机器学习赋能的多维数据查询处理研究综述

马超红 郝新丽 孟小峰 张旭康
(中国人民大学信息学院 北京 100872)

摘 要 多维数据的查询和处理在数据库中普遍存在。高效的多维数据查询处理,一方面依赖于精细的索引结构,例如 R-tree、KD-tree 等被广泛应用;另一方面,也有诸多工作探索利用硬件优势设计高效的数据布局,即研究面向扫描的数据处理策略以及构建数据概要,避免高代价地访问原始数据。然而,随着数字化社会的发展,网络 Web 服务更加普及,传感器网络无处不在,诸如网约车、电子地图等基于位置的服务愈发盛行,使得多维数据正在以前所未有的速度产生,对查询处理提出新的要求,包括更快的查询响应、更低的存储占用。近年来,机器学习包括深度学习算法不断优化,且计算机等硬件环境持续发展,为多维数据查询处理带来更多的优化契机,不仅降低查询执行时间,同时能够节省存储资源,取得显著性优势。因此,机器学习被广泛应用于构建更好的数据管理和数据分析任务解决方案。该文提出机器学习赋能的多维数据查询处理研究框架,一方面介绍机器学习模型对多维索引结构的优化和改进;另一方面,介绍机器学习对不依赖索引结构的查询处理任务的赋能研究,包括数据布局策略和数据概要研究。在总结已有研究现状的基础上,指出该领域面临的挑战和未来研究方向。

关键词 查询处理;多维学习化索引;数据布局;数据概要;机器学习
中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2025.00100

Survey on Machine Learning for Multi-Dimensional Data Query Processing

MA Chao-Hong HAO Xin-Li MENG Xiao-Feng ZHANG Xu-Kang
(School of Information, Renmin University of China, Beijing 100872)

Abstract The query processing and data manipulation for multi-dimensional data are ubiquitous in modern database engines, which are crucial for both businesses and individual users to achieve effective data management, analysis, and decision-making processes. On one hand, efficient query processing for multi-dimensional data hinges on the utilization of sophisticated indexing structures such as R-tree and KD-Tree, which are widely employed in commercial databases to optimize the retrieval and organization of such data. These structures are designed to manage and query multi-dimensional data efficiently, allowing for rapid access and manipulation, enabling the system to quickly locate and retrieve the requested data points from large datasets, significantly enhancing the performance and responsiveness of database queries. On the other hand, many advanced studies explore effective data layout designs that take advantage of hardware characteristics, such as disk access speed and memory hierarchy. For instance, researchers have been focusing on designing scan-oriented data processing strategies to minimize seek time, which is typically the time taken for a hard disk to locate a specific piece of data. Furthermore, constructing data synopses, which are summarizing representations or approximations of data, helps to mitigate the high costs associated with accessing the entire dataset. These synopses can provide quick insights

收稿日期:2023-08-01;在线发布日期:2024-07-16。本课题得到国家自然科学基金项目(62172423)资助。**马超红**,博士研究生,主要研究方向为机器学习化数据库系统、学习化索引及数据管理。E-mail: chaohma@ruc.edu.cn。**郝新丽**,博士研究生,主要研究方向为智能科学发现、时间序列分析和可解释机器学习。**孟小峰**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为数据库系统、数据管理、隐私保护以及社会计算等交叉性研究。E-mail: xfmeng@ruc.edu.cn。**张旭康**,博士研究生,主要研究方向为数据库系统、大数据实时分析、数据库调优、查询优化。

into the data without the need to process the whole dataset, thus saving time and computational resources. Currently, the integration of digital technologies into various facets of society has precipitated a surge in web services, ranging from e-commerce platforms to cloud computing. The digital society has also led to a proliferation of sensor networks and location-based services, such as ride-hailing applications and electronic maps. Consequently, this has resulted in an unprecedented expansion of multi-dimensional data, presenting novel challenges for database systems. These challenges include the imperative for faster query responses and lower storage overhead. Database systems must now handle vast amounts of data more efficiently, ensuring that queries are processed quickly and storage is utilized optimally. In recent years, advancements in machine learning, particularly deep learning, have seen significant improvement, and new hardware environments have continued to evolve. These developments offer enhanced optimization opportunities to query processing for multi-dimensional data, resulting in substantial performance gains. These optimizations not only aim to reduce query execution time but also to efficiently save storage resources, thereby enhancing the overall system efficiency and scalability. As a result, machine learning techniques have been widely applied to build more effective data management and data analysis solutions, from indexing and data layout optimization to data synopsis construction. In this paper, we propose a comprehensive framework for the advanced research on machine learning for multi-dimensional data query processing. Moreover, we introduce the related work on machine learning for multi-dimensional index structures, and the research on non-index structures, including data layout optimization and data synopsis. By reviewing existing literature and current methodologies, we aim to identify key research gaps and propose future directions in this field. Additionally, we highlight anticipated challenges which may be encountered, emphasizing the need for innovative approaches to overcome these obstacles and drive forward advancements in multi-dimensional data query processing.

Keywords　query processing; multi-dimensional learned indexes; data layout; data synopsis; machine learning

1　引　言

机器学习被应用于构建更好的数据管理和数据分析任务解决方案[1-5]，例如查询优化[6]、数据库调优[7]、数据库索引[8]等。多维数据是数据库中普遍存在的数据类型，关联多个维度的数据都可以被称为多维数据[9]。例如，数据库中具有 d 个属性的表可以视为一个 d 维数据集，表中每一行表示 d 维空间中的一个点；再者，地理信息等空间数据都是多维数据。同时，多维数据的查询和处理是数据库中普遍存在的应用，数据库学术界和工业界一直以来都在不断优化算法和数据结构以提高多维数据查询处理的性能。近年来，多维数据，尤其是空间数据，正在以前所未有的速度产生。

多维数据体量的激增，从以下 3 个方面推动机器学习在多维数据管理中的应用，以有效分析并利用多维数据。

首先，多维数据的查询和处理面临更大的挑战。网约车、社交网络、轨迹推荐、数字地图等基于位置服务的兴起和盛行，一方面，要求查询处理在更短时间内被响应，涉及快速查找和插入、高效分析等操作；另一方面，要求系统具备更多资源（如内存、外存）来存储并分析不断增长的大规模数据，包含原始数据及辅助数据结构（索引等）的存储。

其次，传统多维数据查询处理面临瓶颈。传统加速多维数据查询处理的结构，如 R-tree[10]、KD-Tree[11]、多维直方图[12]等，是通用的数据结构；尽管这些数据结构能够处理各种各样的数据类型、数据分布，用于加速查询，但这些通用的结构忽略了应用场景和数据的特征。例如，传统的通用多维索引结构，不仅占用较大的存储空间，选用不当还会带来较高的扫描代价[13]。对于倾斜工作负载，如果能够考虑到数据的稀疏程度，就能够对结构进行优化。空间数

据概要,例如多维直方图、随机抽样等,常用于回答近似查询。然而现有空间数据概要技术的性能很容易受到不同数据或查询负载分布的影响,很难达到最优[14]。然而,借助数据分布能够优化大多数的数据结构[8,15]。

最后,机器学习能够提供解决方案。机器学习包括深度学习,在数据库设计和优化中,得到广泛应用。例如在一维数据上,机器学习模型通过挖掘数据分布构建学习化索引结构[16-20],能够显著提高查询效率,并降低索引结构占用的存储空间;借助机器学习挖掘工作负载特征,优化系统的资源配置和调度算法等[21-23]。多维数据的底层分布,可能存在复杂多变的模式,并且多个维度之间存在关联性,因此将机器学习与多维数据查询处理相结合,从数据和工作负载中挖掘模式,能够实现针对特定数据和场景的优化。

目前已有大量工作将机器学习应用于多维数据,例如应用无监督方法挖掘多维数据关联[24]、构建多维学习化索引加速查询处理[13,25-26]、利用强化学习算法对数据进行分区等[27]。本文主要从数据管理的角度,总结机器学习方法应用于多维数据,构建加速查询处理的解决方案,即机器学习赋能的多维数据查询处理研究。

高效的多维数据查询处理,通常利用大量精细化的索引结构来加速,例如在多维数据上构建R-tree、KD-Tree、网格索引等结构。然而,近年来,随着硬件系统的发展,为充分利用存储设备的顺序扫描带宽,面向扫描的数据处理策略和数据布局等策略也有较多应用,还包括构建数据概要文件以支持近似查询处理。本文将前者统称为多维索引结构研究,后者统称为非索引结构研究。

具体而言,索引结构的主要目的是构建精细化的数据结构,给定查询条件,由数据结构给出一组包含查询结果的候选集[28],从而避免全表扫描。非索引结构[27]缓解了构建索引带来的存储和维护成本。一方面,面向扫描的数据布局策略通过将数据分块存储,避免了精细索引结构,另一方面,构建概要能够获得数据的压缩表示[28],从而减少对原始数据的访问。索引结构和非索引结构,这 2 类方法都能够有效解决查询处理问题,并且互相之间有交叉,例如,每个布局内维护索引结构,以提高查询效率;构建概要数据时,借助索引分区,获得数据的紧凑表示。

针对已有机器学习在多维数据查询处理中的应用,本文提出机器学习赋能的多维数据查询处理研究框架,对现有研究工作进行分类(第 2 节),并从研究框架的角度介绍相关工作(第 3～5 节)。在概括现有工作的基础上,我们总结出机器学习赋能多维数据查询处理面临的挑战,并给出未来研究方向。

本文与其他相关综述性文章的不同如下:

(1)机器学习赋能的多维数据查询处理是当前国内外的研究热点,但缺少针对该方向的综述文章。本文是第一篇系统地从索引结构和非索引结构两方面总结多维数据查询处理的中文综述。

(2)2020 年在线出版的学习索引[29],总结了机器学习在索引结构中的应用,其中第 2 节介绍了面向多维范围查询的学习索引,但没有涉及多维数据查询处理中的数据布局和数据概要等方面。本文更加全面地总结了多维学习化索引,并将现有研究成果分为 3 大类,同时涵盖了 2020 年～2023 年的最新研究文献。

(3)智能分区方法的研究综述[30],侧重于数据驱动的应用分析场景,总结近年来借助机器学习指导数据分区和布局的相关工作,但没有涉及多维索引结构和多维数据概要研究现状的总结。

(4)本文针对多维数据查询处理,梳理并提出了研究框架,从学习化多维索引结构和学习化非索引结构两方面进行总结。其中,多维索引结构分为 3 大类,包括基于降维、基于分区策略和基于属性间关系的方法。非索引结构,根据是否访问原始数据,进一步分为数据布局和数据概要 2 类。本文涵盖了包括 SIGMOD2023、VLDB2023 等在内的最新相关研究工作,提供该方向较为全面的研究进展总结。并根据对现有工作的概括和讨论,总结出 4 个未来值得探索的研究方向。

2 研究框架

机器学习模型应用于多维数据,构建更加高效的查询处理解决方案,是当前热门的研究方向,无论是多维数据长久依赖的索引结构,还是诸如数据布局、数据概要等研究领域都有丰富的研究成果。

本文在对已有相关工作总结的基础上,提出机器学习赋能的多维数据查询处理研究框架。如图 1 所示,高效的多维数据查询处理一方面依赖于精细化的索引结构,另一方面,合理有效的非索引结构不仅能提高查询效率,而且能够降低索引结构带来的存储和维护代价。因此机器学习赋能的多维数据查询处理可分为 2 大类,索引结构和非索引结构。

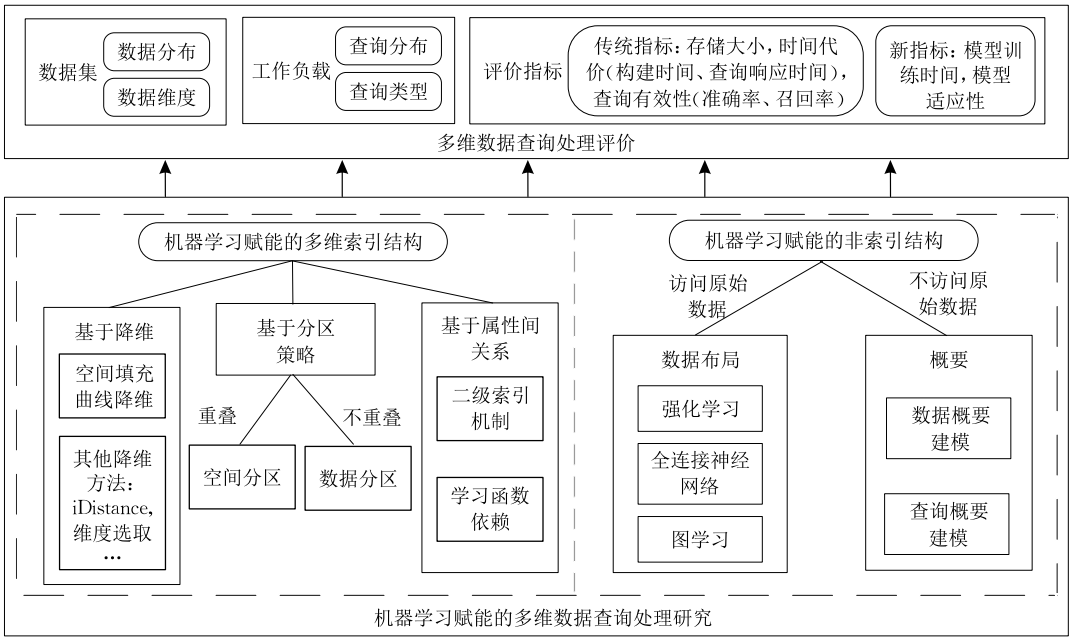


图 1　机器学习赋能的多维数据查询处理研究框架

机器学习赋能的多维索引结构研究包含 3 类，首先是基于降维的多维索引方法，将数据从多维空间映射到一维。在基于降维的方法中，机器学习的应用体现在 2 个角度：(1) 对降维后的数据采用机器学习模型拟合数据分布；(2) 利用机器学习模型对数据进行降维。其次是基于分区策略的索引方法，主要是采用机器学习模型将数据分布和工作负载特征嵌入到分区策略中。最后是基于属性间关系的方法，挖掘不同属性列之间关系，用于辅助查询处理。

