文章编号：2095-2163(2021)12-0012-05 中图分类号：TP399 文献标志码：A

基于提前终止的近似最近邻搜索算法

王执政，杨凯祥，谭宗元
（东华大学 计算机科学与技术学院，上海 201620）

摘 要：近似最近邻搜索是数据库、图像检索等领域的一个重要问题。目前，基于图的近似最近邻搜索算法因其查询速度快、查询精度高而备受关注，该类算法致力于构建高质量的索引，但往往忽略了查询阶段的方法。本文针对基于图的搜索算法的查询阶段，结合自适应的提前终止查询与 L&C 算法，对查询阶段进行改进并验证。实验结果表明，改进后的算法与 L&C 算法相比，平均查询时间最大可减少 76%。

关键词：最近邻搜索；索引；提前终止

Approximate nearest neighbor search algorithm based on early termination

WANG Zhizheng, YANG Kaixiang, TAN Zongyuan
(School of Computer Science and Technology, Donghua University, Shanghai 201620, China)

【Abstract】 Approximate nearest neighbor search is an important issue in the fields of database and image retrieval. At present, the graph-based approximate nearest neighbor search algorithm has attracted much attention because of its fast search speed and high search accuracy. The type of algorithm is committed to constructing high-quality indexes, but often ignores the method of the search phase. Aiming at the search phase of the graph-based search algorithm, this paper combines the adaptive early termination search and L&C algorithm to improve the search phase and verify it. Experimental results show that the improved algorithm can reduce the average search time by up to 76% compared with the L&C algorithm.

【Key words】 nearest neighbor search; index; early termination

0 引 言

随着计算机技术的飞速发展，当今社会已经进入互联网大数据时代。当前的数据量巨大、数据维度之高，给信息检索带来极大挑战。最近邻搜索是信息检索的重要技术，已经在数据库、人工智能、图像搜索等领域广泛应用，其最原始的方法是对数据库中的所有点进行一一比对，找到最近的邻居，然而这种方法只适应于数据量非常小的情况，对于大数据集查询时间较长，无法满足应用需求。近似最近邻搜索成为主流，其主要思想在牺牲较小的查询精度情况下，减少查询时间、提高查询效率。基于图的搜索算法是当前近邻搜索算法中最受关注的，已经被很多大型商业公司广泛应用。

基于图的近似最近邻搜索算法使用的是近似 k 近邻图，其主要思想是将数据集的每个向量通过计算欧氏距离得到 k 个邻居，通过邻居关系建立图索引结构，并在建立的索引的基础上进行查询[1]。该算法一方面使每个点只连接几个邻居，极大的减少了索引的内存；另一方面，通过此图索引可以很快定位到查询点的邻居附近，实现快速查询。近年来，基于图的算法取得了很大的进步，该算法是目前查询速度最快，查询精度最高的近似最近邻搜索算法。

本文通过对基于图的搜索算法查询阶段的分析，将自适应的提前终止算法与 Link and Code (L&C)算法进行结合，对 L&C 算法的查询阶段进行改进。实验结果表明，与原始 L&C 结果相比，在 GIST 数据集上平均查询时间最大减少 76%。

1 相关工作

1.1 相关定义

最近邻搜索离不开距离的计算，距离越小的两个向量越相似。目前常用的度量标准有余弦距离、欧几里得距离(欧氏距离)和汉明距离等，本文采用欧氏距离作为标准，衡量两个向量之间的远近关系。

定义 1 (欧氏距离)在 n 维欧式空间 R^n 中，向量 $\boldsymbol{a}=(a_1,a_2,\cdots,a_n)$ 是其中的一个点，$a_i(i=1,2,\cdots,n)$ 为 $\boldsymbol{a}$ 的第 i 个坐标值，则欧式空间中向量 $\boldsymbol{a}=(a_1,a_2,\cdots,a_n)$、向量 $\boldsymbol{b}=(b_1,b_2,\cdots,b_n)$ 的距离 $d(\boldsymbol{a},\boldsymbol{b})$ 定义为式(1)：

