软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007664] [CSTR: 32375.14.jos.007664]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

粒球计算驱动的智能数据库: 现状和展望*

乔少杰[1], 杨 蕾[1], 夏书银[2,3], 韩 楠[4], 苟浩淞[5], 何 函[1], 袁 冠[6], 唐明靖[7]

[1](成都信息工程大学 软件工程学院, 四川 成都 610225)

[2](网络空间大数据智能安全教育部重点实验室 (重庆邮电大学), 重庆 400065)

[3](重庆邮电大学 人工智能学院, 重庆 400065)

[4](成都信息工程大学 管理学院, 四川 成都 610225)

[5](中国移动通信集团四川有限公司, 四川 成都 610091)

[6](中国矿业大学 计算机科学与技术学院, 江苏 徐州 221116)

[7](云南师范大学 信息学院, 云南 昆明 650500)

通信作者: 韩楠, E-mail: hannan@cuit.edu.cn

摘 要: 传统关系型数据库的关键优化技术在面对海量数据处理、复杂查询及动态负载场景时, 普遍存在估计精度不足、优化决策效率低下以及环境适应性差等瓶颈. 多粒度粒球计算为提升数据库系统性能开辟了新的解决路径, 展现出巨大的研究潜力和应用前景. 首先, 概述了人工智能在驱动数据库智能优化方面的核心方向, 探讨了现有学习型优化方法在模型泛化能力、可解释性以及处理复杂查询与动态数据分布方面所面临的主要挑战. 在此基础上, 系统地综述了数据库优化的现状及关键技术, 结合多粒度粒球计算, 数据库优化技术的核心聚焦于查询优化与配置优化两个方面. 针对查询优化, 关键技术包括基数估计以及连接顺序选择. 在基数估计方面, 传统方法难以有效支持涉及多表复杂连接及嵌套查询的准确评估, 且常带来巨大的存储开销; 基于学习的方法则能更好地处理高维数据关系, 介绍利用多粒度粒球计算技术提取数据分层分布特征并与树结构神经网络结合的新方法, 能显著提高复杂查询基数估计的鲁棒性与精度. 在连接顺序选择方面, 传统方法在多表连接维度下搜索效率低下; 基于历史经验学习的静态方法对新查询模式适应性有限; 动态学习方法虽能支持运行时调整但开销较大; 相比之下, 将连接计划表达为具有几何关系的多粒度粒球结合, 利用其层次结构优化搜索空间并结合深度强化学习进行决策的方法, 为高效寻找全局近似最优连接顺序提供了新思路. 针对数据库配置优化, 参数调优是提升性能的关键. 基于搜索的传统优化技术难以在合理时间内获得全局最优解; 传统机器学习方法虽然能实现自动化调优, 但高度依赖训练数据的质量与覆盖度; 强化学习方法通过与系统环境交互持续改进策略, 仅需少量样本即可实现强大的自适应性调参, 融合多粒度粒球计算方法能够高效表达参数空间特性, 显著提升调优效率与效果. 虽然应用粒球计算技术优化数据库前景广阔, 但实际应用仍面临与现有数据库模型的有效融合、降低计算开销、动态负载变化下的模型稳定性保障等主要挑战. 未来研究需持续深化理论与技术, 推动数据库系统朝着更智能、高效、鲁棒的方向发展.

关键词: 粒球计算; 智能数据库; 基数估计; 连接顺序选择; 查询优化; 参数调优

中图法分类号: TP311

中文引用格式: 乔少杰, 杨蕾, 夏书银, 韩楠, 苟浩淞, 何函, 袁冠, 唐明靖. 粒球计算驱动的智能数据库: 现状和展望. 软件学报. http://www.jos.org.cn/1000-9825/7664.htm

* 基金项目: 国家自然科学基金 (62272066, 62572078); 四川省科技计划 (2025ZNSFSC0044, 2025YFHZ0194); 成都市重点研发计划基础研究项目 (2025-YF12-00019-RC, 2025-YF12-00012-RC, 2025-YF12-00015-RC); 成都重点研发支撑计划产业链协同创新项目 (2025-XT00-00005-GX); 成都市技术创新研发项目重点项目 (2025-YF08-00016-GX); 成都市区域科技创新合作项目 (2025-YF11-00050-HZ); 网络空间大数据智能安全教育部重点实验室开放基金课题 (CBDIS202404)
收稿时间: 2025-08-29; 修改时间: 2025-11-03; 采用时间: 2026-03-05; jos 在线出版时间: 2026-05-27

英文引用格式: Qiao SJ, Yang L, Xia SY, Han N, Gou HS, He H, Yuan G, Tang MJ. Intelligent Database Driven by Granular-ball Computing: Current Status and Prospects. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7664.htm

Intelligent Database Driven by Granular-ball Computing: Current Status and Prospects

QIAO Shao-Jie[1], YANG Lei[1], XIA Shu-Yin[2,3], HAN Nan[4], GOU Hao-Song[5], HE Han[1], YUAN Guan[6], TANG Ming-Jing[7]

[1](School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

[2](Key Laboratory of Cyberspace Big Data Intelligent Security (Chongqing University of Posts and Telecommunications), Ministry of Education, Chongqing 400065, China)

[3](School of Artificial Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

[4](School of Management, Chengdu University of Information Technology, Chengdu 610225, China)

[5](China Mobile Group Sichuan Co. Ltd., Chengdu 610091, China)

[6](School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

[7](School of Information Science and Technology, Yunnan Normal University, Kunming 650500, China)

Abstract: The key optimization technologies of traditional relational databases generally face bottlenecks including insufficient estimation accuracy, low efficiency of optimization decisions, and poor environmental adaptability when dealing with massive data processing, complex queries, and dynamic workload scenarios. Multi-granularity granular-ball computing provides a new solution for improving the performance of database systems, showing great research potential and application prospects. The core directions of artificial intelligence in driving intelligent optimization for databases are outlined, and the main challenges faced by existing learning-based optimization methods in terms of model generalization ability, interpretability, and handling complex queries and dynamic data distributions are explored. On this basis, the current status and key technologies of database optimization are systematically reviewed. Combined with multi-granularity granular-ball computing, database optimization techniques primarily focus on two aspects: query optimization and configuration optimization. For query optimization, the key techniques include cardinality estimation and join order selection. In terms of cardinality estimation, traditional methods are difficult to effectively support accurate estimation of complex joins and nested queries involving multiple tables, and often result in huge storage overhead. Learning-based methods can better handle high-dimensional data relationships, and a new method that uses multi-granularity granular-ball computing techniques to extract hierarchical distribution features of data and combines them with tree-structured neural networks is introduced, which can significantly improve the robustness and accuracy of complex query cardinality estimation. In terms of join order selection, traditional methods have low search efficiency in multi-table join scenarios. Static methods based on historical experience learning have limited adaptability to new query patterns. Although dynamic learning methods can support runtime adjustments, they incur a high cost. In contrast, representing the join plan as a combination of multi-granularity granular-balls with geometric relationships, optimizing the search space through a hierarchical structure, and integrating deep reinforcement learning for decision-making provides an efficient approach to finding a globally approximate optimal join order. For database configuration optimization, parameter tuning plays a key role in improving performance. Traditional optimization techniques based on search are difficult to obtain the global optimal solution within a reasonable time. Although traditional machine learning methods can achieve automated tuning, they highly rely on the quality and coverage of training data. Reinforcement learning methods continuously improve strategies by interacting with the system environment, requiring only a small number of samples to achieve strong adaptive parameter tuning. By integrating multi-granularity granular-ball computing methods, the characteristics of the parameter space can be efficiently represented, thus significantly improving tuning efficiency and effectiveness. Although the prospects of optimizing intelligent databases using granular-ball computing techniques are broad, practical applications still face major challenges such as effective integration with existing database models, reduction of computational overhead, and ensuring model stability under dynamic workload changes. Future research requires continuous development of theories and technologies to promote database systems toward more intelligent, efficient, and robust directions.

Key words: granular-ball computing (GBC); intelligent database; cardinality estimation; join order selection; query optimization; parameter tuning

1 引 言

1.1 研究背景

数据库技术作为信息技术领域的核心分支, 其发展对海量数据的处理与管理至关重要. 随着计算机技术进步与数字化信息激增, 数据库系统成为数据存储、检索和管理的基础设施, 广泛应用于商业、科研等领域. 互联网、物联网及智能设备的普及催生了海量结构化与非结构化数据, 数据量的爆炸式增长给存储、处理和分析带来了严峻挑战. 数据库技术历经了 3 次变革: 第 1 代单机数据库, 如 PostgreSQL、MySQL, 主要解决数据存储与管理问题; 第 2 代集群数据库, 如 Oracle RAC、IBM DB2, 聚焦企业业务的高可用性; 第 3 代分布式与云原生数据库, 如: 亚马逊 Aurora、华为 GaussDB 则应对大数据时代的弹性计算需求[1].

传统的数据库管理系统在面对如此庞大和复杂的数据处理任务时, 越来越难以解决相关问题, 无法满足日益增长的数据分析和处理需求. 其原因在于数据库的执行效率依赖于数据库管理员 (database administrator, DBA) 的人工干预, 而各个数据库实例的应用场景往往存在很大差别, 且同一数据库的工作负载在快速变化, 使得管理员难以动态调整如此多的实例, 也无法保证其可靠性[2,3]. 人工智能技术应用于大数据处理, 带来了诸如大规模异构数据等多种异构计算资源等新的挑战, 传统数据库系统在上述场景下仍存在诸多局限性. 目前, 数据库系统已经在商业中广泛使用, 产品级数据库系统持续稳定地运行, 表示经典的数据库算法已经趋于完善[4].

在人工智能技术被广泛应用到多个科研领域, 通过历史数据和行为中学习经验, 可以找到更好的优化方案, 使模型具有解决复杂问题的能力[5]. Ré等人[1]明确提出关于人工智能与数据库系统结合的思考, 引发了数据库领域对人工智能技术应用的激烈讨论, 人工智能技术以其优异的特性迅速在大数据驱动的应用领域普及并逐渐成为主流. 此外, 机器学习技术已经被广泛运用到科研和生产领域[6−8]. 传统的机器学习模型, 例如线性回归[9]、随机森林[10]、支持向量机[11]以及集成学习[12]等, 能够提高模型解决复杂问题的能力, 但仍存在诸多挑战. 数据库是一个复杂的系统, 数据库优化涉及的数据具有多样性和复杂性, 比如在进行基数估计时, 模型要同时学习表中的样本分布以及查询语句信息或执行计划信息. 此时, 如果使用传统的机器学习模型, 将无法达到较好效果. 例如, 使用简单的线性回归模型时, 无法解决诸如连接顺序选择这样的复杂问题, 因为它无法处理高维参数和连续空间的问题. 随着技术的不断发展, 数据库优化需求不断加强. 数据库管理系统存在查询优化、参数配置等需求, 需要大量数据库管理员的调优经验. 因此, 通过人工智能技术能够快速积累数据库调优经验, 从而应对快速变化的工作负载, 动态地为数据库提供最佳的查询优化配置. 在此背景下, 乔少杰等人[13]系统综述了人工智能在数据库配置优化与查询优化等技术中的应用, 涵盖了索引推荐、视图选择、连接顺序选择等多个关键技术, 为智能数据库优化技术的创新提供了重要理论框架与技术路径. 如图 1 所示, 学习型数据库中基于学习的技术包括: 基数估计[14−16]、视图选择[17−19]、参数调优[20−22]、学习型索引结构[23−25]等.

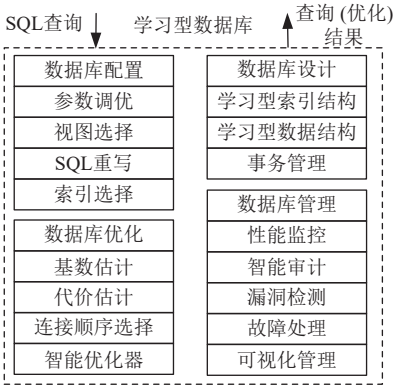

图 1 基于机器学习的数据库技术

学习型数据库通过集成机器学习和人工智能技术, 能够自动优化数据管理和查询处理, 但也存在一些不足. 如

现有学习型基数估计器存在 3 个局限性: 首先, 大多数算法只考虑 SQL 查询本身, 却忽略了 SQL 查询的物理实现, 且无法统一处理不同形式的 SQL 查询以及复杂 SQL 查询; 其次, 大多数算法只专注于基数估计, 忽略了操作算子的代价估计, 导致模型功能单一; 最后, 由于字符串的稀疏性影响, 现有方法对于包含字符串类型的谓词难以处理, 例如模糊查询. 现有学习型连接优化器存在以下局限性: 1) 编码方法未考虑物理运算符编码, 无法体现执行计划结构特征, 导致状态表示不准确; 2) 搜索策略采用简单启发式算法, 难以全局指导连接顺序选择, 易陷入局部最优; 3) 在处理复杂多表连接查询时, 尾部性能较差, 生成的执行计划甚至不如传统优化器. 此外, 在学习型数据库领域, 参数调优作为保障数据库高效运行的关键环节, 同样面临诸多挑战. 现有参数调优机制存在以下不足: 1) 缺乏有效的自动化调优手段, 多数依赖人工经验进行参数设置, 不仅效率低下, 而且容易因个人认知差异导致参数配置不合理, 无法充分发挥数据库性能; 2) 数据库运行环境复杂多变, 现有调优方法难以适应不同的硬件配置、数据规模及业务负载, 难以找到一套通用且高效的参数调优策略, 导致调优效果大打折扣; 3) 现有调优机制缺乏动态调整能力, 无法实时感知数据库运行状态变化, 不能根据实际情况及时优化参数, 在数据库负载波动时, 无法保障性能稳定, 容易出现响应延迟、吞吐量下降等问题.

在此背景下, 多粒度粒球计算[26]作为一种新兴计算范式, 为突破数据库技术瓶颈提供了全新思路. 多粒度粒球计算模拟人类全局优先的认知机制, 通过构建多层次、多尺度的粒球结构, 对数据样本空间进行分层抽象与特征提取. 不同粒度的粒球可分别捕捉数据在宏观、中观和微观层面的特征, 这种处理方式不仅能够有效降低数据复杂度, 还可以增强模型对复杂数据分布的适应性. 在理论研究领域, 多粒度粒球计算已在粗糙集理论、模糊集理论、机器学习等方向取得重要进展. 例如, 基于粒球计算的粗糙集模型能够更灵活地处理数据的不确定性与不完整性; 在机器学习中, 粒球计算可用于数据降维、特征选择, 显著提升算法效率与性能. 在实际应用方面, 多粒度粒球计算在自然语言处理、图像识别、数据挖掘等领域展现出巨大潜力. 例如, 在文本分类任务中, 通过多粒度分析可以更好地理解语义信息; 在图像识别中, 利用粒球结构能快速提取图像的全局轮廓与局部细节特征.

