



BCPQ:基于重心坐标乘积量化的图像检索方法

期刊:	自动化学报
稿件 ID	AAS-CN-2017-0286
稿件类型:	论文
中文关键词:	近似最近邻, 图像检索, 乘积量化, 重心坐标, 软分配
英文关键词:	Approximate nearest neighbor, Content-based image retrieval, product quantization, barycentric coordinates, soft assignment

SCHOLARONETM
Manuscripts

BCPQ: 基于重心坐标乘积量化的图像检索方法

BCPQ : Barycentric coordinates product quantization for image retrieval

Abstract: Fast approximate nearest neighbor (ANN) search technique for high-dimensional feature indexing and retrieval is the crux of large-scale image retrieval. Product quantization is a promising technique in approximate nearest neighbor domain, which indexes high-dimensional image features by decomposing the feature space into a Cartesian product of low-dimensional subspaces and quantizing each of them separately. Despite the promising results reported, their quantization approach follows the typical hard assignment of traditional quantization methods, which may result in large quantization errors, and thus, inferior search precision. This paper puts forward an encoding scheme called barycentric coordinates based soft assignment, which to encoding the high-dimensional feature vectors into sparse representation. By minimizing their quantization errors, making the resulting representation is essentially close to the original data in practice. Experiments show that the combination of barycentric coordinates based soft assignment and product quantization improves the precision of query for ANN search on three public image datasets.

Keywords: Approximate nearest neighbor (ANN) search; image retrieval; product quantization; barycentric coordinates; soft assignment

摘要: 高维特征索引和检索的快速近似最近邻 (ANN) 查询技术是大范围图像检索的关键所在。乘积量化是近似最近邻领域一个有前景的技术, 通过把特征空间分解成低维子空间笛卡尔积的形式, 再分别进行量化来索引高维图像特征。尽管乘积量化有着显著的效果, 但其中的量化过程遵循着传统量化方法的硬匹配, 可能导致大的量化误差, 影响了查询精准性。本文提出了一种基于重心坐标的软分配编码方法, 将高维特征矢量编码成稀疏表示。通过最小化产生的量化误差, 使结果表示更接近于实际中的原始数据。实验表明, 文中提出的算法结合了重心坐标的软分配和乘积量化的优点, 在三个公开的图像数据集上的实验证明本文提出的方法有效地提高了ANN查询的精度。

关键词: 近似最近邻; 图像检索; 乘积量化; 重心坐标; 软分配;

0 引言

图像检索是多媒体应用中的一个重要技术, 例如人脸检索、目标检索、标志识别等。对于大规模图像检索任务来说, 关键技术之一是相似性查询的高效索引方法, 尤其是在高维特征空间中。相似性查询, 亦称为最近邻 (NN) 查询, 由于维数灾难, 对于高维数据而言, 准确的最近邻查询是极具挑战的问题。为了尽可能用较少的查询时间和内存损耗发现最近邻, 从而来解决高维数据的最近邻查询问题, 相关领域出现许多有关近似最近邻 (ANN) 查询方法的研究, 例如哈希、基于树方法、矢量量化等方法。

本文研究的是近似最近邻查询中的矢量量化 (VQ) 方法, 对于 ANN 查询来说是一个典型高效的数据编码方法。其基本工作原理是, 首先构建一个码本 (codebook), 将数据库

中每一个特征矢量用码本中最相似的矢量之一来表示, 典型地称其为码字 (codeword)。然后 VQ 直接使用码字的索引来表示原始数据矢量, 一般只有几位, 有效地减少了特征空间的维度。另外, 查询值和数据库中的数据矢量之间的相似性可以通过计算查询值和码本矢量间的距离来近似。这将极大地减少了计算损耗和查询时间。

通常, VQ 需要更多位数来减少量化误差。但随着编码位数增加, 码本大小也呈指数级增长, 基于 VQ 的方法对于高维数据来说作用效果不明显。为了解决这个问题, 乘积量化(PQ)^[1]给出一个高效索引高维图像特征的范例。不同于其他基于哈希的方法, PQ 采取将高维特征空间分解成低维特征子空间的笛卡尔积的形式, 然后分别进行量化。因为每个子空间维数相对较小, 使用一个小尺寸码本很容易获取到令人满意的查询效果。

尽管将长矢量分解成子部分的形式会使计算损耗明显减少, 但由于高量化误差, PQ 不能在高精度度情况下检索一个查询值的确切最近邻。目前已有一些方法来解决这个问题。Gong 和 Lazebnik^[2]提出一种迭代量化的方法, 将数据映射成二进制编码来进行快速检索。Cartesian K-means^[3]和 Optimized Product Quantization^[4]两种方法采用了旋转原始数据的方法来减小量化误差。但包括 PQ 在内的这些方法均遵循着 VQ 的基本原理, 都不可避免地造成了高量化误差。