机器学习赋能的非索引结构中，按照是否访问原始数据，进一步划分为数据布局和概要。数据布局的研究主要集中在采用不同的机器学习方法优化数据的布局，进而使得查询响应能尽可能少地访问数据块。概要建模的主要目的是采用机器学习方法在数据或查询中学习知识，抽象出数据或查询结果的紧凑表示。在回答查询时通过访问概要获得查询结果，进而避免高代价地访问原始数据。

多维数据查询处理数据结构或算法的性能需要在不同的场景中借助不同指标进行评价，以体现算法的优越性，并为从业者选择算法提供参考。本文从数据集、工作负载和评价指标 3 方面介绍现有多维数据查询处理方法的评价体系，并总结出在机器学习赋能的查询处理场景中所特有的指标。

因此，本文后续章节在研究框架的基础上介绍已有工作。首先，机器学习赋能的多维索引结构研究（第 3 节）主要从基于降维、基于分区和基于属性间关系的方法 3 方面来介绍；接着，机器学习赋能的非索引结构查询处理（第 4 节）主要从学习化数据布局和学习化概要 2 方面来介绍；其次，针对诸多算法和数据结构，第 5 节介绍评价采用的数据集、工作负载及评价指标；最后，文章第 6 节总结全文，并对未来的研究方向进行展望。

3　机器学习赋能的多维索引结构研究

索引结构对于多维数据查询处理至关重要。传统的多维索引结构主要分为：(1) 采用降维方法将多维数据降至一维后构建索引，如 UB-tree[31-32] 采用 Z 阶空间填充曲线将数据降至一维构建 B-tree[33]；(2) 基于空间或数据分区的方法，如基于数据分区的 R-tree[10] 和基于空间分区的网格索引等。

2018 年 SIGMOD 国际会议，Kraska 等人[8] 提出学习化索引（Learned index）的概念，并将其应用于一维数据，通过机器学习模型拟合数据分布，实现关键字到存储位置的映射。如图 2 所示，灰色阴影

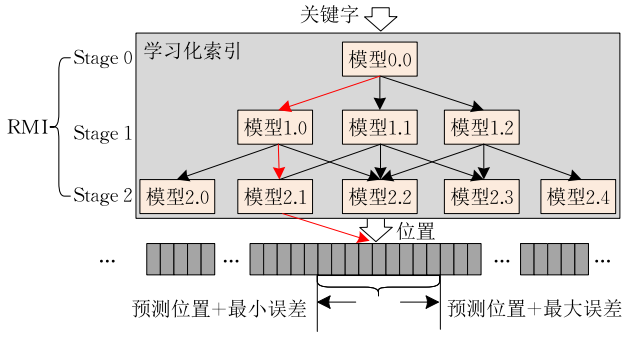


图 2　学习化索引采用模型拟合关键字到位置的映射（以 RMI 为例）

区域表示学习化索引,可以采用任意的模型,模型的输入为关键字,输出为数组中的位置。

具体地,Learned index[8]设计了针对一维数据的递归模型索引(Recursive Model Index,RMI),如图 2 中层级递归模型(模型 0.0,模型 1.0,模型 2.0…)所示。RMI 将数据排序后存入数组,通过模型拟合关键字到数组下标的映射,即数据的累积分布函数(Cumulative Distribution Function,CDF),进而将索引建模为函数f。给定学习函数,通过函数调用来查找关键字,显著降低存储占用,并提高查询性能。

基于此,针对普遍存在的多维数据查询处理,数据库界开始研究通过机器学习模型来构建或优化多维索引结构,以达到降低存储空间占用和加速查询的目的。

然而,在多维数据上构建学习化索引面临着如下挑战。首先,多维数据不存在自然排序(Natural order);其次,多维数据的各属性之间往往存在关联性,属性间不独立。一维学习化索引的基本假设是数据是有序的,通过拟合排序数据的累积分布函数,预测关键字在有序数组中的位置。多维数据由于存在多个维度,其中某一维度上的排序在另一维度上并不适用;并且多个属性之间的关联性使得直接应用一维学习化索引失效或不能达到最优。

多维学习化索引的构建已有部分研究工作,探索解决如上的挑战,主要思想集中在:(1)借助降维方法,为多维数据指定一个排序;(2)基于学习到的数据分布实现更优的多维索引分区;(3)借助属性间的关联优化多维索引。因此,本节从 3 方面介绍多维学习化索引的相关研究:基于降维的方法、基于分区策略的方法以及基于属性间关系的方法。

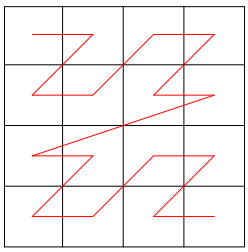
3.1 基于降维的多维学习化索引方法

此类方法首先采用某种降维策略对多维数据降维,使得多维数据中的点能够唯一对应于一维中的某个值;接着基于得到的一维值对数据进行排序;最后在一维排序值上构建一维学习化索引结构,用于优化查询,同时降低存储空间占用。其中最常用的获取多维数据一维排序的方法是采用空间填充曲线[34],因此本节依据采取的降维方法进行分类:空间填充曲线降维和其他降维方法。

3.1.1 空间填充曲线降维

空间填充曲线的主要目的是将高维数据映射到一维空间,同时使得高维空间中近邻的点在一维直线上接近。典型的空间填充曲线包括 Z 曲线、希尔伯特(Hilbert)曲线等,其中 Z 曲线的映射算法简单,但空间近邻保持较差,而希尔伯特曲线的空间近邻特性效果最好,但映射算法复杂[35]。

Learned ZM[36]采用 Z 阶填充曲线[37]将多维数据映射到一维空间得到 Z 阶值(Z-value),图 3 展示了二维空间应用空间填充曲线及获得 Z 阶值示例,首先将二维空间划分为网格,用“Z”形状访问每个单元格,并将所有“Z”首尾相连成一条没有交叉的连续曲线。Learned ZM 对 Z-value 排序后,采用 Kraska 等人[8]提出的递归模型 RMI 构建索引,获得 Z-value 到存储位置的映射。



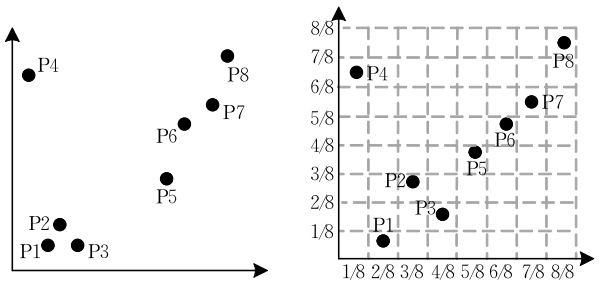
0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

图 3 二维空间中 Z 阶空间填充曲线及 Z 阶值示例

Learned ZM 在执行查询时,首先计算查询点的 Z-value,之后采用 RMI 模型预测位置,在模型的预测误差内对数据进行局部搜索,最终找到精确的查找位置。针对多维数据中常见的范围查询,Learned ZM 首先计算查询范围上界点和下界点的 Z-value,用模型预测出包含查询范围上界点和下界点的存储位置,接着遍历预测范围内的所有点,给出查询结果。Wang 等人[38]将希尔伯特(Hilbert)空间填充曲线与两层 RMI 模型结合,提出学习化 HM 索引(Learned Hilbert Model,Learned HM)。

已有利用空间填充曲线进行降维的方法,都是在原始空间数据上直接应用空间填充曲线,而 RSMI[25]考虑将原始空间中的点映射到秩空间(rank space),在秩空间中应用空间填充曲线对数据进行排序。秩空间转换,采用数据点在原始空间中每个维度上的排序作为秩空间中的坐标。也就是说,秩空间具有和原始数据空间一样的维度数量,数据点 p 在秩空间的坐标为在原始空间中 p 对应维度上的排序,如图 4 所示[25,39],P1 在原始空间(图 4(a))中两个维度上的排序分别为 2 和 1,因此在秩空间(图 4(b))的坐标为(2,1),同样,P4 在秩空间的坐标为(1,7)。

将原始空间转换到秩空间确保了在空间填充曲线的网格中每个行/列只有一个点,使得空间点的曲线值之间具有更均匀的距离,简化函数的拟合。



(a) 原始空间　　(b) 秩空间

图 4　原始数据空间到秩空间的转换

值得注意的是，RSMI 是针对近似查询处理，而非精确查询。同时，RSMI 的预测粒度是磁盘块(block)，并且由于模型的预测误差，导致即便是只查询一个点，也会带来($max\_err+min\_err$)个块的 I/O 和扫描代价，其中 max_err 和 min_err 为模型的最大和最小预测误差。RSMI 能支持数据的插入，但代价是需要标记新插入的块，这会使得在原始 I/O 和扫描代价($max\_err+min\_err$)的基础上增加额外 $O(IB)$的代价，I 为新创建的块个数，B 为每个块中的记录数。另外，RSMI 通过在数据库空闲期执行重新训练来避免大量插入带来的性能下降。

Learned ZM、Learned HM 和 RSMI 均采用固定的空间填充曲线，然而固定的空间填充曲线降维策略并没有从数据分布中学习，并且降维后数据的存储布局缺少优化，即通常将固定数量的数据点存储在一个页(page)中，导致页中数据的最小边界矩形具有大量死区(dead space)或与其他页具有较多重合，使得查询时需要访问大量无关页或数据。为解决如上问题，Gao 等人[40]设计了可学习的单调空间填充曲线(Learned Monotonical Space Filling Curves，LMSFC)以利用数据分布，并基于 LMSFC，提出离线和在线两种优化策略。离线策略用于数据的存储布局，实现最优或次优地将数据存储到页面中，而在线策略递归地将查询拆分为多个子查询。结合离线和在线策略，LMSFC 能够最大限度地减少访问不包含潜在查询结果的页面，从而提高查询性能。

以上介绍的方法，旨在设计端到端的多维索引结构。不同于已有工作，Li 等人[41]提出不依赖于索引结构的分段空间填充曲线(piecewise Space Filling Curves，piecewise SFCs)，能够自然地集成到采用空间填充曲线降维的索引结构中。由于传统 SFC 在整个数据空间内都采用单一的映射模式(mapping scheme)，不能在整个数据空间达到最优，并且没有一个 SFC 能够在所有数据集和工作负载分布上提供最佳性能。因此，为利用数据分布，并在整个数据空间内实现最优性能，Piecewise SFCs 将数据空间划分为不同的子空间，在不同的数据子空间中采用不同映射模式。具体地，Piecewise SFCs 提出位合并树(Bit Merging Tree，BMTree)的方法划分数据子空间并生成子空间相应的 SFC，其中 BMTree 的每个叶子节点对应一个子空间。为实现更好的子空间划分，BMTree 的构建采用强化学习。Piecewise SFCs 能够满足单调性和单射性(即多维数据点和一维映射值具有一一对应关系)。同时，论文实验证明了 piecewise SFCs 具有适应数据插入的能力。另外，作者建议，在查询分布发生较大变化时，需要重新训练 BMTree 以保持性能。

3.1.2　其他降维方法

基于降维的研究工作中，空间填充曲线是最常用的降维方式之一。其次，也存在其他方法，将多维数据映射到一维空间。本节主要介绍已经在多维学习化索引中被采用的基于 iDistance 降维和选取某一维度对数据降维的方法。其中基于 iDistance 的方法能够在降维过程中更好地保持数据在高维空间中的邻近性，而选取某一维度降维的方法具有复杂度低的特点。

Davitkova 等人[42]借鉴 iDistance[43]降维方法提出 ML-index，首先在多维空间中选择 N 个参考点(reference points)，基于参考点及相似度对空间中的点进行聚类产生 N 个分区。接着为每个分区编号，并在每个分区内，根据点到参考点的距离计算一维值。ML-index 改进了 iDistance 计算一维值的方法，使得分区之间的一维值不会出现重复。图 5 展示了包含 2 个参考点的 ML-index 示例图，其中 O1～O2 为参考点，Pi 为空间中的其他点。

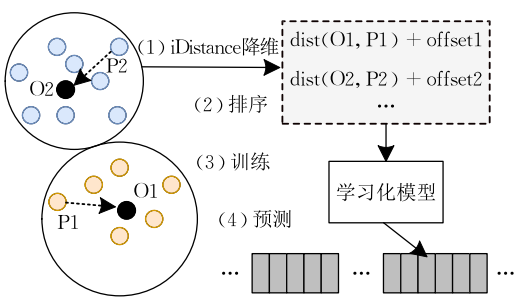


图 5　ML-index 示例

另外，已有的 Learned ZM 和 Learned HM 不支持 K 近邻查询。ML-index[42]考虑同时解决多维数据的点(point)查询、范围(range)查询、K 近邻

(KNN)查询。ML-index 还存在以下问题:(1)没有提及如何处理与参考点距离相等的点;(2)模型训练的好与坏,极大地依赖于超参数(参考点个数、参考点选取、分区排序等);(3)ML-index 降维后的一维值虽然保留了距离参考点的邻近性,但其排序后并不能保证当 $Pi>Pj$ 时,Pi 的每一个维度都大于 Pj 的每一个维度。同时,ML-index 处理查询的复杂度较高,由于要频繁计算距离,ML-index 在大规模数据集上的可扩展性较差。

IF-X[44]采用机器学习模型增强空间索引,同时指出在空间索引中应用机器学习的关键在于找出并理解空间索引中适合采用预测模型来增强的部分。IF-X 没有单独采用降维方法,对多维数据进行降维,而是在每个节点选择一个维度用于排序。具体地,IF-X 着重优化空间索引中叶子结点的访问,在每个叶子结点中,首先根据每个维度的 CDF 选取用于排序的维度,接着在选择的维度上构建机器学习模型。IF-X 的主要思想是保持空间索引(如 R-tree、KD-tree)的上层结点不变,只在叶子结点,选取维度,训练模型。IF-X 较为简单地结合了机器学习模型,能够提高叶子层的查询效率,对于传统空间索引结构的上层并没有优化,同样可能会带来查询性能的下降。

