云南大学学报 (自然科学版), 2002, 24 (5): 345~349 CN 53-1045/N ISSN 0258-7971
Journal of Yunnan University

基于 Context 量化的 Context 模型*

陈建华, 余锦华, 施心陵
(云南大学 电子工程系, 云南 昆明 650091)

摘要: Context 模型广泛地使用在图像编码系统中以提高压缩性能. 然而, 如果没有特殊的处理, 预期的压缩收益将会由于高阶 context 模型引入的模型代价而被抵消. Context 量化是解决这个问题的有效方法. 详细分析了一般 context 量化存在的问题, 提出 context 量化与一般矢量量化问题的相似性, 并且证明了, 如果设定了一个合适的失真度量准则, 运用 Lloyd 迭代算法可设计出最优的 context 量化器.

关键词: Context 模型; Context 量化; 高阶熵编码; Lloyd 迭代算法
中图分类号: TN 919.81 文献标识码: A 文章编号: 0258-7971(2002)05-0345-05

熵编码在图像编码系统中占有很重要的地位, 它通过利用信源符号的统计特性来获得图像的压缩. 然而, 由于高阶熵编码在实现上的复杂性, 使其在图像编码中一直未获得广泛运用, 直到近十几年来算术码技术及高速计算机的广泛运用才使高阶熵编码的实现成为可能. 近年来, 基于通用信源编码理论[1~3]的高阶熵编码引起了越来越多的图像编码研究者的注意[4~8]. 但高阶熵编码的实现面临着一个很严重的问题——模型代价.

根据通用信源编码理论, 如果我们需要将一串符号序列 $x_1, x_2, \cdots, x_n$ 连续地编码, 用于对符号序列编码的最少比特数应为

$$-\log_2 \prod_{i=1}^{n} p(x_i \mid x_{i-1}, \cdots, x_1). \qquad (1)$$

算术码能接近这一码长. 但是, $p(x_i \mid x_{i-1}, \cdots, x_1)$, $1 < i \leqslant n$ 在实际当中通常是未知的, 必须依靠以往编码过程中的观察从实际操作中估计出来. 估计 $p(x_i \mid x_{i-1}, \cdots, x_1)$ 的机制通常称为信源统计模型. 为决定当前符号的条件概率分布而用作条件的过去一系列观察值就称为模型 context. 显然, 该模型决定了我们对符号序列进行编码的效率. 式(1) 描述了对符号序列 $x_1, x_2, \cdots, x_n$ 编码所需的最小码长. 但是, 这并不意味着阶数越高的 context 模型就能获得越短的码长. 因为条件概率 $p(x_i \mid x_{i-1}, \cdots, x_1)$ 必须由相应不同 context 组合状态下的多个符号统计直方图估计, 如果 context 模型的阶数过高, 一幅图像就有可能提供不了足够的样点使得大量的符号统计直方图收敛于 $p(x_i \mid x_{i-1}, \cdots, x_1)$. 换句话说, 在所有可能的模型状态中, 过大的 context 模型将使得对信源符号的计数统计变得过于分散, 从而不能获得准确的条件概率估计. 尽管"条件降低熵值", 但总的结果是码长将增加. 所以, 高阶 context 模型不仅仅带来处理耗时及存储复杂性的问题, 而且还会造成编码效率的降低. 这个问题通常被称为"context 稀释", Rissanen 将此问题以公式化分析为"模型代价"[2].

为了解决模型代价, 人们提出了各种方案. 在 context 算法(AC)[1]和 context 树加权(CTW)[5]这 2 个方案中, 树形结构被用于表示 context 模型. 通过修剪以形成最佳树或者通过将树根的后继节点的概率加权来降低模型代价. 同时, 为了既获得主要的 context 信息又降低模型代价, 对 context 模型的量化是一个很重要的技术. 在文献[6]中, context 变量被量化以减少 context 模型中参数的数量. 然而, 尽管其优化的目标不是最小均方误差, 但 context 变量是作为普通随机变量被标量量化器量化的.