作者简介：王执政(1996-)，男，硕士研究生，主要研究方向：近似最近邻搜索；杨凯祥(1992-)，男，博士研究生，主要研究方向：近似最近邻搜索；谭宗元(1992-)，男，博士研究生，主要研究方向：近似最近邻搜索。

收稿日期：2021-09-14

$$d(\boldsymbol{a},\boldsymbol{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{1}$$

定义 2　(K－最近邻搜索) $Y = \{y_1, y_2, \cdots, y_N\} \in R^d$ 表示 d 维欧式空间中的一个集合(包含 N 个向量)，该空间中的一个查询向量 $\boldsymbol{q}(\boldsymbol{q} \in R^d)$，对于一个整数值 K，满足 $K \leqslant N$，通过近似最近邻搜索算法在 Y 中搜索到 $\boldsymbol{q}$ 的 K 个邻居，根据定义 1 的距离函数 $d(q,y)$，得到 K 个最近邻的公式(2)，返回大小为 $|TopK| = K$ 的向量集合 $TopK \subseteq Y$，同时对 $\forall\ y_q \in TopK$ 和 $y_i \in Y - TopK$，总有 $d(q,y_q) \leqslant d(q,y_i)$。

$$TopK = K - \underset{y \in Y}{argmin} d(q,y) \tag{2}$$

定义 2 是目前基于图索引的搜索算法常用的结果表示，查询点 q 经过搜索返回的 K 个最近邻居的结果示意，如图 1 所示。

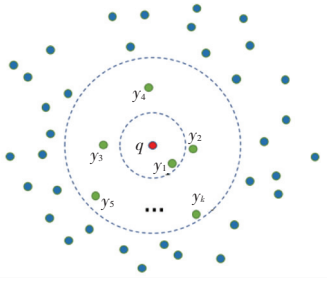

图 1　K-最近邻搜索结果示意

Fig. 1　K-nearest neighbor search result show

1.2　近似最近邻搜索算法

近似最近邻搜索算法主要分为以下几类：基于空间划分的算法、基于哈希的算法、基于量化的算法和基于图的算法。

基于空间划分的方法主要是指基于树的方法，是最近邻搜索的一类经典算法。主要思想：在高维数据的某一维度上，选择该维度的中位数进行划分，数据集上的其他向量在该维度上与此中位数进行比较，这样就可以把数据集划分为左子树和右子树，每次选择一个维度迭代递归多次进行划分，最终生成一个树索引[2]。查找时对于给定查询点，从根节点开始，最终定位到叶子节点，返回查询点的最近邻居。

基于哈希的方法的主要思想：原始空间距离较近的两个点，映射到另一个空间很可能仍然相邻[3]。该类算法将数据集通过哈希函数映射，得到多个桶，每个桶中有若干个数据点，此时建立哈希索引，查询时同样对查询点进行映射，得到映射后所在的桶，在该桶中进行欧氏距离比较，得到最近邻。

基于量化的方法最开始是信号论领域的方法，对信号进行量化处理。目前计算机领域使用较多的是乘积量化的方法，可以有效减少内存占用。乘积量化的思想主要是将高维向量进行分段，在每一段所在的子空间中聚类，在同一子空间中的向量都可以使用聚类后的质心代替，同时对质心编码，这样所有的向量都可以使用编码的质心组合表示，极大的减少了占用内存[4]。查询之前，会预先计算一个预计算表，查询点到来时，直接查表可快速得到结果，但由于量化误差，查询精度会降低。

基于图的近似最近邻搜索，使用近似 k 近邻图表示图之间的一种近邻关系，这种关系通过欧式距离确定，该算法目标是建立一个图索引结构，通过该索引结构可以很快地定位到查询点的邻居附近，从而实现快速搜索。