尽管存在多种可行的映射策略将多维数据点投影到一维,SageDB[45]认为依次沿着数据的维度(例如首先是 x 维度,其次是 y 维度,最后是 z 维度)对数据点进行排序获得的一维空间,能够使得数据的存储布局更容易学习。排序后,SageDB 采用一维学习化索引方法(RMI)对数据进行索引,同时 SageDB 将排序后的点划分到大小均匀的单元中,在范围查询时减少假正类(false positives)。相对于传统数据库,由数据库管理员手动选择维度,SageDB 指出可以利用数据和查询分布自动学习排序维度和每个单元的划分粒度,并且允许数据库采用更加复杂的映射,例如非轴对齐的投影。然而,SageDB 并没有给出具体的细节。

ML-index、IF-X、SageDB 探索了机器学习方法用于多维数据查询处理场景,只针对静态的工作负载,不支持数据的插入和更新。

3.2 基于分区策略的多维学习化索引方法

基于降维的多维学习化索引结构,需要将空间数据转化为一维数据,或者是只在某个维度上采用机器学习模型拟合数据分布,并不能完全利用原始空间数据中的多维特性。因此,基于分区的策略侧重于利用空间数据的多维特性,借助多维数据分布进行分区,而不是将数据降到一维。

基于分区的多维索引结构,依据分区策略,可分为基于空间的分区和基于数据的分区[46]。两种分区策略的主要不同在于,划分的子分区之间是否存在重叠区域[40,47]。基于空间分区的方法,例如 Quad Tree[48]、网格索引[49]等,将空间(数据所在的值域)划分为子分区,然后对这些子分区进行索引,子分区之间不存在重叠;基于数据的分区,例如 R-tree 及其变体[10,50]等,按照数据点之间的距离、密度或聚类等方法,将数据划分为不同的子集,然后对这些子集进行索引,子集所在的子空间区域存在重叠。

3.2.1 基于空间分区的方法

Zhang 等人[26]提出基于网格索引的空间插值函数(Spatial Interpolation Function Based Grid Index, SPRIG),将多维插值函数作为学习模型,可直接应用于多维数据,即不需要将空间数据降到一维。SPRIG 首先从原始数据中抽样,构建自适应网格,在抽样数据之上拟合空间插值函数。在执行查询时,采用插值函数预测位置,再执行局部搜索。为进一步提升 K 近邻查询的查找性能,SPRIG 引入基于枢轴(pivot)的过滤技术。SPRIG 针对空间二维数据,在拟合空间插值函数时,具有较高的训练时间。当维数过高时,多维插值函数很难拟合,导致复杂度更高。SPRIG 在某些情况下具有较高的预测误差。为解决如上问题,Zhang 等人[51]进一步提出 SPRIG+以更好地利用空间插值函数的特性,降低插值函数的拟合时间,同时引入动态编码技术来保证预测误差的上界。

Peng 等人[52]探索将 Learned index[8]的架构应用于多维 K 近邻查询,提出 Learned KD-tree,将 K 近邻查询问题抽象为有监督的多分类问题。首先构建 KD 树,在每个内部结点沿一个维度平分空间,在树的每一层以循环的方式选择维度。借助构建好的 KD 树,获取数据集中每个点的 K 个近邻,构成训练集。接着,对原始数据进行特征提取,训练全连接神经网络模型(Fully Connected Neural Networks, FCN),最后借助模型预测查询点的 K 近邻。

现有多维索引结构存在调优困难的问题,且特定应用场景中性能优越的索引结构,在另外的场景中并不总是优于其他方法。传统通用的多维索引结构,未能充分利用数据分布和工作负载特征,因此缺乏针对特定数据和查询进行自适应优化的能力。

为解决如上针对特定数据和工作负载优化的问

题，Nathan 等人[53]提出 Flood，针对数据分析任务中常见的多维查询，从数据和查询负载中学习网格索引(Grid index)以加速查询。其基本思路是将数据划分到网格单元中。不同于传统的网格索引，所有的 d 个维度都参与划分，Flood 选择 $d-1$ 个维度用于构建网格(Grid)，第 d 个维度用于对网格单元中的数据排序。图 6 上半部分展示了 2 维数据的 Flood 网格索引，维度 1(属性 1)用于网格划分，维度 2(属性 2)用于排序。

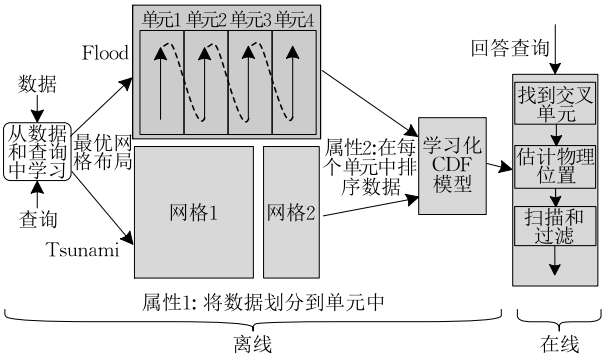


图 6　针对 2 维数据的 Flood 和 Tsunami 网格索引

因此，Flood 在划分网格时，需要确定用于构建网格的 $d-1$ 个维度，每个维度划分的分段数，以及用于排序的维度。这些参数依赖于数据特征和查询工作负载等多方面因素，例如查询中不同属性使用的频率，每个维度选择性的平均值和方差，属性之间的关联性等。不同于传统的网格划分，Flood 训练以查询时间为优化目标的代价模型(cost model)，为减少真实执行查询的代价，采用随机森林回归模型预测 cost model 的参数，并选择代价最小的 grid 布局，该布局包含每个维度划分多少段。每个维度划分的分割值(split value)，借助每一个维度上的累积分布函数来确定。为选择排序维度，Flood 遍历每一个维度，以性能最优的作为排序维度。

尽管 Flood 能够自动地针对特定的工作负载和数据优化，但在数据中存在关联属性及查询工作负载倾斜时，Flood 的性能下降且占用的内存空间增加。因此 Flood 的研究团队在 Flood(洪水)的基础上提出 Tsunami(海啸)[54]。通过识别工作负载的倾斜，Tsunami 将空间划分为多个不相交的区域，每个区域单独构建网格(Grid)，进而采用轻量级的网格树(Grid tree)来处理工作负载倾斜，图 6 下半部分展示了 2 维数据的 Tsunami 索引。在每个区域内部通过函数映射(functional mappings)和条件概率分布函数(conditional CDFs)来挖掘属性间的关联性，并构建增广网格(Augmented Grid)加速查询。Flood 和 Tsunami 能够从查询和数据中学习，两种方法的参数都是通过离线模型确定。

然而，Flood 和 Tsunami 是针对内存数据中读优化的静态工作负载，不支持数据的动态更新，Tsunami 相比于 Flood，确定最优网格结构时需要更多训练，且需要挖掘属性间的关联。其次，Flood 和 Tsunami 都需要有真实的查询负载作为训练数据，以确定参数，而实际中在构建索引时，常常不存在大量可用的查询工作负载样本。并且，网格的最优布局，依赖于构建时从数据和工作负载中抽样的比例和抽样的代表性。最后，Flood 和 Tsunami 尚未用来解决多维数据查询中常见的 K 近邻查询。

为支持多维数据的动态更新，Li 等人[13]提出 LISA。不同于之前介绍的针对内存数据设计的索引结构[53-54]，LISA 是针对磁盘驻留的空间数据。已有学习化索引通过模型预测查询关键字的近似存储位置，并通过局部搜索的方式，弥补预测带来的误差，最终定位到精确存储位置。这对于内存数据是高效的，但对于访问成本较大的磁盘数据，会带来较大的 IO 开销。因此 LISA 的主要思想是利用机器学习模型生成可搜索的数据布局，按照模型生成的布局对数据的存储重新组织，进而增加学习化索引查询的性能和预测的准确性。相对于传统的 R-tree 等空间索引，LISA 能够显著降低存储空间占用并减少磁盘 I/O 代价。

LISA 存在以下局限性：首先，LISA 的预测是粗粒度的，模型预测的是查询关键字存储的片(shard)，shard 由多个页(page)组成，在每个 shard 中，利用局部模型(local model)存储的划分值进行局部查找来定位数据所在的 page，最终在 page 中精确搜索关键字；其次，LISA 需要借助两次局部查找，在大数据量情况下，只适合于外存数据，不适合对性能要求更高的内存数据。LISA 能够支持更新，但其训练好的 shard 预测模型是固定的，能很好地支持已有空间内的数据插入，而超出空间范围(out-of-space)的插入会导致性能急剧下降；最后，文中并没有介绍如何选取 shard 预测模型训练时的超参数。

3.2.2　基于数据分区的方法

这类方法的基本思想是将数据划分到不同的子集，在每个子集内进行索引。

RSMI[25]为解决在大数据集上的扩展性问题，采用递归策略(recursive strategy)对大数据集进行分区，并在每个分区中构建索引。RSMI 是针对磁盘驻留的数据，将数据按照块(block，每个 block 最多存储 B 个点)的方式组织。首先 RSMI 假设存在

一个 M 函数，能够实现以较高的准确率将 N 个点映射到$\lceil N/B \rceil$个 block，即预测 N 个点的 block ID；其次，基于该假设，RSMI 将整个数据集递归划分，每次划分$\lceil N/B \rceil$个分区(cell)，直到每个 cell 最多包含 N 个点，并采用空间填充曲线为每一个 cell 获得一个填充值，对 cell 进行排序，同时训练映射函数 M 来实现将点分组到不同的 cell 中。以上步骤能够重用精确的 M 函数，降低拟合数据分布的代价。最后，在每个分区中应用 M 实现点到 blockID 的映射，并记录每个分区中预测的最大误差和最小误差。

相对于完全替换数据库中的索引结构和查询算法，Gu 等人[55-56]从另一个角度，应用机器学习算法增强多维数据结构，并提出 RLR-tree。传统 R-tree 自提出后，已出现较多变体[57-58]，这些方法大都集中在优化选择要插入数据的子树(ChooseSubtree)和如何分裂子树(Split)的启发式规则上，然而并没有一种启发式规则能完胜其他策略。基于此，RLR-tree 的主要思想是保持传统的 R-tree 结构和其查询算法不变，采用机器学习方法以数据驱动的方式构建 R-tree，即应用强化学习来优化子树选择 ChooseSubtree 及分裂子树 Split。

具体地，RLR-tree 采用强化学习，将 ChooseSubtree 和 Split 视为马尔可夫决策过程，以代替原来的启发式规则，保持 R-tree 结构不变，使得基于学习的索引更容易在数据库中部署。以 ChooseSubtree 为例，在数据插入过程中，RLR-tree 的状态、动作及奖励的计算如图 7 所示。奖励的计算依赖于参考树(reference R-tree)，参考树和 RLR-tree 的结构相同，并同时执行插入，插入后执行查询，计算参考树和 RLR-tree 的代价差值 r 作为奖励($R'-R$)。代价的计算如下公式，其中 T_{rl} 为强化学习 R 树，T_r 为参考 R 树，R 为强化学习 R 树的代价，括号中 R' 为参考 R 树的代价：

$$R(R')=\frac{T_{rl}(T_r)\text{访问节点数}}{T_{rl}(T_r)\text{树高}}。$$

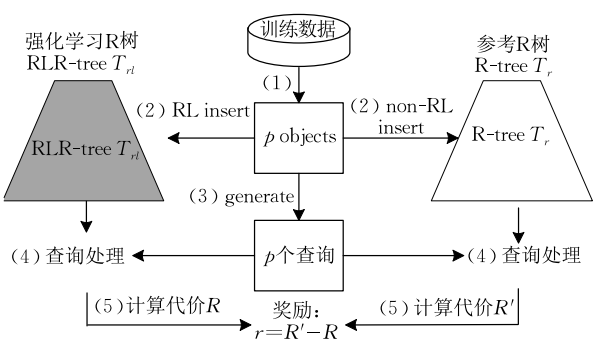


图 7 RLR-tree 训练过程：结合参照 R 树计算奖励

类似于 RLR-tree，即保持 R-tree 的基本结构不变，Yang 等人[59]同样采用机器学习方法增强 R-tree 的结构，提出 PLATON。针对现有的 R-tree 在初始构建(packing/bulk-loading)过程中，依赖于固定的启发式规则，不能适应不同数据分布和查询模式的问题，PLATON 提出基于蒙特卡洛树搜索(Monte Carlo Tree Search，MCTS)的学习型分区策略(Learned partition policy)，并采用分而治之的思想，借助早停和分层抽样的方法，将 MCTS 的复杂度从 $O(k \cdot N^2 \log N)$ 降至线性 $O(k \cdot N)$，其中 N 为数据量，k 为 MCTS 的迭代次数。

RLR-tree 和 PLATON 都保持了 R-tree 的优点，能够有效地支持动态数据插入，同时提高查询性能。但由于保持原来的树结构不变，也继承了原始 R-tree 结构的不足，例如占用存储空间大，尤其在大数据量的情况下。

Kang 等人[60]认为索引一旦建立，查询对象的最近邻分布即是固定的，因此可以采用模型学习该分布。基于此，Kang 等人提出机器学习增强的索引结构(ML-enhanced index)，保持传统的树索引结构不变，训练机器学习模型预测查询对象最近邻在叶子节点出现的可能性，基于预测结果重排扫描叶子节点的顺序，以提高 KNN 查询的召回率。

在轨迹相似性搜索中，R-tree 通过判断查询的最小边界矩形(Minimum Bounding Rectangle，MBR)与节点的 MBR 是否相交，进行剪枝。实际中，针对大数据集，特别当数据分布倾斜时，例如较多轨迹同时覆盖一个公共区域，导致在该区域出现大量 MBR，使得剪枝策略低效。Ramadhan 等人[61]提出基于学习化索引的最小边界矩形剪枝方法 Learned-MBR-prune。该方法首先构建数据集中所有轨迹的 MBR，对于 n 维数据集，需要构建 $2n$ 个 learned index，接着在每一个维度上，对 MBR 的端点进行排序构建 learned index。论文中只给出了主要思想和例子，并没有给出实验评估的结果。由于维数越多，要构建的索引越多，因此可以猜想，Learned-MBR-prune 针对维数较高的多维数据集，代价较大。