为了解决高量化误差问题, 本文提出一种新颖的方法, 称作重心坐标乘积量化(BCPQ)。乘积量化方法可高效地压缩数据空间, 在其基础上改进编码方式, 将任意矢量量化到距离其最近的三个码字所形成单纯形的重心上, 可有效地减少编码所带来的量化误差, 并大幅提高 ANN 的查询精度。

1 相关工作

1.1 近似最近邻查询技术

最近邻查询是在学术上进行大量研究的一个热点课题, 如多媒体应用、图像分类和机器学习等。最近邻查询方法大体上被分为三类: 基于哈希的方法^[6], KD 树^[5]和矢量量化方法^{[1][4]}。

基于哈希的近似最近邻查询方法获得了大量的学术关注。大部分采用了随机映射或者基于学习的方法来形成压缩二进制编码。作为结果, 两个数据矢量间的相似性近似地用它们哈希编码的汉明距离来表示。随机映射是一个保存成对数据点间距离的有效方法。最具有代表性的例子是局部感知哈希 (LSH)^[7]。根据 Jonson Lindenstrauss 定理, LSH 需要 $O(\ln n/\epsilon^2)$ 的随机映射来保存成对距离值, 其中 ϵ 是相对误差。因此, LSH 需要使用长编码位数来增强映射表现, 但这会造成大的计算损耗和内存需求。另一方面, 基于学习的哈希方法尝试学习输入数据的结构。大部分这些算法通过使用数据关联矩阵的谱特性来形成二进制编码, 例如项与项之间的相似性。一些其他的哈希方法也使用了多模态数据或语义信息。尽管用相当短的编码可以实现很好的效果, 但这些方法随着编码长度的增加都不能获得最近邻查询效果。

研究的第二类别是用 KD 树来加速 ANN 查询。KD 树查询的平均复杂度是 $O(D \log n)$, 同时穷举查询的平均复杂度是 $O(nD)$ 。然而, 由于维数灾难, 对于高维数据来说, KD 树不如穷举查询更高效。尽管如此, 随机 KD 树^[8]和分层 K-means^[9]都提升了 KD 树的查询表现。特别地, 这两种方法都包含在 FLANN¹。FLANN 可依赖数据集自动地检索出最好的算法和最优的参数。FLANN 比起其他公开可得的 ANN 查询软件更快。然而, KD 树方法需要完全

¹ FLANN:用于在高维空间中执行快速近似最近邻查询的库。包含已经发现的最适合最近邻查询的算法集合, 以及根据数据集自动选择最佳算法和最优参数的系统。

掌握数据空间的数据结构，因此在查询阶段需要更多的内存。

研究的第三类别是矢量量化方法，尝试用码本中的码字来近似数据矢量。Jégou 等^[1]在 2011 年提出了一个高效的乘积量化 (PQ) 方法。PQ 的关键是把高维特征空间分解成一个低维特征子空间的笛卡尔积的形式，然后使用相应预定义的码本分别进行量化。之后查询相似性矢量可以用对称距离计算 (SDC) 或非对称距离计算 (ADC)。同时，应用倒排文件系统可以进行高效的非穷举查询。实验结果表明，就精确性而言，PQ 明显优于基于哈希的方法。正如在^[1]中被讨论的一样，输入数据底层结构的先验知识对于 VQ 是至关重要的。最近，Ge 等^[4]把 PQ 视作一个优化问题，通过优化查询码本和空间分解来最小量化误差。但由于 VQ^[10]固有的性质，通过硬分配编码方式，将数据矢量量化到最近的码字上，造成了高量化误差影响。

1.2 软分配

词袋 (Bag of Words) 是一个在信息检索中针对文本文件 (网页) 处理的典型模型。已成功地应用到计算机视觉领域，包括基于内容的图像检索，图像分类，精细目标分类。计算机视觉里的词袋模型把局部图像特征视作单词，因此典型地称为视觉词袋 (bag of visual words) 模型。视觉词袋模型首先匹配从图像区域中提取的图像局部描述符，然后把描述符量化到一个单一矢量上，最后根据签名矢量来检索图像。视觉词袋模型与矢量量化紧密相关，把一个描述符编码到最近邻上。然而 VQ 中的硬匹配导致较大的量化误差，降低了图像检索表现。对于这个问题的直观解决方案是设计一个软分配，让编码分布到多个邻近词上。这样一个软分配实质上减小由 VQ 固有的硬分配编码所造成的量化误差，提升了检索精度。

在^[12]中提出一种根据与描述符距离成反比的核函数来分配权重的软分配编码方案，称为核码书编码 (KCB)。基于稀疏和局部编码方案^[13,14]，提出了局部软分配 (LSA) 编码方案^[15]。这个方案使用了局部性和最大池化，比已有的稀疏或局部编码方案实现起来有更好的表现。尽管 KCB 和 LSA 都具有视觉词的不确定性，但这些视觉词的不确定性是“人为的”，因为是通过累加分子使得分母归一化产生的。在本文中提出一个新颖的基于重心坐标的软分配编码方案。其中视觉词的不确定性是“自然的”，因为它是作为区域坐标的重心坐标，一种物理上的几何属性。

