

图象矢量量化器设计的研究

袁 国 流

摘 要

矢量量化是近年用于图象数据压缩的一种高性能的方法,它把信号作为矢量而不是作为标量来编码。在矢量量化器的设计过程中,过程的初始化对于整个设计的计算代价(收敛速度)以及所达到的指标(平均失真)有重大的影响。本文提出了一种“误差函数初始化”(EFI)的方法,这种方法直接针对失真准则,通过求“误差降低函数”的估计量的最大值而使“种子”的选择最佳化。实验结果以及在此基础上的分析表明了EFI作为初始化方法的优点,并且显示了其作为实时矢量量化器刷新算法的前景。

【关键词】 矢量量化; 量化器; 图象压缩; 设计算法; 最佳化

1 概 述

信号的矢量量化方法较早应用于语音信号的压缩。近年来。研究表明它在数字图象编码中也是一个甚为有效的方法^[1]。其原理如下:

一个 K 层的矢量量化器 Q 是一个把某 N 维矢量集 $\{V\}$ 映射为一个有限的码字表 $W = \{W_1, W_2, \dots, W_k\}$ 的映射器。对于任一个输入矢量 V ,量化器的输出是码字表中 K 个码字(或“代表矢量”)中的某一个。这种映射的基础是将输入的矢量空间适当地划分成 K 个互不重叠的部分,并给每一部分指定一个代表矢量。这样,一个矢量量化器就完全由如此划分而成的子空间集 $R = \{R_1, R_2, \dots, R_k\}$ 以及相应的码字表 W 所完全决定,而且有关系

$$R_i = \{V : Q(V) = W_i\},$$

其中 W_i 及 V 都是 N 维矢量。矢量量化器的设计包括在量化误差最小化的准则下输入矢量空间的划分以及相应码字表的产生。

在矢量量化器的实际设计中,通常不是根据信源的分布模型,而是以一个所谓“训练集”(training set,实际上是典型条件下从矢量信源总体中得到的一组足够大的样本)为基础,利用“ K 均值分群算法”(K-means clustering algorithm)来划分信源矢量空间以及产生相应

的码字表的^[2]。K均值分群算法是一个迭代过程，它的目标是按照预先设定的K值（量化器输出层数），把训练集数据分作K“群”且求出相应的各群的中心（下称群心）；这个结果在某一判据的意义上至少是局部最佳的。对于矢量量化来说，判据就是平均量化误差最小，而结果中的K个群心将就是量化器输出端所可能取的矢量值。K均值分群算法的步骤可概述如下：

1. 初始化，也就是采用某种方法选取K个初始的群心（以下称“种子”）。

2. 把输入的（实际上是训练集中的）矢量逐一地分配给它最“接近”的一个群心（所谓接近就是使量化误差最小），这样也就把训练集分成了K群。

3. 求出每个群的平均矢量作为新的群心。

4. 回到第2步循环运算直到过程收敛。

这个算法之所以成为矢量量化器设计的基本方法，是因为在这个算法中，每一步迭代的结果都不会劣于该步迭代前的情况，因此可以保证过程最终必收敛于某种最佳化的结果。不过应该指出下列两点：第一，K均值分群算法的每一次迭代都是相当费时的。对于同一个训练集进行处理，在同一收敛判据之下，可能会经过不同次数的迭代才达到收敛；所需的迭代次数完全取决于初始化。第二，算法过程本身所保证达到的最佳化只是一种局部的最佳化；可能达到的局部最佳结果原则上有多多个；这些结果可以依其最终的平均量化误差来比较优劣。而算法过程最后达到何种结果，即其平均量化误差如何，也是完全取决于初始化。因此，初始化的方法，即如何合理地整个过程选择种子，是十分关键的一步。针对这一点，已有下述几种初始化的方法被提出来：

1. 二进分裂法^[3]：首先根据训练集的统计平均产生两个种子，通过K均值分群算法对这两个种子进行优化之后，使每个一分为二，成为4个种子。然后再优化，再分裂，如此逐次翻番直至所需数目的种子全部决定下来。这个方法的主要问题是算法中包含着K均值分群过程本身，尽管采取了限制迭代次数的措施，时间开销仍然很大。

2. 最大频值法^[4]：首先在训练集上生成一个多维矢量频度分布函数，然后找出其极大值点，并把它们的位置矢量作为种子。这个方法有两方面的问题：从实际的角度看，要生成一个其“详细程度”足以找出必要数目的极大值点的频度分布函数，要求非常巨大的内存空间，往往超出小型机的容量；从理论的角度看，极大频值点的分布同合理的群心分布看不出有什么必然的联系。

3. 参数法^[5]：在多维空间中找出K个线性分布的矢量作为种子，其分布的中心及间隔是分别以训练集各维数据的一阶矩和二阶矩为参数决定的。这个方法要求一个“各维数据强烈相关”的前提，这不能不使其适应性受到很大的局限。