Shi[62]提出学习化空间哈希映射(Learned Spatial Hashmap，LSPH)。LSPH 的主要思路是，选取二维数据中方差较大的维度，在选择的维度上训练单调模型来拟合该维度上数据的累积分布函数(CDF)，获得单调的 learned CDF，并将 learned CDF 用作 hashmap 中的哈希函数(hash function)，即 learned CDF 的预测值作为 hash 值。选择方差较大的维度，有利于提

高 hashmap 的空间利用率，同时解决数据的偏态问题。在动态场景下，如果 learned CDF 预测插入点的 hash 值在范围内，则执行插入，反之，则需要重新构建 hashmap，即重新训练哈希函数。因此 LSPH 在解决超范围(out-of-range)插入时具有较高的代价。

Dong 等人[28]采用 KaHIP 图分割器[63]和神经网络，构建神经位置敏感哈希(neural Locality-Sensitive Hashing，neural LSH)分区。具体地，neural LSH 首先构建 K 近邻图，图中每个顶点即多维空间中的一个数据点，每个顶点与其 K 个近邻点相连构成边。接着采用 KaHIP 将 K 近邻图划分到 m 个分区，尽可能保持每个分区内的顶点数相同，并且跨分区的边尽可能少。最后采用有监督的方法，将点和其对应的分区作为训练数据，训练机器学习模型(可选线性模型或神经网络模型)。最终拟合的模型，被用于预测查询点所在的分区。

Guo 等人[64]为优化时空数据的存储问题，提出新的存储模型 Cymo，用于索引时空多维数据，以适应不同的查询模式。主要思路是将时空数据空间划分到多个子空间中，在每个子空间中，借助机器学习模型，捕捉历史查询负载特征，设计最优的存储模型。Cymo 中采用混合神经网络模型，结合卷积神经网络(Convolutional Neural Network，CNN)和长短期记忆网络(Long Short Term Memory Network，LSTM)，能够同时捕获空间和时间关联。同时 Cymo 设计了中间层，用于解决不同存储设备的异构问题。Guo 等人将 Cymo 集成到了 HBase① 和 GeoMesa②，获得了较好的实验效果。

3.3　基于属性间关系的多维学习化索引方法

数据以关系表的形式存储于数据库，每个关系表包含多个属性(列)，关系表的多个属性之间存在相关关系(correlation relations)或软函数依赖(soft functional dependencies)[65-66]。相关关系的存在，使得其中某一列的值能够借助另一列的值近似估计。因此基于属性间关系的多维学习化索引探索利用列关联性来优化索引结构，以达到降低存储空间和加速查询的目的，主要集中在利用关联性构建二级索引(secondary index)和优化空间分区两方面。

Wu 等人[67]利用列之间的相关性提出 HERMIT，一种快速且简洁的二级索引机制，该机制可以用于回答多维查询。HERMIT 设计了分层回归搜索树(Tiered Regression Search Tree，TRS-Tree)，利用简单的统计回归模型拟合属性列之间隐藏的关联函数。

假设列 M 和列 N 之间存在相关性，且 N 上存在构建好的索引结构，由于查询也会频繁在 M 列上过滤，因此数据库管理员想在列 M 上也构建索引，而利用 HERMIT 能够构建一个简洁的 TRS-Tree 来捕捉 M 和 N 之间的关系，使得在列 M 上的查询经由 TRS-Tree 转化为在列 N 上的查询范围。TRS-Tree 是一棵 k 元树，其构建过程递归地将 M 列划分为 k 个子范围，并判断该子范围内 M 和 N 的关联关系是否能够用一个简单的线性回归模型拟合，若满足，该子范围迭代终止，反之则继续划分。

如图 8 所示，(Time，DJ)上已经构建了索引，而 SP 和 DJ 存在关联，因此构建 TRS-Tree 捕捉 SP 和 DJ 的关系，最终 SP 上的查询转化为 DJ 上的查询，进而利用已有的(Time，DJ)索引。

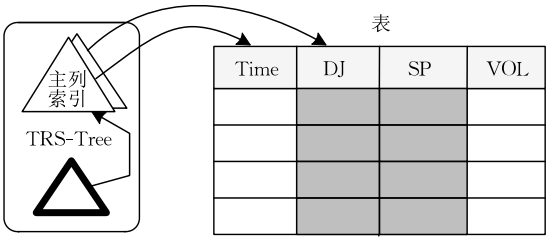


图 8　HERMIT 索引结构利用 DJ 列和 SP 列关联

HERMIT 能够显著降低存储空间占用。对于点查询，HERMIT 需要将列 M 上的查询转化到列 N，带来性能下降，因此 HERMIT 指出 TRS-Tree 较适用于二级索引的构建，因为二级索引上广泛存在范围查询。另外，HERMIT 存在如下局限性，首先其依赖于数据库中存在已经构建好的主列索引，其次要求属性之间存在关联性。由于数据的关联较复杂，同一列上不同范围的数据分布不同，HERMIT 构建的 TRS-Tree 是不平衡树，即其叶子节点的深度可能相差较大，进而带来查询性能的不一致性。

当前 HERMIT 只是考虑利用不同列之间的相关性来降低内存，加速查询。我们认为 HERMIT 机制所依赖的主列索引，可进一步采用机器学习模型拟合，即利用单列上的数据分布，例如借鉴一维学习化索引的相关算法，节约内存资源，提升性能。

在多维数据中，索引结构每增加一个维度就会使得索引的性能进一步恶化。因此，Ghaffari 等人[68]提出方法用于学习数据集中不同列之间的关联，将其称之为学习函数依赖(learned Functional Dependencies，learned FDs)，并采用 learned FDs 对多维数据进行降维，极大地减少用于索引的维度，从而提高索引的查询性能，同时降低内存占用。

① https://hbase.apache.org/
② https://www.geomesa.org/

3.4 本章小结

近年来,机器学习赋能的多维索引结构,涌现出大量研究工作,是当前热门的研究方向。表 1 进一步从数据的存储位置、是否利用属性间关联、支持的查询类型以及是否支持数据更新等方面对现有研究工作进行总结。

表 1 多维学习化索引结构总结

数据存储位置	是否利用属性关联	方法	查询类型	是否支持数据更新	模型类型
内存	否	Learned ZM[36]	点查询、范围查询	否	线性模型
		IF-X[44]	点查询、范围查询		线性模型
		ML-index[42]	点查询、范围查询、K 近邻查询		线性模型
		Learned KD-tree[52]	K 近邻查询否		全连接神经网络模型
		Neual LSH[28]	近似 K 近邻查询		线性模型、神经网络
	是	Flood[53]	点查询、范围查询		线性模型、随机森林
		Tsunami[54]	点查询、范围查询		线性模型、DBSCAN聚类
磁盘	否	RSMI[25]	近似查询:点查询、窗口查询、K 近邻查询	是	线性模型
		LISA[13]	点查询、范围查询		线性模型
		RLR-tree[56]	点查询、范围查询、K 近邻查询		强化学习
		PLATON[59]	点查询、范围查询、K 近邻查询		蒙特卡洛树搜索
		LMSFC[40]	窗口查询	否	贝叶斯优化、线性模型

当下内存的价格不断降低,并且可放入一台机器的主内存量增加,内存数据库变得更加流行[53]。因此多维索引结构的研究针对内存数据较多,但为内存数据设计的学习多维索引不能支持动态更新。

部分方法,如 Flood、Tsunami 依赖于较多参数,这些参数需要在离线训练过程确定。针对磁盘数据的方法能够支持更新,但 RSMI 针对近似查询,并且当频繁数据更新时,需要代价较高地重新构建索引。另一方面,面向磁盘数据设计的多维学习化索引结构,由于访问粒度较粗,因此不适合内存数据。另外的方法,例如 RLR-tree、PLARTON,保持 R-tree 的结构不变,借助机器学习辅助 R-tree 的启发式规则,能实现比传统 R-tree 更好的性能,但同时也继承了传统 R-tree 的不足之处;HERMIT 作为一种二级索引机制支持多维数据查询,利用线性模型拟合属性间的关系,能够显著降低存储资源占用,但 HERMIT 在回答点查询时代价较高,且依赖于已有的主列索引。

现有的多维学习化索引结构大多是针对行式存储结构,Flood 和 Tsunami 是针对内存列式存储结构。在数据分析场景中,列式存储结构已越来越普遍,因此,列式存储方式,是机器学习赋能的多维索引结构研究中值得关注的研究方向。

4 机器学习赋能的数据布局和数据概要研究

随着数据规模的不断增大,传统依赖于大量索引结构的优化策略,给数据管理系统带来极大的挑战,主要在于构建索引导致的存储和维护成本。在某些场景中,空间索引占用的存储空间大小已经超过了数据本身的规模[13]。尤其在海量大数据上构建索引给数据库带来严重的存储和维护成本,同时由于磁盘随机访问造成巨大的 I/O 开销。

基于此,数据库界探索不同的策略,以加速查询处理,本文称之为非索引结构。根据是否访问原始数据,可以分为数据布局和概要构建。接下来首先介绍机器学习在数据布局中的研究工作,接着介绍其在概要方面的应用。

4.1 学习化数据布局

随着数据量的激增,高性能的分析系统,通常要求在秒级时间内回答 TB 级的查询。数据库界逐渐从基于索引的查询处理,转向设计不同的数据分区和组织策略,进而利用硬件的顺序扫描特性,减少随机 I/O 代价[27]。

合理高效的数据布局能够显著提高分析系统的性能,特别是数据密集型的查询,在分布式或云数据库系统中可以由多台机器共同执行[69]。数据分区和布局[29]分别侧重在逻辑层面和物理层面,选择最优的形式在相同物理块或存储介质上组织数据,降低数据访问成本,达到最佳数据库性能。

本节主要介绍机器学习如何应用于数据库构建数据布局,关键思想是利用数据分布和查询工作负载的知识针对不同应用自定义数据布局。

数据布局策略的主要思路是将数据分块存放,以降低扫描的数据读取量。在每个块中,通常构建最大最小索引(min-max index)或分区地图(zone maps),用于在扫描时跳过不相关的数据块。合理调优的数据布局,能够在查询执行期间跳过不相关的块,进而

极大减少访问的块数量[70]。在数据库中，如何设计合理高效的数据布局(data layout)，是近年来的研究热点[71-72]。

为解决将数据记录最优地分配到数据块的问题，Yang等人[27]提出Qd-tree框架，采用强化学习从数据和查询中学习数据布局(Learning data layouts)，如图9所示，Qd-tree主要解决数据组织和查询路由的问题。Qd-tree可以看做是一棵二叉树，其中每个节点对应于整个高维数据空间的一个子空间。树的根对应于整个数据空间。Qd-tree中的数据块能够满足2个重要特性：语义描述和完整性。

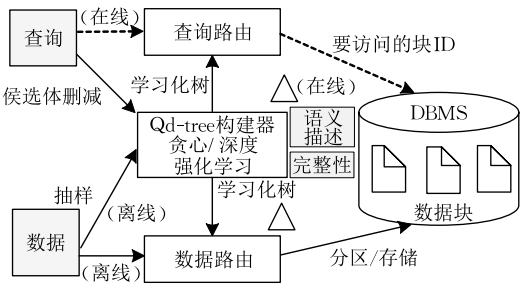


图9　Qd-tree架构

Qd-tree是针对单个表的优化数据布局框架，而实际中，分析工作负载通常包含多个表，并且查询具有不同的连接模式。为解决数据库中多个表的数据布局优化问题，Ding等人[72]在Qd-tree的基础上，提出多表优化(Multi-Table Optimizer，MTO)。在多表数据集和查询工作负载中，MTO利用连接诱导谓词(join-induced predicates)来学习更好的数据布局，使得学习到的布局，能够最大化特定数据集和工作负载上扫描数据块的跳过数量。MTO包含2部分，第1部分实现记录到块的映射，第2部分在执行查询时，确定需要被访问的块和要跳过的块。

不同的分区方案，对于查询处理的性能有显著影响，而传统的分区策略大多依赖于启发式规则，或依赖于查询优化器的估计代价。因此，Hilprecht等人[69,73]针对分析型工作负载，提出基于强化学习的分区工具。强化学习通过尝试不同的分区模式并监控不同工作负载的奖励来学习经验，进而给出分区决策。强化学习的应用，提高了分区工具对于新工作负载的泛化能力，使得分区工具给出的分区策略比仅依赖于成本和规则更准确。

尽管基于强化学习的方法，改进了分区策略，Zhou等人[74]认为在分布式场景下，现有的强化学习方法不能捕捉复杂的数据分布和查询模式，并且为训练模型，需要重复将数据划分到不同的计算节点，以评估分区性能，导致时间和资源的浪费。因此，Zhou等人提出Grep，基于图学习(Graph Learning)的数据库分区系统。Grep采用图模型编码数据和查询模式，以获得向量，并利用图神经网络将分区因子嵌入到向量中，结合查询相关性，最后Grep训练关键字选择模型，用于选择合适的分区关键字，并提出代价模型评估分析性能，而不需要实际对数据分区。Grep已集成到华为分布式数据库GaussDB[75]中。

机器学习应用于数据分区策略的研究还有诸多相关工作，例如Vu等人[76]采用深度学习方法(全连接神经网络和卷积神经网络)为大规模数据给出合适的空间划分技术，Hori等人[77]将强化学习应用于空间数据分区(RL-Spatial Data Partitioning，RLSDP)，借助空间数据特征找到最优分区。Durand等人[78]提出GridFormation，同样也是应用强化学习给出数据分区方案等。读者可以进一步阅读该方向介绍智能化分区和数据布局的详细综述[30]。

4.2　学习化概要

在大规模多维数据库中，即便是简单查询，访问原始数据的代价也非常昂贵[14]，特别是当数据存储在磁盘设备时。为解决该挑战，很多应用程序借助于具有可接受误差的近似查询处理(Approximate Query Processing，AQP)来实现性能和代价的平衡[79]。AQP[79]可分为在线和离线两种处理模式。

