineering, Harbin Institute of Technology, 150001 Harbin, China)

Abstract: The probability estimate method of SLWE can adapt to the non-stationary data. In order to expand its application in entropy coding and code non-stationary data more effectively, an application scheme of SLWE in the range coding is designed. Firstly, to tackle the problem how to map the estimated probability by SLWE to the coding interval, instead of calculating the update probability, we propose to update the range of every character directly based on the idea of SLWE and then adjust the total range according to the computing method of the upper and lower range bounds for range encoding. It not noly combines the advantage of SLWE to cope with the non-stationary data, but also avoids the probability calculation. In addition, the coding range after the update for every character may be less than 1, which causes the loss of the character. To solve this problem, we present a control method to set the minimum range of each character. Experimental results for non-stationary data coding show that the SLWE-based range coder achieves 1% ~ 5% higher than that using traditional probability estimation (e.g. windowing method) in terms of compression ratio.

Keywords: entropy coding; non-stationary data; stochastic learning weak estimator; probability estimate; range coder

随着编码技术的发展,各种熵编码方法应运而生,如香农编码、霍夫曼编码、算术编码、区间编码等. 熵编码的编码性能主要依赖概率估计模型与信源的实际特性相符的程度. 根据待编码数据的统计特性不同,熵编码的概率估计方法也有相应的不同. 当待编码数据特性平稳时,经典的概率估计方法是最大似然法和贝叶斯参数估计法[1];当待编码数据特性非平稳时,主要的概率估计方法为基于“遗忘因子”的方法和加窗法.

贝叶斯参数估计法理论依据较强,先假定待估计参数具有某种先验分布,然后根据不断观测到的值去加强其中某种可能分布的可能性. 由于对于平稳信源,参数的真实分布 θ 可看成不随时间改变的,因而随着观测数据的增加,贝叶斯估计 $\hat{\theta}$ 会以概率 1 收敛于真实分布[2].

文献[3]提出基于“遗忘因子”的想法,当已编码字符总数达到一个事先设定的阈值时,将各种类字符的累积频率乘上一个大于 0 小于 1 的实数,称为“尺度调节(rescaling)”,所乘的实数叫做“遗忘因子(forgetting Factor)”. 而加窗法是通过分析一段特定缓存中的内容去估计信源当前待编码符号的概率分布,这段缓存中存有 W 个之前已编码的字符(即缓存

收稿日期: 2015-11-30.
基金项目: 国家 863 项目 2012AA12A405;国家自然科学基金 61102159.
作者简介: 陈 浩(1978-),男,副教授,博士.
通信作者: 陈 浩,hit_hao@hit.edu.cn.

长度是 W,也可称为窗长),每编码完一个新字符后,缓存中的内容会进行一次移位,新字符存入到缓存中,最早进入缓存的字符被删除. 文献[4]从理论上分析了加窗法在非平稳环境下的编码性能,并讨论了窗长与编码冗余的关系.

近年来,弱估计算法为非平稳数据的概率估计供了新的思路,其中有代表性的是随机学习弱估计 SLWE (stochastic learning weak estimator),它是一种基于学习自动机 LA(learning automata)理论的参数估计方法. LA 属于机器学习中的增强学习,文献[7]提出,最初用于模拟生物在出生时具备很少的先天知识,不断与随机环境交互下的学习过程. Oommen 及其合作者提出了在非平稳环境下估计二项式和多项式分布的方法[5-7]. SLWE 应用在概率估计中,可节省加窗法的窗口中数据缓存的存储空间,同时对于非平稳数据有了更强的适应性,并对多种类型的数据编码均可使用[8].

文献[9]曾尝试将 SLWE 概率估计方法应用 FANO 编码这种并不常用的熵编码中,并给出相应的实现方法. 为拓展 SLWE 原理在熵编码中的应用,更有效的实现对非平稳数据的压缩编码,将其在编码性能更优良的区间编码上进行实现. 区间编码是由 Martin 提出[10],与算术编码本质上类似,尽管在精度上稍微逊色,但比算术编码在频度统计中速度快很多[11]. 将 SLWE 概率估计方法应用到区间编码时,把原本的由概率更新改变各字符所占区间大小,从而通过总区间上下界的变化进行区间更新的方式,改进为将 SLWE 算法直接应用到对各字符所占区间大小的改变上的方法,不先进行概率更新的计算,而是直接用 SLWE 算法中的思想对各字符所占区间大小进行更新,再根据区间编码中总区间上下界计算方法来调整总区间大小,既结合了 SLWE 对非平稳数据概率估计的优势,又避免了概率的具体运算. 从而提高编码过程对非平稳数据压缩编码的运算精度以及编码效率.

1 基于随机学习弱估计的概率估计方法

1.1 二项分布的 SLWE 概率估计过程

二项分布又称伯努利分布,其特征由两个参数决定:试验的次数 n 和每次试验成功的概率 p, 如果将二值信源每次产生二值符号的过程看做一次伯努利试验,那么二项分布的参数估计问题就可看作通过编码二值序列对信源特性进行估计的问题. 假设 X 是一个二项分布的随机变量,取值为“1”或“2”,并且 X 遵循概率分布 $\boldsymbol{Q}=[q_1,q_2]^{\mathrm{T}}$, 即

$$X=\begin{cases}''1'', & \text{概率为 } q_1;\\ ''2'', & \text{概率为 } q_2.\end{cases}$$

其中 $q_1+q_2=1$.