综上所述，现有的乘积量化方法基于 VQ 的硬分配导致较大的量化误差，无法满足更高的检索精度。本文结合重心坐标软分配概念，结合重心坐标软分配和乘积量化方法，可以将检索精度提升到一个满意的结果。

2 模型与定义

本文提出重心坐标乘积量化 (BCPQ) 方法进行图像检索，方法框架如图 1 所示。

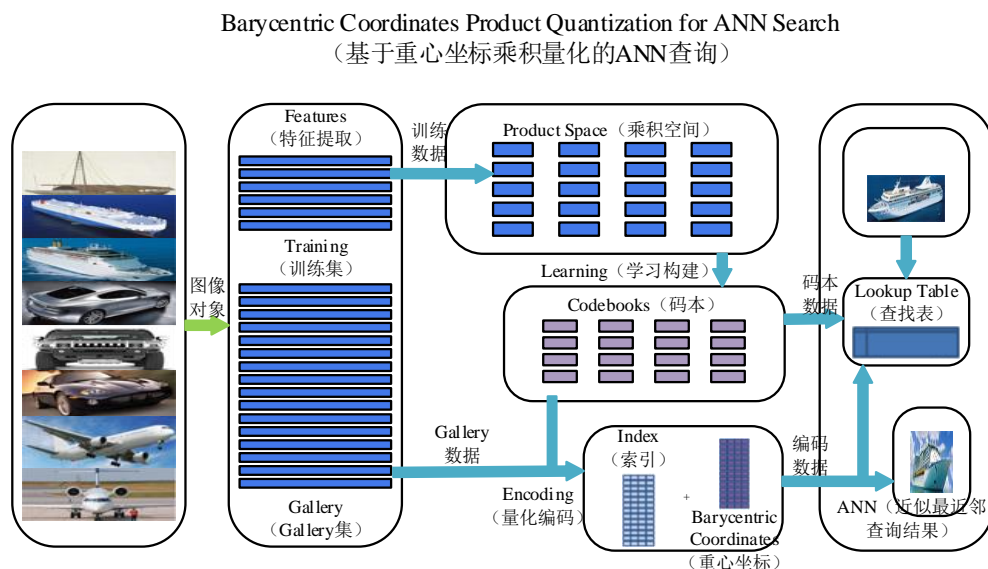


图 1 BCPQ 方法框架

图 1 给出了一个数据集，提取出图像特征，分为训练集和 gallery 集。然后把训练特征集划分成乘积空间形式来构建码本。通过码本和 Gallery 集来进行量化编码。对应编码和码本放在查询表中，给出的查询集图像根据查询表找出最相似的图像。

2.1 矢量量化

定义 1 码本 (codebook) 一个数据集分成一些组，设 x 是数据集中的 D 维矢量，即 $x \in \mathbb{R}^D$ ， c 是将数据集分组后相应组的聚心， I 是一个有穷下标集，令 $I=1 \dots k$ ， q 为一个量化器，令 $q(x) \in C = \{c_i; i \in I\}$ ， C 是长度为 k 的码本。

矢量量化 (VQ) 是数据压缩的一个经典的技术。给出一个向量 $x \in \mathbb{R}^D$ ，VQ 把 x 映射到一个预训练码本 $C = \{c_i, i = 1, 2, \dots, k\}$ 如下：

$$VQ: x \rightarrow q(x) = \arg \min_{c \in C} d(x, c). \quad (1)$$

其中 $d(\cdot)$ 是一种距离标准。在本文中使用的距离是欧氏距离： $d^2(x, c) = \|x - c\|^2$ 。

定义 2 量化误差 输入矢量 x 和它的量化值 $q(x)$ 之间的平方误差。

$$e(x) = d^2(x, q(x)) = \|x - q(x)\|^2 = \min_{c \in C} \|x - c\|^2 \quad (2)$$

称 $e(x)$ 为 x 的量化误差。

通常有两类 ANN 查询方法。一个被称作对称距离计算 (SDC)，其中查询值和数据库向量都被量化成编码。另一个被称作非对称距离计算 (ADC)，其中只有数据库向量被量化。本文采用后者作为 ANN 查询方法。

2.2 乘积量化

遵循着乘积量化 (PQ)，本文把高维空间分解成低维子空间的笛卡尔积形式，然后分别在每个子空间上执行矢量量化。具体地，一个矢量 x 可视为 m 个子矢量的串联： $x = [x^1, x^2, \dots, x^m]$ ，码本可定义为 $C = C^1 \times C^2 \times \dots \times C^m$ 。

