doi:10.3969/j.issn.1001-4616.2024.02.012

基于均衡聚类索引的近似最近邻检索方法

吕宏伟,李　博,刘普凡,刘　识,李继伟,刘俊健

(国家电网大数据中心,江苏 南京 210023)

[**摘要**]　大数据时代,深度学习通过将复杂对象表示为高维特征向量,并使用向量之间的距离度量来衡量样本的相似性,在推荐系统、用户画像、数据中台管理等场景中得到了广泛的应用. 但是,随着数据规模的不断增加,海量特征数据的相似向量检索面临着检索模型占用内容大、特征检索算法召回率较低的严重挑战. 如何在保证检索精度的前提下,设计紧凑型索引图结构,降低特征检索的内存消耗,对于提升大数据系统的近邻检索效率具有重要的作用. 因此,本文提出了一种均衡感知的快速 K 均值近邻聚类的特征数据分桶及其图结构紧凑型索引用于海量数据近邻检索. 首先,设计了均衡感知的快速 K-均值聚类算法,通过在图索引构建过程中海量特征数据的均衡分桶,将高维向量压缩成轻量级紧凑型图索引结构,随后通过量化操作进一步压缩高维向量样本,提升其在候选集上的最近邻检索速度. 在基准数据集上实验验证结果表明,本文提出的方法能够在保证较高检测召回率的同时,有效加快索引构建速度,可以用于支持高维特征数据的高效最近邻检索.

[**关键词**]　大数据检索与分析,最近邻搜索,均衡感知

[**中图分类号**]TP391　[**文献标志码**]A　[**文章编号**]1001-4616(2024)02-0099-10

Balanced Clustering-based Index for Approximate Nearest Neighbor Retrieval

Lü Hongwei,Li Bo,Liu Pufan,Liu Shi,Li Jiwei,Liu Junjian

(Big Data Center of StateGrid Corporation of China,Nanjing 210023,China)

Abstract:In the era of big data,deep learning has been widely applied in recommendation systems,user profiling,and data management by representing complex objects as high-dimensional feature vectors and evaluating their similarities based on vector distance measurements. However,with the continuous growth of data scale,the retrieval of similar feature vectors from massive data faces significant challenges such as large memory consumption of retrieval models and low recall rates of feature retrieval algorithms. It is crucial to design compact index graph structures and reduce memory consumption in feature retrieval to improve the efficiency of nearest neighbor search in large-scale data systems while ensuring retrieval accuracy. Therefore,a balanced-aware distributed K-means clustering-based user feature binning approach and a compact index design algorithm for graph structures are proposed. Firstly,fast balanced-aware K-means clustering algorithm is designed to achieve balanced binning of massive feature data during graph index construction,compressing high-dimensional vectors into lightweight and compact graph index structures. Subsequently,quantization operation is conducted to further compress high-dimensional vectors sample and improve its nearest neighbor search speed in dataset. Experimental results on benchmark datasets demonstrate that the proposed method can effectively accelerate index construction speed while ensuring high accuracy,thus enabling efficient indexing and retrieval of massive data.

Key words:big data retrieval and analysis,nearest neighbor search,balanced perception

随着大数据时代的到来,如何有效地提取数据中的特征成为一个重要问题[1]. 深度学习技术的出现为特征提取提供了全新的方法. 深度学习将复杂的对象如文本[2]、语音[3]和图像[4]编码为高维向量,然后利用向量之间的距离来度量对象之间的相似性. 这种方法在众多领域得到了广泛应用. 例如,在构建搜索推荐系统时,常常需要在大规模、高维度的向量上进行相似性特征检索[5]. 这种方法不仅可以快速准确地

收稿日期:2023-08-29.

基金项目:国家电网有限公司大数据中心自建科技项目(SGSJ0000SJJS2310021).

通讯作者:刘识,硕士,高级工程师,研究方向:最近邻算法大数据检索、智能推荐. E-mail:yetfly092@163.com

找到具有相似性的对象,还可以帮助用户迅速找到感兴趣的内容.

然而,随着数据量的爆发式增长,现有系统在面对海量高维特征相似性检索时面临着显著的时效性和精确性挑战. 例如,海量数据的特征索引构建耗时越来越长,索引的动态更新对计算资源的占用越来越多. 同时,随着新特征的不断增加,如何有效保持特征分桶过程中的均衡性,以及如何平衡相似性检索的精确度和效率等问题变得越发重要. 因此,研究构建性能高、速度快、资源占用少,同时保证准确率的特征检索方法成为一个重要的问题.