本文提出了最优 context 量化问题的综合分析

* 收稿日期: 2002-01-07
基金项目: 云南省自然科学基金资助项目(1999F0003Q).
作者简介: 陈建华(1964-), 男, 云南人, 博士, 副教授, 主要从事图像数据压缩及信号处理方面的研究

以及一种迭代 context 量化器设计算法. 本文余下部分组织如下: 在第2节中, 我们提出通过设定适当的失真度准则, context 量化问题便与普通的矢量量化问题相似. 如果能在一给定划分中找到最优量化值, 那么 context 量化同样能用迭代算法实现. 在第3节中, 我们以相对熵作为失真度量准则, 证明了最优 context 量化通过 Lloyd 迭代算法[9] 能够实现. 实施过程的一些细节考虑则在第4节中给出.

1 问题的描述

设有一随机变量 C, 在给定随机变量 E 的条件下对 C 编码所须的最小比特率为条件熵 $H(C \mid E)$. 如果 E 被量化为 $M = Q(E)$, 那么条件熵变为 $H(C \mid M)$, 并且[10]

$$H(C \mid E) = H(C \mid E, M).$$

由此可得

$$H(C \mid M) = H(C \mid E, M) + I(C; E \mid M) = H(C \mid E) + I(C; E \mid M),$$

并且

$$H(C \mid M) - H(C \mid E) = I(C; E \mid M) \geqslant 0. \tag{2}$$

这就意味着 context 量化增加了熵. 然而, context 量化同时降低了模型代价. 通过使用设计适当的量化器, 结合基于量化 context 的算术编码器, 仍可能获得更好的压缩性能. 由此可见, 优化 context 量化器的目标应是在给定的量化级数的条件下找到一种能够最小化 $H(C \mid M) - H(C \mid E)$ 的 context 量化器.

具体来说, 对某个离散随机变量进行量化时的优化目标是最小化下列失真

$$D = E\,\mathrm{d}(x, Q(x)) = \sum_i \mathrm{d}(x_i, Q(x_i))\, p(x_i), \tag{3}$$

这里 $\{x_i\}$ 是具有非零概率的 x 的值, $Q(x_i)$ 是 x_i 的量化值, $\mathrm{d}(\cdot, \cdot)$ 是给定的失真度量因子.

由于随机变量 E 有 N 个不同的值, 我们可以定义一个集合 $\mathscr{E} = \{e_i, i = 1, \ldots, N\}$, 这样 E 的每一个特定的值由集合中相应的 e_i 值决定. 假定训练样本的数量足够多, 这样条件概率 $p(c \mid e_i)$ 和概率 $p(e_i)$ 的值对所有的 i 均能可靠的估计.

与上面相似, 我们可以为随机变量 M 定义一个集合 $\mathscr{M} = \{m_k, k = 1, \ldots, K\}$. 这里, K 是给定的量化级数. 很明显 $m_k, k = 1, \ldots, K$, 是 $\mathscr{E}$ 的一个子集. $\mathscr{E} = \bigcup_{k=1}^{k} m_k$ 并且 $m_i \cap m_j = \Phi$, 如果 $i \neq j$. 这样, 每个子集 m_k 对应于 e_i 的一种划分, 并且

$$p(m_k) = \sum_{e_j \in m_k} p(e_j).$$

从 $H(C \mid E)$ 和 $H(C \mid M)$ 的定义

$$H(C \mid E) = \sum_{e_i \in \varepsilon} p(e_i) \sum_c p(c \mid e_i) \log_2 \frac{1}{p(c \mid e_i)}, \tag{4}$$

$$H(C \mid M) = \sum_{m_k \in M} p(m_k) \sum_c p(c \mid m_k) \cdot \log_2 \frac{1}{p(c \mid m_k)}, \tag{5}$$

我们可以看到从 E 到 M 的量化对应于用一个 $p(c \mid m_k)$ 来代表所有的 $p(c \mid e_j)$, 这里 $e_j \in m_k$, 并且 $p(c \mid m_k)$ 可视作所有相应 $p(c \mid e_j)$ 的量化值.