对于 PQ，每个子矢量被映射为相应码本的一个子码字

$$x = [x^1, x^2, \dots, x^m]$$

$$PQ: x^i \rightarrow q_i(x) = \arg \min_{c \in C^i} d(x^i, c) \quad (3)$$

$$i = 1, 2, \dots, m$$

其中 q_i 是 x 的第 i 个子向量的量化器。 x 被均等地划分以便全部子向量 $x^i \in \mathbb{R}^{D/m}$, 其中 D 是 m 的倍数。注意到根据不同的码本, 每个子向量被分别编码。在这种情况下, 码本 C 中 x 的任何码字 c 将是 m 个子码字的串联: $c = [c^1, c^2, c^3, \dots, c^m]$, 其中每个 $c^i \in C^i, i \in [1, m]$ 。

令 $Q(x) = [q_1(x), q_2(x), \dots, q_m(x)]$ 标记为 x 的乘积向量。然后通过 PQ, x 的量化误差可定义为 $e(x) = d^2(x, Q(x)) = \sum_{i=1}^m \|x^i - q_i(x)\|^2$ 。

通常, 本文需要量化一组向量 $\chi = \{x_i, i = 1, \dots, n\}$ 而不是单独一个。因此 χ 的量化误差可定义为 $E(X) = \sum_{x \in \chi} e(x)$ 。

正如在等式 (3) 中, 可见 PQ 把等式 (1) 划分成 m 个子 VQ 问题, 从而分别解决。因此, PQ 方法可有效地减小特征空间维度, 压缩数据空间, 而产生 ANN 查询的精准结果。然而, 由于矢量量化^[10]固有的硬分配编码方式导致不可避免的量化误差, 限制了它查询精准性的表现。

下面将介绍一种方法, 能够高效地降低量化误差并提升查询性能。

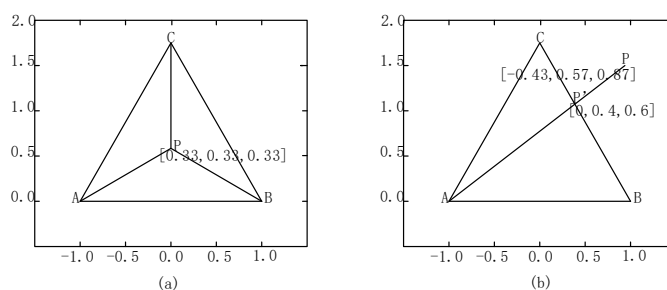


图2 重心坐标与截断

2.3 重心坐标

重心坐标首次被Mobius在1827年^[14]中提出。它们形容一个参考三角形的重心为三个质心的几何聚心。在三角形内, 重心也称为区域坐标, 因为一个三角形的内部点P的重心与这个点及其相对应外部三角形顶点形成的子三角形区域成比例 (如图2(a)中分别为 $\Delta PBC, \Delta PAC, \Delta PAB$)。

定义3 重心坐标 设 Λ 设为一个仿射空间, p_1, \dots, p_K 是 Λ 中的顶点, 可以形成一个单纯形。对于 Λ 中的某点 p , 如果

$$(u_1 + \dots + u_K)p = u_1 p_1 + \dots + u_K p_K \quad (4)$$

满足 u_1, \dots, u_K 中至少一个不等于 0, 称系数 (u_1, \dots, u_K) 为点 p 对应于仿射空间 Λ 中 p_1, \dots, p_K 的重心。

以上定义的重心坐标为同质的, 等价于一个常量。为了获取唯一的重心坐标, 采用归一化:

$$\sum_{k=1}^K u_k = 1 \quad (5)$$

式 (1) 在仿射变换下不变, 故重心坐标具有仿射不变性。

重心坐标定义中 p_1, \dots, p_K 是单纯形的顶点, 则意味着矢量 $p_2 - p_1, \dots, p_K - p_1$ 是线性独立的。

然而这个需求不是一直满足的。因此，根据单纯形定义一个普遍的重心坐标。

假设 Ω 为仿射空间 Λ 中顶点 p_1, \dots, p_K 形成的一个单纯形。如果满足以下三个属性， $u_k: \Omega \rightarrow \mathbb{R}, k = 1, \dots, K$ 可称为一个普遍的重心坐标。

$$u_k(p) \geq 0, \forall k (\text{non-negativity}) \quad (6)$$

$$\prod_{k=1}^K u_k(p) = 1 (\text{summation to unity}) \quad (7)$$

$$\sum_{k=1}^K u_k(p) p_k (\text{reproduction}) \quad (8)$$

注意非负条件(6)只在点 $p \in \Omega$ 有效。如果一个点在单纯形 Ω 外，普遍的重心坐标为负，考虑此种情况如下，采用截断的方式。

2.4 基于重心坐标的软分配

重心坐标软分配的基本想法是把一个数据矢量量化到 3 个局部最近邻形成的单纯形重心上，代替量化到一个最近邻。

令 c_1, \dots, c_K 为码本的 K 个聚心。在重心坐标软分配中让 $C = (c_{i_1}, \dots, c_{i_M}) \in \mathbb{R}^{D \times M}$ 成为一个局部描述符 $d \in \mathbb{R}^D$ 的 M 个最近邻，其中 D 为描述符的维数， u 为描述符 d 的重心坐标。根据重心坐标的定义，以及三个属性(6)(7)(8)将推导出以下等式：