由于在大数据应用中,针对特征的检索通常不需要严格保证的理论最优值,即精确的最相似特征. 因此,随着近些年来大数据技术的发展,近似 K 最近邻搜索方法被提出并广泛应用于图片搜索、人脸识别系统和推荐系统等领域. 根据算法基本思想的不同,近似最近邻搜索算法可以分为四个类别:基于局部敏感哈希方法(locality sensitive hashing,LSH)[6]、基于树形结构划分的方法[7]、基于矢量量化(vector quantization,VQ)的方法[8-9]和基于图的方法. 基于图的方法由于在召回率和召回时间方面的优势,成为目前最主流的近似最近邻搜索算法. Fu 等[10]提出了一种单调相对邻域图的新型图结构以降低搜索复杂度. Subramanya 等[11]通过引入参数优化裁边策略,提高搜索精度. Malkov 等[12]提出了一种基于可控层次结构的可导航小世界图(hierarchical navigable small world,HNSW)的近似 K 最近邻搜索新方法,联合利用尺度分离和启发式算法显著提升了搜索性能. 然而,上述方法仍然存在召回率低以及检索模型占用内容较大的问题. 有学者从压缩原始高维向量和采用更轻量级的架构组织数据两个方向入手减少索引模型的大小[13],但仍无法满足大数据场景的实际需求和规模数据的高效最近邻检索.

因此,本文提出了一种均衡感知的快速 K 均值近邻聚类的用户特征分桶及其紧凑型索引设计算法. 首先,设计了基于快速均衡感知的 K-均值聚类算法,该方法可以在索引构建过程中均衡地分桶海量特征数据,进而实现将高维向量压缩为轻量级紧凑的数据结构的作用. 在基准数据集上的实验结果表明,本文提出的方法能够在保证较高精度的同时,有效加快索引构建速度,支持海量数据的高效索引和检索. 总体来说,本文创新点包括:

(1)提出了一种均衡感知的快速 K 均值近邻聚类算法. 通过在索引构建过程中对海量特征数据进行均衡分桶,实现了高维向量的压缩,并得到轻量级紧凑的数据结构.

(2)设计了紧凑型索引图结构. 在特征分桶的基础上设计了一种紧凑的索引图构建算法,本文设计的高效的图索引构建算法和线量化压缩算法,可以用于高效的基于图搜索的特征检索.

通过在基准数据集上的实验,表明了本文方法在海量数据紧邻检索的过程中能够保证较高精度的同时,有效加快索引构建速度,支持海量数据的高效索引与检索.

1 相关工作

近邻检索作为大数据分析与挖掘领域中重要方法,受到了国内外学者的广泛关注. 针对近似最近邻检索问题,目前已经提出了很多算法. 主要可分为以下四种:矢量量化、基于树的、基于哈希的和基于图的检索方法. 接下来分别从非图类检索方法和基于图的检索方法两个方面介绍国内外相关研究工作.

1.1 非图类的近似最近邻检索方法

矢量量化、树结构和局部敏感哈希是常见的用于数据压缩和近似最近邻检索的算法. 然而,它们在处理高维数据和大规模数据时存在一些限制和挑战.

矢量量化方法通过使用部分向量代替全体向量来实现数据压缩,其中 k- means 算法及其变体[14-15]是常见的实现方式. 这种方法通常与倒排索引集合结合使用,用于近似最近邻检索[7]. 乘积量化(Product Quantization,PQ)及其改进算法是最典型的矢量量化方法[8,16-17]. 树结构算法提供了一种在多个尺度上划分数据集并用于检索的自然方法. kd 树[18]是一种常见的树结构算法,但在数据分布不平衡的情况下性能较差. 因此,出现了改进算法如 M 树[19]和 R 树[20]. 然而,这些算法在处理高维度数据时性能不佳. 局部敏感哈希是一种针对近似最近邻检索问题的哈希算法[21-22]. 它通过构建多个哈希表并将向量映射到较低维度的哈希码来实现高效的比较. 然而,内存限制了可以使用的哈希表数量,并且设计高效且合适的哈希函数往往是比较困难的[23].

综上所述,这些方法都是基于子空间划分的算法,需要首先定位可能出现最近邻的候选区域. 然而,

在高维空间中,候选区域的数量呈指数级增长,使得这些方法无法满足处理大规模高维数据的实际需求,无法实现高效的索引和检索.

1.2 基于图的近似最近邻检索方法

基于图的近似最近邻检索方法通过建模数据之间的邻近关系,以逐步搜索邻近点的方式来实现近似最近邻的检索. 研究显示,这类方法在检索精度和效率方面展现出良好的性能[24]. 在近似最近邻检索的问题中,需将图转化为邻近图[25],例如相对领域图(relative neighborhood graph,RNG)[26]、Delaunay 图[27]等. 然而,这些图在检索效率和精度方面存在一定的限制,因此引入了一系列改进算法,如 HNSW[12]、IVF-HNSW[13]、GNNS[28]等. HNSW 采用多层图索引结构和基于启发式的邻居选择策略,能够降低远距离节点的选取概率,从而有效缩短搜索路径. 该方法在近似最近邻领域广受应用且具备优越性能. 然而,随着数据规模的增大,索引模型的内存占用问题依然存在. 后续的改进算法结合了倒排索引和量化方法,对索引大小进行了压缩,但其在大规模数据上的索引构建速度相对较慢. 因此,在保持检索精度的前提下,如何加速索引构建速度仍然是近似最近邻算法在实际应用中面临的重大挑战.