在线聚集主要是借助抽样算法，选择部分查询元组，求得近似结果。这方面工作主要是对抽样策略的改进，例如在抽样时考虑类别不均衡的问题、权衡抽样数量与查询精度等。PFunk-H[80]将视觉感知模型应用于在线抽样算法，主要思想是借助简单的感知模型自动决定查询结果的置信区间，以提高近似查询处理结果的有效性和准确性。在线聚集领域应用机器学习算法的研究相对较少，因此本文侧重离线概要。

在多维数据场景中，现代分析系统通常以离线方式构建和维护数据概要(data synopsis)，以减少对原始数据的访问。概要数据可以视为对原始完整数据的紧凑表示。通常概要数据存放于内存中，以快速响应查询。在概要数据的基础上，可以在错误度容忍范围内高效地回答特定查询，而无须代价高昂地访问原始数据。例如，多维数据直方图及其变体[14]是空间数据常用的概要形式，可用于回答计数查询，也广泛应用在查询基数估计、选择性估计、空间数据

分割、空间数据挖掘等场景中。

因此，数据概要广泛应用于提高大型空间数据库的查询处理速度。随着机器学习的不断发展，用于近似回答查询结果的数据概要已经不再仅限于对数据建模，也包含对查询进行建模。因此本节从对数据建模和对查询建模两方面进行介绍。

4.2.1 建模数据概要

建模数据概要，最常见的做法之一是采用机器学习模型对多维联合数据分布进行建模。例如 Yang 等人[24]和 Vorona 等人[81]采用深度自回归模型[82]拟合多维数据，用于地理空间查询的近似处理。其中 Yang 等人[24]为捕捉数据中存在的多维分布，提出采用深度自回归模型作为统计模型，并设计蒙特卡洛集成方案，构建数据概要，能够高效处理多维度范围查询。同时，Yang 等人提出的方法相对于传统的数据概要方法具有优势，其不需要任何的独立性假设来近似联合分布。Vorona 等人[81]提出基于深度学习的近似地理数据空间查询引擎 DeepSPACE，以解决空间数据量不断增加对数据处理和分析能力带来的挑战，并实现降低存储资源占用的目的。DeepDB[83]采用离线方式，训练关系和积网络（Relational SumProduct Networks，RSPN）用于学习数据的表示。同时，DeepDB 指出，采用 RSPN 作为数据的表示，不是为取代原始数据，而是为加速查询并在原始数据查询之上增加更丰富的查询功能。

随着维数的增大，神经网络的训练时间急剧增加。为解决卷积神经网络 CNN 训练代价较大的问题，一些方法[84-85]采用极限学习机（Extreme Learning Machine，ELM）来增强 CNN。例如，Zeng 等人[86]为解决 CNN 训练时间过长的问题，提出 ECNN，在模型架构中去掉了池化层，以 ELM 层来代替。

Zhao 等人[87]基于 ECNN 模型，提出 EDense 网络模型，使用密集连接（dense connectivity）集成了 ELM 和 CNN，并扩展了网络的层数，使得在提高训练速度的同时具有较强的学习能力。EDence 能够有效地用于空间数据学习任务中，支持快速且准确的空间数据查询分析。

基于邻近关系的多维数据分析，主要瓶颈为数据检索。K 近邻距离是基于邻近关系的数据分析中最常见的应用之一。尽管已有大量的分析工具，这些方法还是面临大量的数据访问代价。

为避免大量数据访问的代价，Amagata 等人[88]采用机器学习快速预测 K 近邻距离。其主要思路是训练全连接网络 PivNet，实现精确估计。具体地，针对多维数据空间，在空间中选取多个枢轴（pivot），预先计算所有 pivot 的 K 近邻距离。对于给定查询点，首先找到距查询点最近的 pivot，利用 pivot 的 K 近邻进行估计。PivNet 通过确定合理数量的 pivot，并将所有 pivot 及其 K 近邻距离作为训练数据，构建回归网络，在执行查询时不需要访问数据，能够实现以 $O(1)$ 复杂度的模型推测，估计 K 近邻距离。

虽然这些基于深度学习的方法在查询精度上有明显提升，也有部分方法探索降低深度神经网络模型的训练代价，然而，深度学习的训练通常依赖于专家知识且存在大量参数调优。

为解决如上的问题，Liu 等人[14]应用简单模型（包括线性模型、分段线性和多项式模型），构建数据概要，并探索效率和累积分布函数（CDF）拟合模式之间的平衡，提出 LHist。LHist 将一维学习化索引 RMI[8]的思想和多维等深直方图结合，是一种新的学习化多维直方图。

图 10 展示了针对 4 维数据的 3 路 LHist[14]（3 路，即每个维度划分的分区个数）。不同于直接拟合多维数据分布，LHist 首先采用基于模型的分区模式对 $d-1$ 维进行分区，将整个数据集划分到不同的桶中；接着在每个桶内，针对第 d 个维度训练模型拟合 CDF。LHist 中的分区策略和 CDF 拟合都是借助具有层次结构的简单机器学习模型完成，以利用数据分布。同时，LHist 提出模型选择和训练方法，能够在准确率和存储代价之间权衡。

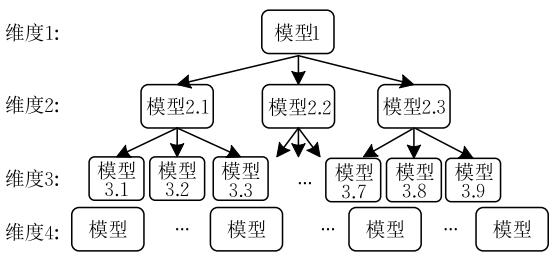


图 10 针对 4 维数据的 3 路 LHist 示例

数据直方图的主要应用场景之一是选择性估计，以帮助数据库针对复杂的多维数据查询选择最佳的查询计划。选择性估计的主要难点在于高效且准确地借助多维数据的分布，估计查询结果的大小，对于查询优化、近似查询处理、数据库调优等至关重要。尽管传统的数据直方图能够实现快速的查询推断，但基于直方图的数据分布需要假设多个维度之间具有独立性，使得估计的准确性较差。针对多个

连续属性且具有大值域范围的场景，为解决基于直方图的方法具有较大估计误差的问题，Meng 等人[89]提出 IAM，集成多个高斯混合模型和深度自回归模型建模数据关联，并学习不同维度的联合数据分布，用于给出更加准确的选择性估计。

另有部分研究采用生成数据的方式，构建数据概要。例如，生成对抗网络（Generative Adversarial Networks，GAN）[90]能够建模高维联合概率分布，因此 Fallahian 等人[91]提出基于 GAN 的数据生成器，基于已有真实数据生成合成数据集，用于构建数据概要，回答近似查询处理请求。采用 GAN 生成的数据更贴近真实数据，因此能够获得较好的性能。其他基于生成数据的方法还包括文献[92-93]。基于生成数据的方法，常用于数据发布场景，旨在保护用户或企业数据隐私，防止数据泄漏。

在数据类型方面，除传统的多维时空数据，视频数据经由目标检测算法生成视频关系表，同样可被视为具有时间和空间关联的多维数据。针对视频数据查询处理，目前也涌现了大量机器学习算法。例如针对前 K 查询（Top-K query），Everest[94]借助卷积混合密度模型提供具有概率保证的查询结果；Noscope、Probabilistic predicates 等[95-96]借助卷积神经网络支持在视频数据上的对象选择查询。

以上的数据概要研究侧重于回答近似查询处理问题。Zhou 等人[97]提出 DeepMapping，一种新的数据抽象（data abstraction），借助神经网络和辅助数据结构，回答精确查询处理，由神经网络实现对大部分数据的记忆，而数据结构记忆少量模型错误分类的数据。具体地，DeepMapping 采用多任务神经网络（multi-task neural networks）建模关键字到多个属性值的映射，作为数据的近似表示，快速响应查询。为满足查询的零误差，DeepMapping 设计辅助数据结构，以实现精确的查找、更新、插入等操作，包含用于表示数据存在性的位向量，以及存储模型错误预测和新插入数据的辅助表。DeepMapping 采用具有共享层（shared layers）的多任务神经网络实现降低模型占用存储空间的目的，并为每一个属性设计私有层（private layer）保证预测性能。然而，当表中的属性数目较多时，模型仍然占用较大的空间。另外，当数据更新频繁时，辅助数据结构的查询性能会显著下降，因此 DeepMapping 需要重新训练模型以保证性能。同时，DeepMapping 集成了数据压缩技术，更进一步地降低存储占用。

4.2.2　建模查询概要

机器学习应用于查询概要建模，旨在挖掘查询语句、查询结果，或者两者之间的关系，用于快速回答查询。

每个查询的结果都会揭示关于其他查询结果的部分知识，即便每个查询访问数据的不同元组和列。基于该观察，Park 等人[98]提出数据库学习（Database Learning，DBL）的概念，并将 DBL 应用于近似查询处理，提出 Verdict。主要解决思路是将过去查询的近似答案视为观察结果，并使用它们来改进对底层数据的后验知识，从而加速未来的查询。

图 11 展示了 Verdict 的工作流程，对于待处理查询，首先获得原始的近似查询结果和误差，后经过查询概要与 DBL 模型，获得改进后的查询结果和误差，支持聚集查询、连接、选择、分组操作等。

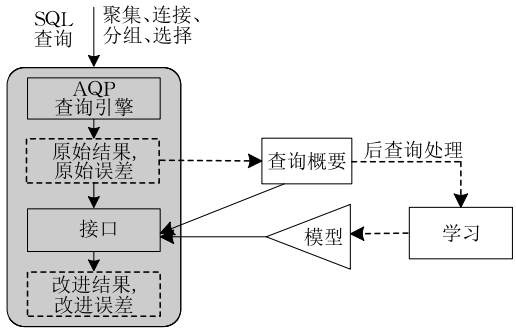


图 11　Verdict 工作流程

Verdict 主要利用查询和查询之间的关系，Regev 等人[99]认为查询和结果之间也存在隐含关系，并基于此，提出 Hunch[99]，借助 LSTM 神经网络学习结构化查询语句（Structured Query Language，SQL）和查询结果之间的关系，用于快速预测查询结果。Hunch 分为 3 个步骤：第 1 步随机生成大量查询，作为训练集；第 2 步采用嵌入方法对查询进行表示；第 3 步训练模型，并用于预测查询结果。Hunch 同样解决近似查询处理问题，因此训练集包含大量分析类型的聚集 SQL 查询。ML-AQP[100]针对云系统服务，从先前回答的查询中提取知识，训练梯度提升模型（Gradient Boosting Machines，GBM），有效估计新查询结果并降低系统资源使用。

回归模型（Regression models）在数据库中的应用十分广泛，例如查询预测、数据分析、数据补全等[101-104]。在众多回归模型中，选择最优模型是挑战，也是亟待解决的问题。Ma 等人[105]实验验证了已有的众多模型在不同数据集，甚至相同数据集不同

子集上的表现都是不同的。基于此，Ma 等人提出以查询为中心的回归模型(Query-centric Regression model，QReg)，QReg 包含已有的众多模型，对于特定查询，选择具有最小估计误差的回归模型。

Hasan 等人[106]探索用深度学习模型解决多维查询中的选择性估计(selectivity estimation)问题，提出 MADE＋Sup。多维查询由于不同的属性间具有关联性，使得该问题更具挑战性。因此，Hasan 等人将选择性估计分别建模为无监督学习和有监督学习问题，提出了两种互补的方法，即 MADE＋Sup。文章提出的方法分 2 个阶段，第 1 阶段为离线阶段，从数据和查询中训练机器学习模型，包括自回归模型和全连接神经网络；第 2 阶段为在线应用阶段，对于给定查询，模型预测其选择性。值得注意的是，对于缺少真实查询训练集的场景，文章中也给出了解决方案。同时，文章提出的有监督学习方法可以在真实数据不可访问的情况下，只借助查询训练集快速训练深度学习模型。同样地，QuickSel[107]从查询训练集中学习混合模型以支持选择性估计。

4.3 本章小结

数据布局和数据概要作为非索引结构，都能够缓解精细化索引结构带来的存储和维护代价。数据布局有利于借助硬件的顺序扫描特性。学习化数据布局，主要借助数据分布和查询分布优化数据存储布局，使得在查询时最大化跳过不相关块，减少扫描代价。数据概要的主要目的是借助机器学习模型构建数据的紧凑表示，避免访问原始数据。

表 2 从是否访问原始数据和模型种类两方面，总结机器学习方法在多维非索引结构中的应用。多维学习化索引中，所采用的机器学习方法，大多为简单的线性模型(如表 1 所示)。而在数据布局、数据摘要等非索引结构中所采用的机器学习方法大多为神经网络。

另外，访问原始数据的方法，支持精确查询结果，也可支持近似查询结果。而不访问原始数据，只访问模型(概要)获得近似查询结果。强化学习在访问原始数据的数据布局研究中应用较多，主要思路是将数据分区存放的问题转换为序列决策问题，进而定义状态、动作、奖励等。数据概要应用场景中，在执行查询时通常不需要访问原始数据，只需要借助在数据或查询上训练获得的概要即可，这使得神经网络模型在该研究领域的应用较多，适合于近似查询处理。其次，在模型拟合的数据概要研究中，如果要回答精确查询处理，需要借助辅助数据结构，例如 DeepMapping[97]采用模型实现快速预测，设计辅助结构弥补模型预测带来的错误结果。

表 2 非索引结构总结

是否访问原始数据	方法	模型种类
访问	Qd-tree[27]、RL-Partition advisor[69]、RLSDP[77]、GridFormation[78]、MTO[72]	强化学习
	Vu[76]	全连接神经网络模型、卷积神经网络
	Grep[74]	图学习
	DeepMapping[97]	多任务神经网络
不访问	Naru[24]、QReg[105]、DeepSPACE[81]	自回归模型/回归模型
	Everest[94]、Noscope[95]、Probabilistic predicates[96]	卷积神经网络模型
	PivNet[88]	全连接神经网络模型
	Verdict[98]	非参数概率模型
	MADE＋Sup[106]	自回归模型、全连接神经网络模型
	Hunch[99]	长短期记忆网络
	ML-AQP[100]	梯度提升模型
	LHist[14]	线性模型、分段线性模型、多项式函数