$$C \cdot u = d \quad (9)$$

$$\text{Subject to } \|u\|_{l^1} = 1, u_m \geq 0, \forall m \quad (10)$$

其中等式(10)结合(9)。让

$$\tilde{C} = \begin{bmatrix} \lambda \\ C \end{bmatrix}, \tilde{d} = \begin{bmatrix} \lambda \\ d \end{bmatrix} \quad (11)$$

其中引进 λ (如: $\lambda = 10^4$)，使其归一化。可得到一个单一等式：

$$\tilde{C} \cdot u = \tilde{d} \quad (12)$$

这样原始公式(1)现在重新被公式化为：

$$\min_u \| \tilde{C} \cdot u - \tilde{d} \|^2 \quad (13)$$

$$\text{Subject to } u_m \geq 0, \forall m \quad (14)$$

以上的等式是一个非负最小二乘法问题，可通过迭代地计算一组基本矢量和相应的对偶矢量来解决[16]。然而这个计算对于大量的描述符的编码效率不高。为了解决这个问题，应用一个简单的截断技术(如图 2(b)中展示的)。重心坐标 $[-0.43, 0.57, 0.87]$ 的点 $P=(1.0, 1.5)$ 首次被截断为 $[0, 0.57, 0.87]$ ，然后用 l1 范数正则化为 $[0, 0.4, 0.6]$ ，表示近似点 $P'=(0.40, 1.05)$ 。

总之，提出的重心坐标软分配方案遵循以下步骤：

- 1) 用非负性约束解决 $\min_u \| \tilde{C} \cdot u - \tilde{d} \|^2$;
- 2) 把负数 u 元素截断为 0: $\forall m \ u_m = \max(0, u_m)$;
- 3) 用 l1 范数重新正则化 u 。

2.5 重心坐标乘积量化

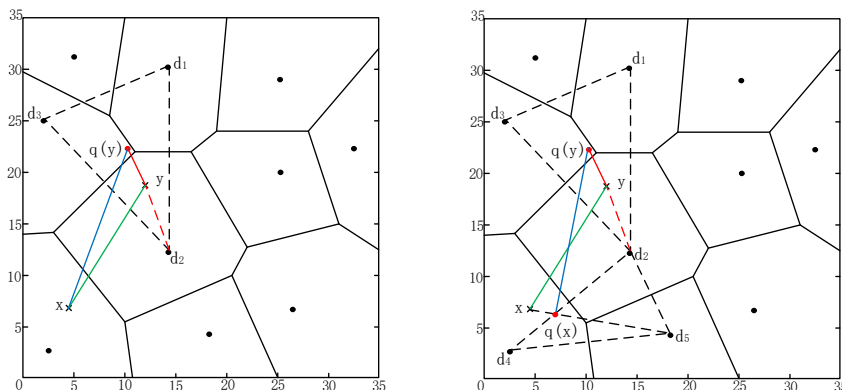


图3 重心坐标矢量量化的 2D-toy 示例。分别为 ADC 和 SDC 两种情况

图3展示了一个 2D-toy 的例子来解释本文提出方法的关键想法。选择 BCPQ 的优势在于：

性质 1 重心坐标乘积量化保留了编码的线性顺序精度

证明：

提出的 BCPQ 方法的一个特性是普遍重心坐标能够保留线性顺序精度。令 $u_k: \Omega \rightarrow \mathbb{R}, k = 1, \dots, K$ 为一个单纯形 Ω 内的重心坐标， $\Phi: \Omega \rightarrow \Omega$ 是任一线性转化：

$$\sum_{k=1}^K u_k(p) \Phi(p_i) = \Phi(p) \quad (15)$$

上述公式直接由普遍重心坐标定义而来，结合公式 (8) 和线性化：

$$\Phi(p) = \Phi\left(\sum_{k=1}^K u_k(p) p_k\right) = \sum_{k=1}^K u_k(p) \Phi(p_i) \quad (16)$$

这个属性理论上证明了提出的 BCPQ 方法对于编码线性转换具有鲁棒性。

证毕□。

在提出的重心坐标矢量量化方法中，本文表示数据库中每个矢量 x 如下： $x \approx C \cdot u$ ，其中 u 为 x 的重心坐标。受乘积量化所推动，在本文中，本文使用所提出的重心坐标乘积量化 (BCPQ) 方法。根据向量划分为子向量方法，将 x 替换成它的子向量 x^i 。因此，本文能够通过等式来近似 $x^i: x^i \approx C^i u^i, i=1, 2, \dots, m$ 。