2 LB-ANNR 近邻检索算法

本章将首先介绍基于均衡聚类索引压缩的海量数据近邻检索算法(LB-ANNR)的总体流程,随后依次介绍均衡聚类算法与索引压缩构建过程、分段聚类量化及最近邻检索过程.

2.1 算法图的近似最近邻检索方法

如图 1 所示,本文设计的基于均衡聚类索引压缩的海量数据近邻检索算法(LB-ANNR)首先利用快速均衡 K-均值聚类算法获取数据样本的聚类中心集合;随后,使用这些聚类中心集合构建查找图,并基于数据的倒排索引和查找图的图索引,构建了一种紧凑型双层索引结构. 由于本文设计的快速均衡 K-均值算法能够有效保持数据聚类过程中每个聚类簇的样本数量均衡性,因此双层索引的构建具有结构紧凑、占用内存小等优点. 同时,采用基于分段聚类的乘积量化方法对原始数据进行压缩. 通过量化压缩,可以降低数据存储开销和距离计算的复杂度. 在快速检索过程中,使用双层索引结构查找候选最近邻,然后通过量化后的向量计算距离,得到候选向量列表. 最后,进行最近邻的重新排序以获得输出结果.

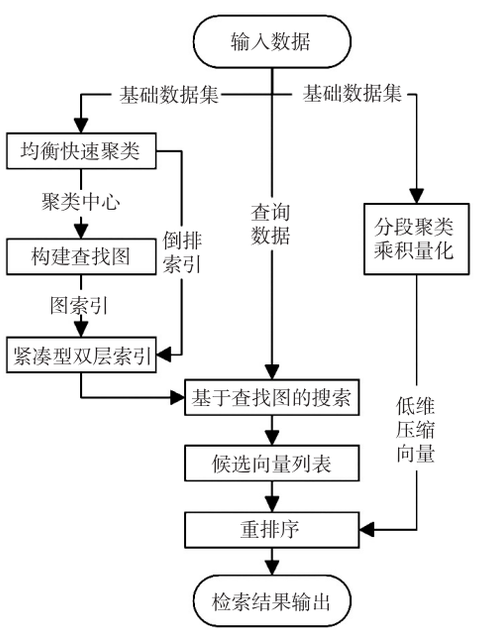


图 1 LB-ANNR 近邻检索算法总体流程图

Fig. 1 The workflow of load balanced approximate nearest neighbor retrieval(LB-ANNR) algorithm

2.2 快速均衡聚类的索引构建

在构建索引的过程中,首先需要使用聚类算法对数据进行分桶,随后用得到的聚类中心构建出倒排索引和查找图,进而由倒排索引和图索引共同构建出紧凑型双层索引,用于下游的数据最近邻快速检索.

然而,在面对海量数据的情况下,传统的 K-均值聚类方法不再切实可行,其速度不仅无法满足要求,而且无法保证不同桶之间数据的均衡性. 在实际大数据场景中,常常会遇到样本特征分布呈现局部密集型的情况,这导致了部分聚类中心样本量过大,从而引发了不同桶之间数据检索时间的显著差异. 这不均衡的情况进一步导致了系统检索请求响应时间的不均匀性. 此外,由于候选列表的最大数量受限,这种聚类不均衡问题还可能导致召回率的下降.

因此本文基于二分 K-Means 提出了一种改进的均衡快速聚类算法,在提升聚类速度的同时,保持每个聚类簇的样本的均衡性. 给定用于构建索引的所有样本,B 表示所有桶的集合,所有桶的数量为 $|B|$,我们用符号 $|\cdot|$ 来表示对应集合中的样本的数量. 对于每一个桶 B_i,我们定义桶 i 的负载系数为:

$$\theta_i = \frac{|B_i|}{\frac{|B|}{\sum_{i=1}|B_i|}{|B|}}, \tag{1}$$

式中,$|B_i|$代表桶 i 中的样本数量,因此 θ_i 代表桶 i 中的实际数据样本数量与所有桶的数据平均量的比值. 数值越接近 1 代表越好. 在聚类过程中,我们使用 Φ 表示负载系数的上限,在实际场景中根据具体应用场景的系统均衡性需求情况进行设置,在本文中设置为 1.05. 基于此,改进型快速均衡聚类算法的流程如下:

算法 1　改进型快速均衡聚类算法

输入:初始数据样本 S,负载系数 Φ,$|B|$

输出:聚类中心点的集合 C

1. 初始化:从 S 中随机选择两个点 C_i 和 C_j 作为初始聚类中心;i←1;j←1;
2. 随机选择一个未聚类的样本点 p,分别计算 p 与 C_i 和 C_j 的距离 $d(p,C_i)$ 和 $d(p,C_j)$;
3. IF $d(p,C_i)<d(p,C_j)$,则 p 更接近 C_i:
4. PROCDURE(i):

IF($\varphi_i \leqslant \Phi$),更新聚类中心 C_i:

$$C_i = \frac{C_i * i + p}{i+1}$$

$$i \leftarrow i+1$$

ELSE:PROCDURE(j):

5. IF $d(p,C_i)>d(p,C_j)$,则 p 更接近 C_j,
6. PROCDURE(j):

IF($\varphi_j \leqslant \Phi$)更新聚类中心 C_j:

$$C_j = \frac{C_j * j + p}{j+1}$$

$$j \leftarrow j+1$$

ELSE:PROCDURE(i):

7. 重复步骤(2)-(7)直至遍历所有样本,利用 C_i 和 C_j 分隔的超平面对数据进行归类;
8. 如果聚类中心的数量$|C|<|B|$,依次分别对 C_i 和 C_j 簇内重复步骤(1)-(7),再次对大聚类的样本簇进行分割;
9. 返回聚类中心集合 C.