接下来, 类似于上面(3)式提到的失真, 我们可以给出 context 量化的量化失真如下

$$D = \sum_{e_i \in \varepsilon} \mathrm{d}(p(c \mid e_i), Q(p(c \mid e_i)))\, p(e_i), \tag{6}$$

context 量化的优化目标变为: 对于给定的量化级数 K, $(K < N)$, 为所有的 e_i, $i = 1, \ldots, N$ 找到一个最优划分算法, 然后为所有的 K 个划分计算最优量化值 $Q(p(c \mid e_i))$, 由此, (6) 式所示的失真被最小化.

由量化器优化理论[11] 可知, 如果最近近邻条件和质心条件这两个准则能够满足, 我们就能用 Lloyd 算法或者神经网络方法[12] 来找到最优的(至少是局部最优的)量化器. 换句话说, 如果我们能选择一种合理的失真度量方法来估计两种概率分布之间的距离, 并且在这个失真度量的基础上, 我们能为特定的划分找到最优"概率分布表示"(量化值), 我们就能用 Lloyd 迭代算法解决 context 量化的问题. 同时, 如果我们能证明方程(6)的最小化等价于方程(2)的最小化的话, 用迭代算法设计出的量化器就是最优量化器.

2 Context 量化

我们选择相对熵

$$D(p(c \mid e_i) \parallel q_k(c)) = \sum_c p(c \mid e_i) \log_2 \frac{p(c \mid e_i)}{q_k(c)} \tag{7}$$

作为我们的量化失真度量标准. 这里, $q_k(c)$ 是所有 $p(c|e_i)$ 的量化值 $e_i \in m_k$. (6) 式中的量化失真变为

$$D = \sum_{e_i \in \varepsilon} D(P(c|e_i) \| q_k(c)) p(e_i), \qquad (8)$$

这里必须注意因为相对熵的非对称性, 所以它并不是 2 个概率分布之间的真实距离. 我们之所以选择它作为失真度量标准是因为它反映了用 $q_k(c)$ 代替 $p(c|e_i)$ 的损失. 另一方面, 可以证明使用这种失真度量, 在 e_i 的最优划分算法已经确定的情况下, 很容易计算出不同的量化值 $q_k(c)$. 然而, 由于相对熵的非对称性, 量化值 $q_k(c)$ 应该总是被放在分母上, 不同的距离才能被适当地比较.

命题 1　对于给定的量化级数 K, 如果为所有的 e_i 都找到了最优划分算法, 只有当量化值 $p(c|m_k)$ 对于每一个划分 m_k 都有以下的形式, (8) 式中的量化失真才被最小化

$$p(c|m_k) = \frac{\sum_{e_j \in m_k} p(e_j) p(c|e_j)}{\sum_{e_j \in m_k} p(e_j)}. \qquad (9)$$

证明　(8) 式中的量化失真可以被写为

$$D = \sum_{m_k \in M} \sum_{e_j \in m_k} D(p(c|e_j) \| p(c|m_k)) p(e_j), \qquad (10)$$

因为 $D(p(c|e_j) \| p(c|m_k)) p(e_j) \geqslant 0$, 如果

$$\frac{1}{\sum_{e_j \in m_k} p(e_j)} \sum_{e_j \in m_k} D(p(c|e_j) \| p(c|m_k)) p(e_j),$$

对于每一个 $k \in \{1, \ldots, K\}$ 都被最小化, 那么 D 将会被最小化. 常数 $\frac{1}{\sum_{e_j \in m_k} p(e_j)}$ 对于上述失真的最小化是没有影响的. 将 $\sum_{e_j \in m_k} p(e_j)$ 计作 L, 那么

$$\frac{1}{L} \sum_{e_j \in m_k} D(p(c|e_j) \| p(c|m_k)) p(e_j) =$$

$$\frac{1}{L} \sum_{e_j \in m_k} p(e_j) \sum_c p(c|e_j) \log_2 \frac{p(c|e_j)}{p(c|m_k)} =$$