3 算法

3.1 量化编码

重心坐标乘积量化编码阶段算法 1:

Algorithm 1 Encoding stage in barycentric

coordinates product quantization

Input:

The database $X = [x_1, \dots, x_n]$, the codebook size k ,
the subspace number s

Output: barycentric coordinates u

- 1) Sample a subset X_s of X .
 - 2) For each subspace X_s^i of X_s do
 - 3) Using the fast stochastic online algorithm
to train a codebook C^i of size k on X_s^i .
 - 4) end for
 - 5) for each subspace X^i of X do
 - 6) Solve $\min_u \|\tilde{C} \cdot u - \tilde{d}\|^2$ without the
non-negativity
 - 7) Truncate the negative u components to
0: $\forall m \ u_m = \max(0, u_m)$
 - 8) Re-normalize u with l_1 norm.
 - 9) **end for**
-

算法复杂度分析: 编码部分包括提取数据库中子集后进行的码本学习和在数据库的每个子空间下进行重心坐标的求解。让 D 表示每个特征矢量的维度, 对于一个给定的查询, 编码部分算法复杂度为 $O(nD + kD)$, 取决于数据库中的总项数 n 和码本尺寸 k 。

3.2 近似最近邻查询

本文将提到的 BCPQ 方法应用到针对大范围图像检索任务的 ANN 查询。本文提出的 BCPQ 方法的 ADC 版本的查询阶段算法 2。

特别地, 为了加快 ANN 查询, 本文使用提出的 BCPQ 方法来编码库中的全部数据矢量。然后本文计算一个查询 q 和数据库中数据的距离, 使用两种距离测量: ADC 和 SDC。

根据定义, ADC 公式化为:

$$\begin{aligned} d^2(q, x) &= \sum_{i=1}^m \|q^i - x^i\|^2 \\ &= \sum_{i=1}^m \|x^i\|^2 + \|q^i\|^2 - 2 \langle x^i, q^i \rangle \\ &= \|x\|^2 + \|q\|^2 - 2 \sum_{i=1}^m \langle C^i u^i, q^i \rangle \end{aligned} \quad (17)$$

至于 SDC 计算, 本文使用重心坐标乘积量化来近似查询矢量 $q: q^i \approx C^i \cdot v^i, i = 1, 2, \dots, m$, 相似地, SDC 可计算为:

$$\begin{aligned} d^2(q, x) &= \sum_{i=1}^m \|q^i - x^i\|^2 \\ &= \sum_{i=1}^m \|x^i\|^2 + \|q^i\|^2 - 2 \langle x^i, q^i \rangle \\ &= \|x\|^2 + \|q\|^2 - 2 \sum_{i=1}^m \langle C^i u^i, C^i v^i \rangle \end{aligned} \quad (18)$$

为了更好地解释, 图 3 展示了一个 2D 距离计算的 ADC 和 SDC 版本。

Algorithm 2 Query stage in barycentric coordinates
product quantization

Input: The query set $Q = [q_1, \dots, q_m]$ and the number
of NN p

Output: The top p ANN indexes I , the top p ANN

distances D

- 1) for each query q of Q
- 2) for each subspace q^i of q do
- 3) Precompute lookup table T^i with q^i and C^i
- 4) Using u and T^i to compute the approximate distances E^i to the database on this subspace.
- 5) end for
- 6) Sum up the approximate distances $E = \sum_i E^i$.
- 7) Search the top p NNs based on E and save them to I and D .
- 8) End for

算法复杂度分析：查询部分中输入查询集和最近邻数，为每一个查询值的每一个子空间进行 lookup 表查询，时间复杂度为 $O(ms)$ ，其中 s 为每个查询值矢量的子空间数， m 为查询集的大小。

4 实验分析

为了验证本文所提出的近似最近邻查询模型 BCPQ 的高效性，将进行一组大规模实验，验证在三个数据集上的表现，同时与几个先进的 ANN 方法进行比较。本实验拟从误差度 (distortion)、准确率 (precision)、召回率 (recall)、平均正确率 (mAP) 四个指标来评价算法的性能。本文利用 matlab 语言实现了本文的 BCPQ 算法，实现环境如表 1 所示。

表 1 实验环境

操作系统	CPU	内存	硬盘
Windows10	Core(TM)i5-4570	4.0	750GB
64 位	3.20GHz	GB	

4.1 数据集

本文采用的三个数据集均为公开可得的图像特征集 SIFT、GIST 以及 MNIST。每个数据集被划分成三个部分：训练集，gallery set 和查询集。这些数据集总结如下：