5 多维数据查询处理方法评价

多维查询处理算法的评价，通常借助算法在不同数据集和工作负载上的性能表现来评价。本节首先介绍常用的多维数据集、工作负载，接着，概括文献中采用的评价指标，并总结在机器学习赋能的查询处理场景中所特有的指标。

5.1 多维数据查询处理数据集

表 3 列出了多维数据查询处理研究中常用的数据集，包含数据的维度、数据规模、使用的研究文献以及对该数据集的简短介绍。

真实数据集与合成数据集在多维数据查询处理中都得到广泛使用。其中合成数据集，常见的分布有均匀(uniform)、倾斜(skewed)、高斯(Gaussian)、正态(normal)、齐普夫(Zipfian)等。在使用合成数据集的文献中都写明了数据集的合成规则。空间数据集的维度为 2，其他数据集的维度呈现 2～70 不等。在数据规模上，大多数的数据集在百万级以上，这也符合当前大数据和网络时代数据的特征。不同的数据集通常具有不同的分布，并且随着时间推移，相同的数据集也会出现不同的分布，这些数据分布对数据结构的优化具有重要意义。

表 3　数据集及介绍

数据集	维数	数据集中的记录数	采用数据集的文献	数据集描述
POST	2	123 593	Learned ZM	美国东北部邮局位置数据集
sales	6	30 million	Flood	某大型大公司在匿名条件下捐赠的销售数据集
osm①	6	105 million	Flood, RSMI, LHist, RLR-tree, LMSFC, PLATON	美国东北部的开放地图(OpenStreetMap)项目提供的数据集
perfmon	6	230 million	Flood, Tsunami	美国一所大学管理的所有机器一年内的日志
ErrorLog-Int	50	100 million	Qd-tree	大型软件供应商收集的内部客户使用过程中的错误日志
ErrorLog-Ext	58	81 million	Qd-tree	大型软件供应商收集的外部客户(应用程序)使用软件的崩溃转储日志
Tiger	2	17 million	RSMI, piecewise SFCs	代表美国东部 18 个州地理特征的数据集,每个地理位置采用矩形框表示
stocks②	7	210 million	Tsunami, LMSFC	从 1970 年到 2018 年,6000 只股票的每日历史股价
Taxi③	9	184 million	Tsunami, LMSFC	2018 年和 2019 年美国纽约市黄色出租车的出行记录
imis-3months	2	98 million	LISA	由 IMIS Hellas SA 公司收集的交通数据,并捐赠用于研究目的
ImageNet	6	72 million	LISA	ImageNet 数据集中每个图片所有像素的三通道值
wesad④	12	63 million	LHist	由可穿戴设备记录的生理和运动数据集
Twiter⑤	2	20 million	SPRIG, IAM	推特网站中推文位置的数据集
Census⑥	14	50 K	MADE+Sup	美国统计局人口普查数据集
IMDB⑦	17	8.7 million	MADE+Sup, IAM	包含大量的关于电影和相关演员、导演、制作公司等信息的数据集,这些数据可免费用于非商业用途
TPCH⑧	up to 70	based on the scale factor	Flood, Qd-tree, Tsunami, LHist, MADE+Sup, DeepMapping	由一组面向业务的临时查询和数据修改操作组成,是公开的决策支持基准
Synthetic datasets	—	—	RSMI, Learned ZM, LISA, LHist, RLR-tree, PLATON	合成数据集,数据集的分布包含均匀、正态、倾斜、齐普夫、高斯等

5.2　多维数据查询处理工作负载

工作负载有多种分类方式。例如,依据是否支持数据的插入更新以及插入更新的比例,可以将工作负载分为只读型(read-only)、读多型(read-heavy)、读写平衡型(balanced)、写多型(write-heavy)以及只写型(write-only)。

多维数据处理中,另一种常见方式是依据查询任务对工作负载进行分类。常见的查询任务包括:点查询(point query)、范围查询(range query)、窗口查询(window query)、K 近邻查询(K Nearest Neighbor query,KNN)等。同时还包括在数据集上进行的分析任务,例如均值、计数、求和等聚集查询任务[91]。根据查询结果的精度可分为精确查询和近似查询。

从这些分类中,可以看出多维数据查询处理的工作负载具有多样性。在实际应用中,不同的应用场景可能具有不同的负载,因此设计工作负载感知的索引结构具有重要价值。

5.3　多维数据查询处理评价指标

多维数据查询处理作为数据库管理系统中普遍存在的应用,其性能评价指标通常沿用数据库的通用指标。评价一般分为 3 方面:其一是存储代价(storage overhead),即数据结构占用的存储空间大小;其二是时间代价(time cost),包括数据结构的构建时间、执行插入或查询的延迟(latency)等;其三是查询有效性(efficiency),即在近似查询处理中查询的准确率和召回率等。

表 4 总结了现有多维查询处理中采用的评价指标。大部分方法都考虑了数据结构占用的存储代价。在查询时间的评估方面,有不同的评价方法,例如平均查询执行时间,即批量执行查询,计算平均每个查询的执行时间(总执行时间/总查询数);吞吐量(throughput),即单位时间内执行的查询数量(总查询数/总执行时间);I/O 代价,即获取查询结果需要访问磁盘页或块的数量。

在这些评价指标的基础上,也存在诸多延伸,以进一步评价数据结构的性能。例如 HERMIT[67] 针对查询时间进行分解(performance breakdown),将查询时间划分为 TRS-Tree 推理时间、主列索引查询时间及验证时间 3 部分;Flood[53]、Tsunami[54]、

① https://planet.openstreetmap.org/
② https://www.kaggle.com/ehallmar/daily-historical-stock-prices-1970-2018
③ https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
④ https://archive.ics.uci.edu/ml/machine-learning-databases/00465/
⑤ https://developer.twitter.com/en
⑥ http://www.census.gov
⑦ https://www.imdb.com/
⑧ http://www.tpc.org/tpch/

表 4 多维数据查询处理评价指标

分类	度量标准	评估该标准的方法
存储代价	索引大小或结构大小(占用内存或占用磁盘)	Learned ZM、LISA、SPRIG、LHist、RSMI、HERMIT、RLR-tree、DeepMapping
时间代价	数据结构的构建时间、模型训练时间	LHist、Tsunami、HERMIT、SPRIG、RSMI、Qd-tree、QReg、RLR-tree、PLATON
	查询处理时间、查询响应时间	Flood、Tsunami、Learned ZM、LHist、RSMI、QReg、piecewise SFCs、PLATON、DeepMapping
	吞吐量	Qd-tree、HERMIT
	访问磁盘的 I/O 代价	RLR-tree、LISA、RSMI、piecewise SFCs、PLATON
有效性	估计的准确率	LHist、QReg
	召回率(窗口查询或 K 近邻查询中)	RSMI、ML-enhanced index

piecewise SFCs[41]等考虑了不同数据规模以及数据分布或工作负载偏移对结果的影响;Qd-tree[27]评估了在不同线程数量时吞吐量的变化;在评价范围查询的性能时,SPRIG[50]、LHist[14]等考虑了查询选择性对结果的影响。

5.4 学习化数据结构评价新指标

在机器学习赋能的场景下,传统多维数据查询处理的评价指标,虽依然适用,但存在欠缺。作为数据库中学习化的组件,机器学习赋能的数据结构,旨在借助机器学习模型来适应不断变化的工作负载和数据分布。因此,需要设计新的指标来评价并捕捉学习化组件的新特性[108]。

模型的训练时间,是衡量机器学习赋能数据结构的重要指标之一。如表 4 所示,多数方法在时间代价方面,都测试了模型训练时间。如 RLR-Tree[56]指出其在大数据集上的训练较慢,因此 RLR-tree 提出使用小规模数据集作为训练集,并在模型训练时间和查询性能之间寻找平衡。另外,RLR-tree 指出可以借助图形处理单元(Graph Processing Units, GPU)来加速训练过程。LHist[14]指出在百万级的数据集上,由于额外的模型训练,LHist 构建多维直方图的时间略高于传统的多维等宽直方图,但相对基于深度无监督学习方法的 Naru[24],LHist 的时间代价要低得多。

查询搜索时间依赖于模型的预测准确率,而学习化索引结构通常采用分而治之的思想,包含多个模型,每个模型负责不同的数据分片,这使得不同查询的响应时间不同,甚至由于数据分布差异,导致模型预测准确率相差较大。因此,简单地以平均查询时间作为指标不足以衡量这一特性。基于此,学习化索引结构已经开始采用批量执行查询,统计查询时间的中位数,75%、90%及 95%分位数的查询时间来衡量查询性能,例如针对超内存数据库的一维学习化索引[109]在实验中评估了中位数和 95%分位数,针对磁盘数据库的一维学习化索引[110]采用尾延迟(tail latency)来衡量查询性能的一致性。而在学习化多维数据结构依然沿用平均查询时间,还缺少对于数据分布带来的性能差异度量,即衡量学习化结构对不同数据或查询分布的适应性。

6 总结与展望

综上,本文进行总结与展望,以期给出该方向面临的挑战和未来值得探讨的研究方向。

6.1 总 结

本文介绍了机器学习赋能的多维数据查询处理,提出该方向研究框架,并总结该领域最新的相关工作。机器学习赋能的多维数据查询处理是当前较热的研究方向,我们在概括现有研究现状的基础上,总结出以下不足和挑战。

首先,机器学习赋能的多维数据查询处理中所考虑的数据类型较为单一,大部分集中在多维点数据,并且较少工作考虑查询语句的语义信息;其次,当前学习化多维数据结构解决的场景较少,普遍集中在传统的数据库领域;再者,学习化多维数据结构将机器学习方法嵌入到数据结构中,极少利用不断发展的新硬件,并且现有工作尚未考虑机器学习模型的资源问题;最后,机器学习赋能的多维数据查询处理面临着评价困难的问题。

6.2 展 望

基于如上的总结,我们认为机器学习赋能多维数据查询处理还存在诸多未探索且有价值的方向,值得学术界和工业界进一步研究。

6.2.1 复杂数据类型与语义信息

(1)非点类型空间数据。已有研究大多针对空间中的点数据,然而,非点(non-point)空间数据在实际应用中同样普遍存在,如多边形、线段等[111]。非点数据面临着独有的挑战,例如,在网格划分中,需要对跨边界的同一个空间对象进行切分或重复存放。因此,借助机器学习模型优化非点数据的查询

处理在实际应用场景中具有研究价值，并且面临更大挑战。目前，机器学习赋能的非点数据查询处理研究较少，是值得关注的研究方向。

(2) 建模语义描述。在查询和数据分析等场景中，查询语句通常包含语义(semantics)，而原始的数据并不能直接反映语义信息。现有的研究工作中能够建模语言描述的还较少，Qd-tree[27] 通过保持语义描述的完整性实现查询剪枝操作。由于语言信息的复杂性，使得该问题具有挑战性。设计建模语言描述的多维数据结构，是值得探讨的未来研究方向，大语言模型的发展，或许能提供解决思路。

6.2.2　新应用场景的多维数据结构与算法

(1) 并行处理场景。现有研究，尤其是机器学习赋能的多维索引结构，只考虑了单线程场景。实际应用场景中，对多维数据的并发访问和处理是提高性能的重要手段，因此机器学习赋能的多维数据结构中，考虑并发处理十分必要。

(2) 新型数据库场景。随着机器学习算法的发展，不断涌现出机器学习与数据库技术结合的新型解决方案，诸如模型数据库[112]、向量数据库[113]、神经数据库[114-115] 等。在软件 2.0 时代，机器学习模型成为系统的重要组成部分，随着大模型的发展，数据的嵌入(embedding)表示形式——向量(vector)变得更加重要，促成了新型向量数据库①；神经网络模型的不断发展和应用，促成了模型数据库和神经数据库②。多维数据查询处理如何利用这些新型数据库技术，并借助机器学习模型优化新型数据库中的多维数据处理，是值得探讨的未来研究方向。

(3) 内存动态场景。当前内存空间不仅变得越来越大，价格也在逐渐下降，并且在很多大数据应用程序中，空间数据可以很容易地放入内存中[111]。而针对大规模空间数据的内存管理研究关注还较少。因此，针对内存多维数据，借助机器学习模型，优化数据的查询处理，具有广泛的研究空间。从研究方向上，动态场景下的多维数据处理不仅面临查询需要，也面临插入、删除、修改的需求。LISA 考虑了数据驻留在磁盘时的动态场景，但在内存数据场景下，这方面的研究较少。

6.2.3　新硬件特性与机器学习算法

(1) 结合新硬件。新的硬件在不断发展，如 GPU、TPU 等硬件加速单元，以及非易失存储等设备。多维数据查询处理中，如何结合这些新硬件特性，需要在设计数据结构时重新考虑，以利用新硬件的优势。例如现代 CPU 的计算能力相对于内存带宽的改善更加显著，因此算法应考虑最大限度使用 CPU 的算力，减少数据的访问。此外，传统的数据结构，已经有诸多方法利用硬件特性，例如单指令流多数据流(Single Instruction Multiple Data，SIMD)、缓存局部性(cache locality)等。应用机器学习模型优化多维数据结构时，也可以从这些方面考虑优化，并在设计时考虑硬件特性。

(2) 低资源低成本。机器学习赋能的多维数据结构，尤其数据概要等研究，通常依赖于大量的训练数据，并且需要代价较高的训练过程。线性模型代价较低，也有部分工作探讨采用多个线性模型集合辅助多维数据查询处理，但线性模型的拟合能力有限。数据结构的构建时间、占用存储空间是多维数据查询处理中常用的评价指标。因此，探索低资源、低成本的模型架构，用于构建多维数据结构是值得关注的未来研究方向。