设 $X(n)$ 是随机变量 X 在时刻 n 的具体值,根据 SLWE 理论,对于 $X(n)$ 的概率估计是一个动态的过程,设对 $X(n)$ 概率估计为 $\boldsymbol{P}(n)=[p_1(n),p_2(n)]^{\mathrm{T}}$, 如果有参数 λ 称为学习因子 $(0<\lambda<1)$, 则其概率更新方式如下:

$$\begin{aligned}p_1(n+1)&\leftarrow 1-\lambda p_2(n),\ \text{当 } x(n)=1 \text{ 时};\\ &\leftarrow \lambda p_1(n),\ \text{当 } x(n)=2 \text{ 时}.\end{aligned}\tag{1}$$

1.2 多项分布的 SLWE 概率估计过程

与二项分布一样,多项分布的特征也由两个参数决定,即试验的次数 n 和决定试验结果的一个矢量,该矢量描述了一个事先给定的事件集合中某一特定事件发生的概率. 同样的,可将多项分布的参数估计问题看成多值信源的特性估计问题. 假设 X 是一个服从多项分布的随机变量,其真实概率分布为 $\boldsymbol{Q}=[q_1,\cdots,q_r]^{\mathrm{T}}$, 其中 $X=''i''$ 的概率为 q_i, $\sum_{i=1}^{r}q_i=1$.

同样,令 $x(n)$ 为变量 X 在 n 时刻的实际观测值,假设 n 时刻对 $\boldsymbol{Q}$ 的概率估计矢量为 $\boldsymbol{P}(n)=[p_1(n),\cdots,p_r(n)]^{\mathrm{T}}$, 其中 $p_i(n)$ 表示在 n 时刻对 q_i 的估计,则根据 SLWE 理论, $p_i(n)$ 更新方式如下

$$\begin{aligned}p_i(n+1)&\leftarrow 1-\lambda\sum_{j\neq i}p_j(n),\ \text{当 } x(n)=i;\\ &\leftarrow \lambda p_i(n),\ \text{当 } x(n)\neq i.\end{aligned}\tag{2}$$

式中 i,j 为多项分布 X 中的任意变量,且 $j\neq i$, 参数 λ 称为学习因子, $0<\lambda<1$.

将运用多项分布情况下的 SLWE 概率估计方法,在信源信息为非平稳数据的情况下,对区间编码进行改进,以提高其编码性能.

2 区间编码方法

一般的区间编码以各种符号的统计频率来估计概率,其编码端和解码端工作原理如下.

编码端:输入是信源符号集合 $\boldsymbol{S}$ 和待编码序列 $\boldsymbol{Z}$, 输出是编码产生的码流 $\boldsymbol{Y}$. 处理步骤如下:

步骤 1 统计信源基本信息. 以字节为单位读取待编码数据,统计待编码数据字节长度 B_{BSIZE}, 字符种类数 N.

步骤 2 初始化概率估计模型. 设定在 n 时刻总区间大小为 $R(n)$, 信源符号集合 $\boldsymbol{S}$ 中各符号的频率为 $f_i(n)$, 其中 i 表示 $\boldsymbol{S}$ 中第 i 种字符($0\leqslant i\leqslant N-1$), 所有编码数据的总频率为 $F_{\mathrm{TF}}(n)$ (Total Frequency),

并且 $F_{\mathrm{TF}}(n)=\sum_{i=0}^{N-1}f_i(n)$. 根据概率估计模型,初始总区间大小 $R(0)$, 令区间阈值为 R_0, 各符号的初始频率 $f_i(0)=1$, 总频率为

$$F_{\mathrm{TF}}(0)=\sum_{i=0}^{N-1}f_i(0)=N.$$

步骤 3　根据序列 $\boldsymbol{Z}$ 中读入的符号，假设在 n 时刻待编码字符为第 i 种字符，设定 $c_{\mathrm{cumf}}(n)$ 为前 0 到 $i-1$ 种字符的累计频率，计算 $c_{\mathrm{cumf}}(n)=\sum_{j=0}^{i-1}f_j(n)$.

在 $n=0$ 时刻，初始化累积频率为 $c_{\mathrm{cumf}}(0)=0$.

步骤 4　编码和区间规格化. 首先定义在 n 时刻待编码字符 i 的概率为 $p_i(n)=f_i(n)/F_{\mathrm{TF}}(n)$，编码顺序在字符 i 之前的字符 0 到 $i-1$ 的累计概率为 $c_{\mathrm{cump}}(n)=c_{\mathrm{cumf}}(n)/F_{\mathrm{TF}}(n)$，则当前总区间按照各个符号的概率进行更新，区间更新方法为

$$\begin{aligned}L(n+1)&\leftarrow L(n)+R(n)\times c_{\mathrm{cump}}(n),\\ R(n+1)&\leftarrow R(n)\times p_i(n).\end{aligned}\tag{3}$$

式中：$L(n)$ 为总区间下界，$R(n)$ 为区间范围，$L(n+1)$ 为更新后的总区间下界，$R(n+1)$ 为更新后的区间范围，当更新后的区间范围小于指定阈值 R_0 时，或者以字节为单位比较新区间的上下界并且上下界的高位字节相等时，移出高位的字节作为输出码流，同时对区间进行规格化处理.

步骤 5　频率表更新，根据 n 时刻已编码字符 i 更新频率表. 在字符 i 频率 $f_i(n)$ 的基础上加一个给定频率增量 M，得 $n+1$ 时刻的频率 $f_i(n+1)=f_i(n)+M$，此时其他 $N-1$ 种字符的频率不变，则总频率 $F_{\mathrm{TF}}(n)=f_i(n+1)+\sum_{j=0,j\neq i}^{N-1}f_j(n+1)$，并且，

同时对累积频率进行更新：

$$c_{\mathrm{cumf}}(n+1)=\sum_{j=0}^{i-1}f_j(n+1).$$

按照步骤 3~5 所述依次对所有 8bit 字符进行编码. 为能够准确的进行解码操作，需要对最后一个符号的映射区间范围完全移出，同时保留该区间内某一个二进制数 V，形成码流 $\boldsymbol{Y}$，并将其传递给解码器.