- 1) SIFT 数据集由 1,000,000 个 128 维的局部 SIFT 特征组成，其中 100,000 个样本被用来学习码本。全部 1,000,000 个样本被用来作为 gallery set，10,000 个样本被用来作为评估。其中训练集和 gallery set 间没有重合，因为前者从 Flickr 图像提取的，后者是从 INRIA Holidays 图像中提取的；
- 2) GIST 由 960 维全局特征组成。有 50,000 个样本被使用来学习码本。相似地，在数据库中的 1,000,000 个样本被视作 gallery set，1,000 个样本被用作查询评估。它们分别从 tiny 图像集，Holidays 图像集，Flicker1 M Holiday 集中提取的；
- 3) MNIST 是全部 70,000 幅 784 维手写数字的图像集。在本文的实验中，本文随机地采用 1,000 幅图像作为查询，余下的数据被视作 gallery set。为了学习码本，本文随机地从 gallery set 获取 7,000 幅；

表 2 总结了在本文实验中使用的数据集的统计数字。

dataset	SIFT	GIST	MNIST
d	128	960	784
Training	100K	10K	10K
Gallery	1M	1M	60K
Query	10K	1K	1K

4.2 对比方法

本文用以下主流的方法,包括乘积量化(PQ)、优化乘积量化(OPQ)、笛卡尔积 K-means (CK-means),以及 FLANN,与本文提出的 BCPQ 方法进行比较。

- 1) product quantization (PQ) 试图在笛卡尔积空间构建码本,被看作基线。IVFPQ 引用倒排文件结构在 PQ 上。在实验 PQ 的全部结果从原始实现中复现。
- 2) optimized product quantization (OPQ) 针对发现 PQ 的一个优化空间分解。相似地,本文采用他们自己的实现作为默认配置。
- 3) Cartesian K-means (CK-means) 是另一种来发现 PQ 最优的空间分解方式。
- 4) FLANN 是基于搜索树框架最受欢迎的开源 ANN 查询工具箱。能够自动地选取最好的算法和参数对于给定的数据集。

4.3 与基于矢量量化的其他方法相比

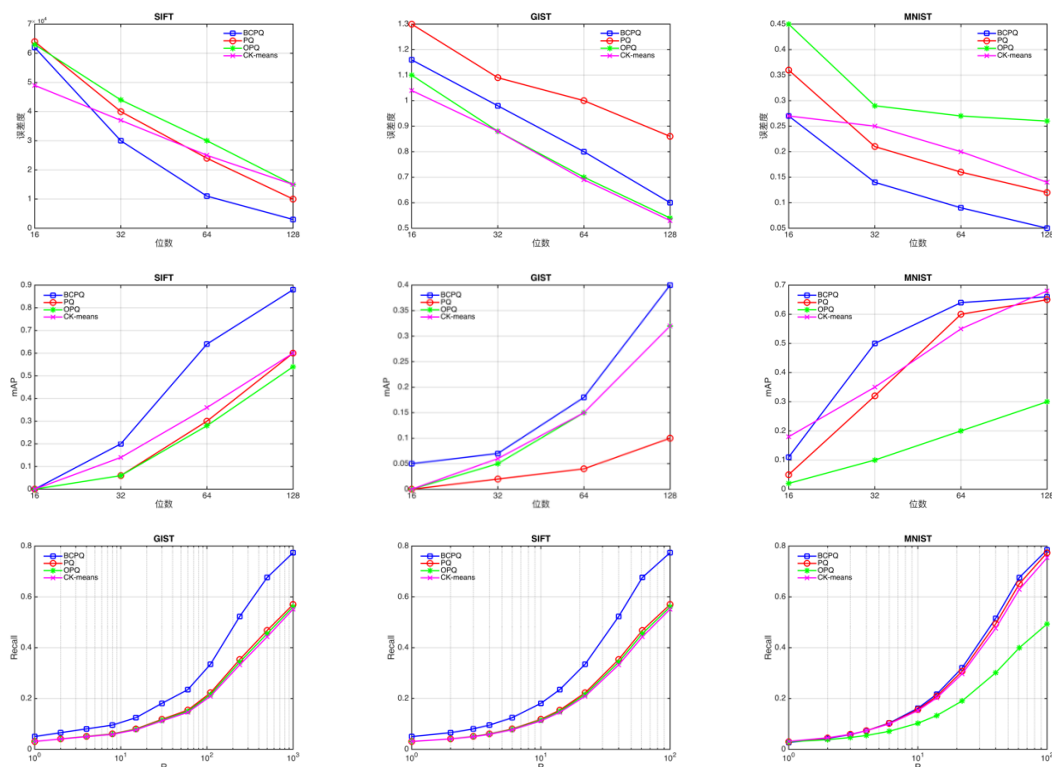


图 4 在不同数据集上测试比较。第一列到第三列分别是 SIFT, GIST 和 MNIST 数据集。第一排呈现的是误差 vs 编码位数的测试比较; 第二排呈现的是 mAP vs 编码位数的测试比较; 第三排呈现的是当发现 50 个最近邻时 R vs Recall 的测试比较。