4. 最大分隔法^[6]：在训练集中找出K个互相间距离尽可能大的矢量作为种子。该方法因完全未利用数据的统计特性而带有较大的随机性。

本文提出一种称为“误差函数初始化方法”（以下简称EFI）的初始化技术。在这种方法中每决定一个种子都是以前面已经决定的种子的分布为基础的。简要地说，在现有种子分布的条件下，每增加一个种子，一般将使总量化误差降低，但降低的量与新种子的位置有关。如果对不同位置产生的量化误差减量进行估计并求其最大值，则可求得在估计的意义上最佳的新种子的位置。由于这种方法是直接与失真的量度相联系的，因而作为初始化技术，它具有更好的性能，并且其初始量化误差小的特色对于实时自适应矢量量化系统是有特别意义

的。

2 误差函数初始化方法 (EFI)

一种好的初始化技术应该具有下列 3 个特征：首先，它应能使算法过程达到较小的量化误差；第二，它应能使过程收敛较快；第三，初始化本身的代价要小（主要由计算时间衡量）。下文提出的 EFI 方法在这几方面都有较好的性能。

EFI 的基本思想是针对量化误差准则对所选择的种子逐个进行最佳化处理。假设对于一个给定的训练集，已经存在 i 个种子。我们可以认为这 i 个种子构成了一个 i 层的矢量量化器 $Q(i)$ ，每个种子都是其输出的代表矢量值。把 $Q(i)$ 应用于训练集将产生一定的量化误差（或失真） E_i 。现在，如果要通过给 $Q(i)$ 增加一个输出代表矢量的方式来构成一个 $i+1$ 层的量化器 $Q(i+1)$ ，则新的量化器所产生的失真与第 $i+1$ 个种子的位置有关，表示为 $E_{i+1}(p)$ ，其中 p 为新种子的位置。显然，有 $E_{i+1}(p) \leq E_i$ ；而二者之差 $R_{i+1}(p) = E_i - E_{i+1}(p)$ 量度了由于量化器增加了一个输出层而使失真降低的程度，称为“误差降低函数”，它随第 $i+1$ 个种子的位置 p 而变化。我们总是希望通过适当地决定 p ，使得 R_{i+1} 的值为最大。

企图对矢量空间的每一点来计算 $R_{i+1}(p)$ 的值以求得函数最大值是不实际的，计算代价将太大。一个合理的途径是构造并计算另外一个函数，其特征应与 $R_{i+1}(p)$ 密切相关但易于计算。为了构造这样的函数我们注意到下列事实：每当量化器增加一个新的输出层的时候，仅有一部分输入矢量的量化结果受到影响，这些受影响的矢量必定是在矢量空间中相对靠近新的输出代表矢量的。这个事实意味着 $R_{i+1}(p)$ 等于位置 p 附近的输入矢量在相继两次量化中失真减少量的累加值。那么问题变成：当一个新的输出代表矢量出现在位置 p 的时候，附近有多少输入矢量的量化结果要受到影响以及影响的大小如何？

如果能够估计出受影响的输入矢量的数目以及每一矢量受影响的“平均”程度，则 $R_{i+1}(p)$ 的估计值可以表达成这两个因数的乘积。其中受影响矢量的数目是与输入矢量出现在 p 位置邻域的频度密切相关的。为了估计它可以把矢量空间离散成计算单元，在训练集上统计出每一单元上输入矢量出现的频度。至于受影响的平均程度，可以用一个恰好出现在 p 位置上的输入矢量 V_p 所受的影响程度来代表。按照均方误差准则，不难理解， V_p 在第 $i+1$ 个种子出现以前的量化误差等于 p 位置与原来的 i 个种子中最靠近的一个种子的距离 $D_i(p)$ 的平方；而在新种子出现（在 p 位置）之后，其量化误差为零。即是说 V_p 受影响的程度就是 $D_i^2(p)$ 。其他矢量受影响的程度可能比 $D_i^2(p)$ 大，也可能较小，所以按 $D_i^2(p)$ 来估计 p 点附近输入矢量受影响的平均程度（至少对于用作比较的相对程度）是合理的。综上所述， $R_{i+1}(p)$ 可以由下式定义的“拟误差减少函数” $Q_{i+1}(p)$ 来估计：

$$Q_{i+1}(p) = f(p) \cdot D_i^2(p),$$

其中 $f(p)$ 是 p 点所在的矢量空间计算单元内输入矢量出现的频度， $D_i(p)$ 是 p 点与原来最靠近的一个种子的距离。 $Q_{i+1}(p)$ 函数构成的几何意义见图 1 所示。实际上 p 点取的都是每一计算单元的中心。