解码端：输入是编码产生的码流 $\boldsymbol{Y}$，信源符号集合 $\boldsymbol{S}$，输出是解码序列 $\boldsymbol{Z}$. 处理步骤如下：

步骤 1　读取信源基本信息以及待解码文件，得到原数据长度 B_{BSIZE}，符号种类数 N，以及编码时保留的最后一个符号区间内的二进制数 V.

步骤 2　初始化概率模型，方法与编码端相同.

步骤 3　计算累积频率估计值 $c_{\mathrm{cumfe}}(n)$，由公式

$$\begin{aligned}&L(0)+R(0)\times c_{\mathrm{cumf}}(0)/T_{\mathrm{TF}}(0)\leqslant V,\\ &V<L(0)+R(0)\times(c_{\mathrm{cumf}}(0)+f_i(0))/F_{\mathrm{TF}}(0).\end{aligned}$$

从而有

$$\begin{aligned}&c_{\mathrm{cumf}}(0)\leqslant(V-L(0))/(R(0)/T_{\mathrm{TF}}(0)),\\ &(V-L(0))/(R(0)/T_{\mathrm{TF}}(0))<(c_{\mathrm{cumf}}(0)+f_i(0)).\end{aligned}$$

令 $c_{\mathrm{cumfe}}(0)=(V-L(0))/(R(0)/\mathrm{T}_{\mathrm{TF}}(0))$，得到累积概率的初始估计值，并且有估计不等式：

$$c_{\mathrm{cumf}}(0)\leqslant c_{\mathrm{cumfe}}(0)<(c_{\mathrm{cumf}}(0)+f_i(0)).$$

每编码一个字符，保持在当前 n 时刻的估计式 $c_{\mathrm{cumf}}(n)\leqslant c_{\mathrm{cumfe}}(n)<(c_{\mathrm{cumf}}(n)+f_i(n))$，从而根据此式估计 $c_{\mathrm{cumfe}}(n)$ 值.

步骤 4　解码器解码与编码器编码保持一致，根据累计频率估计值，输出所在区间的字符，并更新当前区间范围，即

$$\begin{aligned}L(n+1)&\leftarrow L(n)+R(n)\times c_{\mathrm{cumfe}}(n)/F_{\mathrm{TF}}(n),\\ R(n+1)&\leftarrow R(n)\times f_i(n)/F_{\mathrm{TF}}(n).\end{aligned}$$

并判断是否需要区间规格化.

步骤 5　根据步骤 4 中解码出的字符更新频率表，并根据当前区间范围以及新的频率表重新为各字符分配所属区间.

按照上述步骤 3~5 所述依次对所有 8 bit 字符进行编码，直到解码结束.

本文重点研究如何用 SLWE 算法实现编码端和解码端步骤 5 中概率更新的过程，并对实现过程中所遇到的相关问题给出解决和改进方法.

3　基于 SLWE 的区间编码设计方法

首先通过对原区间编码过程的分析.

1) SLWE 应用到区间编码中，核心是概率估计模型替换的过程，但同时会面临新的概率模型所估计出的概率如何映射到区间的问题.

在原区间编码中，各种字符在 n 时刻的估计概率是 $p_i(n)=f_i(n)/F_{\mathrm{TF}}(n)$，可知各种字符的概率更新是由频率的更新得到. 并由式(3)，每编码一个字符，按照相应各字符概率的变化来计算新的区间下界，从而将概率映射到区间，进行区间更新，最后根据区间的变化得到输出码流.

而 SLWE 应用到区间编码时，考虑区间编码的输出码流是由区间的不断更新变化得到的. 因此，可将式(3)中 $R(n)\times c_{\mathrm{cump}}(n)$ 设定为 n 时刻待编码字符 i 之前的字符 $0\sim i-1$ 的累积区间，即：$C_{\mathrm{CUMR}}(n)=R(n)\times c_{\mathrm{cump}}(n)$，同时将 $R(n)\times p_i(n)$ 设定为 n 时刻待编码字符 i 所占区间的大小，即

$$r_i(n)=R(n)\times p_i(n),$$

从而

$$C_{\mathrm{CUMR}}(n)=\sum_{j=0}^{i-1}r_j(n).$$

式中 j 为字符 0 到 $i-1$. 由此，在编码实现过程中，将各字符的概率 $p_i(n)$ 为成各字符所占区间 $r_i(n)$，将各字符的总概率 1 等价于总区间范围 $R(n)$，即 $R(n)=\sum_{i=0}^{N-1}r_i(n)$. 并结合 SLWE 概率估计方法，将各

种字符概率的更新转化为对各种字符所占区间的更新,即 n 时刻待编码字符为 i, 则非字符 i 的各字符所占区间更新为 $r_j(n+1)=\lambda\times r_j(n)$, 其中, $j=0,\cdots,N-1, j\neq i$, 字符 i 的区间更新为

$$r_i(n+1)=R(n+1)-\sum_{j=1,j\neq i}^{N} r_j(n+1).$$

同时,将 SLWE 应用到区间编码时,由于各种字符的估计概率是用区间代替的,在每编码一个字符时,总区间都会发生变化,相应的各种字符所占区间也需要进行同比例的缩放,从而才能保证当前各种字符的估计概率不变. 因此,在每编码完一个字符时,记录下编码前后总区间的变化情况,编码前总区间为 $R(n)$, 编码后总区间大小为 $R(n+1)$, 每次编码之后记录下 $m=R(n+1)/R(n)$, 则非待编码字符 i 所占区间更新为 $r_j(n+1)=\lambda\times[m\times r_j(n)]$, 其中, $j=0,\cdots,N-1, j\neq i$.