近年来，比较高效的算法有 HNSW 算法和 NSG 算法等，目标是建立占用内存较小、同时搜索时间较短且查询准确度较高的索引。HNSW 算法是分层可导航小世界算法，该算法采用分层的方式，每一层是一个索引图，分层结构提供了出色的性能，复杂度为对数级别[5]。查询过程从顶层开始，依次向下查找，可以快速准确定位到查询点的最近邻附近；NSG 是查询导向拓展图算法，算法从图构建的连通性、平均出度、路径长度等几个方面入手，通过综合考虑这些因素，得出一种高效的图剪枝策略，索引的复杂度大大降低，同时该算法使用数据集的近似中心作为导航节点，对于搜索带有导向作用，同时搜索从该节点开始，可实现快速查询[6]。

尽管以上两种算法查询时间快且精度高，但对于大数据集，其索引结构占用内存较大，一台内存有限的服务器无法满足实验。Link and Code(L&C)算法结合基于图的算法和量化编码两种算法的优势，在 HNSW 算法的基础上，对原始的数据库向量编码存储，极大减少了索引内存[7]。同时，该算法为了减少编码量化误差，使用一组量化回归系数改进向量近似表示的方法，并且对查询向量的候选列表重新排序，可以提高查询精度。该算法可以在一台内存有限制的服务器上实现亿级别数据集的实验。

2　自适应的提前终止查询算法

基于图索引结构的搜索算法，工作重点在于构建高质量的图索引结构进行搜索，如 HNSW、NSG 算法构建的索引，查询速度快、精度高。基于图的搜索算法有两个阶段：构建索引和查询。高质量的索引

可以提高查询效率,但查询方法同样可以影响查询效率。本文结合自适应的提前终止查询与 L&C 算法,对 L&C 索引结构进行分析,对查询阶段进行改进并通过实验验证。

2.1 问题分析

很多基于图的搜索算法在查询阶段使用固定的查询终止条件进行查询,当查询达到这个条件时查询终止。例如:HNSW 算法会设置固定的查询轮次(efSearch),该查询轮次可以认为查询走过的跳数;在倒排索引算法中,只搜索查询点的前几个聚类中心(nprobe),该做法可以减少搜索时间,同时精度损失较小,这些终止条件的大小可以根据精度的要求设置,查询终止条件越大,搜索时间越长,同时搜索精度也会越高。

但所有的查询点使用固定的查询终止条件,会出现这个固定的终止条件对于一些查询点来说过大的问题,可能较小的终止条件就能满足查询精度的需求,这就会产生很多额外的查询时间,导致最终的查询时间变长。原因在于设置的查询终止条件必须满足绝大部分查询点在查询结束时能够得到对应的精度,才能保证在最终的平均查询精度达到对应的精度值。这就需要将终止条件变大,此时一些查询点本来可以很快结束搜索,但最终还是多花费了过多的时间。

通过分析可以看出不同查询点存在不同的查询终止条件。由于 L&C 算法与 HNSW 算法有类似的索引结构,故本文对 L&C 算法的查询阶段使用提前终止算法进行改进,通过预测得到不同查询点的终止条件,提高查询的效率。

2.2 算法实现

本文在 L&C 算法索引上使用自适应的提前终止查询算法对查询阶段进行改进,通过预测提前终止查询[8]。该算法主要分为:

(1)特征的选择:本文对 L&C 图索引结构进行分析,结合索引的特点,选择一些特征。选择查询点作为一类特征,查询点的每一维都作为一个特征,查询过程中的中间搜索结果是一个重要指标,因此选择中间结果作为另一类特征。