随后,基于得到的聚类中心集合,构建基于 KNN 的图索引. 在海量数据情况下,当聚类中心集合 C 也较大时,直接精确构建 KNN 索引图计算耗时较大,因此本文使用基于 NN-descent 的近似 KNN 方法构建索引图(图 2).

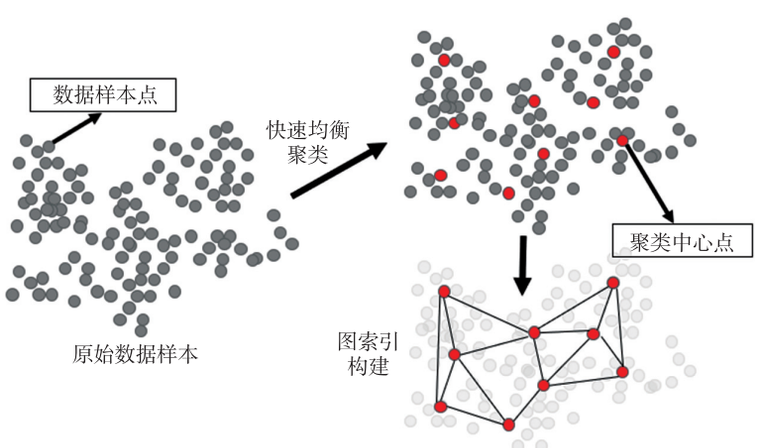


图 2　基于快速均衡聚类的图索引构建

Fig. 2　Construction of graph index based on fast balanced clustering

给定聚类算法返回的聚类中心集合 C,首先计算聚类中心点集合的质心:

$$c=\frac{1}{|C|}\sum_{x\in C}x, \tag{2}$$

式中,x 代表属于聚类中心集合 C 中的所有的中心点. 此时,选择离中心点 c 最近的聚类中心点作为起始点,依次在图上执行搜索. 由于固定开始点之后,只需要确保图中的任一点可达,即可满足最近邻搜索过程中所有点均可访问的要求,无需索引图完全连通,从而可以降低索引的大小. 具体操作是,针对每一个点 x,它来自聚类中心集合 C,我们需要计算其与最近的 k 个点的距离,然后按照距离升序排序这些点,并创建新的边将点 x 与最近的 k 个点相连接. 由于聚类中心的数量远远少于总样本数量,因此这个步骤大大降低了计算量.

索引图构建完成后,需要确保图中的任何起始点都是连通的. 为了实现这一目标,我们使用广度优先搜索算法在图中查找与起始点相连的节点. 如果发现某些点 y 不连通,运用贪婪搜索算法找出距离点 y 最近的 k 个点,并建立连接,然后循环执行这个过程,直到广度搜索可以遍历所有的聚类中心点,确保索引图中的连通性. 在最近邻检索过程中,从起始点开始,在索引图上进行搜索,与构建过程相反. 由于索引图无需满足强连通性的要求,因此可以减小索引的大小,降低内存消耗. 这个索引构建和检索过程的设计考虑到了海量数据处理的需求,以提高检索效率和减小资源消耗.

2.3 特性数据线量化

针对高维特征向量,通过线量化压缩,将高维数据映射至聚类中心所在空间上的投影点,通过计算投影点与聚类中心点之间的距离来近似表示原始高维特征的样本点与聚类中心点之间的距离. 通常采用的量化算法不同,产生的距离计算误差和压缩效率也不同. 本文使用基于乘积量化树(PQT)的方法对高维特征数据进行线量化操作,相比与点量化,基于 PQT 的量化方法误差更小.

具体来说,对于给定样本点 p,用于量化的两个聚类中心分别为 c_i 和 c_j,$\varphi(p)$ 为样本点 p 在聚类中心 c_i 和 c_j 连线上的投影. 用 λ 表示 $\varphi(p)$ 离聚类中心点 c_i 的距离占 $\varphi(p)$ 与 c_i 和 c_j 距离之和的比值,则 $\lambda<0.5$ 代表其离 c_i 更近,反之则代表离 c_j 更近. 令函数 $d(\cdot)$ 代表样本点之间的空间距离,由空间三角关系可以得出:

$$\mathrm{d}(p,c_j)=\sqrt{a^2+b^2-2a\cos\frac{\lambda b}{a}}, \tag{3}$$

式中,$a=\mathrm{d}(p,c_j)$,$b=\mathrm{d}(c_i,c_j)$,即样本点至对应聚类中心的距离可以由三角关系定理进行计算,由此可以推出:

$$\lambda=-\frac{\mathrm{d}(p,c_j)^2-\mathrm{d}(p,c_i)^2-\mathrm{d}(c_i,c_j)^2}{2\mathrm{d}(c_i,c_j)^2}. \tag{4}$$