2)由于设定了各字符所占区间,而计算机实现时,区间是用整型数据表示,在对个字符所占区间进行更新时,更新后的各种字符所占区间大小可能出现小于 1 的情况,从而可能导致编码字符丢失的问题.

对此,对每种字符设定了最小区间 $r(n)_{\min}=R(n)\times p_{\min}\geqslant 1$ 作为阈值,其中 $R(n)$ 为当前总区间, $P_{\min}$ 为经验参数,从而保证每编码一个字符,总区间大小更新后,都会给出一个具体的最小区间,使得各字符所占区间的变化是有下限的,不能无限减小. 当某一时刻计算出的某种字符所占的区间大小小于 $r(n)_{\min}$ 时,以 $r(n)_{\min}$ 作为该种字符的新区间.

综上所述,结合区间编码的整体框架,设计如图 1 扭不的编码端其工作流程如下:

步骤 1 统计信源基本信息. 以字节为单位读取待编码数据,统计待编码数据长度 B_{BSIZE}, 对不同种类符号编号的最大值为 m_0, 符号编号的最小值为 m_1, 符号种类数 $N=m_0-m_1+1$, 各符号种类分别用索引 $0,\cdots,N-1$ 表示;

步骤 2 初始化. 对 32 位系统,可初始化区间上界为 $H=$ 0xffffffff, 区间下界为 $L=$ 0x00000000, 则原始区间大小为 $R(0)=$ 0xffffffff, 初始化各个符号占据的初始区间长度为 $r_i(0)=R(0)/N, i=0,\cdots,N-1$, 初始化区间规格化时的临界阈值 $R_0=$ 0x00001000.

步骤 3 根据读入字符计算累计区间长度. 初始化累积区间长度 $C_{\mathrm{CUMR}}(0)=0$, 假设当前待编码字符索引为 i, 计算 $C_{\mathrm{CUMR}}(n)=\sum_{j=0}^{i-1} r_j(n)$.

步骤 4 编码和区间规格化. 记录当前区间长度 $R(n)$, 然后根据当前待编码字符对区间进行更新,更新方式为:

$$L(n+1)\leftarrow L(n)+C_{\mathrm{CUMR}}(n), R(n+1)\leftarrow r_i(n).$$

当新区间长度 $R(n+1)\leqslant R_0$ 或以字节为单位比较新区间的上界和下界,上下界的高位字节相等时,移出高位的字节作为输出码流,同时对区间进行规格化处理,具体做法为将 $R(n+1)$ 和 $L(n+1)$ 分别左移 8 位,即 $R(n+1)\leftarrow R(n+1)\ll 8, L(n+1)\leftarrow L(n+1)\ll 8$. 最后计算编码后区间和编码前区间的比值 $m=R(n+1)/R(n)$;

步骤 5 更新概率估计表. 根据 SLWE 算法,对各字符所在区间大小进行更新,具体过程与编码端相同.

Step 1 根据信源基本信息,给定参数 $p_{\min}$, 并根据更新后的区间 $R(n+1)$ 计算各字符所占区间的下限,即最小区间 $r(n+1)_{\min}=R(n+1)\times p_{\min}$, 初始化学习因子 λ,设 $n+1$ 刻除字符 i 外的其他字符所占区间为 $S_{\mathrm{SUMR}}(0)=0$.

Step 2 依次计算序号为 $j=0,\cdots,N-1, j\neq i$ 的字符所占区间大小.

$r_j(n+1)=\max\{\lambda\times[m\times r_j(n)], r(n+1)_{\min}\}$. 其中 $j=0,\cdots,N-1, j\neq i$.

Step 3 计算 $S_{\mathrm{SUMR}}(n+1)=\sum_{j=1,j\neq i}^{N} r_j(n+1)$.

Step 4 计算 $r_i(n+1)=R(n+1)-S_{\mathrm{SUMR}}(n+1)$.

步骤 6 编码收尾阶段. 按照步骤 3~5,对所有待编码数据进行编码. 编码结束时,移出映射区间内所有的位. 分别将码流和待编码数据的相关信息以文件形式保存.

同时设计了如图 2 所示的解码端框架,工作流程如下:

步骤 1 读取信源基本信息文件,得到原数据长度 B_{BSIZE}, 对不同种类符号最大值 m_0, 符号编码最小值 m_1, 计算符号种类数 $N=m_0-m_1+1$.

步骤 2 与编码过程中的初始化过程一样初始化区间上界 $H=$ 0xffffffff、区间下界 $L=$ 0x00000000, 初始化各个符号占据的初始区间长度 $r_i(0)=R(0)/N, i=0,\cdots,N-1$, 初始化区间规格化时的临界阈值 $R_0=$ 0x00001000. 并以字节为单位读取码流文件,设定初始标识符 $T=$ 0x00000000, 通过计算机运算,将每 4 个字节码流信息存储于 T.