将多粒度粒球计算引入数据库系统, 有望从根本上改变传统数据库的数据处理模式. 通过构建基于粒球计算的数据库架构, 可实现对海量异构数据的高效存储与智能管理; 借助粒球计算的多粒度分析能力, 能够显著提升数据库的查询性能和数据分析效率; 人工智能技术可赋予数据库自主优化与智能决策能力, 满足复杂业务场景的多样化需求. 因此, 开展多粒度粒球计算驱动的智能数据库研究, 对推动数据库技术的创新发展具有重要理论价值与现实意义.

1.2 智能数据库热点研究

1.2.1 基数估计

基数估计是数据库查询优化的核心环节, 其准确性直接决定查询执行计划的优劣. 传统基数估计方法[27−31]主要依赖于统计信息 (如直方图、基数表) 进行估算, 但在处理复杂数据分布、高维数据及动态查询条件时, 常出现较大误差. 例如, 当查询涉及多个连接条件、复杂谓词或非均匀分布数据时, 传统方法难以精准预测结果集大小, 可能导致查询优化器选择次优执行计划, 造成查询性能严重下降[32]. 多粒度粒球法是一种新的基数估计方法. 通过对数据进行粒化处理, 可以更加精细地刻画出数据的分布特性和内部规律.

1.2.2 查询优化

数据库查询优化旨在生成高效的查询执行计划, 以最小资源消耗满足用户需求. 传统查询优化器[33−35]多采用基于规则的启发式算法或基于代价的优化方法, 但在面对复杂查询 (如多表连接、嵌套子查询、递归查询) 及动态变化的查询负载时, 难以找到全局最优解[36]. 随着数据规模与查询复杂度的持续攀升, 传统优化方法的局限性愈发显著. 多粒度粒化算法能够实现多层次、多粒度的数据分析, 为用户的查询优化提供更多的信息量. 例如, 在多个表格的连通性查询中, 通过粒球算法可以迅速发现表格之间的关键连接情况和数据分布特性, 并对连接次序进行优化和算法选取.

1.2.3 参数调优

数据库系统性能高度依赖参数配置[37], 不同应用场景与工作负载需匹配不同参数设置. 传统参数调优[38−54]主

要依赖 DBA 的经验手动调整, 这种方式效率低、耗时长, 且难以找到全局最优参数组合. 随着数据库规模与业务需求的多样化, 手动调参已无法满足实际需求. 多粒度粒化算法能够充分利用多粒度粒子的历史数据和当前任务的特点, 为优化算法的优化提供理论基础. 比如, 对数据存取规律和资源利用率进行多粒度分析, 找出影响性能的关键参数, 并对其进行动态调节.

1.3 面临挑战

将粒球计算融入数据库以解决其核心难题, 关键在于用粒球的多粒度抽象、几何化建模特性适配查询优化、数据处理等场景需求, 结合当前技术适配与落地现状, 核心挑战集中在以下方面.

1.3.1 数据结构与特征表示的映射难题

数据库任务对数据表征、语义解析有明确要求, 与粒球的几何化模型存在映射障碍. 粒球的中心-半径模型依赖纯度与汇聚度拟合数据分布, 但数据库多表关联的复杂拓扑关系难以通过单一粒球参数精准映射, 易放大基数估计等任务的误差; 粒球生成的抽象语义特征与数据库查询优化器依赖的数值型特征差异显著, 现有转换方法难以兼顾多粒度信息保留与代价模型对接; 针对文本、向量等非结构化数据的粒球生成算法尚未成熟, 抽象过程易丢失原生语义, 无法支撑跨类型数据的统一分析.

1.3.2 资源消耗与效率平衡的兼容难题

在数据库环境与动态变化的工作负载下, 基于多粒度粒球计算的模型面临泛化能力与稳定性的双重考验. 数据库系统的硬件配置、软件版本、数据分布、查询模式等因素差异显著. 上述变化可能导致粒球计算模型的性能大幅下降甚至失效. 例如, 不同硬件环境下模型的计算效率差异明显, 数据分布变化会影响模型的预测准确性. 此外, 业务需求的动态演变使数据库工作负载具有不确定性, 粒球计算模型需具备自适应能力. 现有研究虽通过改进模型结构、优化算法等方式提升泛化能力, 但在复杂多变场景下, 模型的稳定性与鲁棒性仍难以保障.

1.3.3 模型泛化能力与稳定性的适配难题

数据库运行环境的动态变化对粒球模型提出严峻考验. 数据层面, 新类别数据涌入会导致原有粒球覆盖度不足, 重新生成易丢失历史分布特征, 影响分析连续性. 查询层面, 动态查询需求的随机性使固定粒度粒球难以适配, 粗粒度虽高效但易遗漏数据, 细粒度则抵消优化价值. 部署层面, 硬件算力差异导致粒球生成速度波动, 且现有模型缺乏硬件感知的自适应机制, 跨环境性能稳定性难以保障.

1.4 本文贡献

1.4.1 系统梳理研究现状

本文对多粒度粒球计算在数据库领域的研究进行全面、系统的综述, 涵盖基数估计、查询优化、参数调优这 3 个核心方向. 通过深入分析现有研究成果, 总结各方向的技术路线、方法优势与局限性, 为后续研究提供清晰的理论框架与研究脉络. 同时, 对比不同研究方法的适用场景与性能表现, 为选择合适的技术方案提供参考.

1.4.2 明确关键研究问题与挑战

结合数据库技术发展趋势与实际应用需求, 提炼出多粒度粒球计算在数据库应用中的关键研究问题, 并深入剖析其技术难点. 针对每个研究问题, 分析当前研究的不足与尚未解决的挑战, 明确未来研究的重点与方向. 此外, 对技术融合过程中模型适配、资源消耗、泛化能力等共性挑战进行系统性总结.

1.4.3 粒球计算与数据库技术融合

粒球计算与数据库技术的融合, 主要围绕基数估计、连接顺序选择、参数调优这 3 个方面展开, 并针对技术融合中的共性挑战提出发展建议. 首先, 对于基数估计, 借助粒球计算对查询数据进行分层处理, 提取不同粒度粒球的统计、分布、结构特征并编码为向量; 结合 Tree-GRU 模型, 对查询执行计划树结构加以处理, 通过学习层次特征输出基数估计值. 其次, 在连接顺序选择上, 基于粒球计算, 结合深度强化学习与 TreeLSTM, 将查询计划以粒球形式呈现; 通过强化学习选择粒球连接顺序, 经 TreeLSTM 处理生成最优查询计划, 借助代价训练和延迟调整两步来优化. 最后, 针对参数调优, 数据库端利用粒球对状态和参数进行多粒度表示, 算法端结合深度强化学习与粒球搜索算法, 依据多粒度表示生成调优策略, 同时辅以规则、训练、深度强化学习这 3 种调优模式. 后文将分章节

详述上述融合思路在各个场景的具体实现.

1.5 本文结构

为助力学者与研究人员洞悉智能数据库的发展趋势, 本文第 2 节阐述粒球计算的背景知识. 第 3 节阐述数据库基数估计的研究现状及其与粒球结合面临的挑战. 第 4 节阐述数据库连接顺序选择的研究现状及其与粒球结合面临的挑战. 第 5 节阐述数据库参数调优的研究现状及其与粒球结合面临的挑战. 第 6 节通过结合具体的应用实例, 对数据库与粒球计算融合的发展进行展望.

2 粒球计算

2.1 粒球计算的定义

粒球计算 (granular-ball computing, GBC)[26]是在多粒度认知计算基础上提出的人工智能理论, 是多粒度认知计算的重要实现形式. 其核心思想是模拟大范围优先的机制, 将数据或问题空间中的信息以粒球 (granular-ball, *GB*) 作为基本信息单元进行表示和处理. 通过多粒度粒球的生成、分裂与优化等操作, 粒球计算能够突破传统点式计算的局限, 实现对复杂数据的高效、鲁棒且可解释的处理.

2.1.1 粒球的形式化

粒球是对数据库元素 (如数据分布、查询语义等) 的多粒度几何抽象模型, 核心特征在于通过低维几何参数实现高维信息表征. 使用粒球进行多粒度特征表示的基本方法, 原因在于球体几何形状完全对称, 在任何维度空间下都具有最简洁的、统一的数学模型表达形式:

$$\{x \mid (x-c)^d \leqslant r^d\} \tag{1}$$

其中, x 为粒球内样本, c 是粒球中心, 表示粒球所抽象的目标对象, 如某部分数据分布、特定查询语义对应的信息集合的聚集核心点, 是该粒球覆盖范围内样本的核心表征坐标; r 是粒球半径, 表示为从中心 c 到粒球几何覆盖范围边界的距离, 用于界定该粒球所覆盖的样本范围, 任何维度下都只需要两个数据来表征: 中心和半径. 因此, 有利于扩展到高维数据空间.

2.1.2 粒球计算的优化目标

粒球计算的核心优化目标是在满足质量约束与覆盖约束的前提下, 实现粒球数量最少、表征精度最高、计算开销最小的多目标优化, 给定数据集 $D=\{x_i, i=1, 2,\ldots, n\}$, 其中, x_i 表示 D 中的第 i 个样本, n 表示样本总数. GB_i ($i=1, 2,\ldots, k$) 表示在 D 上生成第 i 个粒球, k 表示 D 上生成的所有粒球数量. 多粒度粒球计算的标准模型如下所示:

$$f(x, \vec{\alpha}) \to g(GB, \vec{\beta}) \tag{2}$$

$$\text{s.t. } \min \frac{n}{\sum_{i=1}^{k} |GB_i|} + k + loss(GB_i) \tag{3}$$

$$\text{s.t. } quality(GB_i) \geqslant T \tag{4}$$

该模型体现了粒球计算的范式创新: $f(x, \vec{\alpha})$ 表示现有的以点 x 为输入的学习模型; $\vec{\alpha}$ 为模型参数; $g(GB, \vec{\beta})$ 是粒球计算的学习模型, 以粒球 GB 和模型参数 $\vec{\beta}$ 为输入, 当学习模型的输入从传统的样本点 x 转变为粒球 GB 时, 计算模型也从 f 转换为 g. 公式 (2)–(4) 为核心优化公式, 其目标函数反映粒球对样本的覆盖程度. k 表示控制粒球数量; $loss(GB_i)$ 为粒球质量损失项; 约束条件 $quality(GB_i) \geqslant T$ 要求每个粒球的质量不低于阈值 T, 确保粒球的最小粒度与表征可靠性.

2.2 粒球计算的应用

粒球计算凭借多粒度几何化表征核心优势, 已广泛应用于人工智能多个领域, 形成一系列高效、鲁棒且可解释的技术方案[55]. 例如: 1) 粒球支持向量机 (granular-ball-support vector machine, GB-SVM)[56]突破传统点式输入局限, 以粒球为基本学习单元, 通过多粒度分层处理机制在数亿级数据分类任务中实现效率与精度的协同优化, 同时

具备强噪声抵抗能力. 2) 粒球聚类技术依托自适应多粒度生成机制, 无需人工预设参数, 可动态适配不同分布数据, 自动过滤孤立噪声点并降低高维数据计算压力, 显著提升聚类结果的可靠性与适配性. 3) 粒球图粗化 (granular-ball graph coarsening, GBGC)[57]通过节点聚合生成多粒度图结构, 在保留核心拓扑关系的同时缩减图规模, 使图粗化与图分类处理速度较传统方法提升数十至数百倍, 同步强化任务可解释性. 4) 粒球粗糙集技术融合多粒度思想与粗糙集理论, 通过自适应粒化实现属性空间分层约简, 降低属性组合爆炸开销, 适配离散、连续及混合类型数据, 提升特征选择与规则提取的效率和鲁棒性; 5) 在神经网络领域, 粒球驱动模型以粒球替代传统像素点或特征向量输入, 通过结构化表征实现轻量化训练, 在提升模型对抗攻击防御能力的同时, 增强深度神经网络的可解释性.

2.3 粒球计算在数据库中的应用价值

将粒球计算应用到智能数据库的必要性显著, 核心价值体现在以下方面: 其一, 智能数据库需处理海量、高维、异构数据, 传统单粒度表示难以兼顾效率与精度. 粒球的多粒度自适应机制可动态调整粒度, 既降低存储计算开销, 又保障查询分析准确性, 适配复杂数据的存储、检索与分析需求; 其二, 智能数据库对查询优化、基数估计等任务高效性要求高. 粒球通过几何化中心-半径模型, 将数据库特征映射为粒球参数, 利用粒球间距离等关系简化流程, 降低计算复杂度, 提升响应速度与可扩展性; 其三, 粒球质量评价体系基于纯度、汇聚度量化指标, 可保障数据表示有效性, 为数据库知识发现、数据挖掘等智能任务提供可靠的多粒度基础, 助力智能数据库在效率、可解释性与智能化水平方面全面提升, 满足智能数据库高效处理、精准分析、智能决策的核心需求. 总之, 粒球的多粒度特性为解决数据库难题提供了新路径, 后文将详细展开各场景的应用介绍.

3 基数估计

基数估计是数据库查询优化的核心技术, 其准确性直接影响查询计划的选择和执行效率. 传统方法主要包括直方图法[27,28,58]、数据画像法[29,59]和采样法[30,31,60]. 这些方法基于静态统计假设构建数据模型, 在处理多列关联分析和动态数据变化时存在局限性, 复杂查询误差率可达 10 倍以上[61]. 近年来提出的学习型方法 (包含监督学习和无监督学习) 通过数据驱动建模提高了估计精度, 但仍存在依赖标注数据、泛化能力不足以及可解释性差等问题[32]. 表 1 对基数估计的相关工作进行了总结.

表 1 基数估计方法对比

类别		方法	核心技术	估计	编码
传统方法		直方图法	直方图[27,28,58]	选择性	—
		数据画像法	位图、哈希[29,59]	基数	—
		采样法	采样[30,31,60]	基数	—
基于学习的方法	监督学习	混合模型	混合模型	选择性 & 基数	谓词
		集成模型	XGBoost[62]	选择性	谓词
		全连接网络	LocalNN[63]	基数	表 & 谓词
		卷积神经网络	CNN[32]	基数	SQL查询
		循环神经网络	Plan-RNN[64]	代价	元数据 & 操作符
			Tree-LSTM[65]	基数 & 代价	完整计划
	无监督学习	核密度估计	KDE[66]	选择性	样本
		概率图模型	DeepDB[67]	选择性	样本 & 谓词
			贝叶斯网络[68]	选择性	样本 & 谓词
		深度自回归模型	自回归模型[69,70]	选择性	样本 & 谓词

3.1 传统基数估计方法

传统基数估计方法通过不同统计策略刻画数据特征, 为查询优化提供基数估算的依据, 其优缺点如后文表 2 所示.