$$\frac{1}{L} \sum_{e_j \in m_k} p(e_j) \sum_c p(c|e_j) \log_2 \frac{1}{p(c|m_k)} -$$

$$\frac{1}{L} \sum_{e_j \in m_k} p(e_j) \sum_c p(c|e_j) \log_2 \frac{1}{p(c|e_j)} =$$

$$\sum_c \frac{1}{L} \sum_{e_j \in m_k} p(e_j) p(c|e_j) \log_2 \frac{1}{p(c|m_k)} -$$

$$\frac{1}{L} \sum_{e_j \in m_k} p(e_j) \sum_c p(c|e_j) \log_2 \frac{1}{p(c|e_j)}.$$

$$\beta_k(c) \stackrel{\text{def}}{=} \frac{1}{L} p(e_j) p(c|e_j),$$

那么 $\beta_k(c)$ 也是在 C 上的概率分布, 并且我们有

$$\frac{1}{L} \sum_{e_j \in m_k} D(p(c|e_j) \| p(c|m_k)) p(e_j) =$$

$$\sum_c \beta_k(c) \log_2 \frac{\beta_k(c)}{p(c|m_k)} + \sum_c \beta_k(c) \log_2 \frac{1}{\beta_k(c)} -$$

$$\frac{1}{L} \sum_{e_j \in m_k} p(e_j) \sum_c p(c|e_j) \log_2 \frac{1}{p(c|e_j)}, \qquad (11)$$

因为 $p(e_j)$ 和 $p(c|e_j)$ 是已知的, $\beta_k(c)$ 也就已知, 并且

$$\sum_c \beta_k(c) \log_2 \frac{1}{\beta_k(c)} -$$

$$\frac{1}{L} \sum_{e_j \in m_k} p(e_j) \sum_c p(c|e_j) \log_2 \frac{1}{p(c|e_j)}$$

将是恒定的. 对于 (11) 中的第一项, 我们有

$$\sum_c \beta_k(c) \log_2 \frac{\beta_k(c)}{p(c|m_k)} \geqslant 0,$$

当且仅当

$$p(c|m_k) = \beta_k(c) = \frac{\sum_{e_j \in m_k} p(e_j) p(c|e_j)}{\sum_{e_j \in m_k} p(e_j)}$$

时, 等式成立.

证毕.

命题 2　如果对每一划分 m_k 量化值被表示为

$$p(c|m_k) = \frac{\sum_{e_j \in m_k} p(e_j) p(c|e_j)}{\sum_{e_j \in m_k} p(e_j)},$$

并且失真度量被定义为

$$D(p(c|e_i) \| q(c)) =$$

$$\sum_c p(c|e_i) \log_2 \frac{p(c|e_i)}{q(c)}$$

方程式 (2) 和方程式 (8) 中的表达式是相同的, 也即

$$H(C|M) - H(C|E) = D.$$

证明　因为

$$p(m_k) = \sum_{e_j \in m_k} p(e_j),$$

$$H(C|M) = \sum_{m_k \in M} p(m_k) \sum_c p(c|m_k) \cdot$$

$$\log_2 \frac{1}{p(c|m_k)} =$$

$$\sum_{m_k \in M} p(m_k) \sum_c \left[\frac{\sum_{e_j \in m_k} p(e_j) p(c|e_j)}{\sum_{e_j \in m_k} p(e_j)} \right] \cdot$$

$$\log_2 \frac{1}{p(c \mid m_k)} = \sum_{m_k \in M} \sum_{c} \sum_{e_j \in m_k} p(e_j)\, p(c \mid e_j) \cdot \log_2 \frac{1}{p(c \mid m_k)} = \sum_{m_k \in M} \sum_{e_j \in m_k} p(e_j) \sum_{c} p(c \mid e_j) \cdot \log_2 \frac{1}{p(c \mid m_k)},$$

而没有量化的 context 的条件熵是

$$H(C \mid E) = \sum_{e_i \in \varepsilon} p(e_i) \sum_{c} \log_2 \frac{1}{p(c \mid e_i)} \sum_{m_k \in M} \sum_{e_j \in m_k} p(e_j) \sum_{c} p(c \mid e_j) \cdot \log_2 \frac{1}{p(c \mid e_j)}.$$

那么,

$$H(C \mid M) - H(C \mid E) = \sum_{m_k \in M} \sum_{e_j \in m_k} p(e_j) \sum_{c} p(c \mid e_j) \log_2 \frac{p(c \mid e_j)}{p(c \mid m_k)} = \sum_{m_k \in M} \sum_{e_j \in m_k} D(p(c \mid e_j) \parallel p(c \mid m_k))\, p(e_j) = D$$

证毕.