步骤 3 根据 T 和当前区间下界 L 以及各符号的区间长度进行解码,解码过程中得到当前符号的索引 i. 具体做法为寻找满足下述关系的 i:

$$\sum_{j=0}^{i-1} r_j(n)\leqslant T-L<\sum_{j=0}^{i} r_j(n).$$

步骤 4 首先记录当前区间长度 $R(n)$,然后根据 i 对区间进行更新,更新方式与编码过程中的更新方式一致,即

$L(n+1) \leftarrow L(n) + C_{\mathrm{CUMR}}(n), R(n+1) \leftarrow r_i(n)$.

更新后的区间长度为第 i 种字符所在的区间长度 $r_i(n+1)$，即 $R(n+1) \leftarrow r_i(n+1)$. 当新区间长度 $R(n+1)$ 小于指定阈值 R_0 或以字节为单位比较新区间的上界和下界，上下界的高位字节相等时，对区间进行规格化处理，即：将 $R(n+1)$ 和 $L(n+1)$ 分别左移 8 位，$R(n+1) \leftarrow R(n+1) \ll 8$，$L(n+1) \leftarrow L(n+1) \ll 8$，同时以字节为单位读取码流文件并与 T 进行或运算，之后将 T 左移 8 位. 计算编码后区间和编码前区间的比值 $m = R(n+1)/R(n)$.

步骤 5　根据 SLWE 算法对各字符所在区间大小进行更新，具体过程如下：

Step 1　根据信源基本信息，给定参数 $P_{\min}$，并根据更新后的区间 $R(n+1)$ 计算各字符所在区间的下限 $R(n+1)_{\min} = R(n+1) \cdot P_{\min}$，初始化学习因子 λ，定义变量 $S_{\mathrm{SUMR}}(0) = 0$.

Step 2　依次计算序号为，$j = 0, \cdots, N-1$，$j \neq i$ 的字符所占区间大小

$$r_j(n+1) = \max\{\lambda \times [m \times r_j(n)], r(n+1)_{\min}\}.$$

Step 3　计算 $S_{\mathrm{SUMR}}(n+1) = \sum_{j=1, j\neq i}^{N} r_j(n+1)$.

Step 4　计算 $r_i(n+1) = R(n+1) - S_{\mathrm{SUMR}}(n+1)$. 输出此次解码后得到的字符，形成解码有序.

步骤 6　当解出的数据总长度小于原数据长度 B_{BSIZE} 时，重复进行步骤 3、4、5.