见表 1, $\boldsymbol{F}1$ 是特征查询向量 $\boldsymbol{q}$; $F2$ 表示 $\boldsymbol{q}$ 与索引结构第 0 层的查询起始点之间的欧氏距离;$F3$ 代表 $\boldsymbol{q}$ 与经过查询后得到的最近邻之间的距离;$F4$ 表示 $\boldsymbol{q}$ 与经过查询后得到的第 10 个最近邻居之间的距离;$F5$ 是 $F3$ 与 $F2$ 的距离比值;$F6$ 是 $F4$ 与 $F2$ 的比值,其中 $F3$,$F4$,$F5$ 和 $F6$ 中间结果特征。

表 1 特征

Tab. 1 Feature

特征	描述
$\boldsymbol{F}1$: $\boldsymbol{q}$	查询向量
$F2$: d_entry	$d(\boldsymbol{q}$, 0th layer entry-point)
$F3$: d_nn	$d(\boldsymbol{q}$, nearest neighbor)
$F4$: d_10th	$d(\boldsymbol{q}$, 10th neighbor)
$F5$: nn_to_entry	$F3/F2$
$F6$: 10th_to_entry	$F4/F2$

(2)模型的选择与训练:该算法选择梯度提升决策树(GBDT)作为训练和预测的模型。选择该模型有以下几点原因:首先,训练时间短,同时在训练过程中上一次迭代产生的残差数据作为特征可以在下一次迭代中继续使用,使得模型具有一定的反馈调节能力,并且最终的模型所占空间较小;其次,该算法在训练时,对于小数据集使用该数据集的 10%进行训练,对于上亿级别的大数据集使用 100 万大小进行训练,训练集超过 100 万训练时间变长且效果相差不大。训练的输入是上述提到的特征,输出是在 L&C 索引上查询过程中的最小搜索量,查询终止条件与该值高度相关。

(3)L&C 索引上模型的整合与预测:查询的过程在 L&C 索引结构上,将训练得到的预测模型加载到该索引上,进行 L&C 索引和预测模型衔接,在搜索过程直接预测,实现查询点的提前终止查询。

3 实验

本文使用 4 个公开的数据集进行实验,分别是 ImageNet、SIFT10M、GIST 和 DEEP100M 数据集,见表 2。

表 2 数据集

Tab. 2 Datasets

数据集	数据维度	数据集大小	查询点个数
ImageNet	150	2,340,373	200
SIFT10M	128	10,000,000	10,000
GIST	960	1,000,000	1 000
DEEP100M	96	100,000,000	10,000

搜索算法得到的最近邻越准确,表明算法越好,衡量标准为召回率,其含义是查找到的精确近邻个数与实际精确近邻个数的比值,式(3):

$$recall@\,k = \frac{|\,S' \cap S\,|}{|S|} \tag{3}$$

其中,S 是查询点的精确近邻集合,S' 是算法搜索得到的近邻集合。

比较不同算法时不仅要求搜索准确，同时要求时间花费较少，可以采用固定召回率比较时间或者固定时间比较召回率两种方法。

ImageNet、SIFT10M、GIST 和 DEEP100M 4 个数据集对应的平均查询时间和召回率的实验结果如图 2~5 所示，基准算法对应的是 L&C 算法（图 2~5 中的紫色曲线，baseline），浅蓝色曲线（adaptive）是本文在 L&C 算法上使用的自适应提前终止算法的实验结果。