首先本文检测在不同编码长度下不同方法的量化误差。通过图 4 可以看出, 本文提出的 BCPQ 方法在全部数据集上与其他矢量量化方法相比均有更小的量化误差。然后本文使用了 mAP 作为测量标准。结果如图 4, 可见本文的方法优于其余方法。最后本文测试了当查询 50 个真实最近邻时的召回率。同样从结果中清晰可见, 本文的方法优于其他相比较的方法。

4.4 与 FLANN 相比

本实验本文采用 SIFT 作为实验数据集, 评估了在给定查询时间下的精准率。如参照 PQ^[15], 本文采用一个倒排文件来加速 BCPQ 方法。FLANN 包括重排序阶段来计算参考最近邻的确切距离, 本文也在 BCPQ 方法里添加了重排序功能。通过实验, 本文的方法在 6s 的时间里获得了 95.4% 的准确率, FLANN 花费 6.4 秒获得了 84.2% 的准确率。图 5 展示了实验结果:

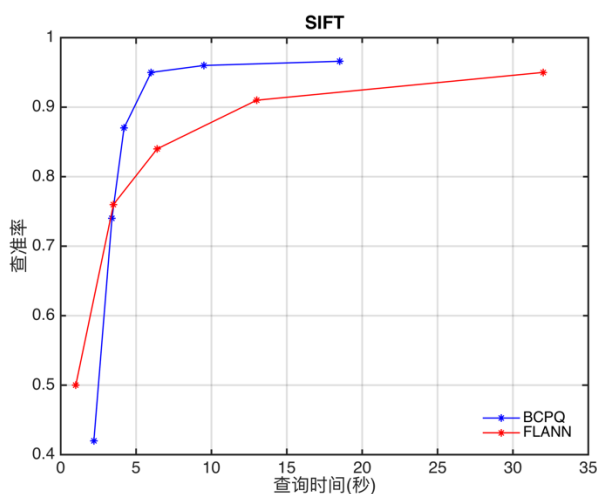


图 5 与 FLANN 相比, 查询时间 (秒) vs 查准率

5 总结

本文提出一个新颖的重心坐标乘积量化的方法把高维特征空间编码成稀疏表示。两个矢量间使用量化后的编码计算欧氏距离, 来获得最近邻。使用数据矢量稀疏表示的方式最小化了量化误差, 使结果表示更接近实际中的原始数据。本文已经在三个公开数据集上进行了大量的实验来评估提出的重心坐标乘积量化方法。实验结果表明本文的方法更快更精确。今后的研究方向, 进一步缩短内存和计算损耗的同时, 来提高查询精准度。

参考文献

- [1] Jegou H, Douze M, Schmid C. Product Quantization for Nearest Neighbor Search[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2011, 33(1):117-28.

- [2] Gong Y, Lazebnik S, Gordo A, et al. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(12):2916-29.
- [3] Norouzi M, Fleet D J. Cartesian K-Means[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2013:3017-3024.
- [4] Ge T, He K, Ke Q, et al. Optimized Product Quantization[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(4):744-755.
- [5] Sung C L, Gramacy R B, Haaland B. Potentially Predictive Variance Reducing Subsample Locations in Local Gaussian Process Regression[J]. 2016.
- [6] Zhen Y, Gao Y, Yeung D Y, et al. Spectral Multimodal Hashing and Its Application to Multimedia Retrieval[J]. IEEE Transactions on Cybernetics, 2016, 46(1):27-38.
- [7] Andoni A, Laarhoven T, Razenshteyn I, et al. Lower Bounds on Time-Space Trade-Offs for Approximate Near Neighbors[J]. 2016.
- [8] Lepetit V, Laguerre P, Fua P. Randomized Trees for Real-Time Keypoint Recognition[C]// Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005:775-781 vol. 2.
- [9] Liu C, Yuen J, Torralba A. Nonparametric Scene Parsing via Label Transfer[M]// Dense Image Correspondences for Computer Vision. Springer International Publishing, 2016:2368-82.
- [10] Vector quantization[J]. Assp Magazine IEEE, 1984, 1.
- [11] Ge T, He K, Sun J. Product Sparse Coding[C]// Computer Vision and Pattern Recognition. IEEE, 2014:939-946.
- [12] Van Gemert J C, Veenman C J, Smeulders A W M, et al. Visual word ambiguity[J]. IEEE transactions on pattern analysis and machine intelligence, 2010, 32(7): 1271-1283.
- [13] Everingham M, Eslami S M A, Gool L V, et al. The Pascal, Visual Object Classes Challenge: A Retrospective[J]. International Journal of Computer Vision, 2015, 111(1):98-136.
- [14] Oquab M, Bottou L, Laptev I, et al. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks[C]// Computer Vision and Pattern Recognition. IEEE, 2014:1717-1724.
- [15] Liu L, Wang L, Liu X. In defense of soft-assignment coding[C]// International Conference on Computer Vision. IEEE Computer Society, 2011:2486-2493.
- [16] Lawson C L, Hanson R J. Solving least squares problems[M]. Prentice-Hall, 1974.