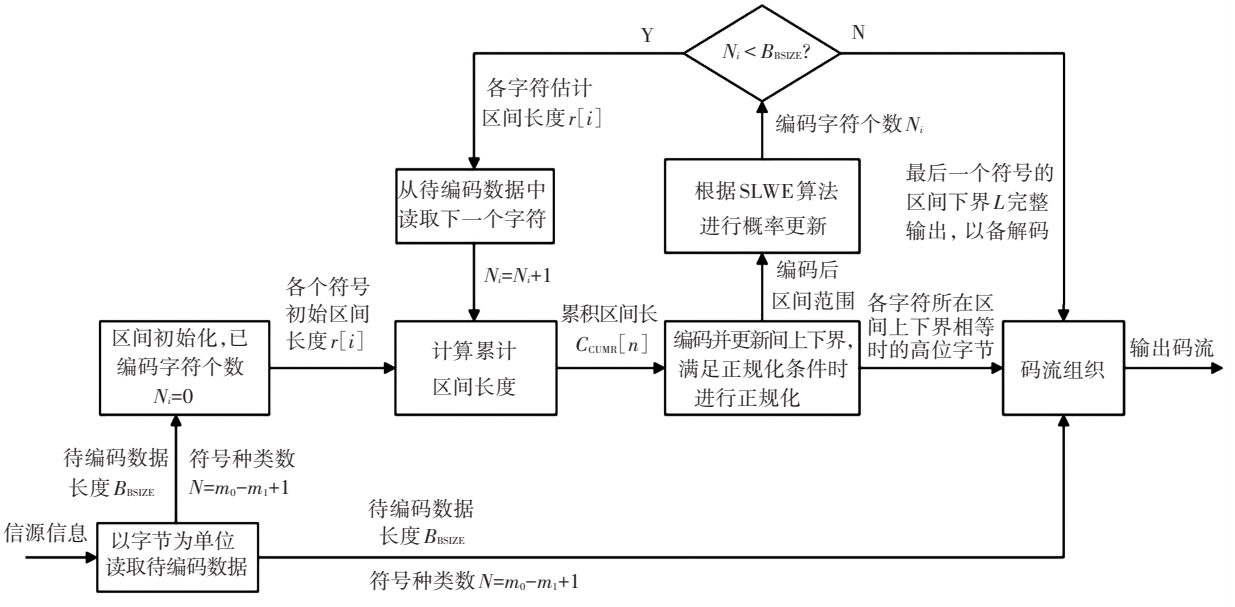


图 1　编码流程图

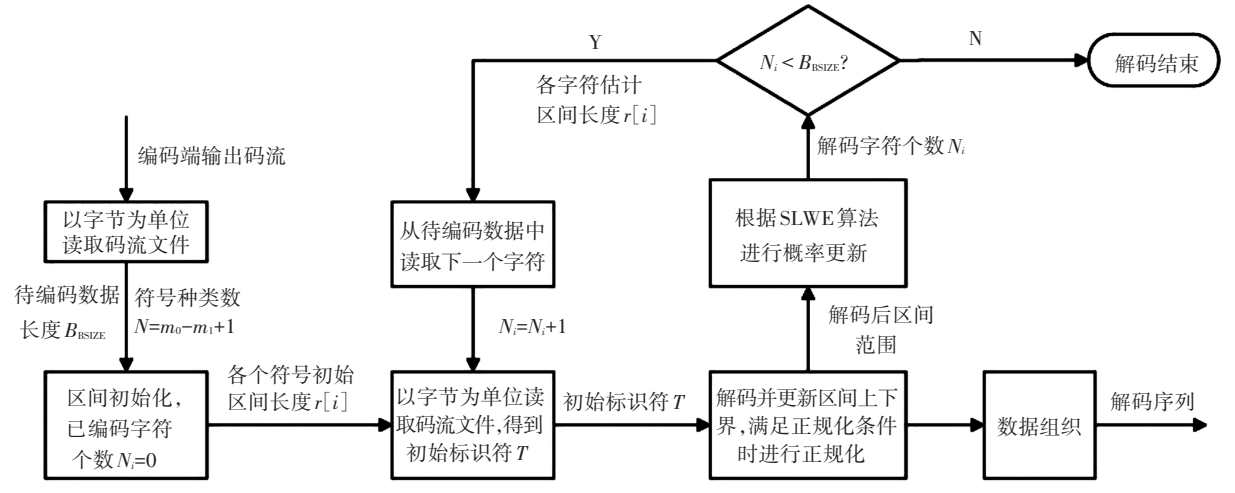


图 2　解码流程图

4　实验结果及讨论

实验中，对仿真数据在不同概率估计方法下的熵值进行分析和比较，验证基于 SLWE 的概率估计方法对信源熵值估计的有效性. 并在试验中考察学习因子 λ 在何种范围下，概率估计的效果较好. 对真实数据的编码实验

中,应用已有的基于不同概率估计方法的区间编码原理,分别对不同的数据类型进行编码实验,并对相应的压缩率进行比较和有针对性的分析,从而验证基于SLWE的区间编码方法,在编码非平稳数据时的优越性.

4.1 面向仿真数据的概率估计实验

本部分实验主要根据各概率估计算法对仿真数据的估计熵与其实际熵的偏离程度评价各算法的适用情况. 表1是对不同特性仿真数据的真实熵值,静态模型以及add-half准则的估计熵,和用SLWE方法的估计熵之间的比较情况. 其中,对于静态模型,在编码过程中概率模型不随时刻的改变而改变. 静态概率模型的建立往往需要信源的先验知识,若没有相关的先验知识,可通过遍历整个待编码字符串,统计其分布特征来建立概率模型.

"add-half"法则是由Krichevskii提出的对贝叶斯参数估计法的一种实现方法[12],从通用编码(Universal coding)角度来说,在贝叶斯参数估计法中,"add-half"法则是最优的.

仿真数据为二值伪随机序列,L表示某种给定概率特性的信源持续作用的时间,N为不同种类信源的个数,L越大说明某种信源持续作用的时间越长,表示其统计特性越平稳,N越大说明该仿真数据特性变化的越频繁,表示其统计特性越不平稳. 另外,为了更贴近实际数据的特性变化,上述过程中的L可设置为可变的,这种情况下用L_{max}为产生的数据中可能的最大平稳段数据长度.

实验中首先产生指定特性的仿真数据,然后分别计算仿真数据的真实熵以及用各种概率估计算法得到的概率模型下的估计熵(λ取0.9、0.95、0.99),为了消除仿真数据的偶然性因素影响,每组实验进行100次,取均值.

实验结果见表1,当数据特性非平稳时($N=3$,$N=5$的情况下),SLWE算法的估计熵更接近真实熵,并且学习因子λ不同,逼近程度也不一样.

表1 不同特性二值仿真数据的真实熵与各概率估计方法下的估计熵

信源特性	真实熵(比特/符号)	估计熵(比特/符号)				
		静态模型		自适应模型		
		遍历方式	add-half	SLWE算法		
				λ=0.9	λ=0.95	λ=0.99
$L=200$, $N=1$	0.750 5	0.742 5	0.750 1	0.799 9	0.782 1	0.823 4
$L=200$, $N=3$	0.698 5	0.908 2	0.931 3	0.761 1	0.748 0	0.829 7
$L=200$, $N=5$	0.703 0	0.941 6	0.957 3	0.766 2	0.753 2	0.834 2
$L=200$, $N=1$	0.711 0	0.703 3	0.754 2	0.764 6	0.742 3	0.771 7
$L=200$, $N=3$	0.735 3	0.917 5	0.934 3	0.791 3	0.772 1	0.819 0
$L=200$, $N=5$	0.723 6	0.949 4	0.956 3	0.784 6	0.764 8	0.818 6
$L=200$, $N=1$	0.709 2	0.707 5	0.777 0	0.762 4	0.736 7	0.747 6
$L=200$, $N=3$	0.727 1	0.915 0	0.905 1	0.781 3	0.758 4	0.781 3
$L=200$, $N=5$	0.723 7	0.947 2	0.949 1	0.778 4	0.756 0	0.781 3
$L_{max}=500$, $N=1$	0.709 7	0.697 5	0.753 6	0.768 0	0.753 9	0.798 9
$L_{max}=500$, $N=3$	0.738 3	0.869 1	0.889 5	0.796 7	0.778 5	0.816 2
$L_{max}=500$, $N=5$	0.714 8	0.933 5	0.920 2	0.774 2	0.758 3	0.814 0

图3给出信源特性不同时,SLWE方法中不同学习因子λ下概率估计算法的概率估计曲线与反映信源真实特性的概率曲线之间的关系. 从图中可以看出,λ的大小主要影响信源特性发生突变时概率估计曲线相对于信源真实特性变化曲线的滞后情况,λ较小时,概率估计曲线滞后较小,而λ较大时(接近1),概率估计曲线滞后较大. 但λ较小时,概率估计曲线的波动较大,说明此时概率模型易受已编码符号中短暂的异常分布影响,而λ较大时,这种影响就很弱,表现为曲线波动较小.