在计算机实现中，EFI 算法包括下列步骤：

第 1 步。把矢量空间划分成一定数目的计算单元，并在训练集上统计每一单元内输入矢

量出现的频度 $f(p)$ 。

第 2 步, 对频度设置一阈值, 筛分出高于该阈值的单元来候选, 舍弃其余所有的单元。

第 3 步, 选取具有最大频度的单元的中心作为第一个种子。

第 4 步, 在其余的单元中, 找出具有最大 $Q_i(p)$ 值的单元的中心作为下一个种子。

第 5 步, 重复第 4 步直至产生出所需数目的种子。

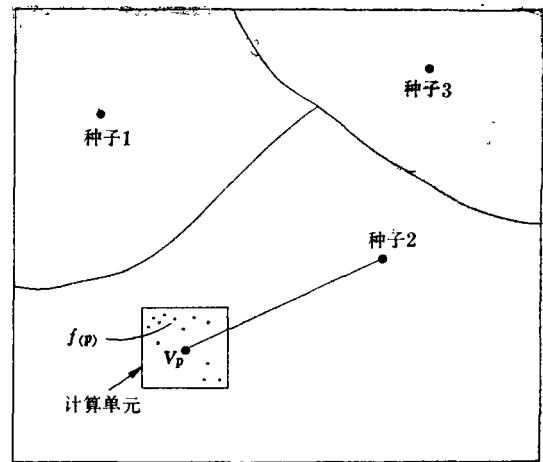


图 1 拟误差减少函数在二维空间中 (当 $i=3$ 时) 构成的示意图

3 实验数据与结果

对上述的 EFI 方法, 我们通过实验对其性能进行了研究。在实验中还包括对下列 3 种初始化方法在同等条件下进行的试验, 以便与 EFI 进行比较: 二进分裂法、参数法、最大分隔法。至于最大频度法, 则由于其对计算系统的要求太高而未能试验。

为了考察各种技术对于不同类型数据的表现, 我们采用两个训练集, 分别取自两组具有不同统计特性的原始数据。第 1 组称为“相关数据”, 来自一幅单色人面图象, 每一矢量由一个 2×2 像素的图象子块形成。第 2 组称为“去相关数据”, 来自卫星地面图象; 图象首先划分成 3×3 像素的子块, 每子块的 9 个灰度值经用该子块内的灰度均值及离散度进行归一化以降低相关性之后形成一个矢量。人面图象主要由大片的灰度渐变块面构成, 数据的空间相关性较强。卫星地面图象则含有丰富的细节, 再经过去相关的预处理, 子块内灰度的相互依赖性就更弱了。显然, 就矢量各维之间的相关程度而言, 这两组数据有显著的差别。两个训练集的数据量分别是 512×512 像素和 510×510 像素。

实验结果在表 1 和表 2 中给出。其中“二进分裂法(2)”和“二进分裂法(4)”分别对应两种二进分裂法方案, 一种在其初始化过程中每步引用 K 均值分群迭代的次数限于 2, 而另一种限于 4。“初始失真”和“最终失真”分别指在初始化完成之后和在量化器设计完成之后所达到的均方误差 (MSE)。“收敛所需迭代次数”指的是经初始化之后, 开始实行 K 均值分群算法直到达到收敛判限所进行的迭代次数; 但对于二进分裂法, 这个次数包括了初始化时所需的迭代次数。收敛判限采用下式定义的“量化误差相对差” (QED):

$$QED = (\sqrt{MSE_{t-1}} - \sqrt{MSE_t}) / \sqrt{MSE_t},$$

其中 $t-1$ 和 t 分别指前一次和目前一次迭代。表 1 采用的 QED 判限为 0.001 而表 2 为 0.005。

上列实验结果表明, EFI 能适用于相关程度不同的数据; 与其他 3 种初始化技术相比, EFI 一般能使 K 均值分群算法以较少的迭代次数收敛于较低的最终误差。值得一提的是, 用于 EFI 的计算代价比用于 K 均值分群算法一次迭代的计算量还要小得多。

EFI 的一个显著特征是其初始误差普遍地、明显地比参数法和最大隔离法的要小 (只有二进分裂法是例外, 这种方法的初始误差往往也较小。但显然这是由于这种初始化过程本身

表 1 各种初始化方法性能比较 (对于“相关数据”)