很显然, 如果定义式(7) 中条件熵作为失真度量, 最近近邻条件就满足了. 有了式(9) 中定义的每一划分的量化值, 质心条件就满足了. 因此, 如果我们知道了概率分布 $p(e_i)$ 和 $p(c \mid e_i)$ 我们就能找到用 Lloyd 迭代算法设计 context 量化器的最优解决方案. 通过命题二, 我们知道(8) 式表示的失真最小化也就等价于(2) 的最小化. 所以, 上述 Lloyd 算法所设计的 context 量化器就是我们所需要的最优量化器.

Lloyd 算法概述如下:

步骤 1: 用足够多的图像来训练条件概率分布 $p(c \mid e_i)$, $i = 1, \ldots, N$, 由此得到对 $p(c \mid e_i)$ 和 $p(e_i)$ 的准确估计.

步骤 2: 建造一个初始的 context 量化器, $Q_c^1 = p(c \mid m_k)$, $k = 1, \ldots, K$. 这里 K 是量化划分的数量, 设 $n = 1$.

步骤 3: 为每一个集合 m_c 中的 e_j 设定 context 量化器 $Q_c^n = p(c \mid m_k)$, $k = 1, \ldots, K$, 对所有的集合 m_k, $k = 1, \ldots, K$ 计算相对熵

$D(p(c \mid e_j) \parallel p(c \mid m_k))$,

寻找 $D(p(c \mid e_j) \parallel p(c \mid m_k))$ 的最小值. 如果 $l \neq c$, 把 e_j 从集合 m_c 移到集合 m_l. 这一步骤重复至所有的集合 m_c, $c = 1, \ldots, K$ 都被处理.

步骤 4: 对所有的集合 m_k 用(9) 计算量化值 $p(c \mid m_k)$ 来获得新的 context 量化器 Q_c^{n+1}.

步骤 5: 对 Q_c^{n+1} 计算(10) 式中的失真. 如果它与上一步迭代相比只改变很小的数值, 就停止, 否则将 $n+1$ 赋值给 n, 然后转到步骤 3.

3 实现考虑

3.1 $\mathscr{E}$的大小　假设我们有一个 context 事件的集合 $c_1, \ldots, c_n$ 并且 $c_i \in \mathscr{T}$. 令 N_t 等于集合 $\mathscr{T}$的大小并且令 $\bar{c}_i$ 是 c_i 在 $\mathscr{T}$中的索引. 作为条件的随机变量 E 可被视作 $\bar{c}_1, \ldots, \bar{c}_n$ 的组合. 为了简化起见, 我们用如下公式计算集合 $\mathscr{E}$中 e_i 的索引值 i

$$i = \bar{c}_n \times N_t^{n-1} + \overline{c_{n-1}} \times N_t^{n-2} + \cdots + \bar{c}_2 \times N_t^1 + \bar{c}_1, \tag{12}$$

因此, 每个特定 $\bar{c}_1, \ldots, \bar{c}_n$ 的组合将会对应集合 $\mathscr{E}$中的一个特定的 e_i, 这样, 所有 i 的可能值 N 就等于 N_t^n. 因为我们需要为每一个 e_i 建立一个统计直方图, 我们的计算机应该具有存储所有的直方图的存储量. 然而, 如果没有特别的处理来限制 $\mathscr{T}$的大小以及 n 的值, N 将变得很大, 我们的计算机将不能为所有的 $p(c \mid e_i)$ 存储所有的直方图. 实际中, 我们能选择降低 context 事件 n 的数量, 或者 $\mathscr{T}$的大小, 或者两者一起减小, 由此, N 就被限制在一个适当大小的范围内了.