另外,对于表1中给出的几种情况,均为$\lambda=0.95$时,估计效果最佳. 图4是对于$L_{max}=10\,000$, $N=10$和$L_{max}=10\,000$, $N=5$两种特性的二值仿真数据分

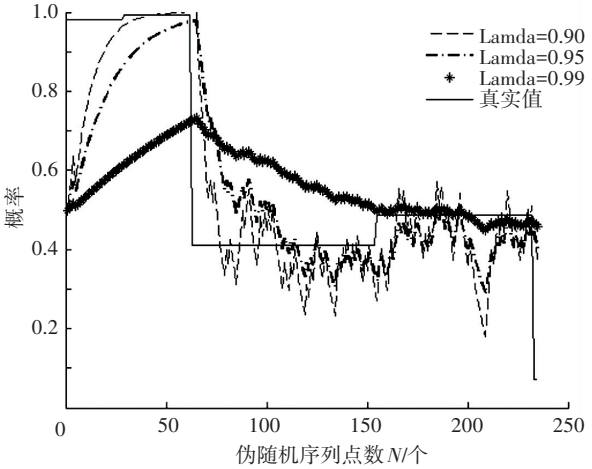
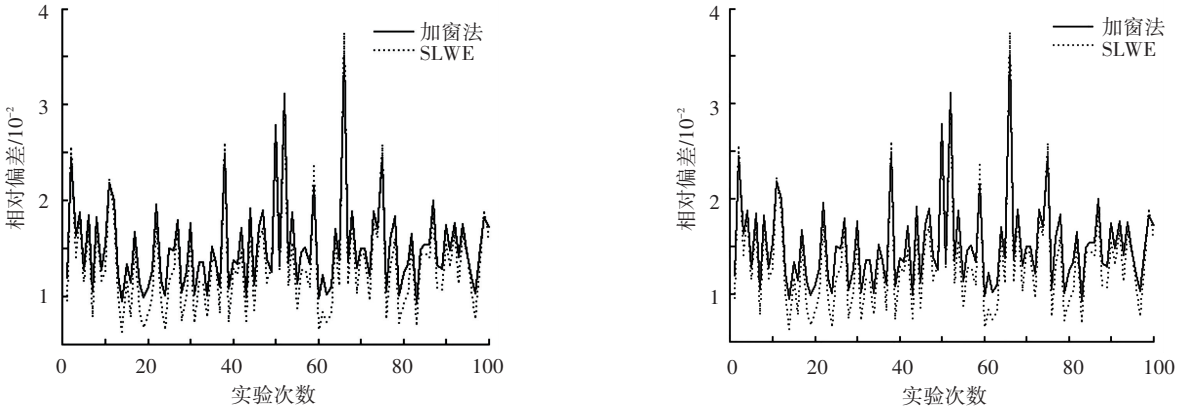


图3 不同学习因子λ下SLWE概率跟踪曲线



(a) L = 10 000, N = 10　　(b) L = 10 000, N = 5

图 4　仿真数据两种方法选取最优参数时相对偏差曲线

别在上述两种方法在给定参数选择范围下均选取最优参数时估计熵与真实熵的相对偏差曲线比较情况. 从图中可看出,对于指定特性的仿真数据,两种方法在均选取最优参数的情况下,相对偏差曲线非常接近,SLWE 算法略小于加窗法.

4.2　面向真实数据的编码实验

本部分利用第 3 节中设计的基于 SLWE 概率估计算法的区间编码对真实数据进行编码实验,并与基于静态模型的区间编码和基于遗忘因子法的区间编码进行比较.

为验证本文研究的概率估计算法应用于实际熵编码器对于编码效果的影响,本文选取几种常见格式的真实数据进行熵编码实验,相关介绍见表 2.

数据的选取参考文献[13]中的数据选取方法,主要包括 bmp 图像、计算机中的系统文件、微软 word 文档和文本文档,所选数据中前 8 种数据属于非平稳数据,后两种 Price.txt 和 Rabinaranath Tagore.txt 属于平稳数据. 由于 SLWE 法与遗忘因子法的概率估计效果均与参数有关,为了公平起见,实验中采用如下参数选取方式:遗忘因子法中设置β = 0.5,$N_{\max} = 2^{14}$, 通过改变 M 调节算法的自适应能力, M 调节范围为 1~20,在给定的参数选择范围中选取压缩率最高的 M 值,记录下实验结果;对于 SLWE 法,设置 $P_{\min}$ = 0.001, 通过改变 λ 调节算法的自适应能力, λ 的选取范围为 0.9 到 0.99,间隔为0.01,同样在给定参数范围内选择压缩率最高的 λ 值,同样记录下实验结果. 表中 l_y 表示码流长度,B 表示 bytes, ρ 为压缩率,最后一栏 Ave/tol 列出所有实验数据的总大小和压缩后数据的总大小以及对各数据压缩率的均值.