量化器分层数	方 法	收敛所需迭代次数	初始失真 (MSE)	最终失真 (MSE)
4	E	4	247.2	229.1
	P	14	1140.6	229.8
	S	5	1029.7	230.0
	B2	7	1180.2	229.5
	B4	8	982.0	230.5
8	E	6	94.7	79.7
	P	21	126.7	93.6
	S	20	327.4	92.1
	B2	9	209.0	79.7
	B4	13	207.8	79.7
16	E	13	47.8	44.8
	P	9	72.5	52.1
	S	15	287.0	48.9
	B2	6	71.3	44.6
	B4	8	70.1	44.6
32	E	20	35.8	30.1
	P	21	53.9	31.0
	S	27	181.9	30.6
	B2	20	39.6	29.4
	B4	25	39.5	28.8
64	E	17	32.4	19.8
	P	16	48.2	22.0
	S	19	75.4	21.4
	B2	19	32.1	19.8
	B4	24	30.9	19.8
128	E	17	31.6	13.8
	P	19	46.2	14.7
	S	21	45.4	14.9
	B2	20	24.0	13.6
	B4	17	20.5	13.6

注: E—误差函数法, P—参数法, S—最大分隔法,

B2—二进分裂法(2), B4—二进分裂法(4)

表 2 各种初始化方法性能比较 (对于“去相关数据”)

量化器分层数	方 法	收敛所需迭代次数	初始失真 (MSE)	最终失真 (MSE)
4	E	4	0.696	0.567
	P	3	1.096	0.939
	S	4	0.878	0.590
	B2	11	0.919	0.567
	B4	10	0.812	0.572
8	E	4	0.554	0.470
	P	10	1.003	0.471
	S	6	0.754	0.472
	B2	7	0.689	0.479
	B4	10	0.588	0.472
16	E	4	0.457	0.381
	P	9	0.983	0.385
	S	6	0.639	0.399
	B2	8	0.507	0.383
	B4	10	0.486	0.380
32	E	5	0.431	0.315
	P	11	0.979	0.322
	S	7	0.413	0.317
	B2	8	0.412	0.320
	B4	9	0.392	0.318
64	E	6	0.383	0.262
	P	10	0.979	0.263
	S	6	0.424	0.264
	B2	8	0.346	0.262
	B4	10	0.328	0.262
128	E	8	0.356	0.214
	P	11	0.979	0.216
	S	7	0.336	0.216
	B2	8	0.285	0.215
	B4	10	0.274	0.214

已经包括有若干步 K 均值分群算法的迭代运算，即它的这种效果是以很大的运算量为代价的)。EFI 的这个特色意味着，采用这种技术，可以在量化器的设计中，在一开始就使量化器达到相对地接近于最终性能的指标。这一特点可能很有意义。在某些场合下，用 EFI 算法取代费时的 K 均值分群算法来作为一种近似的然而非常快速的矢量量化器设计方法也许是可取的。比如在数据统计特性剧烈变化的条件下，有可能利用 EFI 技术构成一种自适应的矢量量化器。自适应意味着要经常地刷新量化器，即频繁地对量化器重新进行设计。在这种环境下，耗费大量计算代价使量化器精确地切合某一时刻的数据特性并无多大意义，重要的是使量化器灵敏地跟上变化着的数据特性。在这种背景下，EFI 不仅能作为初始化方法，而且看来具有作为一种快速的矢量量化器设计方法的潜力。

4 结 论

本文提出并通过实验初步验证了一种用于矢量量化器设计的“误差函数初始化”(EFI)方法。从理论角度看其特点与量化误差准则直接相联系；从性能上看，它能比其他初始化方法使量化器的设计过程以较少的迭代次数达到较优的失真指标。从应用的角度看，它不仅是一种较好的初始化方法，而且由于它兼有快速而初始误差小的性能，因而有可能超出初始化技术的范围，而在实时的以及自适应的系统中用作量化器的快速刷新算法。

参 考 文 献

- [1] Gray R M. Vector Quantization. IEEE ASSP Mag., April 1984. 4~29
- [2] Tou J T, Gonzalez R C. Pattern Recognition Principles. Addison-Wesley Publishing Company, 1974.
- [3] Linde Y, Buzo A and Gray R M. An Algorithm for Vector Quantizer Design. IEEE Trans. Communications, 1980, Com-28, (1): 84~95
- [4] Warton Stephen W. A Generalized Histogram Clustering Scheme for Multidimensional Image Data. Pattern Recognition, 1983, 16(2): 193~199
- [5] Hilbert E E. Cluster Compression Algorithm, a Joint Cluster/Data Compression Concept. Jet Propulsion Laboratory Publication, 77~43
- [6] Sun H F and Goldberg M. Image Sequence Coding Using Vector Quantization. IEEE Proceeding of Intl. Communic. and Energy Conf., 1984. 266~269

An Approach to the Design of the Vector Quantizer for Image Signals

Yuan Guoliu

Abstract

In image data compression vector quantization has demonstrated noticeable performance. The design method for an optimized vector quantizer, however, is rather an expensive process. The paper presents a technique, which, besides being able to improve the optimizing design algorithm in the design cost and the performance index, can also serve as a fast design method for sub-optimal vector quantizers in real-time systems.

[Key words] vector quantization; quantizer; image compression; design algorithm; optimization