表 2 传统的基数估计技术

方法	优点	缺点	使用场景	表现	适应性
直方图法	实现简单, 广泛使用	多列关联误差大, 难以适应动态数据	单列、简单查询	中	高
数据画像法	存储轻量, 基数计算快	复杂场景适应差	基数快速统计	中	低
采样法	处理大数据快, 无需全量	采样随机致偏差, 连接采样耗时	大规模数据简算	中	中

3.1.1 直方图法

直方图法的基本思想是根据边界集合 $[B_1, B_2,\ldots, B_m]$ 将列数据划分成若干份, 并统计属性位于边界 B_{i-1} 和 B_i 范围内的元素行数以及位于此范围内不同的元素行数. Lu 等人[27]提出基于总误差最小化的直方图构建方法, 通过优化桶的划分方式使直方图更接近真实数据分布. 该方法更注重直方图与实际数据的匹配程度, 在不同数据场景中均表现良好, 尤其在低维数据中效果显著. Poosala 等人[28]提出改进的直方图方法, 首先系统分析现有方法并建立分类体系, 然后引入新的分区规则和近似技术. 该方法结合采样技术降低构建成本, 在范围查询选择性估计方面表现更优, 能够更准确预测结果集大小, 帮助查询优化器选择更优执行计划. Zhang 等人[58]提出基于密度模型的多维直方图 (density-model-based multidimensional histogram, DMMH) 方法, 旨在解决多维数据选择率估计问题. 该方法通过结合等宽直方图与机器学习模型, 在保证准确性的同时降低空间成本. 与传统方法相比, DMMH 通过聚类识别数据密集区域并进行密度建模, 同时备份非密集区域的非空桶, 有效提升了空间效率. DMMH 在多维数据选择率估计中表现优于同类方法, 尤其在高维数据和低选择率查询场景中优势显著. 该数据驱动方法未来可与基于查询的机器学习模型结合, 进一步优化选择率估计的精度和效率.

3.1.2 数据画像法

数据画像法的基本思想是将位图初始化为零, 然后进行全表扫描, 将每个元素对应位置置为 1, 统计 1 的数量, 即为不同元素的数量. Heule 等人[29]针对 HyperLogLog 基数估计算法进行改进, 采用 64 位哈希函数、偏差校正、稀疏表示等技术, 以降低内存消耗并提升特定范围的基数估计精度. 改进后的 HyperLogLog++算法优化了哈希处理和数据表示方式, 在保证估计精度的同时提高了内存利用效率. 该算法在大型基数场景中具有更高的估计精度和更低的内存占用, 适用于列存储中的字典编码, 能够减少字典大小, 为大规模数据处理提供更高效的基数估计支持, 进而提升查询效率. Durand 等人[59]提出鲁棒基数估计算法, 针对大规模数据集基数估计问题, 采用基于哈希与随机投影的混合模型. 该方法通过创新设计, 仅需少量辅助内存即可精确估计数据集中不同元素的数量. 与传统方法相比, 鲁棒基数估计器显著降低了内存依赖, 能帮助数据库管理系统 (database management system, DBMS) 执行引擎更准确地预测查询操作的基数. 这种准确性可避免选择低效执行方案, 使 DBMS 有效选取最优查询执行路径, 从而提升大规模数据场景下的查询性能.

3.1.3 采样法

采样法的基本思想是从数据集中随机抽取部分样本, 将查询语句应用于上述样本, 然后根据样本命中比例估计出整体的结果数目. Lipton 等人[30]提出基于自适应随机采样的基数估计算法, 用于解决数据库查询结果的估计问题. 该方法分析了采样次数与估计精度的关系, 为实际应用提供了理论支持. 研究引入“sanity bounds”概念, 专门处理数据倾斜和结果查询非常小的场景, 增强了估计的鲁棒性. 该算法能够准确估计查询基数且时间开销低, 与现有方法相比具有效率优势, 并为数据库执行引擎选择高效执行方案提供了可靠支持. Qiu 等人[60]针对 SP (select-project) 和 SPJ (select-project-join) 查询的基数估计问题, 提出加权独特采样 (weighted distinct sampling, WDS) 算法. 该算法根据投影属性的值频率, 按频率平方根比例为每个值分配样本预算, 以提高基数估计的准确性. 与传统均匀采样方法相比, WDS 算法在理论分析和实验中表现更好, 尤其在数据倾斜时优势明显. 同时, 提出基于随机游走的采样算法, 可在单次遍历数据时收集样本, 无需计算完整连接结果, 有效提升了 SPJ 查询的估计效率.

3.2 学习型基数估计法

直方图法[58]处理单列的数据分布效果很好, 并且被广泛应用, 但是对于多列或者多表连接误差较大. 数据画像法[59]必须在误差与时空代价上综合考虑. 采样法[60]具有随机性, 对于连接操作相关的样本采样时间较长. 为了

解决上述局限性, 学习型基数估计方法已逐渐替代传统基数估计方法. 学习型基数估计方法比传统基数估计方法更能适应多表连接等复杂情况, 能更好地捕获数据分布特性. 学习型基数估计方法使用人工智能技术有效地捕获负载与数据的特性, 从而优化数据库查询.

3.2.1 基于监督学习的基数估计

基于监督学习的基数估计流程图如图 2 所示. 首先, 查询特征提取器提取数据集或者数据库中的 SQL 查询的特征, 使用真实的基数作为标签. 然后, 使用查询编码器对特征进行进一步编码. 最后, 将特征编码传入模型进行训练, 预测真实基数. 模型可通过参数优化器优化自身的参数.

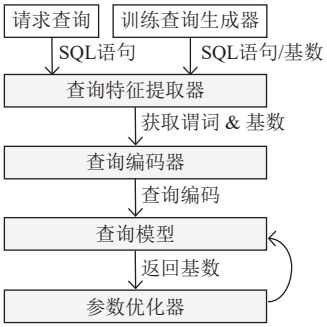

图 2 基于监督学习的基数估计工作流程

基于监督学习的基数估计方法主要包括如下模型.

(1) 混合模型. 基本思想是一方面对谓词、连接类型等查询特征进行编码, 另一方面独立建模数据分布特征; 随后将特征编码结果与分布建模输出结合, 学习两者的关联关系; 最终利用融合后的特征估计查询结果基数, 有效处理多列、多表连接等复杂场景, 提升估计准确性.

Park 等人[71]提出基于混合模型的 QuickSel 方法, 用于解决传统选择性估计中准确性和效率不足的问题. 该方法将数据分布表示为多个子群体的组合, 通过训练学习各子群体权重以准确估计查询选择性. 研究采用均匀混合模型 (uniform mixture model, UMM) 将数据分布近似为多个均匀分布的加权组合, 利用最小二乘误差度量距离并通过高效算法确定最优权重. 与传统方法相比, 能有效避免参数数量指数级增长和训练效率低的问题, 为数据库查询优化提供了可靠的选择性估计支持.

(2) 集成模型. 基本思想是使用基于树结构的 Boosting 方法, 由一组回归模型组成, 每个模型均拟合先前模型产生的误差, 每个回归模型递归地分割具有最高增益的特征维度, 真实基数集中在每个分区中. Dutt 等人[62]利用回归技术进行多维范围谓词选择性估计, 提出回归标签转换和特征工程两种模型设计策略, 该方法可以实现快速估计. 真实数据集上的实验结果显示, 选择性估计的准确性明显优于传统直方图、采样及核密度估计方法.

(3) 全连接神经网络 (fully-connected neural network, FNN). 基本思想是将查询分为不同的结构, 分别编码为 one-hot 向量, 然后将不同结构的 one-hot 向量拼接为一个长的 one-hot 向量, 最后将该 one-hot 向量放入 FNN 中, 得到估计值. 局部神经网络[63]考虑连接路径上的不同谓词, 与任意查询的表示学习相比, 学习连接路径更容易, 查询空间更小. 此外, 如果连接条件是固定的, 则连接表查询的关键特征仅是谓词, 使用简单的多层感知机进行建模.

(4) 卷积神经网络 (convolutional neural network, CNN). 基本思想是将查询分为不同的结构, 分别编码为 one-hot 向量, 然后将不同结构的 one-hot 向量分别传入 CNN 中进行学习, 将学习到的中间向量进行拼接和平均池化, 最后进行估计. Kipf 等人[32]使用 CNN 进行基数估计, 模型将查询分为 3 个部分: 表、连接条件和谓词, 并对涉及的列进行采样操作, 将样本压缩为位图, 拼接到对应的表向量中. 每个部分均由一个卷积网络来处理, 并在平均池化之后连接起来, 可以以较低的计算开销给出准确的基数. 然而, 这种方法仅能处理一些简单的查询, 无法处理带有复杂字符串的谓词以及嵌套查询等.

(5) 循环神经网络 (recurrent neural networks, RNN). 基本思想是基于 SQL 每个部分的时序关系, 通过使用 RNN 进行基数估计. 由于执行计划为树形结构, 所以使用树形结构的循环神经网络是一个很好的选择. 将执行计划的每一个节点都作为循环单元的输入, 高层节点学习到来自低层节点的信息. Marcus 等人[64]提出了一种基于 RNN 的代价模型, 用以预测查询性能, 无需人工设计特征, 能够精确匹配并预测任意优化器选择的查询执行计划的延迟, 通过捕捉查询操作符与输入关系之间的复杂交互, 实现了查询延迟的高精度预测. Sun 等人[65]提出了一种基于端到端学习的成本估计框架, 可以同时估计成本和基数, 相比传统方法具有更好的性能, 基于树形结构模型, 提出了有效的特征提取和编码技术, 并设计了一种基于模式的编码方法, 以提高谓词匹配的泛化能力.

3.2.2 基于无监督的基数估计技术

无监督学习的基数估计通过拟合数据的联合分布间接推断基数, 无需显式的查询基数标签. 基于无监督学习的基数估计器的基本思想是直接从数据中学习联合数据分布. 已知表 T 的联合数据分布是 T 中所有元组 t_i 概率的联合表示, 记为 $P(t_1, t_2,\ldots, t_m)$, 基于无监督学习的基数估计流程如图 3 所示.

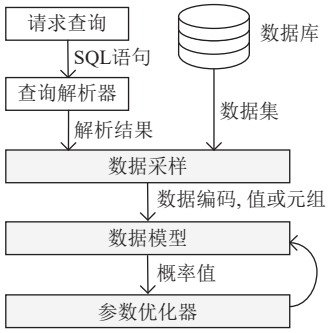

图 3 基于无监督学习的基数估计工作流程

基于无监督学习的基数估计方法主要包括如下模型.

(1) 核密度模型. 基本思想是在样本上构建平滑的核模型, 随机元组的概率是所有模型输出的总和. Heimel 等人[66]提出一种基于核密度估计器 (kernel density estimation, KDE) 的选择性估计器, 该估计器通过优化带宽选择、自适应模型维护和高效的样本维护, 显著提高了估计质量, 并能够适应数据库更新和查询负载的变化, 但无法处理多表连接查询.

(2) 概率图模型. 基本思想是通过图/树结构表示数据关联特性, 其中节点对应数据子集 (如数据列或记录行), 边表征不同部分的条件依赖关系. 在基数估计任务中, 主要应用以下 3 类方法.

1) 和积网络 (sum product network, SPN) 采用分层建模策略: 针对元组分布差异进行聚类划分, 使相同子节点内的元组具备相似分布特征, 通过 SUM 运算符聚合簇内基数; 同时依据列间相关性实施属性分组, 使同组属性保持强关联, 利用 Product 运算符融合各组基数得到最终结果, 典型应用如 DeepDB[67], 该方法通过样本数据拟合联合概率分布, 学习 SUM 节点的连接权重以量化不同分区的贡献度.

2) 贝叶斯网络 (Bayesian) 通过有向无环图建模属性间依赖关系: 以数据表属性为节点, 基于统计搜索算法 (如条件互信息计算) 确定父子节点约束关系, 子节点分布受父节点制约. Chow 等人[68]利用该网络学习数据分布特性, 依据条件概率推导完成基数估计.

3) 深度自回归模型. 基本思想是基于历史值序列预测后续数值, 生成输入元组的条件概率分布列表, 最终整合样本概率实现基数估计. Yang 等人[69]提出的自回归密度模型可表征多列联合分布, 其创新性在于采用渐进抽样机制, 利用学习到的密度分布指导样本选择, 有效支持范围谓词并适应偏斜数据分布. 但该方法存在维度限制, 无法处理高维数据或多表连接场景. 针对此局限, Yang 团队进一步提出 NeuroCard 模型[70], 突破单表建模约束, 直接学习全库跨表关联关系 (无需属性独立性假设), 同步支持范围查询与多表连接操作.

3.3 粒球驱动的基数估计

传统方法如直方图法、概率法和采样法依赖静态数据分布假设, 在处理多表连接、高维关联等复杂查询时, 因难以捕获属性间的动态相关性, 常导致估计误差显著放大. 随着人工智能技术的发展, 基于深度学习的方法逐渐在查询优化中获得应用, 尤其是乔少杰等人[72]提出的一种基于树型门控循环单元 Tree-GRU 的基数和代价估计方法, 通过结合查询和执行计划的特征提取, 利用 Tree-GRU 模型进行特征编码和估计, 并针对字符串类型值进行神经网络提取与嵌入, 能够在处理复杂查询时同时准确估计基数和代价, 优于现有基数估计技术. 然而, 现有基于学习的方法通常仅能处理简单查询, 对于多表连接或嵌套查询等复杂情况的处理能力有限.

基于上述挑战, 结合粒球计算理论的高效、鲁棒和可解释性, 可以将粒球计算应用于数据库基数估计中, 将数据划分为不同粒度的粒球, 并与 Tree-GRU 模型结合, 实现对复杂查询基数的高效、鲁棒估计. 为清晰对比不同方法的差异, 对传统方法、基于学习的方法及粒球驱动的方法进行比较, 具体优缺点如表 3 所示.

表 3　基数估计方法优缺点对比

方法	优点	缺点	使用场景	表现	适应性
传统方法	实现简单, 静态数据有效	依赖静态假设, 复杂查询误差大	静态简单查询	中	低
基于学习的方法	简单查询精度高	复杂查询处理差, 可解释性低	简单查询	中	中
粒球驱动的方法	处理复杂查询, 动态适应	大规模验证不足	复杂查询	高	高

笔者所在团队近期提出的粒球驱动的基数估计模型架构如图 4 所示, 具体工作步骤描述如下所示.