6.2.4　多维学习化结构评估基准

机器学习赋能的索引结构研究，是当前较热的研究方向，涉及一维数据和多维数据。针对一维结构已经有较多工作对现有的方法进行实验评估，例如 Marcus 等人[116] 采用只读型工作负载评测一维学习化索引的性能，GRE[117]、TLI[118]、CLIP[119] 对一维学习化索引进行了更为全面的实验评估，实验涉及不同的工作负载(不同的读写比例)和数据集，并考虑了并发场景下各算法的性能优劣。同时 GRE 提出数据硬度(data hardness)来度量数据集被拟合的难易程度，CLIP 在非易失存储设备中评估学习化索引的性能。一维学习化索引结构有效的理论证明[120-121] 分别发表在 ICML2020 和 ICML2023 国际会议。然而，多维学习化索引结构研究中还缺少全面的评估。具体体现在，多维学习化索引研究已有较多工作[122-123]，Pandey 等人[124] 从实验角度初步评估了学习化空间索引的性能，比较了 Flood 和 5 种传统索引的性能，尚未考虑其他结构。其次，索引的性能依赖于数据分布、工作负载类型、硬件环境等，目前还缺少系统的实验评估。系统完整的实验评估能够为数据库管理员、系统开发人员等提供索引选择上的指导。正如第 5.4 节所述，学习化多维数据

① https://www.pinecone.io/
② https://www.thirdai.com/neural-databases-a-next-generation-context-retrieval-system-for-building-specialized-ai-agents-with-chatgpt/

查询处理,需要设计新的评价指标,例如衡量数据结构对数据分布或查询分布偏移时的适应能力[125],并设计新的基准测试(benchmark)用于衡量机器学习赋能场景下数据结构的性能。

参 考 文 献

[1] Sabek I, Mokbel M F. Machine learning meets big spatial data (revised)//Proceedings of the 2021 22nd IEEE International Conference on Mobile Data Management (MDM). Toronto, Canada, 2021: 5-8

[2] Wang W, Zhang M, Chen G, et al. Database meets deep learning: Challenges and opportunities. ACM SIGMOD Record, 2016, 45(2): 17-22

[3] Ré C, Agrawal D, Balazinska M, et al. Machine learning and databases: The sound of things to come or a cacophony of hype?//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Melbourne, Australia, 2015: 283-284

[4] Meng Xiao-Feng, Ma Chao-Hong, Yang Chen. Survey on machine learning for database systems. Journal of Computer Research and Development, 2019, 56(9): 1803-1820(in Chinese)
(孟小峰, 马超红, 杨晨. 机器学习化数据库系统研究综述. 计算机研究与发展, 2019, 56(9): 1803-1820)

[5] Sun Lu-Ming, Zhang Shao-Min, Ji Tao, et al. Survey of data management techniques powered by artificial intelligence. Journal of Software, 2020, 31(3): 600-619(in Chinese)
(孙路明, 张少敏, 姬涛等. 人工智能赋能的数据管理技术研究. 软件学报, 2020, 31(3): 600-619)

[6] Wu C, Jindal A, Amizadeh S, et al. Towards a learning optimizer for shared clouds. Proceedings of the VLDB Endowment, 2018, 12(3): 210-222

[7] van Aken D, Pavlo A, Gordon G J, et al. Automatic database management system tuning through large-scale machine learning//Proceedings of the 2017 ACM SIGMOD International Conference on Management of Data. New York, USA, 2017: 1009-1024

[8] Kraska T, Beutel A, Chi Ed H, et al. The case for learned index structures//Proceedings of the 2018 ACM SIGMOD International Conference on Management of Data. Houston, USA, 2018: 489-504

[9] Hu Huan. Efficient Algorithms for Approximate Aggregation and Nearest Neighbor Queries Over Multi-Dimensional Data [Ph.D. dissertation]. Harbin Institute of Technology, Harbin, 2021(in Chinese)
(胡欢. 多维数据上近似聚集和最近邻查询的高效算法[博士学位论文]. 哈尔滨工业大学, 哈尔滨, 2021)

[10] Guttman A. R-trees: A dynamic index structure for spatial searching//Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data. Waterloo, Canada, 1984: 47-57

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

[12] Muralikrishna M, DeWitt D J. Equi-depth histograms for estimating selectivity factors for multi-dimensional queries//Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data. Austin, USA, 1988: 28-36

[13] Li P, Lu H, Zheng Q, et al. LISA: A learned index structure for spatial data//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland, USA, 2020: 2119-2133

[14] Liu Q, Shen Y, Chen L. LHist: Towards learning multi-dimensional Histogram for massive spatial data//Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE). Chania, Greece, 2021: 1188-1199

[15] Behrend A, Dignös A, Gamper J, et al. Period index: A learned 2D hash index for range and duration queries//Proceedings of the 16th International Symposium on Spatial and Temporal Databases. Vienna, Austria, 2019: 100-109

[16] Galakatos A, Markovitch M, Binnig C, et al. FITing-tree: A data-aware index structure//Proceedings of the 2019 ACM SIGMOD International Conference on Management of Data. Amsterdam, The Netherlands, 2019: 1189-1206

[17] Ferragina P, Vinciguerra G. The PGM-index: A fully-dynamic compressed learned index with provable worst-case bounds. Proceedings of the VLDB Endowment, 2020, 13(8): 1162-1175

[18] Tang Chuzhe, Wang Youyun, Dong Zhiyuan, et al. XIndex: A scalable learned index for multicore data storage//Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. San Diego, USA, 2020: 308-320

[19] Wu Jiacheng, Zhang Yong, Chen Shimin, et al. Updatable learned index with precise positions. Proceedings of the VLDB Endowment, 2021, 14(8): 1276-1288

[20] Li Pengfei, Hua Yu, Jia Jingnan, Zuo Pengfei. FINEdex: A fine-grained learned index scheme for scalable and concurrent memory systems. Proceedings of the VLDB Endowment, 2022, 15(2): 321-334

[21] Saxena G, Rahman M, Chainani N, et al. Auto-WLM: Machine learning enhanced workload management in Amazon Redshift//Proceedings of the 2023 International Conference on Management of Data. Seattle, USA, 2023: 225-237

[22] Mao H, Schwarzkopf M, Venkatakrishnan S B, et al. Learning scheduling algorithms for data processing clusters//Proceedings of the ACM Special Interest Group on Data Communication. Beijing, China, 2019: 270-288

[23] Kaftan T, Balazinska M, Cheung A, Gehrke J. Cuttlefish: A lightweight primitive for adaptive query processing. arXiv preprint arXiv:1802.09180, 2018

[24] Yang Zongheng, Liang E, Kamsetty A, et al. Deep unsupervised cardinality estimation. Proceedings of the VLDB Endowment, 2019, 13(3): 279-292

[25] Qi J, Liu G, Jensen C S, Kulik L. Effectively learning spatial indices. Proceedings of the VLDB Endowment, 2020, 13(12): 2341-2354

[26] Zhang S, Ray S, Lu R, et al. Spatial interpolation-based learned index for range and kNN queries. arXiv preprint arXiv:2102.06789, 2021

[27] Yang Z, Chandramouli B, Wang C, et al. Qd-tree: Learning data layouts for big data analytics//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland, USA, 2020: 193-208

[28] Dong Y, Indyk P, Razenshteyn I P, Wagner T. Learning space partitions for nearest neighbor search//Proceedings of the ICLR 2020. Vienna, Austria, 2002(online)

[29] Zhang Zhou, Jin Pei-Quan, Xie Xi-Ke. Learned indexes: Current situations and research prospects. Journal of Software, 2021, 32(4): 1129-1150(in Chinese)
(张洲，金培权，谢希科. 学习索引：现状与研究展望. 软件学报，2021，32(4)：1129-1150)

[30] Liu Huan, Liu Peng-Ju, Wang Tian-Yi, et al. Survey of intelligent partition and layout technology in database system. Journal of Software, 2022, 33(10): 3819-3843(in Chinese)
(刘欢，刘鹏举，王天一等. 智能数据分区与布局研究. 软件学报，2022，33(10)：3819-3843)

[31] Bayer R. The universal B-tree for multidimensional indexing: General concepts//Proceedings of the International Conference on Worldwide Computing and Its Applications. Tsukuba, Japan, 1997: 198-209

[32] Ramsak F, Markl V, Fenk R, et al. Integrating the UB-Tree into a database system kernel//Proceedings of the Conference on Very Large Databases. Cairo, Egypt, 2000: 263-272

[33] Comer D. The ubiquitous B-tree. ACM Computing Surveys, 1979, 11(2): 121-137

[34] Sagan H. Space-Filling Curves. New York: Springer, 1994. https://doi.org/10.1007/978-1-4612-0871-6

[35] Mokbel M F, Aref W G, Kamel I. Analysis of multi-dimensional space-filling curves. GeoInformatica, 2003, 7(3): 179-209

[36] Wang Haixin, Fu Xiaoyi, Xu Jianliang, Lu Hua. Learned index for spatial queries//Proceedings of the 2019 20th IEEE International Conference on Mobile Data Management (MDM). Hong Kong, China, 2019: 569-574

[37] Lee K C K, Zheng B, Li H, Lee W-C. Approaching the skyline in Z-order. Proceedings of the VLDB Endowment, 2007, 7(1): 279-290

[38] Wang Ning, Xu Jianqiu. Spatial queries based on learned index//Proceedings of the 1st Internatial Conference of SpatialDI. 2020: 245-257(online)

[39] Qi J, Tao Y, Chang Y, et al. Theoretically optimal and empirically efficient R-trees with strong parallelizability. Proceedings of the VLDB Endowment, 2018, 11(5): 621-634

[40] Gao Jian, Cao Xin, Yao Xin, et al. LMSFC: A novel multi-dimensional index based on learned monotonic space filling curves. Proceedings of the VLDB Endowment, 2023, 16(10): 2605-2617

[41] Li Jiangneng, Wang Zheng, Cong Gao, et al. Towards desigining and learning piecewise space-filling curves. Proceedings of the VLDB Endowment, 2023, 16(9): 2158-2171

[42] Davitkova A, Milchevski E, Michel S. The ML-index: A multidimensional, learned index for point, range, and nearest-neighbor queries//Proceedings of the EDBT. Copenhagen, Denmark, 2020: 407-410

[43] Jagadish H V, Ooi B C, Tan K-L, et al. iDistance: An adaptive B^+-tree based indexing method for nearest neighbor search. ACM Transactions on Database Systems, 2005, 30(2): 364-397

[44] Hadian A, Kumar A, Heinis T. Hands-off model integration in spatial index structures. arXiv preprint arXiv:2006.16411, 2020

[45] Kraska T, Alizadeh M, Beutel A, et al. SageDB: A learned database system//Proceedings of the 2019 Conference on Innovative Data Systems Research(CIDR). Asilomar, USA, 2019: 117-127

[46] Eldawy A, Alarabi L, Mokbel M F. Spatial partitioning techniques in SpatialHadoop. Proceedings of the VLDB Endowment, 2015, 8(12): 1602-1605

[47] Moti M H, Simatis P, Papadias D. Waffle: A workload-aware and query-sensitive framework for disk-based spatial indexing. Proceedings of the VLDB Endowment, 2022, 16(4): 670-683

[48] Finkel R A, Bentley J L. Quad trees: A data structure for retrieval on composite keys. Acta Informatica, 1974, 4: 1-9

[49] Nievergelt J, Hinterberger H, Sevcik K C. The grid file: An adaptable, symmetric multikey file structure. ACM Transactions on Database Systems, 1984, 9(1): 38-71

[50] Beckmann N, Kriegel H-P, Schneider R, Seeger B. The R*-tree: An efficient and robust access method for points and rectangles//Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data. Atlantic City, Tennessee, 1990: 322-331

[51] Zhang S, Ray S, Lu R, et al. Efficient learned spatial index with interpolation function based learned model. IEEE Transactions on Big Data, 2022, 9(2): 733-745

[52] Peng Yongxin, Zhou Wei, Zhang Lin, et al. A study of learned KD tree based on learned index//Proceedings of the 2020 International Conference on Networking and Network Applications (NaNA). Haikou, China, 2020: 355-360

[53] Nathan V, Ding J L, Alizadeh M, Kraska T. Learning multi-dimensional indexes//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland, USA, 2020: 985-1000

[54] Ding J, Nathan V, Alizadeh M, Kraska T. Tsunami: A learned multi-dimensional index for correlated data and skewed workloads. Proceedings of the VLDB Endowment, 2021, 14(2): 74-86

[55] Gu T, Feng K, Cong G, et al. A reinforcement learning based R-tree for spatial data indexing in dynamic environments. arXiv preprint arXiv:2103.04541, 2021

[56] Gu T, Feng K, Cong G, et al. The RLR-Tree: A reinforcement learning based R-tree for spatial data//Proceedings of the ACM on Management of Data (SIGMOD). Seattle, USA, 2023, 1(1): 1-26

[57] Greene D. An implementation and performance analysis of spatial data access methods//Proceedings of the 1989 International Conference on Data Engineering (ICDE). Los Angeles, USA, 1989: 606-607

[58] Ang C H, Tan T C. New linear node splitting algorithm for R-trees//Proceedings of the International Symposium on Spatial Databases. Berlin, Heidelberg, 1997: 337-349

[59] Yang Jingyi, Cong Gao. PLATON: Top-down R-tree packing with learned partition policy//Proceedings of the 2023 ACM SIGMOD International Conference on Management of Data. Seattle, USA, 2023: 1-26

[60] Kang R, Wu W, Wang C, et al. The case for ML-enhanced high-dimensional indexes//Proceedings of the 3rd International Workshop on Applied AI for Database Systems and Applications. Copenhagen, Denmark, 2021: 1-10

[61] Ramadhan H, Kwon J. Learning minimum bounding rectangles for efficient trajectory similarity search//Proceedings of the 2020 IEEE International Conference on Big Data (Big Data). Atlanta, Georgia, 2020: 5810-5812(online)

[62] Shi H. Learned Hashmap for Spatial Queries [M. S. dissertation]. The University of Melbourne, Melbourne, Australia, 2020

[63] Sanders P, Schulz C. Think locally, act globally: Highly balanced graph partitioning//Proceedings of the 12th International Symposium on Experimental Algorithms. Berlin, Germany, 2013, 7933: 164-175

[64] Guo Y, Shao Z. Cymo: A storage model with query-aware indexing for spatio-temporal big data//Proceedings of the 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS). Bologna, Italy, 2022: 122-132