同时,由上文中定义知:

$$\lambda=\frac{\mathrm{d}(\varphi(p),c_i)}{\mathrm{d}(\varphi(p),c_i)+\mathrm{d}(\varphi(p),c_j)}. \tag{5}$$

则通过使用(λ,c_i,c_j)即可表示出样本 p 量化压缩后的空间向量 $\varphi(p)$. 再次运用空间三角关系定理,可以推出:

$$\mathrm{d}(p,\varphi(p))=\sqrt{\mathrm{d}(\varphi(p),c_i)^2+\lambda^2\mathrm{d}(c_i,c_j)^2+\lambda D}, \tag{6}$$

$$D=\mathrm{d}(\varphi(p),c_j)^2-\mathrm{d}(\varphi(p),c_i)^2-\mathrm{d}(c_i,c_j)^2. \tag{7}$$

由公式(6)、(7)可以得出,计算 $\mathrm{d}(p,\varphi(p))$ 只依赖于三个数值(c_i,c_j,λ),无需保留原始样本的高维空间向量,因此实现了对原始数据样本的线量化操作,压缩样本数据大小的同时提升了最近邻算法计算的效率.

2.4 基于图索引的检索

完成图索引和线量化操作之后,即可在建立好的索引图上进行最近邻检索. 如上文所述,检索的过程与图索引构建的过程是反向的. 在索引构建的过程中,我们首先对原始样本进行均衡快速聚类,随后基于聚类中心构建出紧凑型图索引. 而在检索的过程时,首先是在索引图上进行检索,随后利用均衡聚类的聚

类结果进行倒排查找. 对于一个查询样本 q,最近邻检索过程的步骤如下:

(1)首先,从索引图的某个起始点开始,通常是选择与查询样本 q 最近的聚类中心点作为起始点. 利用图上的边和距离信息,我们进行广度优先搜索或其他适当的搜索算法,以找到与查询样本 q 最接近的点集合. 这一步骤在索引图上进行,其目标是迅速确定一组潜在的最近邻候选点.

(2)接下来,利用均衡聚类的聚类结果,执行倒排查找. 这些聚类中心具有代表性,可以减少搜索空间. 对于每个聚类中心,我们使用倒排索引查找与查询样本 q 最接近的点,通常是根据距离升序排序. 选择距离最近的 k 个点,这些点将成为最终的最近邻候选集合.

(3)使用本文 3.3 节的乘积线量化方法,依次计算查询样本 q 与候选集合 Q 中的每一个样本的近似距离,随后对 Q 中的样本按照距离升序排序,取出距离最小的 k 个样本即为近似最近邻检索的输出.

由此,经过上述三个步骤,即可根据给定的查询向量 p,输出其距离最近的 k 个近邻. 在第(1)步中通过使用紧凑型图索引进行快速检索,提升了查找最可能近邻聚类中心的速度,同时降低了索引文件的大小,同时由于使用了本文设计的均衡快速聚类,使得每个聚类中心包含的样本点较为接近,降低了系统检索的速度差异. 随后通过第(2)步在近邻聚类中心的样本中进行检索,大幅降低了检索候选集的大小. 最后通过第(3)步线量化距离计算,进一步提升检索效率. 由此在面向海量高维特征样本时,本文设计的方法能够在保证精度的前提下实现快速近邻检索.

3 实验与分析

3.1 实验设置与数据集

本文实验数据集来源于 SIFT[29] 和 GIST[30] 两个基准特征数据. 其中,SIFT1M 数据集包括 100 万个 SIFT 特征样本,其数据维度为 128 维,数值类型为浮点类型. GIST1M 数据集也是 100 万个全局特征数据,每个样本的数据维度为 960 维,也是浮点数类型. 在实验中,测试机器环境的 CPU 为 Intel E5-2630,512 G 内存,操作系统为 CentOS 7.5,编译器使用 g++5.4. 本文实验的对比方法主要包括:

kGraph:一种经典的基于 kNN 检索图的近邻搜索方法. 基于 kNN 生成的图使用图搜索算法进行近邻近似检索.

NSG:一种搜索起始点固定的图检索算法. 也是通过 kNN 来构建图索引.

Efanna:一种基于随机树的检索算法,使用近似 kNN 图进行最近邻近似检索.

HNSW:一种广泛使用的图检索算法,通过构建多层搜索图索引,依次逐层进行近邻检索.

LB-ANNR:本文提出的基于均衡聚类图索引压缩的最近邻搜索算法.