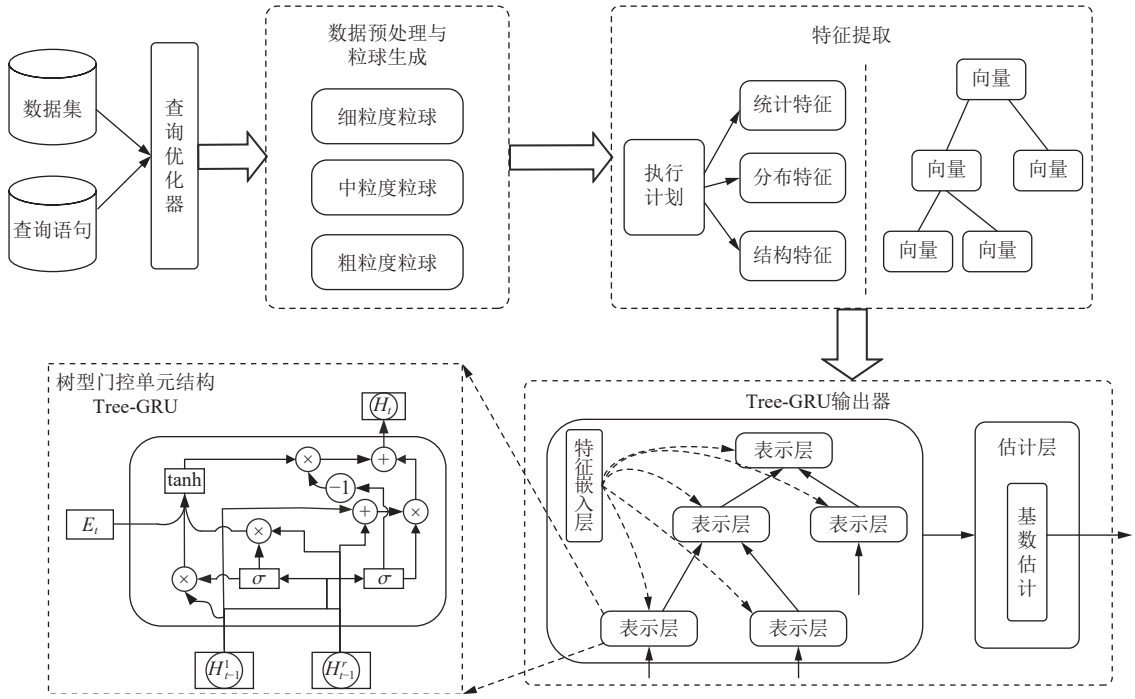

图 4　基于粒球的基数估计模型

(1) 数据预处理与特征提取: 利用粒球计算方法对查询数据进行分层表示, 提取结构化特征. 首先, 从数据库查询优化器中获取 SQL 查询语句, 解析该语句的查询执行计划, 获取数据表的元数据信息, 包括表结构、列属性、索引信息等. 然后, 采用粒球计算方法, 将数据库表数据组织成不同粒度的粒球. 细粒度粒球表示单条记录的数据分布, 用于查询优化的微调; 中粒度粒球表示一组数据的特征, 用于常规查询; 粗粒度粒球表示数据的全局分布, 用于快速估计. 每个粒球对应一个特征向量, 包括: 均值、方差、最大值、最小值等统计特征; 频率直方图、熵等分

布特征; 表间关系、索引分布等结构特征. 并且, 对特征信息分别进行编码, 将其转化为对应的节点向量; 在此基础上, 进一步将粒球节点向量整合转换, 最终形成树型向量, 完成数据结构从节点层面到树状结构的过渡; 最后, 将特征输入 Tree-GRU 估计器, 增强其对数据分布的感知能力.

(2) 粒球生成与优化: 通过自适应粒球计算生成不同粒度的粒球, 动态调整粒度. 根据查询谓词 (如: WHERE 条件) 和数据分布, 对数据进行自适应粒球划分. 在查询条件对应的区域内, 对较为密集的数据区域采用小粒度粒球, 稀疏区域则使用大粒度粒球, 从而在保证估计精度的同时提高计算效率. 使用基于核密度估计或聚类方法进行动态粒球划分, 使其适应不同查询负载. 其中, 基于核密度估计的方法是: 采用高斯函数, 使用 Silverman 经验公式计算自适应带宽; 计算各数据点局部密度后, 以核密度峰值点为中心, 密集区生成半径 h 的小粒球, 稀疏区生成半径 $2h$ 的大粒球; 使用聚类方法时, 则采用 K-means++算法确定初始聚类中心, 即先随机选取 1 个数据点作为第 1 个中心, 之后每次选择距已选中心距离最远的数据点作为新中心, 重复至选够初始聚类数, 以此提升初始聚类的合理性. 若两个粒球的特征相近, 则进行合并, 通过计算粒球之间的相似度, 减少冗余计算并提升模型的泛化能力.

(3) Tree-GRU 基数估计器: 结合 Tree-GRU 进行基数估计, 提高查询优化的准确性. 对于基数估计, 采用 Tree-GRU 模型, 该模型能够处理复杂查询的执行计划树结构. 将查询执行计划转换为树结构, 使用叶节点表示表扫描、索引扫描, 内部节点表示连接、排序等操作, 然后对执行计划中的每个节点进行 one-hot 编码和粒球特征映射, 作为 Tree-GRU 的输入. Tree-GRU 通过自下而上的方式传播信息, 结合执行计划的各节点特征, 进行基数估计. 当查询条件对应的粒球历史数据较少时, 则采用贝叶斯校正进行调整, 当查询条件落在多个粒球之间时, 则进行插值估计, 提高泛化能力. 模型通过学习执行计划的层次结构特征, 输出查询的基数估计值. 通过粒球特征的嵌入, Tree-GRU 不仅能够增强对不同查询的处理能力, 而且能够提高模型对查询模式的适应性.

(4) 模型训练与优化: 通过训练和优化提高模型的泛化能力和计算效率. 在训练阶段生成大量基于真实数据库的数据, 包含查询的执行计划和真实基数 (执行查询得到的实际结果行数), 将粒球特征和查询计划特征作为输入, 真实基数作为标签进行监督学习. 训练数据被用来训练 Tree-GRU 模型, 采用对数均方误差作为损失函数, 优化模型参数. 为了提高训练的效率和准确性, 采用 AdamW 优化器和正则化策略来防止过拟合. 训练后的模型能够处理多表连接、嵌套查询等复杂查询, 并在不同查询负载下提供精确的基数估计.

4 连接顺序选择技术

连接查询优化是查询优化领域中极具挑战性的问题. 连接查询优化主要为复杂的多表连接查询选择最佳连接顺序, 对于数据库管理系统获得良好的查询性能至关重要. 连接顺序选择方法分为 3 类: 传统方法、静态学习方法、动态学习方法. 传统方法中, 动态规划算法[33]通过减少冗余和缩小搜索范围, 通常能选出最佳计划, 但计算成本较高. 启发式方法[34,35]可以更快地生成查询计划, 不过容易产生低质量的方案. 近年来, 基于机器学习的方法被提出以提升连接顺序选择性能. 这类方法通过学习已有连接操作, 解决了传统方法中基数/代价估计不准确导致的偏差. 静态学习方法[73−75]借助神经网络和强化学习, 通过学习历史查询来优化连接顺序, 虽然效率较高, 但存在明显不足: 难以有效表示连接树的结构特征, 无法处理数据库模式更新问题, 也不能在查询执行过程中进行动态调整. 动态学习方法[76,77]中的自适应强化学习, 能够在查询执行时更改连接顺序, 具备一定适应性, 但存在训练时间长、泛化能力较弱以及表征能力不足等问题. 连接顺序选择方法的优缺点如后文图 5 所示.

4.1 传统连接顺序选择方法

传统方法包括动态规划算法和启发式算法, 其目标是为多表连接查询确定高效的执行顺序.

4.1.1 动态规划算法

基于动态规划方法的连接查询优化[78], 核心思想是依次构建 2 个表连接的部分连接树、3 个表连接的部分连接树直到全部表连接的完整连接树, 并为完整连接树指定物理运算符, 如表扫描运算符和连接运算符等, 进而生成执行计划. 动态规划方法尽可能枚举具有不同连接顺序的执行计划, 并结合基数估计器和代价估计器对执行计划进行评估, 最终找到具有最小代价的执行计划.

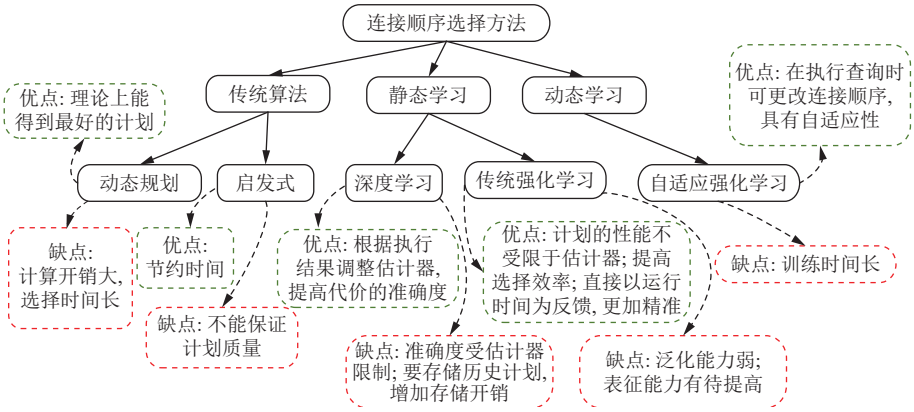

图 5 连接顺序选择方法对比

Ioannidis 等人[33]提出分阶段枚举连接顺序并存储最优子计划的策略. 该策略核心过程包括单表初始化、中间结果递归组合和全局计划回溯, 依靠动态规划表避免重复计算, 理论上可穷举左深树策略空间内的所有可能方案, 确保生成全局最优执行计划. 但该方法的搜索空间随表数量指数级扩大, 导致计算成本高昂, 仅适用于小规模查询. 动态规划方法的性能与策略空间结构密切相关: 在仅包含左深树的 L 空间中, 成本函数呈现复杂多峰形态, 动态规划虽能系统性搜索, 却难以突破指数级开销瓶颈; 高瑞伟等人[79]提出了一种基于异步 Dueling DQN (deep Q-network) 和计划时间预测网络的连接优化器, 通过集成新型编码方法和设计计划时间预测网络, 改进了 Dueling DQN 模型, 并引入异步更新机制, 有效提升了数据库查询性能并减少了规划时间.

Moerkotte 等人[80]提出一种高效的动态规划算法, 目的是生成不含交叉乘积的最优连接树. 该算法通过递归枚举连接子图、扩展子图邻域、构建互补子图连接并计算连接成本, 支持内连接、外连接、反连接等多种连接类型, 以及包含多个关系的复杂连接谓词. 算法将时间复杂度优化到接近理论下界, 解决了传统动态规划算法处理复杂连接谓词和非内连接类型的局限, 能有效处理多元连接场景, 保障复杂查询优化的可行性, 适用于数据库管理系统、数据仓库和大数据分析等场景, 为复杂查询的执行计划优化提供了高效且通用的解决方案.

4.1.2 启发式算法

基于启发式方法的连接查询优化[5], 核心思想是借助预先设定的启发式规则, 按优先级挑选高效的连接顺序与操作方式, 以此减少中间结果的数据量和计算复杂度, 进而提升查询执行效率.

启发式算法通过简化搜索策略, 大幅提高了数据库连接计划选择的计算效率, 尤其适用于大规模表连接场景. 以 QuickPick-1000 算法[35]为例, 其通过随机采样生成候选计划, 经数据库代价估计器评估后选出最优解. 这种随机性使其能够探索动态规划难以覆盖的状态空间, 在处理复杂查询时, 生成计划的时间仅为动态规划的极小部分. 但随机采样存在盲目性, 可能导致遗漏更优解, 极端情况下甚至会生成比传统方法更差的计划. 例如, 在高度相关的表连接中, 随机生成的计划可能因缺乏对数据分布的理解, 而选择高代价的连接顺序.

为改善随机搜索的不足, 基因优化器[81]引入进化机制, 通过模拟生物遗传过程生成候选计划. 该算法应用于 PostgreSQL 中, 用于优化左深树结构的连接顺序, 通过交叉和变异操作对种群进行迭代优化. 比如, 初始种群中的计划可能为 A-B-C-D 和 B-A-D-C, 交叉操作生成 A-B-D-C, 变异则随机调整表的顺序. 这种策略通过概率选择机制平衡探索与开发, 但收敛速度受种群大小和迭代次数影响, 可能因过早收敛而陷入局部最优.

另一类贪心算法如 GOO (greedy optimization for ordering)[82], 采用短视决策, 每一步选择当前代价最小的连接操作, 例如优先合并中间结果集较小的两张表, 逐步构建完整的连接计划. 这种贪心策略的时间复杂度为 $O(n^3)$, 显著低于动态规划的指数级复杂度, 但可能因忽略后续连接的累积代价, 导致全局次优. 例如, 早期选择较小的中间结果, 可能导致后续连接需要处理更大的数据量, 反而增加整体执行时间.

4.2 基于学习的连接顺序选择

基于学习的方法可以在较短时间内有效地选择更好的方案. 通常将现有基于学习的方法分为静态学习法[73−75]和动态学习法[76,77]. 静态学习法通过从历史查询中学习以提高未来查询的性能. 动态学习法可以在查询执行过程中动态学习并调整连接顺序.

4.2.1 静态学习法

在数据库查询优化领域, 静态学习法是机器学习驱动的计划选择算法的重要组成部分. 该方法通过分析历史查询数据进行学习, 训练模型指导未来查询执行, 从而提升查询连接顺序的优化效率. 静态学习法主要包括深度学习和强化学习两个方向, 其核心目标是从历史执行计划中挖掘潜在模式, 为新查询生成更优的执行策略.

Marcus 等人[73]提出了一种基于受限的多臂老虎机的学习型优化策略 Bao (Bandit-optimizer), 旨在解决查询优化中的计划选择和物理算子优化问题. 该方法通过在现有数据库优化器之上构建学习层, 利用强化学习技术在运行时根据历史执行经验自动选择最优查询计划. Bao 算法通过对查询特征和性能数据的持续学习, 能够针对特定数据集和工作负载实现显著的性能提升. 其最大的优势在于良好的可集成性, 能够以“零修改”方式轻松嵌入 PostgreSQL 等现有数据库管理系统中, 在处理复杂查询和动态负载场景时表现出优异的鲁棒性和环境适应性.

Marcus 等人[74]还提出深度强化学习优化方法 ReJOIN, 针对传统算法在连接顺序优化中无法利用历史经验的问题, 采用策略梯度方法, 利用近端策略优化 (proximal policy optimization, PPO) 算法训练神经网络. 策略网络根据当前状态预测动作的概率分布, 执行时从概率分布中采样选择动作, 平衡新动作的探索与已知动作的利用. 通过观察多个查询, 记录每次查询的策略参数和奖励, ReJOIN 更新神经网络权重以优化策略. 性能评估显示, ReJOIN 生成计划的成本和执行时间与 PostgreSQL 优化器相当或更优, 有助于提升查询性能和系统效率.