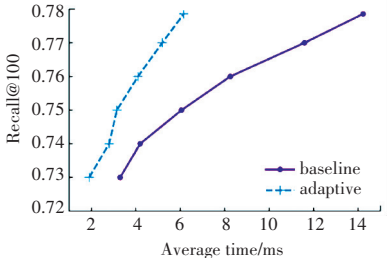

图 2　平均时间-召回率（ImageNet）

Fig. 2　Average time-recall（ImageNet）

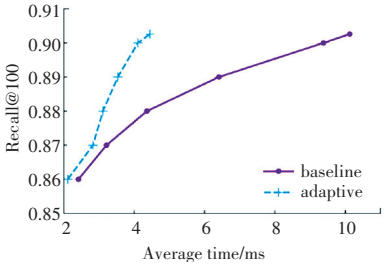

图 3　平均时间-召回率（SIFT10M）

Fig. 3　Average time-recall（SIFT10M）

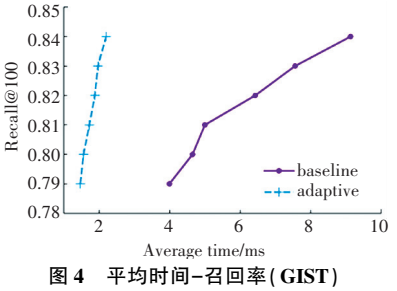

图 4　平均时间-召回率（GIST）

Fig. 4　Average time-recall（GIST）

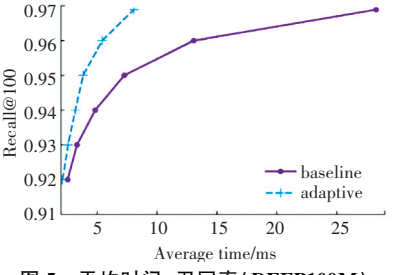

图 5　平均时间-召回率（DEEP100M）

Fig. 5　Average time-recall（DEEP100M）

在 ImageNet 数据集上，召回率为 0.73 时，基准算法查询花费时间为 3.295 ms，本文在 L&C 上的提前终止查询花费时间为 1.901 ms，减少了 42%；而当召回率为 0.778 6 时，基准算法花费时间为 14.226 ms，本文在 L&C 上提前终止查询时间为 6.155 ms，减少了 57%，在召回率不断变大的过程中，减少的时间百分比也在不断变大。在 SIFT10M 数据集上，召回率为 0.86 和 0.87 时，两种算法的差距不大，但随着召回率的变大，本文在 L&C 上提前终止查询的算法优势越来越明显，最大减少 56%。在 GIST 数据集上，本文的实验效果较好，在召回率较低为 0.79 时，减少 63%，在召回率为 0.84 时，减少百分比最大为 76%，在每一个召回率上平均时间减少百分比都比较大，整体效果比较好。在 DEEP100M 数据集上，0.92 和 0.93 的召回率时，本文的实验结果优势不大，但随着召回率的变大，优势越来越明显，召回率为 0.968 9 时，平均时间减少 71%。

从上述实验结果分析中，可以看出本文在 L&C 索引上的提前终止查询算法优于原始的 L&C 算法。本文在 L&C 索引上实验的优势在高召回率上，召回率越高实验效果提升越大，随着召回率升高，两种算法平均查询时间的差距越来越大，提升效果越来越好。

4　结束语

目前基于图的最近邻搜索算法均使用固定的终止条件，但对于一些查询点来说，小于这个固定的终止条件就能满足查询精度的需求，此时使用这个固定的终止条件会产生很多额外的查询时间。为了减少额外的查询时间，本文在 L&C 索引结构上使用了自适应的提前终止查询算法，对查询阶段的查询方法进行改进，同时在 ImageNet、SIFT10M、GIST 和 DEEP100M 4 个数据集上实验验证。实验结果表明，改进后的效果明显优于 L&C 算法。

参考文献

[1] PAREDES R, CHÁVEZ E. Using the k-nearest neighbor graph for proximity searching in metric spaces [C]//International Symposium on String Processing and Information Retrieval. Springer, Berlin, Heidelberg, 2005: 127-138.

[2] BENTLEY J L. Multidimensional binary search trees used for associative searching[J]. Communications of the ACM, 1975, 18(9): 509-517.

[3] GIONIS A, INDYK P, MOTWANI R. Similarity search in high dimensions via hashing[C]//Vldb. 1999, 99(6): 518-529.

[4] HERVÉ JÉGOU, DOUZE M , SCHMID C. Product Quantization for Nearest Neighbor Search [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2010, 33(1):117-128.

（下转第 21 页）