3.2　步骤 3 中 $D(p(c \mid e_j) \parallel p(c \mid m_k))$ 的计算

在寻找 $D(p(c \mid e_j) \parallel p(c \mid m_k))$ 的最小值的过程中, 当 $e_j \in m_k$, $l \neq k$ 时, $D(p(c \mid e_j) \parallel p(c \mid m_k))$ 通常会小于 $D(p(c \mid e_j) \parallel p(c \mid m_l))$ 特别是当 m_l 的尺寸小的时候. 为了对所有 m_k 都能适当比较 $D(p(c \mid e_j) \parallel p(c \mid m_k))$, 我们假设对所有 m_k, $e_j \in m_k$, 然后, 对所有 m_k 重新计算 $p(c \mid m_k)$ 和 $D(p(c \mid e_j) \parallel p(c \mid m_k))$. 经过这样处理, 我们找到的最小的 $D(p(c \mid e_j) \parallel p(c \mid m_k))$ 将更精确.

4 结　论

Context 量化是实现高阶熵编码同时降低模型代价的重要技术. 我们介绍了最优 context 量化的综合分析. 通过选择一种合适的失真度量准则, 我们证明了 context 量化与矢量量化的相似性, 并且

提出在给定量化级数的情况下, 用 Lloyd 迭代算法来寻找最优 context 量化器的方法.

参考文献:

[1] RISSANEN J. A universal data compression system[J]. IEEE Trans Info Theory, 1983, 29(5): 656—664.

[2] RISSANEN J. Universal coding, information, prediction and estimation[J]. IEEE Trans Info Theory, 1984, 30: 629—636.

[3] WEINBERGER M J, RISSANEN J, FEDER M. A universal finite memory source[J]. IEEE Trans Info Theory, 1995, 41(3): 643—652.

[4] WEINBERGER M J, RISSANEN J, ARPS R B. Applications of universal context modeling to lossless compression of gray-scale images[J]. IEEE Trans Image Process. 1996, 5(4): 575—586.

[5] WILLEMS F M, SHTARKOV Y M, TJALKENS T J. The context-tree weighting method: basic properties[J]. IEEE Trans Info Theory, 1995, 41(3): 653—664.

[6] WU X. Lossless compression of continuous-tone images via context selection, quantization, and modeling[J]. IEEE Trans Image Processing, 1997, 6(5): 656—664.

[7] ORDENTLICH E, WEINBERGER M, SEROUSSI G. A Low-complexity modeling approach for embedded coding of wavelet coefficients[J]. Proc of IEEE Data Compression Conference, 1998: 408—417.

[8] RISSANEN J, LANGDON G. Arithmetic coding[J]. IBM J Res and Devl, 1979, 23(2): 149—162.

[9] LLOYD S P. Least squares quantization in PCM[J]. IEEE Trans Info Theory, 1982, 28(2): 129—137.

[10] CONER T M, THOMAS J A. Elements of information theory[M]. USA: John Wiley & Sons Inc, 1991.

[11] GERSHO A, GRAY R M. Vector quantization and signal compression[M]. Netherlands: Kluwer Academic Publishers, 1992.

[12] KOSKO B. Neural networks for signal processing[M]. USA: Prentice-Hall Inc Englewood Cliffs, New Jersey, 1992.

Context modeling based on context quantization

CHEN Jian-hua, YU Jin-hua, SHI Xin-ling
(Department of Electronic Engineering, Yunnan University, Kunming 650091, China)

Abstract: Context modeling is widely used in image coding to improve the compression performance. However, with no special treatment, the expected compression gain will be cancelled by the model cost introduced by high order context models. Context quantization is an efficient method to deal with this problem. The general context quantization problem is analyzed in detail and showed that context quantization is similar to a common vector quantizer can be designed by a Lloyd style iterative algorithm.

Key words: context modeling; context quantization; high order entropy coding; Lloyd style iterative algorithm