尽管上述方法能生成低代价连接顺序并保持较高效率, 但存在神经网络结构简单的问题, 无法有效表示连接树结构特征, 也难以处理数据库模式更新. 为解决上述问题, Yu 等人[75]提出一种基于树结构长短期记忆网络 (Tree-LSTM) 的强化学习优化器 RTOS (reinforcement learning with Tree-LSTM for join order selection), 用于解决数据库连接顺序选择问题. 传统优化方法在处理复杂查询时存在扩展性和准确性问题, 现有深度强化学习方法因采用固定长度向量编码, 难以捕捉连接树结构信息, 对数据库模式变化的适应性较弱. RTOS 集成创新改进: 1) 利用 Tree-LSTM 对连接树进行编码, 动态捕捉其结构特征, 避免不同结构连接计划的编码混淆; 2) 借助动态图特征和共享权重机制, 支持表/列添加及多别名表处理, 提升模型对模式变化的适应能力.

4.2.2 动态学习法

与静态学习法不同, 动态学习法侧重于使用自适应查询处理来学习连接顺序, 其最大的特点是即使在查询执行过程中, 也可以更改连接顺序, 以适应不断变化的查询环境和数据特征.

Avnur 等人[76]为解决大型分布式数据库中传统静态查询优化技术在动态环境下的不足, 提出了 Eddy 连续自适应查询处理机制. 该机制通过动态调整查询操作符的顺序, 实现细粒度的在线优化, 能够实时适应计算资源、数据选择性和用户偏好的变化. Eddy 的核心设计基于“对称性时刻”和“同步屏障”特性, 利用 Ripple Join 家族算法的高频对称性与自适应屏障机制, 支持元组以灵活顺序流经操作符. 该机制通过维护元组的“就绪位”和“完成位”实现动态路由, 并采用两种策略进行优化: Naive Eddy 基于流体动力学背压机制, 按操作符处理速率分配元组; Lottery Eddy 通过彩票调度算法学习操作符效率, 平衡成本与选择性差异. 但这两种策略依赖简单反馈机制, 缺乏对动态环境中多元因素 (如数据源延迟、语义约束) 的综合建模能力, 限制了其在复杂场景下的优化精度. 针对上述问题, Trummer 等人[77]提出了新型数据库系统 SkinnerDB, 通过强化学习实现可靠的连接顺序优化, 其系统架构如图 6 所示. 与传统方法不同, SkinnerDB 不依赖数据统计和成本模型, 而是利用强化学习算法在查询执行过程中动态探索最优连接顺序. 该系统将查询执行划分为多个时间切片, 在不同切片中尝试多样化的连接顺序, 并通过评估各切片内的执行进度 (如中间结果生成效率、资源消耗) 筛选高潜力策略, 从而提升查询性能并降低执行成本. 这种分阶段试错与动态优化的模式, 避免了对先验知识的依赖, 为数据驱动的自适应查询优化提供了新范式, 弥补了 Eddy 在复杂动态环境中的不足, 展现了强化学习在数据库内核优化中的应用.

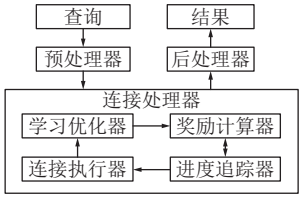

图 6 SkinnerDB 系统组件与处理流程图

4.3 粒球驱动的连接顺序选择

现有基于深度强化学习的连接顺序选择方法 (如 ReJOIN、DQ) 虽然能利用历史执行数据优化策略, 但是普遍采用扁平化编码方式, 导致无法充分建模执行计划树的结构特征[73,74]. 现有研究尝试使用 Tree-LSTM 模型来捕捉树结构特征 (如 RTOS), 然而在处理高维异构数据时, 仍然存在粒球间几何关系抽象不足的问题[75].

针对这一问题, 近期研究提出了基于粒球计算的连接顺序选择方法, 通过多粒度粒球表示、动态几何关系优化和领域知识融合这 3 个创新方向解决现有瓶颈. 为对比不同方法的差异, 现对传统方法、基于学习的方法及粒球驱动的方法进行比较, 具体优缺点及工作原理分别见表 4.

表 4 连接顺序选择方法优缺点对比

方法	优点	缺点	使用场景	表现	适应性
传统方法	简单、成本低	复杂查询误差大	简单稳定数据	中	低
基于学习的方法	在基数估计上表现优异	复杂查询处理弱	简单有规律数据	中	中
粒球驱动的方法	提升效率、鲁棒性	与现有系统集成需调整	复杂高维数据	高	高

笔者团队提出的粒球驱动的连接顺序选择方法[13]工作原理如图 7 所示.

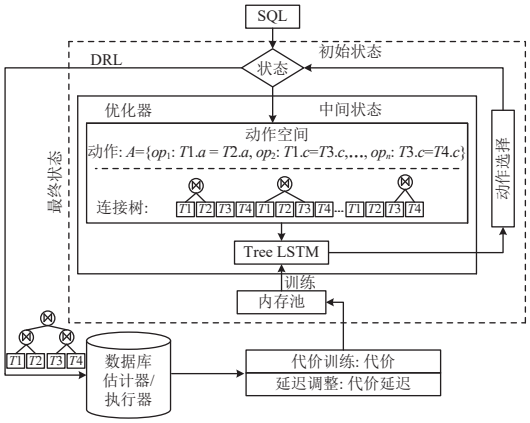

图 7 基于粒球计算的连接顺序选择方法工作原理

该方法将查询计划树的表示方式从传统的节点表示方式转换为粒球表示, 每个粒球表示一个连接操作或连接顺序, 粒球的半径和中心根据查询的连接路径和操作关系自适应生成. 具体来说, 当有 SQL 查询输入时, 系统会先判断状态, 若为初始状态, 则初始化一个空状态 s_0, 其中仅包含查询的基本信息; 若为中间状态, 则基于之前的粒球选择结果形成由多个粒球组成的查询计划森林. 粒球之间的关系表示为几何结构, 极大降低了计算量, 提升了计算效率. 并且, 通过粒球的粗粒度特性, 模型能够在查询计划选择中减少不必要的计算, 提升处理复杂查询时的效率和鲁棒性. 深度强化学习策略通过 Q 网络选择最优的粒球连接顺序, Q 值函数根据当前状态和所选粒球进行计算. 同时利用树型 LSTM 处理每个粒球的表示, 捕捉粒球之间的几何关系和结构信息, 减少了传统方法中计算点距离的复杂度.

在强化学习的训练过程中, 算法会根据当前状态选择下一个粒球, 数据库估计器或执行器则基于所选粒球连接计划对查询执行的代价和延迟进行估计或实际执行, 生成量化的性能反馈并存入内存池, 用于模型训练, 这一过程不断循环, 直到整个查询计划的粒球覆盖完成进入最终状态. 粒球的生成和选择遵循大范围优先的认知规律, 从粗粒度到细粒度逐步优化查询计划. 通过粒球的分裂和合并, 模型能够高效地选择最优连接顺序和操作, 从而避免过多的细粒度计算, 确保查询计划优化的高效性. 对于并非所有表都连接在一起的中间状态 s_i, 计划状态是一个计划森林 F, 根层捕获森林中的信息 $R(s_i)=root(R(tree_1), R(tree_2),\ldots, R(tree_n))$, 其中每个 $tree$ 表示一个粒球生成的查询计划. 通过深度优先搜索遍历粒球计划树, 并利用树型 LSTM 进一步处理后, 最终获得最优的查询计划.

将训练过程分为两个步骤: 代价训练和延迟调整. 在代价训练阶段, 粒球表示的查询计划将用于训练树型 LSTM 神经网络, 得到初步的查询计划; 在延迟调整阶段, 仅选择少数延迟较大的计划进行微调, 进一步优化粒球的质量和连接顺序, 以提升查询执行的整体性能. 通过引入粒球计算方法, 查询计划选择模型能够在保持高效性的同时, 提升其鲁棒性和可解释性. 粒球计算方法自适应地调整粒度, 减少细粒度计算的需求, 使得查询优化过程更加高效和可靠. 不仅提升了深度强化学习策略的性能, 还进一步降低了查询计划生成过程中的计算复杂度, 确保在处理大规模查询时能够找到最优连接顺序.

5 参数调优技术

数据库参数调优技术是指通过调整数据库内部参数来提高系统性能的一种方式[83]. 在大数据时代, 数据量和数据处理需求不断增加, 数据库系统的参数变得越来越复杂, 给定特定工作负载, 寻找最优配置的问题已被证明是 NP-hard 问题[84], 数据库在调优时会面临巨大挑战. 此外, 在实际运用中, 参数调优是运行和管理数据库的一个重要部分, 数据库系统调节到一个合适的参数配置会使得系统性能大幅度提高. 图 8 为李江敏等人[54]提出的新型数据库参数调优框架.

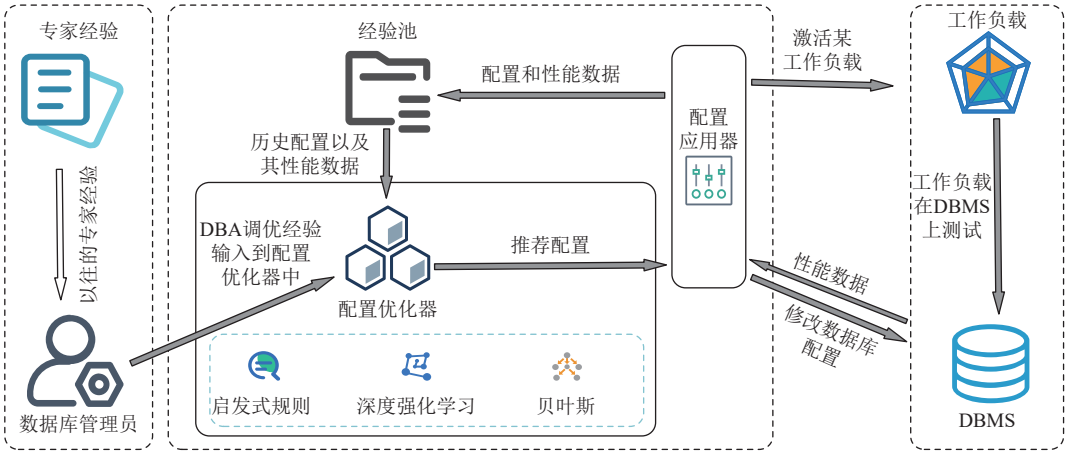

图 8 数据库参数调优过程

5.1 传统参数调优技术

最早的数据库参数调优方法是依靠人工经验和试错, DBA 利用长期调优经验根据现有系统的需要进行调优, 根据负载特点和硬件环境来调整内存分配、缓存大小、I/O 优化等参数, 其优点在于有经验保障. 但传统的 DBA 调优方式劣势明显. 在面对复杂的数据库环境以及大规模数据时, 会存在以下不足: 1) 参数众多且会相互影响, 难以捕捉系统复杂性; 2) 同一参数在不同工作负载下重要性不同, 缺乏普适性; 3) 人工检验的手工配置耗时耗力, 且难以与实际场景同步, 在面对复杂的数据库环境以及大规模数据时, DBA 手动调优的方式往往不能根据特定的工作负载进行有效调优. 因此, 学者相继提出新的调优方法[38−53]来解决上述问题. 后文图 9 展示了近 20 年来参数调优方法的时间线.

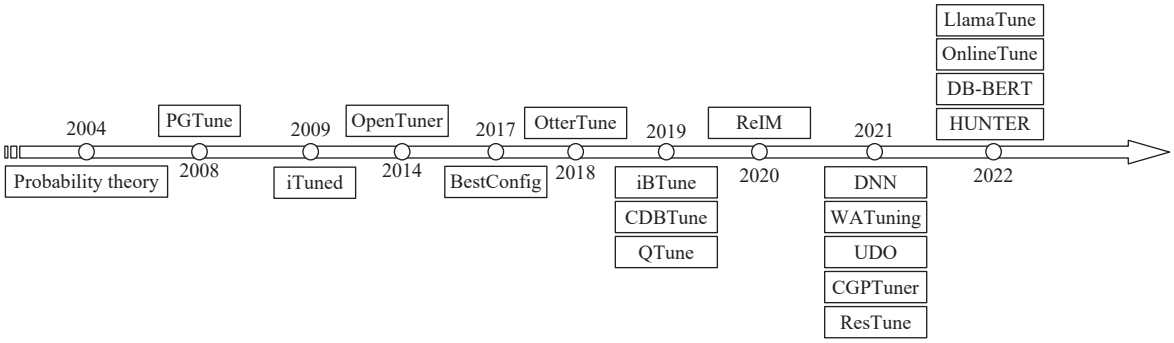

图 9　代表性参数调优方法

5.2　基于学习的参数调优

除了传统的数据库参数调优以外, 近年来, 基于学习的参数调优方法逐渐受到关注. 这类方法利用机器学习技术, 从过去的优化经验中学习, 逐步提高参数搜索的效率. 目前, 基于学习的参数调优分为 3 类: 1) 基于启发式规则的参数调优方法; 2) 基于贝叶斯优化的调优方法; 3) 基于深度强化学习的调优方法.

5.2.1　基于启发式规则的参数调优

基于启发式规则[85]是常用的数据库参数调优方法. 这类方法借鉴了经验丰富的专家或 DBA 的调优经验, 通过构建人工制定的规则来指导参数配置. 其核心思想在于利用现有的知识库或文档中的成功案例, 将调优过程形式化为一系列规则, 从而实现自动化调优. 虽然该方法具有较高的直观性和易于实现的特点, 但这既是优点, 又是缺点, 因为该方法对规则的依赖较大, 会使调参过程难以适应不断变化的系统环境和复杂多变的配置空间.

基于启发式方法可以分为基于规则和基于搜索的方法. 基于规则的方法实则是模拟了手动调优的方式[38], 从 DBA 的经验或数据库使用说明书中构建规则, 随后用算法实现规则进而指导调优过程. 而基于搜索的方法则将配置空间划分为多个子空间, 在子空间中运行给定的工作负载, 随后探索相邻子空间对性能的提升效果. 迭代地, 找到最佳的参数配置. 这两类方法分别适用于不同场景, 同时也存在一些不可忽视的缺点: 采样和搜索过程耗时; 在调优时, 因为某些搜索空间可能被忽略, 所以采样过程可能会错过更好的参数配置.