在精度评估指标方面,本文使用召回率 R 来评价,召回率表示方法查询的最近邻结果在真实最近邻中所占的比例,是检索算法准确性评估基准指标. 其定义为,假设算法返回的 k 个最近邻结果集为 K',真实的 k 个最近邻结果集为 K,则召回率的计算公式为:

$$R=\frac{|K'\cap K|}{|K|} \tag{8}$$

在本文实验中,针对一组查询,召回率可表示为所有查询结果召回率的平均值. 同时,针对前 k 个结果输出的精准率,我们还使用 $R@k$ 来表示前 k 个检索结果的准确率.

3.2 近似检索精度和效率对比

在检索精度和检索时间实验中,为了对比不同算法的性能差异,所有方法在两个数据集中都返回 10 个检索结果. 对于 kGraph 方法,检索候选队列设置为 100,构建 50 个近邻图. NSG 算法中,我们设置图的最大出度为 50,候选结果集大小为 100. 对于 Efanna 方法,候选队列大小也设置为 100,构建近邻图为 50. 对于 HNSW 算法构件图的最大出度设置为 32,检索时候选列表为 100. 针对本文提出的 LB-ANNR,我们设置最大图出度为 32,检索候选列表大小为 100.

表 1 展示了上述不同方法在两个基准数据集上的对比结果. 可以看出,在 SHIFT1M 数据集中,虽然 kGraph 检索耗时最少,但是其无法保证检索精度,Recall 值是所有算法中最低的. 相比于 HNSW 和 NSG 算法,本文提出的 LB-ANNR 在取得相近的召回率的同时,检索查询时间显著减少,从 0.3 ms 左右大幅减低至 0.21 ms. 体现了本文设计的方法在保证检索精度的同时,可以提升检索效率. 在 GIST1M 的数据集中,

可以看出,kGraph 的召回率只有 76.5%,显著低于其他方法. 在所有召回率高于 80%的算法中,本文设计的方法检索时间是最少的. 进一步体现了 LB-ANNR 算法的优越性. 相比与精度最高的基准方法 NSG,LB-ANNR 算法取得了高于其召回率的同时,减少了查询时间,证明了其平衡聚类算法在图索引时能够在保持精准度的同时提升效率的优越性能表现.

表 1　各个算法检索精度与效率在不同数据集中的对比

Table 1　Comparison of retrieval accuracy and efficiency of each algorithm in different datasets

Algorithm	SIFT1M		GIST1M	
	Recall/%	Time/ms	Recall/%	Time/ms
KGraph	0.946	0.12	0.765	0.85
NSG	0.994	0.31	0.897	1.82
Efanna	0.990	0.23	0.865	2.01
HNSW	0.991	0.30	0.851	1.95
LB-ANNR	0.993	0.21	0.903	1.72

3.3　检索构建效率和内存占用

随后,本文开展了索引构建的效率测试及消耗的时间测试. 实验设置同 3.2 节相同,各个算法的结果如表 2 所示.

表 2　各个算法索引大小与构建时间对比

Table 2　Comparison of the size and construction time of retrieval index of each algorithm in different datasets

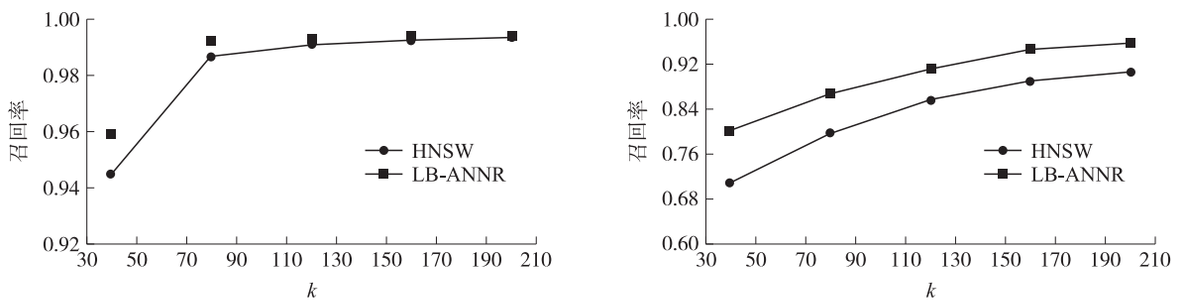
Algorithm	SIFT1M		GIST1M	
	Index Size/MB	Time/s	Index Size/MB	Time/s
KGraph	398	156	398	1 235
NSG	120	329	87	3 084
Efanna	444	148	451	392
HNSW	141	115	141	551
LB-ANNR	76	53	58	372

由于 KGraph 的 kNN 图设置没有变化,因此其索引文件大小在两个数据集中相同. Efanna 的索引大小为其树结构和图结构索引的大小之和. 可以看出,在两个数据集中,本文提出的方法索引文件大小均是最小的,比最好的基本方法 NSG 在索引大小内存占用上降低了约 40%,体现了 LB-ANNR 紧凑型索引结构在内存消耗方面的优势. 更小的索引结构带来的索引构建时间消耗优势更加明显. 例如在 SIFT1M 数据集中,本文提出的 LB-ANNR 比最好的基准方法时间消耗方面降低了 50%以上. 总体来说,本文设计的方法具有更小的索引结构,内存消耗更低,同时构建时间也更短.