[65] Ilyas I F, Markl V, Haas P, et al. CORDS: Automatic discovery of correlations and soft functional dependencies//Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. Paris, France, 2004: 647-658

[66] Kimura H, Huo G, Rasin A, et al. Correlation maps: A compressed access method for exploiting soft functional dependencies. Proceedings of the VLDB Endowment, 2009, 2(1): 1222-1233

[67] Wu Yingjun, Yu Jia, Tian Yuanyuan, et al. Designing succinct secondary indexing mechanism by exploiting column correlations//Proceedings of the 2019 ACM SIGMOD International Conference on Management of Data. Amsterdam, The Netherlands. 2019: 1223-1240

[68] Ghaffari B, Hadian A, Heinis T. Leveraging soft functional dependencies for indexing multi-dimensional data. arXiv preprint arXiv:2006.16393, 2020

[69] Hilprecht B, Binnig C, Röhm U. Learning a partitioning advisor for cloud databases//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland, USA, 2020: 143-157

[70] Sudhir S, Tao W, Laptev N, et al. Pando: Enhanced data skipping with logical data partitioning. Proceedings of the VLDB Endowment, 2023, 16(9): 2316-2329

[71] Li Z, Yiu M L, Chan T N. PAW: Data partitioning meets workload variance//Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE). Kuala Lumpur, Malaysia, 2022: 123-135

[72] Ding J, Minhas U F, Chandramouli B, et al. Instance-optimized data layouts for cloud analytics workloads//Proceedings of the 2021 International Conference on Management of Data. Xi'an, China, 2021: 418-431

[73] Hilprecht B, Binnig C, Röhm U. Towards learning a partitioning advisor with deep reinforcement learning//Proceedings of the 2nd International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. Amsterdam, The Netherlands, 2019: 1-4

[74] Zhou X, Li G, Feng J, et al. Grep: A graph learning based database partitioning system//Proceedings of the 2023 ACM SIGMOD on Management of Data. Seattle, USA, 2023: 1-24

[75] Huawei Technologies Co., Ltd. Basic Knowledge of Databases. Database principles and technologies—Based on Huawei GaussDB. Beijing: Posts & Telecom Press, 2022: 41-86

[76] Vu T, Belussi A, Migliorini S, et al. Using deep learning for big spatial data partitioning. ACM Transactions on Spatial Algorithms and Systems (TSAS), 2020, 7(1): 1-37

[77] Hori K, Sasaki Y, Amagata D, et al. Learned spatial data partitioning//Proceedings of the 6th International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. 2023: 1-8

[78] Durand G C, Pinnecke M, Piriyev R, et al. GridFormation: Towards self-driven online data partitioning using reinforcement learning//Proceedings of the 1st International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. Houston, USA, 2018: 1-7

[79] Li Kaiyu, Li Guoliang. Approximate query processing: What is new and where to go? A survey on approximate query processing. Data Science and Engineering, 2018, 3(4): 379-397

[80] Alabi D, Wu E. PFunk-H: Approximate query processing using perceptual models//Proceedings of the HILDA@SIGMOD. San Francisco, USA, 2016: 1-6

[81] Vorona D, Kipf A, Neumann T, Kemper A. DeepSPACE: Approximate geospatial query processing with deep learning//Proceedings of the 2019 International Conference on Advances in Geographic Information Systems (SIGSPATIAL/GIS). Chicago, USA, 2019: 500-503

[82] Germain M, Gregor K, Murray I, Larochelle H. MADE: Masked autoencoder for distribution estimation//Proceedings of the 2015 ICML. Lille, France, 2015: 881-889

[83] Hilprecht B, Schmidt A, Kulessa M, et al. DeepDB: Learn from data, not from queries!. Proceedings of the VLDB Endowment, 2020, 13(7): 992-1005

[84] Song G, Dai Q. A novel double deep ELMs ensemble system for time series forecasting. Knowledge Based Systems, 2017, 134: 31-49

[85] Liu Q, Zhou S, Zhu C, et al. MI-ELM: Highly efficient multi-instance learning based on hierarchical extreme learning machine. Neurocomputing, 2016, 173: 1044-1053

[86] Zeng Y, Xu X, Fang Y, Zhao K. Traffic sign recognition using deep convolutional networks and extreme learning machine//Intelligence Science and Big Data Engineering. Image and Video Data Engineering. Suzhou, China, 2015: 272-280

[87] Zhao Xiangguo, Bi Xin, Zeng Xiangyu, et al. EDense: A convolutional neural network with ELM-based dense connections. Neural Computing and Applications, 2023, 35: 3651-3663

[88] Amagata D, Arai Y, Fujita S, et al. Learned k-NN distance estimation//Proceedings of the 2022 International Conference on Advances in Geographic Information Systems (SIGSPATIAL/GIS). Seattle, USA, 2022: 1-4

[89] Meng Z, Wu P, Cong G, et al. Unsupervised selectivity estimation by integrating Gaussian mixture models and an autoregressive model//Proceedings of the 2022 International Conference on Extending Database (EDBT). 2022: 2: 247-2: 259(online)

[90] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets//Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal, Canada, 2014: 1-9

[91] Fallahian M, Dorodchi M, Kreth K. GAN-based tabular data generator for constructing synopsis in approximate query processing: Challenges and solutions. arXiv preprint arXiv: 2212.09015, 2022

[92] Choi E, Biswal S, Malin B, et al. Generating multi-label discrete patient records using generative adversarial networks//Proceedings of the 2017 Machine Learning for Healthcare Conference. Boston, USA, 2017: 286-305

[93] Park N, Mohammadi M, Gorde K, et al. Data synthesis based on generative adversarial networks. arXiv preprint arXiv: 1806.03384, 2018

[94] Lai Z, Han C, Liu C, et al. Top-*K* deep video analytics: A probabilistic approach//Proceedings of the 2021 International Conference on Management of Data. Xi'an, China, 2021: 1037-1050

[95] Kang D, Emmons J, Abuzaid F, et al. NoScope: Optimizing deep CNN-based queries over video streams at scale. Proceedings of the VLDB Endowment, 2017, 10(11): 1586-1597

[96] Lu Y, Chowdhery A, Kandula S, Chaudhuri S. Accelerating machine learning inference with probabilistic predicates//Proceedings of the 2018 International Conference on Management of Data. Houston, USA, 2018: 1493-1508

[97] Zhou L, Candan K S, Zou J. DeepMapping: Learned data mapping for lossless compression and efficient lookup//Proceedings of the 2024 IEEE 40th International Conference on Data Engineering (ICDE). Utrecht, Netherlands, 2024

[98] Park Y, Tajik A S, Cafarella M, et al. Database learning: Toward a database that becomes smarter every time//Proceedings of the 2017 ACM International Conference on Management of Data. Chicago, USA, 2017: 587-602

[99] Regev N, Rokach L, Shabtai A. Approximating aggregated SQL queries with LSTM networks//Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN). 2021: 1-8

[100] Fotis S, Anagnostopoulos C, Triantafillou P. ML-AQP: Query-driven approximate query processing based on machine learning. arXiv preprint arXiv:2003.06613, 2020

[101] Thiagarajan A, Madden S. Querying continuous functions in a database system//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. Vancouver, Canada, 2008: 791-804

[102] Schleich M, Olteanu D, Ciucanu R. Learning linear regression models over factorized joins//Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data. San Francisco, USA, 2016: 3-18

[103] Ma Q, Triantafillou P. DBEst: Revisiting approximate query processing engines with machine learning models//Proceedings of the 2019 International Conference on Management of Data. Amsterdam, The Netherlands, 2019: 1553-1570

[104] Ma Q, Shanghooshabad A M, Kurmanji M, et al. Learned approximate query processing: Make it light, accurate and fast//Proceedings of the 2021 Conference on Innovative Data Systems Research (CIDR). Chaminade, USA, 2021: 15-26

[105] Ma Q, Triantafillou P. Query-centric regression. Information Systems, 2022, 104: 101736

[106] Hasan S, Thirumuruganathan S, Augustine J, et al. Deep learning models for selectivity estimation of multi-attribute queries//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland, USA, 2020: 1035-1050

[107] Park Y, Zhong S, Mozafari B. QuickSel: Quick selectivity learning with mixture models//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland, USA, 2020: 1017-1033

[108] Bindschaedler L, Kipf A, Kraska T, et al. Towards a benchmark for learned systems//Proceedings of the 2021 IEEE 37th International Conference on Data Engineering Workshops. Chania, Greece, 2021: 127-133

[109] Ma C, Yu X, Li Y, et al. FILM: A fully learned index for larger-than-memory databases. Proceedings of the VLDB Endowment, 2022, 16(3): 561-573

[110] Lan H, Bao Z, Culpepper J S, et al. A fully on-disk updatable learned index//Proceedings of the 2024 IEEE 40th International Conference on Data Engineering(ICDE). Utrecht, The Netherlands, 2024

[111] Tsitsigkos D, Lampropoulos K, Bouros P, et al. A two-layer partitioning for non-point spatial data//Proceedings of the 2021 International Conference on Data Engineering (ICDE). Chania, Greece, 2021: 1787-1798

[112] Kumar A, McCann R, Naughton J, Patel J M. Model selection management systems: The next frontier of advanced analytics. ACM SIGMOD Record, 2016, 44(4): 17-22

[113] Guo Rentong, Luan Xiaofan, Xiang Long, et al. Manu: A cloud native vector database management system. Proceedings of the VLDB Endowment, 2022, 15(12): 3548-3561

[114] Thorne J, Yazdani M, Saeidi M, et al. Neural databases. arXiv preprint arXiv: 2010.06973, 2020

[115] Thorne J, Yazdani M, Saeidi M, et al. From natural language processing to neural databases. Proceedings of the VLDB Endowment, 2021, 14(6): 1033-1039

[116] Marcus R, Kipf A, van Renen A, et al. Benchmarking learned indexes. Proceedings of the VLDB Endowment, 2020, 14(1): 1-13

[117] Wongkham C, Lu B, Liu C, et al. Are updatable learned indexes ready?. Proceedings of the VLDB Endowment, 2022, 15(11): 3004-3017

[118] Sun Z, Zhou X, Li G. Learned index: A comprehensive experimental evaluation. Proceedings of the VLDB Endowment, 2023, 16(8): 1992-2004

[119] Ge Jiake, Shi Boyu, Chai Yanfeng, et al. Cutting learned index into pieces: An in-depth inquiry into updatable learned indexes//Proceedings of the 2023 IEEE 39th International Conference on Data Engineering(ICDE). Anaheim, USA, 2023: 315-327

[120] Ferragina P, Lillo F, Vinciguerra G. Why are learned indexes so effective?//Proceedings of the 2023 International Conference on Machine Learning (ICML). 2020: 3123-3132 (online)

[121] Zeighami S, Shahabi C. On distribution dependent sub-logarithmic query time of learned indexing//Proceedings of the 2023 International Conference on Machine Learning (ICML). Hawaii, USA, 2023: 40669-40680

[122] Al-Mamun A, Wu H, Aref W G. A tutorial on learned multi-dimensional indexes//Proceedings of the 28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL/GIS). Seattle, USA, 2020: 1-4

[123] Al-Mamun A, Wu H, He Q, et al. A survey of learned indexes for the multi-dimensional space. arXiv preprint arXiv: 2403.06456, 2024

[124] Pandey V, Renen A, Kipf A, et al. The case for learned spatial indexes. arXiv preprint arXiv:2008.10349, 2020

[125] Kurmanji M, Triantafillou P. Detect, distill and update: Learned DB systems facing out-of-distribution data// Proceedings of the 2023 ACM SIGMOD on Management of Data. Seattle, USA, 2023: 1-27

MA Chao-Hong, Ph. D. candidate. Her main research interests include machine learning for database systems, learned indexes, and data management.

HAO Xin-Li, Ph. D. candidate. Her main research interests include intelligent scientific discovery, time series analysis, and interpretable machine learning.

MENG Xiao-Feng, Ph. D., professor, Ph. D. supervisor. His main research interests include database management systems, data management, privacy protection, and interdisciplinary research like social computing.

ZHANG Xu-Kang, Ph. D. candidate. His primary research interests include database systems, real-time analysis for big data, database tuning, and query optimization.

Background

Multi-dimensional data processing has always been one of the most crucial research areas in database management. Recently, machine learning for multi-dimensional data processing has gained hot interest, and received plenty of successful work.

On one hand, modern applications generate multi-dimensional data at an unprecedented speed, thus, it becomes more crucial to improve the performance of efficient operations on multi-dimensional data. However, the traditional multi-dimensional data structures for fast query processing face challenges. For example, big data volume entails large R-trees. And sometimes, the size of the R-tree is larger than the original data leading to serious storage overhead.

On the other hand, the research on machine learning algorithms provides more opportunities for the query and processing performance of multidimensional data. For instance, some works apply machine learning models to learn from the data and query workload, then optimize the index structures or the data layouts, which achieve improved performance.

However, there is no comprehensive literature review to summarize the existing research efforts, and thus provide guidelines for the researchers and developers in this area.

This paper aims to fill this gap. First of all, we discuss the research challenges in multi-dimensional data processing and summarize existing works that apply machine learning algorithms to address these challenges. In addition, this paper categorizes the research works into two aspects: index structures and non-index structures.

Secondly, this paper reviews how to evaluate the performance in multi-dimensional data processing, including the datasets, the workloads run on the datasets, and the various metrics to evaluate.

Finally, this paper provides the future research challenges and opportunities in this field, to facilitate the development of machine learning for multi-dimensional data processing.

Before this paper, a series of related works are published in VLDB, ICDE, TKDE, EDBT, etc. More specifically, the authors have done a survey to summarize the research works in machine learning for database management. Additionally, they have proposed a learned index to accelerate the query processing and reduce the storage overhead as a research paper. In the process of participating in the National Natural Science Foundation of China project "Intelligent Analysis of Astronomical Big Data for Large Field-of-View Short-Timescale Sky Survey", the authors have further addressed the challenges arising from the management for the spatio-temporal data which is one of the important types in multi-dimensional data.

This work was partially supported by grants from the National Natural Science Foundation of China (No. 62172423).