5.2.2　基于贝叶斯优化的参数调优

为了获得高质量的参数配置, 可以使用贝叶斯优化方法来引导调优过程. 贝叶斯优化 BO (Bayesian optimization)[86,87]是一种顺序模型建模迭代算法. 贝叶斯优化的数据库参数调优方法可以通过探索-开发策略有效地找到高质量的参数配置, 并且优于启发式方法, 如 OtterTune[42]、ResTune[44]. 它使用代理模型来近似目标函数, 并通过迭代观察到的新数据点来顺序更新该模型. BO 使用抽样数据点初始化代理模型. 抽样方法包括随机抽样和拉丁超立方体抽样. 对于每次迭代, BO 使用获取函数来决定在哪里采样新点, 对于参数调优问题, 如图 10 所示, BO 中的探索-开发策略可以描述如下: 1) 基于当前配置对参数值进行轻微修改得到估计值; 2) 尝试探索未访问空间中的参数值. 具体来说, 贝叶斯优化旨在优化未知目标函数 $f(x)$, 如图 10, 求取最大值. 通过在特定点, 即当前配置参数, 获取真实观测值, 依据观测值利用贝叶斯方法估计目标函数在其他点的值, 即估计值 f, 同时包括反映估计值可信程度的置信度. 利用采集函数权衡探索与利用, 选择其最大值位置作为下一次采样点, 迭代寻找目标函数最优值. 实际中, 这一方法用于数据库性能与配置参数关系等复杂关系优化.

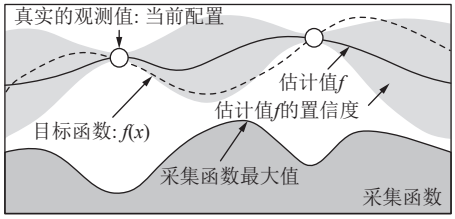

图 10　贝叶斯优化

基于贝叶斯优化的方法是利用采样的参数配置及其性能作为调优的起始点, 重复建模和更新步骤, 直到找到满意的参数配置或达到终止条件, 如: 时间、资源约束. 然而, 基于贝叶斯优化的方法, 特别是采用高斯过程的方法不能有效地扩展到大的配置空间并且会陷入次优解.

5.2.3 基于深度强化学习的参数调优

为了在没有太多历史数据的情况下改进大型配置空间的性能, 研究者提出了基于强化学习的方法[53,54,88], 使用试错策略探索可以让数据库性能有更好的配置空间, 使用深度神经网络使强化学习能够扩展到大型配置空间, 而不限制参数的数量. 此外, 强化学习使学习者能够在特定场景中采取动作, 并按离散时间与环境进行交互并学习. 如 Zhang 等人[53]提出 CDBTune+, 其使用深度确定性策略梯度方法 DDPG (deep deterministic policy gradient) 来优化高维配置空间中的参数设置. 但它无法捕获即将到来的查询工作负载信息, 并且推荐的参数设置可能不会在新的工作负载上实现高性能. 此外, 李江敏等人[54]提出了面向数据库参数调优的协作型多智能体模型, 能够分功能、分参数级别对数据库参数进行调优, 在模型训练初始阶段便能达到主流算法的性能.

总的来说, 基于学习的数据库参数调优的方法一直在平衡效率和适应性. 以上 3 种方法各有合适的使用场景和特点, 对比情况如表 5 所示.

表 5 基于学习的参数调优方法对比

方法	原理	探索方式	成本	适用场景	收敛速度
启发式规则	经验规则, 简单搜索调参	预定义策略, 缺灵活	低	低维、规则明确问题	依赖规则, 可能慢
贝叶斯优化	概率模型预测选优参	平衡探索开发提效	中	中等维度、有不确定性问题	收敛快, 高效寻优
深度强化学习	智能体交互, 依奖励优化	试错积累, 动态调策	高	高维、复杂动态问题	需大量数据, 收敛慢

5.3 基于粒球的参数调优

数据库系统的性能优化一直是数据库领域的关键研究方向, 而参数调优在提升数据库性能中占据重要地位. 传统的数据库参数调优方法, 如基于经验的手动调整和基于规则的自动调整, 在面对日益增长的数据规模和复杂的查询需求时, 逐渐暴露出效率低下、准确性不足等问题. 基于学习的数据库参数调优方法仍存在平衡效率和适应性的问题. 针对以上问题, 夏书银等人[26,55−57]在近期提出了粒球驱动的思想, 针对数据库参数调优的方面能够作为一种新兴的多粒度计算方法, 为对比不同方法的差异, 对传统方法、基于学习的方法及粒球驱动的方法进行比较, 其优缺点如表 6 所示.

表 6 参数调优的优缺点对比

方法	优点	缺点	使用场景	表现	适应性
传统方法	实现简单, 静态场景有效	难应对复杂动态变化	静态、参数简单	中	低
基于学习的方法	简单场景调优精度高	复杂场景弱, 可解释性低	高精度简单参数	中	中
粒球驱动的方法	处理复杂关系, 动态适应强	研究阶段, 大规模验证不足	复杂参数	高	高

数据库系统通常有数百个可调参数, 手动调整这些参数不仅耗时且效率低下, 传统的自动调优方法在大规模数据集上往往需要重新训练模型, 限制了其有效性. 为了解决这一问题, 结合强化学习与粒球计算方法, 笔者团队提出一种混合调优方案, 基于粒球计算的参数调优方法[13], 如图 11 所示. 通过自适应粒度的调整和深度强化学习算法来优化数据库参数选择, 提高数据库的自动调优效率, 进一步提高数据库系统性能.

该方法能够将粒球计算深度融合到数据库状态表示中, 实现多粒度驱动的参数调优. 数据库代理通过提取实例的特征, 利用粒球计算方法对当前数据库的状态和参数进行多粒度表示. 其中, 多粒度划分的规则是按参数影响范围与状态波动频率来划分的: 粗粒度粒球对应全局参数与低频波动状态, 例如数据库整体连接数的长期变化; 中粒度对应模块级参数与中频状态, 像某业务模块 SQL 的响应时间波动; 细粒度对应局部参数与高频状态, 比如单条热点 SQL 的执行延迟峰值, 以此来明确各粒度的覆盖对象. 每个粒球通过自适应调整其粒度, 能够根据查询的特性和数据库的实际负载, 准确捕捉数据库参数调整的关键因素. 粒度表示使得在调优过程中可以根据不同的计

算粒度进行优化, 减少冗余计算, 提高优化效率.

调优过程包括 4 个主要部分: 数据库、算法端、调整端和工作负载. 数据库通过数据库代理提取特征后, 生成的多粒度粒球数据输入算法端; 算法端结合了深度强化学习[88]和基于粒球的搜索算法, 一方面利用粒球的多粒度表示进行调优模式判断, 另一方面粒球计算能为每个参数配置提供粗粒度和细粒度的估计, 让深度强化学习可在短时间内通过多粒度优化策略快速收敛到最佳参数. 调整端每一步调整的输入均为自适应的多粒度表示, 依次生成新的参数推荐并设置至数据库主机; 工作负载通过压力测试和基准任务运行评估调优效果, 将执行性能反馈给调整端, 同时其粒球多粒度引导搜索能加速算法端强化学习的收敛, 进一步优化参数.

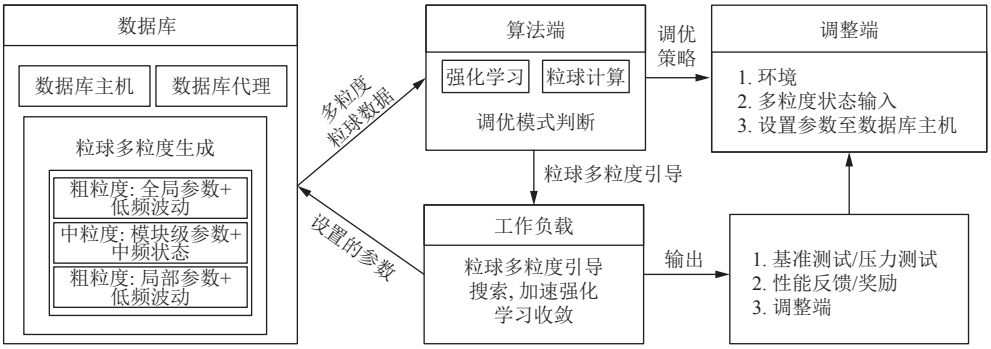

图 11 基于粒球计算的参数调优方法

为了实现在线调优, 提供了 3 种调优模式: 规则模式、训练模式和深度强化学习模式. 1) 规则模式根据专家经验生成初步参数推荐, 缩小调优空间, 提高调优效率; 2) 训练模式通过强化学习不断迭代优化参数, 并利用工作负载测试生成训练样本, 进一步提高调优模型的准确性; 3) 深度强化学习模式结合粒球计算对参数进行全局搜索与优化, 凭借粒球的多粒度表示支持更精确的调优.

6 研究展望与未来趋势

6.1 基于粒球的基数估计

尽管多粒度粒球计算在数据库基数估计领域展现出一定优势, 其特性优势可有效处理复杂查询中的多表关联和高维数据特征[55], 为查询优化提供更准确的基数信息. 同时, 粒球的分层表示和几何特性能更精准捕捉数据分布特征, 显著提高基数估计准确性. 但是, 在处理超高维数据时, 粒球的分层表示和几何特性会受到维度影响, 导致对数据分布特征的捕捉精度下降, 尤其在数据稀疏的高维空间中, 基数估计误差容易被放大[56]. 同时, 粒球的动态调整机制在面对数据分布突变时, 响应速度和调整效率仍有欠缺, 难以在短时间内完成基数的重新估计, 无法满足实时性要求较高的场景. 此外, 现有粒球与机器学习技术的结合多停留在浅层融合, 未能充分发挥两者的协同优势, 导致基数估计模型的泛化能力不足, 在不同类型的数据库和查询场景中适应性较差. 未来, 可以研究如何利用粒球的动态调整特性, 在数据分布发生显著变化时快速重新估计基数; 也可探索如何将粒球与其他先进的机器学习技术相结合, 进一步提升基数估计模型的精度和适应性, 有望为数据库查询优化领域带来突破性的进展, 使基数估计更加准确、高效, 从而提升整个数据库系统的性能.

6.2 基于粒球的连接顺序

在数据库查询优化的连接顺序选择中, 多粒度粒球计算虽提供了创新思路, 利用粒球之间的几何关系和层次结构, 可更高效地搜索最优连接顺序, 并且粒球的多粒度特性和自适应能力能动态调整连接顺序[55], 以适应不同层次的数据关联和查询需求, 但目前仍存在诸多局限. 当面对包含大量表的复杂查询时, 粒球的层次结构会变得异常庞大, 导致搜索最优连接顺序的效率大幅降低, 难以在合理时间内找到全局最优解. 而且, 粒球间几何关系的计算复杂度会随着表数量的增加呈指数级增长, 进一步加剧了查询计划生成的耗时问题. 此外, 现有粒球的自适应能

力主要针对静态数据关联场景设计, 在动态变化的数据库环境中, 如数据频繁更新、表结构动态调整等情况, 粒球对连接顺序的动态调整效果不佳, 容易出现查询计划与实际数据分布不匹配的问题[57]. 未来, 可以深入研究粒球在连接顺序选择中的应用, 如何利用粒球的层次结构来处理更复杂的查询场景, 或者如何结合强化学习等技术来进一步优化连接顺序的选择过程. 利用上述技术有望显著提高查询计划的全局最优性, 减少查询执行时间和资源消耗, 将有助于推动数据库查询优化技术的发展, 使数据库系统在处理大规模复杂查询时更加高效和智能.

6.3 基于粒球的参数调优

粒球计算在数据库参数调优中具有独特价值, 能够通过缩减参数搜索空间, 也能在一定程度上提高调优效率, 为数据库参数配置提供有针对性的指导, 助力提升数据库系统性能[56], 但目前的粒球计算方法主要适配结构化数据的参数调优, 对于图像、文本等非结构化数据场景下的参数优化, 尚未形成成熟的解决方案, 难以满足多元化数据环境的调优需求. 并且面对日益复杂的数据库架构, 现有粒球计算方法对复合参数组合、多级参数关联等复杂场景的支持能力不足, 难以精准处理参数间的耦合关系[57], 影响调优结果的全面性. 此外, 在算法性能方面, 尽管粒球计算缩减了参数搜索空间, 但在大规模数据库系统中, 调优算法的计算开销仍较为可观, 尤其在实时性要求较高的场景下, 参数推荐的时效性有提升的空间. 未来, 可以研究如何利用粒球计算来构建更智能的参数调优模型, 以适应不同规模和类型的数据库系统; 探索如何将粒球与其他参数调优技术相结合, 如遗传算法等, 以进一步提高调优的效果. 上述研究将有助于推动数据库参数调优技术的发展, 使数据库系统能够自动、智能地调整参数配置, 以实现最佳性能.

6.4 基于粒球的 SQL 查询

在 SQL 查询优化领域, 多粒度粒球计算展现出显著应用优势, 其通过粒球化数据预处理、多粒度探索策略可直接对 SQL 查询进行重写与优化, 分层表示特性还能帮助查询优化器高效识别查询关键部分[55], 针对性调整可以提升执行效率. 但目前仍存在诸多局限: 一是复杂 SQL 语法语义的粒球化适配难度大, 面对嵌套子查询、多表关联的复合查询时, 粒球的拆分逻辑易与 SQL 语义脱节, 导致优化方向偏离实际需求; 二是实时查询场景下粒球动态调整的响应滞后, 粒球分裂合并的耗时可能超过查询优化带来的性能提升, 反而降低整体效率; 三是与现有数据库 SQL 优化器的集成成本高, 传统优化器依赖固定规则与成本模型, 难以直接兼容粒球的多粒度决策逻辑. 未来, 可以深入研究 SQL 语法语义与粒球结构的映射机制, 解决复杂查询的粒球化适配问题; 也可探索低延迟的粒球动态调整算法, 提升实时查询场景下的协同效率; 还可设计轻量化粒球-SQL 优化器接口, 降低现有系统的改造难度. 这些研究有望为数据库查询语言创新提供新方向, 进一步提高 SQL 查询的性能与灵活性.

6.5 基于粒球的索引选择

多粒度粒球计算能为数据库索引选择领域提供创新支撑, 可以动态依据数据分布与查询负载变化选择最优索引配置, 通过粒球纯度、覆盖度等量化指标精准评估索引价值[56], 搭配优化算法可快速收敛到最优或近似最优索引方案. 然而当前仍面临显著挑战: 一是多类型索引, 如 B+树索引、哈希索引、全文索引, 其粒球表征差异大, 难以构建统一的粒球评估模型, 导致不同索引的优化效果不均衡; 二是高并发查询场景下, 粒球评估索引价值的计算开销较高, 易占用过多系统资源, 影响查询实时性; 三是数据频繁更新时, 粒球动态调整索引的滞后性明显, 易出现索引配置与数据分布不匹配的问题, 降低查询性能. 未来, 可以深入研究多类型索引的统一粒球表征模型, 实现不同索引的均衡优化; 也可探索轻量化的粒球索引评估算法, 减少高并发场景下的计算开销; 还可结合机器学习技术构建粒球驱动的索引预测模型, 提前预判查询负载变化以调整索引配置. 这些研究将助力提升数据库查询效率, 降低存储与维护成本, 推动索引技术突破发展.