3.4　不同候选集大小的精度测试

本文进行了召回率(R@k)随着前 k 个检索结果变化的实验,在两个基准数据集中进行观察,实验结果如图 3 所示.

在实验中,我们选择了前 40、80、120、160 和 200 个候选集进行测试,并观察它们对召回率的影响. 较高的召回率值表示更好的性能. 图 3(a)展示了在 SIFT1M 数据集上的结果. 可以看出,随着检索候选集的



(a) SIFT1M数据集上的*R*&*k*值　　(b) GIFT1M数据集上的*R*&*k*值

图 3　不同候选集大小的召回率测试

Fig. 3　Recall@k experiments in different dataset

增加,召回率也逐渐提高,在候选集大小超过 100 时,召回率已经超过了 99%. 同时,我们的算法在所有候选集大小上都表现出优势,特别是在较小的候选集合下,这一优势更加显著. 这表明我们的方法在较小的检索候选集上能够获得出色的召回率精度. 图 3(b)展示了 HNSW 算法和我们的算法在 GIST1M 数据集上的比较. LB-ANNR 在所有候选集大小上都明显优于 HNSW,平均召回率提高了超过 5%. 在较小的候选集合下,这一提升更为显著,接近 10%. 这进一步证明了我们的方法在较小的检索候选集上表现出卓越的召回率精度. 在 GIST1M 数据集中,当数据维度从 SIFT1M 的 128 增加至 960 时,这种精度提升效果更加显著. 这表明我们的方法在处理高维度数据时具有更明显的优势. 通过上述实验,我们验证了 LB-ANNR 算法在不同候选集大小上的召回率优势.

3.5 大数据场景下的实验测试

随着数据集的增长,对于大规模数据的处理,索引构建所需的时间变得至关重要. 在真实场景下,全国电力数据存在着规模大、分布广的特点. 为了验证本文描述的方法在处理大数据场景下的可行性和实用性,我们收集了两个内部电力文本大数据数据集,分别是 GridCorpus01 和 GridCorpus02,其中 GridCorpus01 数据集包含 973 万个样本,GridCorpus02 数据集包含 1526 万个样本. 每个样本都具有 128 个特征维度. 我们在这两个数据集上进行了索引构建效率和耗时的测试,实验设置与第 3.2 节相同,各个算法的对比结果如表 3 所示.

表 3 各个算法索引大小与构建时间对比

Table 3 Comparison of the size and construction time of retrieval index of each algorithm in different datasets

Algorithm	GridCorpus01		GridCorpus02	
	Recall/%	Time/ms	Recall/%	Time/ms
KGraph	0.825	3.83	0.803	2.57
NSG	0.902	9.04	0.924	6.23
Efanna	0.873	7.91	0.879	5.79
HNSW	0.886	8.86	0.891	5.85
LB-ANNR	0.904	7.52	0.922	5.48

可以看出,在这两个数据集上,kGraph 构建的检索耗时最短,但检索精度相对较低,其召回率是所有算法中最差的. 在 GridCorpus01 数据集上,与 HNSW 和 NSG 算法相比,本文提出的 LB-ANNR 方法在保持良好召回率的同时,检索时间分别减少了 16.8%和 15.1%. 这表明本文所描述的方法在大数据场景下仍然能够保持较好的检索精度和高效的检索速度,进一步证明了本文方法在实际应用中的可行性和实用性. 在 GridCorpus02 数据集上,与除 kGraph 外的算法相比,本文所描述的 LB-ANNR 方法的检索时间减少了 5.3%. 在所有召回率高于 85%的算法中,我们设计的方法具有最少的检索时间. LB-ANNR 在召回率方面也同样是表现较好的算法之一. 在大数据场景下的实验测试结果表明 LB-ANNR 在不同数据条件下都表现出了较好的鲁棒性和适应性,进一步证明了其在大数据场景的可行性和实用性.

4 结论

本论文针对大规模数据的近似最近邻检索问题,通过研究基于图的近似最近邻方法,提出了一种均衡感知的快速 K 均值近邻聚类的特征数据分桶及其紧凑型图索引构建算法. 我们通过设计快速均衡感知的 K 均值聚类算法,实现了海量特征数据的均衡分桶,并将高维向量压缩为轻量级紧凑型图索引数据结构,从而降低了特征检索过程中的内存消耗. 与传统方法相比,本文提出的算法在保证较高的检索精度的同时,显著提高了索引构建速度,支持了大规模数据的高效索引和检索. 通过开源数据和真实场景数据实验证明,我们的方法取得了良好的性能表现,展示了在海量数据场景中的潜在应用价值. 在未来工作中,我们将进一步探索在实时和批量新增数据的场景下,基于增量聚类的均衡索引设计与研究.

[参考文献]

[1] LI X, ZHANG W, DING Q. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction[J]. Reliability engineering & system safety, 2019, 182(C): 208-218.

[2] WU H, LIU Y, WANG J. Review of text classification methods on deep learning[J]. Computers, materials & continua, 2020,

63(3):1309-1321.