表 2　实验数据及相关描述

文件名	描述
lena.bmp	图像压缩相关文献中常用的测试图像
bandon.bmp	图像压缩相关文献中常用的测试图像
opera.bmp	图像压缩相关文献中常用的测试图像
twain.dll	动态链接库文件,用于支持获取图像,系统版本 Windows 7,6.5a
timesbi.ttf	新罗马粗斜体字体文件,系统版本 Windows 7,6.5a
ariali.ttf	Arial Italic font 字体文件,系统版本 Windows 7,6.5a
coding_style.doc	微软 Word 文件,介绍 H.26L 参考软件的统一编码风格
Tofel.doc	微软 Word 文件,Tofel 写作关键词及短语汇总手册
Price.txt	文本文档,傲慢与偏见英文版
Rabinaranath Tagore.txt	文本文档,泰戈尔抒情诗选电子版,吴岩译

从表 3 中可看出 SLWE 算法与遗忘因子法一样,在对非平稳数据进行概率估计时能较好的适应数据特性的变化,显著的优于静态模型的概率估计效果,并且与遗忘因子法相比,在选定的数据范围内,SLWE 算法也优于遗忘因子法. 这说明 SLWE 算法非常适合非平稳环境下的概率估计问题.

表3 不同概率估计模型下的区间编码编码结果

文件名	原始大小/B	基于静态模型		基于遗忘因子法		基于SLWE算法	
		l_y/B	ρ/%	l_y/B	ρ/%	l_y/B	ρ/%
lena.bmp	263 224	250 160	4.96	243 087	7.65	240 255	8.73
bandon.bmp	263 224	213 600	18.85	176 190	33.06	161 858	38.51
opera.bmp	263 224	241 350	8.31	210 538	20.01	210 386	20.07
twain.dll	94 784	79 066	16.58	70 119	26.02	69 753	26.41
timesbi.ttf	621 296	535 390	13.82	485 197	21.90	484 515	22.02
ariali.ttf	557 004	475 330	14.66	425 455	23.62	422 049	24.23
coding_style.doc	54 272	33 423	38.41	27 646	49.06	27 417	49.48
Tofel.doc	55 297	22 242	59.77	18 750	66.09	18 087	67.29
Price.txt	32 590	23 374	28.28	22 260	31.70	21 588	33.76
Rabinaranath Tagore.txt	278 353	210 750	24.28	210 830	24.26	210 721	24.29
Ave/tol	2 483 268	2 084 685	19.12	1 890 072	31.38	1 866 629	33.03

5 结 论

本文以提高熵编码的编码性能为目的,以压缩率为指标,对熵编码中概率估计模型进行改进. 编码器参考的概率模型是否符合信源的实际特性会直接影响最后的编码性能,当待编码数据特性非平稳时,概率模型的建立问题将变得很复杂,原始概率模型往往会与信源的实际特性有所偏离,影响编码性能,若概率模型能及时的反映待编码数据这种特性变化,理论上可取得更好的编码效果.

因此,本文设计基于SLWE的区间编码方法为使概率模型所估计出的概率更好的映射到区间,在实现过程中,用区间累加替代原算法中的概率累加. 并且,通过设置最小区间避免了区间大小小于1的问题. 在对不同特性的非平稳数据的概率估计仿真实验和实际编码实验中,本文方法效果比加窗法和遗忘因子法等方法更好,这说明SLWE算法比较适合用于非平稳数据的概率估计,同时改进后的区间编码的编码效率更高.

对于本文SLWE算法中的学习因子λ,通过仿真实验得到最佳数值. 未来重点研究在于学习因子λ的选择方法,实现学习因子λ可调的概率估计方法.

参考文献

[1] DUTTWEILER D L, CHAMZAS C. Probability estimation in arithmetic and adaptive-Huffman entropy coders[J]. IEEE Transactions on Image Processing, 1995, 4(3): 237-246.

[2] MERHAV N, FEDER M. A strong version of the redundancy-capacity theorem of universal coding[J]. IEEE Transactions on Information Theory, 1995, 41(3): 714-722.

[3] GALLAGER R G. Variations on a theme by Huffman[J]. IEEE Transactions on Information Theory, 1978, 24(6): 668-674.

[4] SSNEHAG P, SHAO W, HUTTER M. Coding of non-stationary sources as a foundation for detecting change points and outliers in binary time-series[C]//Proceedings of the Tenth Australasian Data Mining Conference-Volume 134. Australian Computer Society, Inc., 2012: 79-84.

[5] YAZIDI A, OOMMEN B J, GRAMMO O C. A novel stochastic discretized weak estimator operating in non-stationary environments [C]//IEEE International Conference on Computing, Networking and Communications, 2012: 364-370.

[6] TSETLIN M L. On the behavior of finite automata in random media[J]. Avtomatika I Telemekhanika. 1961, 22(10): 1345-1354.

[7] OOMMEN B J, RUEDA L. Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments[J]. Pattern Recognition, 2006, 39(3): 328-341.

[8] SUDIP M, SUKHCHAIN S, MANAS K. MIRACLE: Mobility Prediction Inside a Coverage Hole Using Stochastic Learning Weak Estimator [J]. IEEE Transactions on Cybernetics , 2015, PP(99): 1-12.

[9] RUEDA L, OOMMEN B J. Stochastic automata-based estimators for adaptively compressing files with nonstationary distributions [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2006, 36(5): 1196-1200.

[10] MARTIN G N N. Range encoding: an algorithm for removing redundancy from a digitized message[C]//Proc. Institution of Electronic and Radio Engineers International Conference on Video and Data Recording. [S. l.]: [s. n.], 1979:1-11

[11] EVGENY B, LIU Kai, LI Yunsong. An Efficient Adaptive Binary Range Coder and Its VLSI Architecture[J]. IEEE Transactions on Circuits and Systems for Video Technology. 2015, 25(8):1435-1446.

[12] KRICHEVSKIV R E. Universal Compression and Retrieval [M]. Dordrecht, the Netherlands: Kluwer, 1994:217.

[13] ROS M, CUELLAR M P, DELGDO M, VILA A. Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal windows[J]. Information Sciences, 2013, 220(10):86-101.

(编辑 王小唯 苗秀芝)