6.6 基于粒球的视图优化

针对数据库视图优化的核心需求, 多粒度粒球计算可以凭借高效、鲁棒的特性展现出突出应用价值[57], 其通过将查询与数据表征为粒球, 利用粒球间的几何关系和层次结构可高效识别等价子查询、筛选最优物化视图, 并且粒球的动态调整能力能实时更新物化视图以适配查询负载变化. 不过目前仍存在不少局限: 一是复杂查询粒球化后的语义一致性难以保障, 等价子查询的粒球匹配易因语义偏差出现误判, 导致物化视图选择失误; 二是多粒度

物化视图的存储资源占用过高, 不同粒度的视图粒球并存会大幅增加存储开销, 降低系统资源利用率; 三是多视图协同优化时粒球关联计算复杂, 视图间的粒球拓扑关系分析耗时较长, 易延长整体查询优化周期. 未来, 可以研究粒球驱动的等价子查询语义校验机制, 提升物化视图选择的准确性; 也可探索基于粒球的物化视图存储压缩策略, 通过动态调整粒度减少存储消耗; 还可设计基于深度强化学习[88]的粒球-视图协同优化框架, 简化多视图间的关联计算逻辑. 这些研究可切实优化数据库系统的整体运行表现, 为复杂查询高效执行提供有力保障.

除了本文提出的基于粒球的基数估计、连接顺序、参数调优、SQL 查询、索引选择、视图优化等方向外, 多粒度粒球计算还可在数据库的存储与分区、数据安全隐私保护和数据流的实时处理等方面展开进一步研究. 未来, 粒球计算凭借其多粒度表征、动态调整与几何优化的核心优势, 有望成为数据库系统的核心技术, 推动数据库技术在智能化决策、高效性处理、高适应性适配等方面的进一步发展. 后续可进一步探索粒球计算与智能数据库在这些领域的深度融合, 通过攻克复杂场景下的技术痛点, 切实提升数据库系统性能, 为数据库领域的技术革新提供切实助力.

References

[1] Ré C, Agrawal D, Balazinska M, Cafarella M, Jordan M, Kraska T, Ramakrishnan R. Machine learning and databases: The sound of things to come or a cacophony of hype? In: Proc. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data. Melbourne: ACM, 2015. 283–284. [doi: 10.1145/2723372.2742911]

[2] Li GL, Zhou XH. XuanYuan: An AI-Native database systems. Ruan Jian Xue Bao/Journal of Software, 2020, 31(3): 831–844 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5899.htm [doi: 10.13328/j.cnki.jos.005899]

[3] Chen JJ, Chen Y, Chen ZB, Ghazal A, Li GL, Li SH, Ou WJ, Sun Y, Zhang MY, Zhou MQ. Data management at Huawei: Recent accomplishments and future challenges. In: Proc. of the 35th IEEE Int'l Conf. on Data Engineering. Macao: IEEE, 2019. 13–24. [doi: 10.1109/ICDE.2019.00009]

[4] Meng XF, Ma CH, Yang C. Survey on machine learning for database systems. Journal of Computer Research and Development, 2019, 56(9): 1803–1820 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2019.20190446]

[5] Li GL, Zhou XH, Sun J, Yu X, Yuan HT, Liu JB, Han Y. A survey of machine learning based database techniques. Chinese Journal of Computers, 2020, 43(11): 2019–2049 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2020.02019]

[6] Yuan HT, Li GL, Feng L, Sun J, Han Y. Automatic view generation with deep learning and reinforcement learning. In: Proc. of the 36th IEEE Int'l Conf. on Data Engineering. Dallas: IEEE, 2020. 1501–1512. [doi: 10.1109/ICDE48307.2020.00133]

[7] Qiao SJ, Han N, Ding ZM, Jin CQ, Sun WW, Shu HP. A multiple-motion-pattern trajectory prediction model for uncertain moving objects. Acta Automatica Sinica, 2018, 44(4): 608–618 (in Chinese with English abstract). [doi: 10.16383/j.aas.2017.c160575]

[8] Li RY, He HJ, Wang RB, Huang YC, Liu JW, Ruan SJ, He TF, Bao J, Zhang Y. JUST: JD urban spatio-temporal data engine. In: Proc. of the 36th IEEE Int'l Conf. on Data Engineering. Dallas: IEEE, 2020. 1558–1569. [doi: 10.1109/ICDE48307.2020.00138]

[9] Wang B, Li WF, Li ZM, Liao QM. Adaptive linear regression for single-sample face recognition. Neurocomputing, 2013, 115: 186–191. [doi: 10.1016/j.neucom.2013.02.004]

[10] Gallo G, Torrisi A. Random forests based WCE frames classification. In: Proc. of the 25th IEEE Int'l Symp. on Computer-based Medical Systems. Rome: IEEE, 2012. 1–6. [doi: 10.1109/CBMS.2012.6266362]

[11] Jiang H, Ching WK, Chu DL. Discriminant analysis in pairwise kernel learning for SVM classification. Int'l Journal of Bioinformatics Research and Applications, 2012, 8(3–4): 305–321. [doi: 10.1504/ijbra.2012.048963]

[12] Han M, Liu B. Ensemble of extreme learning machine for remote sensing image classification. Neurocomputing, 2015, 149: 65–70. [doi: 10.1016/j.neucom.2013.09.070]

[13] Qiao SJ, Li Z, Han N, Xu QQ, Wu T, Yuan G, Wu XD. Artificial intelligence enabled relational database optimization techniques: The state of the art and prospects. Chinese Journal of Computers, 2025, 48(7): 1639–1669 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2025.01639]

[14] Zhu YQ, Liu JX, Guo MY, Bao YG, Ma WL, Liu ZY, Song KP, Yang YC. BestConfig: Tapping the performance potential of systems via automatic configuration tuning. In: Proc. of the 2017 Symp. on Cloud Computing. Santa Clara: ACM, 2017. 338–350. [doi: 10.1145/3127479.3128605]

[15] Negi P, Wu ZN, Kipf A, Tatbul N, Marcus R, Madden S, Kraska T, Alizadeh M. Robust query driven cardinality estimation under changing workloads. Proc. of the VLDB Endowment, 2023, 16(6): 1520–1533. [doi: 10.14778/3583140.3583164]

[16] Zhang JT, Zhang C, Li GL, Chai CL. AutoCE: An accurate and efficient model advisor for learned cardinality estimation. In: Proc. of the

39th IEEE Int'l Conf. on Data Engineering. Anaheim: IEEE, 2023. 2621–2633. [doi: 10.1109/ICDE55515.2023.00201]

[17] Han Y, Li GL, Yuan HT, Sun J. AutoView: An autonomous materialized view management system with encoder-reducer. IEEE Trans. on Knowledge and Data Engineering, 2023, 35(6): 5626–5639. [doi: 10.1109/TKDE.2022.3163195]

[18] Han Y, Li GL, Yuan HT, Sun J. An autonomous materialized view management system with deep reinforcement learning. In: Proc. of the 37th IEEE Int'l Conf. on Data Engineering. Chania: IEEE, 2021. 2159–2164. [doi: 10.1109/ICDE51399.2021.00217]

[19] Budiu M, Chajed T, McSherry F, Ryzhyk L, Tannen V. DBSP: Automatic incremental view maintenance for rich query languages. Proc. of the VLDB Endowment, 2023, 16(7): 1601–1614. [doi: 10.14778/3587136.3587137]

[20] Zhang XY, Wu H, Li Y, Tang ZJ, Tan J, Li FF, Cui B. An efficient transfer learning based configuration adviser for database tuning. Proc. of the VLDB Endowment, 2023, 17(3): 539–552. [doi: 10.14778/3632093.3632114]

[21] Kanellis K, Ding C, Kroth B, Müller A, Curino C, Venkataraman S. LlamaTune: Sample-efficient DBMS configuration tuning. Proc. of the VLDB Endowment, 2022, 15(11): 2953–2965. [doi: 10.14778/3551793.3551844]

[22] Zhang XY, Wu H, Li Y, Tan J, Li FF, Cui B. Towards dynamic and safe configuration tuning for cloud databases. In: Proc. of the 2022 ACM SIGMOD Int'l Conf. on Management of Data. Philadelphia: ACM, 2022. 631–645. [doi: 10.1145/3514221.3526176]

[23] Li PF, Lu H, Zhu R, Ding BL, Yang L, Pan G. DILI: A distribution-driven learned index. Proc. of the VLDB Endowment, 2023, 16(9): 2212–2224. [doi: 10.14778/3598581.3598593]

[24] Gao J, Cao X, Yao X, Zhang G, Wang W. LMSFC: A novel multidimensional index based on learned monotonic space filling curves. Proc. of the VLDB Endowment, 2023, 16(10): 2605–2617. [doi: 10.14778/3603581.3603598]

[25] Cai P, Zhang SM, Liu PR, Sun LM, Li CP, Chen H. An overview of learned index technologies for intelligent database. Chinese Journal of Computers, 2023, 46(1): 51–69 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2023.00051]

[26] Xia SY, Liu YS, Ding X, Wang GY, Yu H, Luo YG. Granular ball computing classifiers for efficient, scalable and robust learning. Information Sciences, 2019, 483: 136–152. [doi: 10.1016/j.ins.2019.01.010]

[27] Lu X, Guan JH. A new approach to building histogram for selectivity estimation in query processing optimization. Computers & Mathematics with Applications, 2009, 57(6): 1037–1047. [doi: 10.1016/j.camwa.2008.10.056]

[28] Poosala V, Haas PJ, Ioannidis YE, Shekita EJ. Improved histograms for selectivity estimation of range predicates. In: Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data. Montreal: ACM, 1996. 294–305. [doi: 10.1145/233269.233342]

[29] Heule S, Nunkesser M, Hall A. HyperLogLog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm. In: Proc. of the 16th Int'l Conf. on Extending Database Technology. Genoa: ACM, 2013. 683–692. [doi: 10.1145/2452376.2452456]

[30] Lipton RJ, Naughton JF, Schneider DA. Practical selectivity estimation through adaptive sampling. In: Proc. of the 1990 ACM SIGMOD Int'l Conf. on Management of Data. Atlantic City: ACM, 1990. 1–11. [doi: 10.1145/93597.93611]

[31] Müller M, Moerkotte G, Kolb O. Improved selectivity estimation by combining knowledge from sampling and synopses. Proc. of the VLDB Endowment, 2018, 11(9): 1016–1028. [doi: 10.14778/3213880.3213882]

[32] Kipf A, Kipf T, Radke B, Leis V, Boncz P, Kemper A. Learned cardinalities: Estimating correlated joins with deep learning. In: Proc. of the 9th Biennial Conf. on Innovative Data Systems Research. Asilomar: CIDR Press, 2019. 1–8.

[33] Ioannidis YE, Kang YC. Left-deep vs. bushy trees: An analysis of strategy spaces and its implications for query optimization. In: Proc. of the 1991 ACM SIGMOD Int'l Conf. on Management of Data. Denver: ACM, 1991. 168–177. [doi: 10.1145/115790.115813]

[34] Chande SV, Sinha M. Genetic optimization for the join ordering problem of database queries. In: Proc. of the 2011 Annual IEEE India Conf. Hyderabad: IEEE, 2011. 1–5. [doi: 10.1109/indcon.2011.6139336]

[35] Waas F, Pellenkoft A. Join order selection (good enough is easy). In: Proc. of the 17th British National Conf. on Databases: Advances in Databases. Exeter: Springer, 2000. 51–67. [doi: 10.1007/3-540-45033-5_5]

[36] Kossmann J, Papenbrock T, Naumann F. Data dependencies for query optimization: A survey. The VLDB Journal, 2022, 31(1): 1–22. [doi: 10.1007/s00778-021-00676-3]

[37] Mozaffari M, Dignös A, Gamper J, Störl U. Self-tuning database systems: A systematic literature review of automatic database schema design and tuning. ACM Computing Surveys, 2024, 56(11): 277. [doi: 10.1145/3665323]

[38] Zheng S, Geng H, Bai C, Yu B, Wong MDF. Boosting VLSI design flow parameter tuning with random embedding and multi-objective trust-region Bayesian optimization. ACM Trans. on Design Automation of Electronic Systems, 2023, 28(5): 74 [doi: 10.1145/3597931]

[39] Ansel J, Kamil S, Veeramachaneni K, Ragan-Kelley J, Bosboom J, O'Reilly UM, Amarasinghe S. OpenTuner: An extensible framework for program autotuning. In: Proc. of the 23rd Int'l Conf. on Parallel Architectures and Compilation. Edmonton: ACM, 2014. 303–316. [doi: 10.1145/2628071.2628092]

[40] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. Journal of Machine Learning Research, 2012, 13: 281–305. [doi: 10.1109/icitacee50144.2020.9239164]

[41] Duan SY, Thummala V, Babu S. Tuning database configuration parameters with ituned. Proc. of the VLDB Endowment, 2009, 2(1): 1246–1257. [doi: 10.14778/1687627.1687767]

[42] Zhang BH, van Aken D, Wang J, Dai T, Jiang SL, Lao J, Sheng SY, Pavlo A, Gordon GJ. A demonstration of the OtterTune automatic database management system tuning service. Proc. of the VLDB Endowment, 2018, 11(12): 1910–1913. [doi: 10.14778/3229863.3236222]

[43] Cereda S, Valladares S, Cremonesi P, Doni S. CGPTuner: A contextual Gaussian process bandit approach for the automatic tuning of it configurations under varying workload conditions. Proc. of the VLDB Endowment, 2021, 14(8): 1401–1413. [doi: 10.14778/3457390.3457404]

[44] Zhang XY, Wu H, Chang Z, Jin SW, Tan J, Li FF, Zhang TY, Cui B. ResTune: Resource oriented tuning boosted by meta-learning for cloud databases. In: Proc. of the 2021 Int'l Conf. on Management of Data. ACM, 2021. 2102–2114. [doi: 10.1145/3448016.3457291]

[45] Zhang XY, Chang Z, Li Y, Wu H, Tan J, Li FF, Cui B. Facilitating database tuning with hyper-parameter optimization: A comprehensive experimental evaluation. Proc. of the VLDB Endowment, 2022, 15(9): 1808–1821. [doi: 10.14778/3538598.3538604]

[46] van Aken D, Yang DS, Brillard S, Fiorino A, Zhang BH, Bilien C, Pavlo A. An inquiry into machine learning-based automatic configuration tuning services on real-world database management systems. Proc. of the VLDB Endowment, 2021, 14(7): 1241–1253. [doi: 10.14778/3450980.3450992]