[3] PURWINS H,LI B,VIRTANEN T,et al. Deep learning for audio signal processing[J]. IEEE journal of selected topics in signal processing,2019,13(2):206-219.

[4] LITJENS G,KOOI T,BEJNORDI B E,et al. A survey on deep learning in medical image analysis[J]. Medical image analysis,2017,42:60-88.

[5] DA'U A,SALIM N. Recommendation system based on deep learning methods:a systematic review and new directions[J]. Artificial intelligence review,2020,53(4):2709-2748.

[6] ZHENG B,ZHAO X,WENG L. et al. PM-LSH:a fast and accurate in-memory framework for high-dimensional approximate NN and closest pair search[J]. The VLDB journal,2022,30(6):1339-1363.

[7] BABENKO A,LEMPITSKY V. The inverted multi-Index[J]. IEEE transactions on pattern analysis and machine intelligence,2015,37(6):1247-1260.

[8] GE T Z,HE K M,KE Q F,et al. Optimized product quantization[J]. IEEE transactions on pattern analysis and machine intelligence,2014,36(4):744-755.

[9] KALANTIDIS Y,AVRITHIS Y. Locally optimized product quantization for approximate nearest neighbor search[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus:IEEE,2014:2321-2328.

[10] FU C,XIANG C,WANG C,et al. Fast approximate nearest neighbor search with the navigating spreading-out graph[J]. Proceedings of the VLDB endowment,2019,12(5):461-474.

[11] SUBRAMANYA J S,DEVVRIT F,SIMHADRI H V,et al. DiskANN:fast accurate billion-point nearest neighbor search on a single node[C]//Neural Information Processing Systems. Canada:IEEE,2019,32.

[12] MALKOV Y A,YASHUNIN D A. Efficient and robust approximate near-est neighbor search using hierarchical navigable small world graphs[J]. CoRR,2016.

[13] BARANCHUK D,BABENKO A,MALKOV Y. Revisiting the inverted indices for billion-scale approximate nearest neighbors[C]//Proceedings of the European Conference on Computer Vision(ECCV). Munich:Springer-Verlag,2018:202-216.

[14] MOROUZI M B A,FLEET D J. Cartesian k-means[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Portland:IEEE,2013:3017-3024.

[15] WANG J,WANG J,SONG J,et al. Optimized cartesian k-means[J]. IEEE transactions on knowledge and data engineering,2014,27(1):180-192.

[16] LINDE Y,BUZO A,GRAY R. An algorithm for vector quantizer design[J]. IEEE transactions on communications,1980,28(1):84-95.

[17] ZHAN J,MAO J,LIU Y,et al. Jointly optimizing query encoder and product quantization to improve retrieval performance[C]//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. Santonni:ACM,2021:2487-2496.

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

[19] CIACCIA P,PATELLA M,ZEZULA P. M-tree:an Efficient access method for similarity search in metric spaces[C]//Proceedings of the 23rd VLDB Conference. Athens,Greece:ACM,1997:426-435.

[20] GUTTMAN A. R-trees:A dynamic index structure for spatial searching[C]//Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data. Boston:ACM,1984:47-57.

[21] ANDONI A,INDYK P,NGUYEN H L,et al. Beyond locality-sensitive hashing[C]//Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics. PA:ACM,2014:1018-1028.

[22] DATAR M,IMMORLICA N,INDYK P,et al. Locality-sensitive hashing scheme based on p-stable distributions[C]//Proceedings of the Twentieth Annual Symposium on Computational Geometry. New York:ACM,2004:253-262.

[23] PAULEVE L,JEGOU H,AMSALEG L. Locality sensitive hashing:a comparison of hash functiontypes and querying mechanisms[J]. Pattern recognition letters,2010,31(11):1348-1358.

[24] HARWOOD B,DRUMMOND T. Fanng:fast approximate nearest neighbour graphs[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York:IEEE,2016:5713-5722.

[25] JAROMCZYK J W,TOUSSAINT G T. Relative neighborhood graphs and their relatives[J]. Proceedings of the IEEE,1992,80(9):1502-1517.

[26] TOUSSAINT G T. The relative neighbourhood graph of a finite planar set[J]. Pattern recognition,1980,12(4):261-268.

[27] AURENHAMM F. Voronoi diagrams—a survey of a fundamental geometric data structure[J]. ACM computing surveys(CSUR), 1991,23(3):345-405.

[28] HAJEBI K,ABBASI-YADKORI Y,SHAHBAZI H,et al. Fast approximate nearest-neighbor search with k-nearest neighbor graph[C]//Twenty-Second International Joint Conference on Artificial Intelligence. Catalonsa,Spain:ACM,2011.

[29] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision,2004,60(2):91-110.

[30] DOUZE M,JÉGOU H,Sandhawalia H,et al. Evaluation of gist descriptors for webscale image search[C]//Proceedings of the ACM International Conference on Image and Video Retrieval. Santorini,Greece:ACM,2009:1-8.

[责任编辑:杜忆忱]