

基于云边多数仓架构和代价预测模型的查询调度

高叙宁¹, 杨松², 李明哲¹, 张岩峰³

1. 东北大学医学与生物信息工程学院, 辽宁 沈阳 110169;
2. 沈阳市纪检监察大数据监督保障中心, 辽宁 沈阳 110179;
3. 东北大学计算机科学与工程学院, 辽宁 沈阳 110169

摘要

随着云计算和大数据的发展, 传统的本地数据仓库面临着扩容困难、数据处理效率低下的问题, 由此云边架构的数据仓库应运而生。该架构的数据仓库分布在云中心和边缘端, 使数据存储和处理更加灵活, 在保证查询效率的前提下, 为数据安全、数据隐私和跨地域数据共享等业务提供支持。设计基于云边多数仓的调度框架, 集成以机器学习技术为核心的查询代价预测模型, 实现了在多个查询粒度上的云边协同执行和云边择优执行, 提高了系统的整体性能和查询效率。此外, 提出了有利于查询代价信息增强的多特征融合和特征筛选方法。此调度框架和优化算法在SSB和TPC-DS数据集上均表现出显著的性能提升, 为云边多数仓架构下的数据仓库查询调度提供了有效的解决方案。

关键词

云边多数仓; 调度框架; 查询代价预测; 随机森林; 特征筛选

中图分类号: TP319

文献标志码: A

doi: 10.11959/j.issn.2096-0271.2025011

Query scheduling based on cloud-edge multi-data warehouse architecture and cost prediction model

GAO Xuning¹, YANG Song², LI Mingzhe¹, ZHANG Yanfeng³

1. College of Medicine and Biological Information Engineering, Northeastern University, Shenyang 110169, China
2. Shenyang Discipline Inspection and Supervision Big Data Supervision and Security Center, Shenyang 110179, China
3. School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China

Abstract

With the development of cloud computing and big data, traditional local data warehouses are difficult to expand and have low data processing efficiency. As a result, the data warehouse of cloud edge architecture comes into being. The architectures data warehouses are distributed in the cloud center and at the edge, making data storage and processing more flexible and supporting business such as data security, data privacy and cross-geographic data sharing while ensuring query efficiency. This paper designed a scheduling framework based on cloud edge multi-data warehouses, integrated the query cost prediction model with machine learning technology as the core, and realized cloud edge

collaborative execution and cloud edge selective execution on multiple query granularity, so as to improve the performance and query efficiency of the whole system. In addition, the multi-feature fusion and feature selection method is proposed to enhance the query cost information. The scheduling framework and optimization algorithm achieve significant performance improvement on SSB and TPC-DS datasets. It provides an effective solution for the query scheduling of data warehouse under the cloud edge multi-data warehouses architecture.

Key words

cloud-edge multi-data warehouses, scheduling framework, query cost prediction, random forest, feature selection

0 引言

随着互联网的发展和物联网技术的普及,企业面临前所未有的数据涌入形势。数据已成为企业竞争的核心资源,这些数据不仅包含传统的结构化数据,还包括来自传感器、社交媒体、日志文件等的非结构化或半结构化数据。随着数据量的增长和数据来源的多样化,企业面临着数据管理和分析的挑战。因此,企业需要处理不断增长和多样化的数据流。数据仓库是一个存储历史数据的数据集合,为企业提供全面的数据分析和决策支持^[1]。数据仓库的概念起源于20世纪80年代,用于支持组织在决策过程中的分析需求^[2],主要用于查询和分析。数据仓库集成了不同数据源的数据,对数据进行预处理、转换和清洗,以使用户进行数据挖掘和分析。随着云计算技术的发展,数据仓库架构经历了从本地到云中心再到云边多数仓的演进过程。

在数据仓库架构的演进过程中,传统本地数据仓库是企业早期选择的方案,使用传统硬件和软件在本地数据中心部署^[3-5],存在一些局限性,例如,数据传输延迟高、数据存储成本高、对实时数据处理的限制等,这些都限制了企业对数据