[47] Tan J, Zhang TY, Li FF, Chen J, Zheng QX, Zhang P, Qiao HL, Shi Y, Cao W, Zhang R. iBTune: Individualized buffer tuning for large-scale cloud databases. Proc. of the VLDB Endowment, 2019, 12(10): 1221–1234. [doi: 10.14778/3339490.3339503]

[48] Chai YF, Ge JK, Chai YP, Wang X, Zhao BX. XTuning: Expert database tuning system based on reinforcement learning. In: Proc. of the 22nd Int'l Conf. on Web Information Systems Engineering. Melbourne: Springer, 2021. 101–110. [doi: 10.1007/978-3-030-90888-1_8]

[49] Ge JK, Chai YF, Chai YP. WATuning: A workload-aware tuning system with attention-based deep reinforcement learning. Journal of Computer Science and Technology, 2021, 36(4): 741–761. [doi: 10.1007/s11390-021-1350-8]

[50] Li GL, Zhou XH, Li SF, Gao B. QTune: A query-aware database tuning system with deep reinforcement learning. Proc. of the VLDB Endowment, 2019, 12(12): 2118–2130. [doi: 10.14778/3352063.3352129]

[51] Trummer I. DB-BERT: A database tuning tool that "reads the manual". In: Proc. of the 2022 Int'l Conf. on Management of Data. Philadelphia: ACM, 2022. 190–203. [doi: 10.1145/3514221.3517843]

[52] Zhang J, Liu Y, Zhou K, Li GL, Xiao ZL, Cheng B, Xing JS, Wang YT, Cheng TH, Liu L, Ran MW, Li ZK. An end-to-end automatic cloud database tuning system using deep reinforcement learning. In: Proc. of the 2019 Int'l Conf. on Management of Data. Amsterdam: ACM, 2019. 415–432. [doi: 10.1145/3299869.3300085]

[53] Zhang J, Zhou K, Li GL, Liu Y, Xie M, Cheng B, Xing JS. CDBTune$^+$: An efficient deep reinforcement learning-based automatic cloud database tuning system. The VLDB Journal, 2021, 30(6): 959–987. [doi: 10.1007/s00778-021-00670-9]

[54] Li JM, Qiao SJ, Han N, Wu T, Gao RW, Peng YH, Xie TC, Ran LQ. A collaborative multi-agent model for database parameter tuning. Acta Electronica Sinica, 2024, 52(11): 3751–3756 (in Chinese with English abstract). [doi: 10.12263/DZXB.20230724]

[55] Xia SY, Peng DW, Meng DY, Zhang CQ, Wang GY, Giem E, Wei W, Chen ZZ. Ball *k*-Means: Fast adaptive clustering with no bounds. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2022, 44(1): 87–99. [doi: 10.1109/TPAMI.2020.3008694]

[56] Xia SY, Lian XY, Wang GY, Gao XB, Chen JC, Peng XL. GBSVM: An efficient and robust support vector machine framework via granular-ball Computing. IEEE Trans. on Neural Networks and Learning Systems, 2025, 36(5): 9253–9267. [doi: 10.1109/TNNLS.2024.3417433]

[57] Xia SY, Wang G, Xu GJ, Zhao S, Wang GY. GBGC: Efficient and adaptive graph coarsening via granular-ball computing. In: Proc. of the 34th Int'l Joint Conf. on Artificial Intelligence. Montreal: IJCAI. org, 2025. 3489–3497. [doi: 10.24963/ijcai.2025/388]

[58] Zhang MF, Wang HZ. Selectivity estimation with density-model-based multidimensional histogram. Knowledge and Information Systems, 2021, 63(4): 971–992. [doi: 10.1007/s10115-021-01547-7]

[59] Durand M, Flajolet P. Loglog counting of large cardinalities. In: Proc. of the 11th Annual European Symp. on Algorithms. Budapest: Springer, 2003. 605–617. [doi: 10.1007/978-3-540-39658-1_55]

[60] Qiu Y, Wang YL, Yi K, Li FF, Wu B, Zhan CQ. Weighted distinct sampling: Cardinality estimation for SPJ queries. In: Proc. of the 2021 Int'l Conf. on Management of Data. ACM, 2021. 1465–1477. [doi: 10.1145/3448016.3452821]

[61] Leis V, Radke B, Gubichev A, Kemper A, Neumann T. Cardinality estimation done right: Index-based join sampling. In: Proc. of the 8th Biennial Conf. on Innovative Data Systems Research. Santa Cruz: CIDR, 2017. 8–11.

[62] Dutt A, Wang C, Nazi A, Kandula S, Narasayya V, Chaudhuri S. Selectivity estimation for range predicates using lightweight models. Proc. of the VLDB Endowment, 2019, 12(9): 1044–1057. [doi: 10.14778/3329772.3329780]

[63] Woltmann L, Hartmann C, Thiele M, Habich D, Lehner W. Cardinality estimation with local deep learning models. In: Proc. of the 2nd

Int'l Workshop on Exploiting Artificial Intelligence Techniques for Data Management. Amsterdam: ACM, 2019. 5. [doi: 10.1145/3329859.3329875]

[64] Marcus R, Papaemmanouil O. Plan-structured deep neural network models for query performance prediction. Proc. of the VLDB Endowment, 2019, 12(11): 1733–1746. [doi: 10.14778/3342263.3342646]

[65] Sun J, Li GL. An end-to-end learning-based cost estimator. Proc. of the VLDB Endowment, 2019, 13(3): 307–319. [doi: 10.14778/3368289.3368296]

[66] Heimel M, Kiefer M, Markle V. Self-tuning, GPU-accelerated kernel density models for multidimensional selectivity estimation. In: Proc. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data. Melbourne: ACM, 2015. 1477–1492. [doi: 10.1145/2723372.2749438]

[67] Hilprecht B, Schmidt A, Kulessa M, Molina A, Kersting K, Binnig C. DeepDB: Learn from data, not from queries! Proc. of the VLDB Endowment, 2020, 13(7): 992–1005. [doi: 10.14778/3384345.3384349]

[68] Chow CK, Liu CN. Approximating discrete probability distributions with dependence trees. IEEE Trans. on Information Theory, 1968, 14(3): 462–467. [doi: 10.1109/TIT.1968.1054142]

[69] Yang ZH, Liang E, Kamsetty A, Wu CG, Duan Y, Chen X, Abbeel P, Hellerstein JM, Krishnan S, Stoica I. Deep unsupervised cardinality estimation. Proc. of the VLDB Endowment, 2019, 13(3): 279–292. [doi: 10.14778/3368289.3368294]

[70] Yang ZH, Kamsetty A, Luan SF, Liang E, Duan Y, Chen X, Stoica I. NeuroCard: One cardinality estimator for all tables. Proc. of the VLDB Endowment, 2020, 14(1): 61–73. [doi: 10.14778/3421424.3421432]

[71] Park Y, Zhong SC, Mozafari B. QuickSel: Quick selectivity learning with mixture models. In: Proc. of the 2020 ACM SIGMOD Int'l Conf. on Management of Data. Portland: ACM, 2020. 1017–1033. [doi: 10.1145/3318464.3389727]

[72] Qiao SJ, Yang GP, Han N, Qu LL, Chen H, Mao R, Yuan CA, Gutierrez LA. Cardinality and cost estimator based on tree gated recurrent unit. Ruan Jian Xue Bao/Journal of Software, 2022, 33(3): 797–813 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6448.htm [doi: 10.13328/j.cnki.jos.006448]

[73] Marcus R, Negi P, Mao HZ, Tatbul N, Alizadeh M, Kraska T. Bao: Making learned query optimization practical. In: Proc. of the 2021 Int'l Conf. on Management of Data. ACM, 2021. 1275–1288. [doi: 10.1145/3448016.3452838]

[74] Marcus R, Papaemmanouil O. Deep reinforcement learning for join order enumeration. In: Proc. of the 1st Int'l Workshop on Exploiting Artificial Intelligence Techniques for Data Management. Houston: ACM, 2018. 3. [doi: 10.1145/3211954.3211957]

[75] Yu X, Li GL, Chai CL, Tang N. Reinforcement learning with Tree-LSTM for join order selection. In: Proc. of the 36th IEEE Int'l Conf. on Data Engineering. Dallas: IEEE, 2020. 1297–1308. [doi: 10.1109/ICDE48307.2020.00116]

[76] Avnur R, Hellerstein JM. Eddies: Continuously adaptive query processing. In: Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data. Dallas: ACM, 2000. 261–272. [doi: 10.1145/342009.335420]

[77] Trummer I, Wang JX, Wei ZY, Maram D, Moseley S, Jo S, Antonakakis J, Rayabhari A. SkinnerDB: Regret-bounded query evaluation via reinforcement learning. ACM Trans. on Database Systems, 2021, 46(3): 9. [doi: 10.1145/3464389]

[78] Selinger PG, Astrahan MM, Chamberlin DD, Lorie RA, Price TG. Access path selection in a relational database management system. In: Proc. of the 1979 ACM SIGMOD Int'l Conf. on Management of Data. Boston: ACM, 1979. 23–34. [doi: 10.1145/582095.582099]

[79] Gao RW, Qiao SJ, Han N, Min SJ, Li H, Qin X, Zhang T, Yuan CA. A join optimizer based on asynchronous dueling DQN and plan latency prediction network. Acta Electronica Sinica, 2023, 51(7): 1868–1874. (in Chinese with English abstract). [doi: 10.12263/dzxb.20220490]

[80] Moerkotte G, Neuman T. Dynamic programming strikes back. In: Proc. of the 2008 ACM SIGMOD Int'l Conf. on Management of Data. Vancouver: ACM, 2008. 539–552. [doi: 10.1145/1376616.1376672]

[81] Horng JT, Kao CY, Liu BJ. A genetic algorithm for database query optimization. In: Proc. of the 1st IEEE Conf. on Evolutionary Computation and IEEE World Congress on Computational Intelligence. Orlando: IEEE, 1994. 350–355. [doi: 10.1109/ICEC.1994.349926]

[82] Fegaras L. A new heuristic for optimizing large queries. In: Proc. of the 9th Int'l Conf. on Database and Expert Systems Applications. Vienna: Springer, 1998. 726–735. [doi: 10.1007/BFb0054528]

[83] Jiang LL, Gao JT. Survey of machine learning for database parameter tuning techniques. Computer Engineering and Application, 2024, 60(3): 1–16 (in Chinese with English abstract). [doi: 10.3778/j.issn.1002-8331.2304-0101]

[84] Sullivan DG, Seltzer MI, Pfeffer A. Using probabilistic reasoning to automate software tuning. In: Proc. of the 2004 Joint Int'l Conf. on Measurements and Modeling of Computer Systems. New York: ACM, 2004. 404–405. [doi: 10.1145/1005686.1005739]

[85] Huang CW, Li YX, Yao X. A survey of automatic parameter tuning methods for metaheuristics. IEEE Trans. on Evolutionary Computation, 2020, 24(2): 201–216. [doi: 10.1109/TEVC.2019.2921598]

[86] Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: Proc. of the 26th Int'l Conf. on

Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2012. 2951–2959.

[87] Shahriari B, Swersky K, Wang ZY, Adams RP, de Freitas N. Taking the human out of the loop: A review of Bayesian optimization. Proc. of the IEEE, 2016, 104(1): 148–175. [doi: 10.1109/JPROC.2015.2494218]

[88] Wang JX, Trummer I, Basu D. UDO: Universal database optimization using reinforcement learning. Proc. of the VLDB Endowment, 2021, 14(13): 3402–3414. [doi: 10.14778/3484224.3484236]

附中文参考文献

[2] 李国良, 周煊赫. 轩辕: AI 原生数据库系统. 软件学报, 2020, 31(3): 831–844. http://www.jos.org.cn/1000-9825/5899.htm [doi: 10.13328/j.cnki.jos.005899]

[4] 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. [doi: 10.7544/issn1000-1239.2019.20190446]

[5] 李国良, 周煊赫, 孙佶, 余翔, 袁海涛, 刘佳斌, 韩越. 基于机器学习的数据库技术综述. 计算机学报, 2020, 43(11): 2019–2049. [doi: 10.11897/SP.J.1016.2020.02019]

[7] 乔少杰, 韩楠, 丁治明, 金澈清, 孙未未, 舒红平. 多模式移动对象不确定性轨迹预测模型. 自动化学报, 2018, 44(4): 608–618. [doi: 10.16383/j.aas.2017.c160575]

[13] 乔少杰, 李洲, 韩楠, 徐泉清, 吴涛, 袁冠, 吴信东. 人工智能赋能关系型数据库优化技术: 现状与展望. 计算机学报, 2025, 48(7): 1639–1669. [doi: 10.11897/SP.J.1016.2025.01639]

[25] 蔡盼, 张少敏, 刘沛然, 孙路明, 李翠平, 陈红. 智能数据库学习型索引研究综述. 计算机学报, 2023, 46(1): 51–69. [doi: 10.11897/SP.J.1016.2023.00051]

[54] 李江敏, 乔少杰, 韩楠, 吴涛, 高瑞玮, 彭钰寒, 谢添丞, 冉黎琼. 面向数据库参数调优的协作型多智能体模型. 电子学报, 2024, 52(11): 3751–3756. [doi: 10.12263/DZXB.20230724]

[72] 乔少杰, 杨国平, 韩楠, 屈露露, 陈浩, 毛睿, 元昌安, Gutierrez LA. 基于树型门控循环单元的基数和代价估计器. 软件学报, 2022, 33(3): 797–813. http://www.jos.org.cn/1000-9825/6448.htm [doi: 10.13328/j.cnki.jos.006448]

[79] 高瑞玮, 乔少杰, 韩楠, 闵圣捷, 李贺, 覃晓, 张桃, 元昌安. 基于异步 Dueling DQN 和计划时间预测网络的连接优化器. 电子学报, 2023, 51(7): 1868–1874. [doi: 10.12263/dzxb.20220490]

[83] 姜璐璐, 高锦涛. 面向机器学习的数据库参数调优技术综述. 计算机工程与应用, 2024, 60(3): 1–16. [doi: 10.3778/j.issn.1002-8331.2304-0101]

作者简介

乔少杰, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为人工智能数据库, 时空数据库.

杨蕾, 硕士生, 主要研究领域为人工智能数据库.

夏书银, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为粒球计算.

韩楠, 博士, 副教授, 主要研究领域为人工智能数据库, 时空数据库.

苟浩淞, 博士, 教授级高级工程师, 主要研究领域为人工智能数据库.

何函, 硕士生, 主要研究领域为人工智能数据库.

袁冠, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为人工智能, 大数据.

唐明靖, 博士, 教授, CCF 专业会员, 主要研究领域为深度学习, 图表示学习, 智能计算.