的实时分析和应用。在当前的商业环境中,企业需要更快速地洞察数据,并做出实时的决策。传统数据仓库架构往往无法满足这种需求,因此需要一种新型的数据仓库架构来支持数据的多样性处理和实时性分析。随着云计算技术的发展,云中心数据仓库越来越受欢迎,它具有弹性、可扩展、低成本、可靠和灵活等特点^[6-8]。企业可以利用云端和边缘节点的计算和存储资源,更好地处理数据。云计算提供了弹性、可扩展的计算和存储资源,而边缘计算则使数据可以在距离数据源更近的地方被处理和分析,从而降低了延迟并提高了实时性。最新的趋势是采用云边多数数据仓库方案,结合云计算和边缘计算技术,将数据分布在本地和云端多个地方,该方案具备分布式存储、异构数据处理、灵活和可扩展等优势,满足了互联网时代下数据处理多样性和高实时性的要求,这种方案将在未来的数据仓库市场中发挥重要作用。云边多数数据仓库是一种新型的数据仓库解决方案,它在云端和边缘同时部署多个数据仓库,同时支持多种数据存储和处理方式。云边多数数据仓库的特点包括分布式存储、异构数据处理、灵活性和可扩展性、数据安全、成本低^[9-10]。在云边多数仓中,查询调度对于确保整个系统的性能、效率和可扩展性至关重要。调度指的是智能决策过程,根据一系列预定的标准

和算法，动态地决定数据查询应当在哪个位置（云中心或边缘节点之一）执行，以实现数据处理任务的最优分配。这一过程涉及对查询的特性、数据的位置、网络状况、资源可用性等多方面因素的综合考虑。云边多数仓架构调度框架的核心目的是提高数据处理效率，降低延迟，优化资源利用，并支持高并发处理和实时数据分析的需求。

本文的主要研究内容为构建云边多数仓架构调度框架，实现多查询粒度的云边协同，对比分析查询执行代价预测算法，查询执行代价多特征融合与筛选，利用自适应提升（AdaBoost）算法改进基于随机森林的代价预测模型。首先，构建云边多数仓架构调度框架以实现多查询粒度的云边协同，分别在云中心搭建一个数据仓库、在边缘搭建多个数据仓库，建立基础的云边多数仓架构。接下来，构建查询调度框架，通过调度框架将不同的查询分别调度到云中心和边缘数仓去执行。具体而言，云端数据仓库应该承担计算复杂的全局查询任务，而边缘端数据仓库应该主要用于边缘计算。云端数据仓库应该存储所有数据，而边缘端数据仓库只存储部分数据或者缓存部分数据。此外，还需要考虑如何将数据从云端传输到边缘端，以及如何将边缘端处理过的数据传输到云端。然后，对比分析查询执行代价预测算法，在预测查询执行代价场景中，通常需要综合考虑多个因素来选择最合适的机器学习算法。本文主要考虑4个方面。①数据类型和数据量：不同的机器学习算法适用于不同类型和规模的数据；②数据特征：在选择机器学习算法时，需要考虑特征工程的难度和可行性，不同算法对数据特征的要求不同；③精度和速度：机器学习算法的选择还应考虑精度和速度之间的平衡；④算法

的可解释性：有些算法难以解释其内部工作原理，而有些算法则可以提供直观的结果。选择哪种算法还取决于应用场景的要求。本文通过多次试验，利用交叉验证的方式，从常见的机器学习分类算法中选择在实际场景中表现最好的算法作为预测模型的内核。在基于随机森林算法构建模型时，可以使用特征融合^[11]和筛选^[12]来优化模型的性能。在代价预测模型中，AdaBoost算法和随机森林都是常见的集成学习算法^[13]。

尽管随机森林在许多场景下表现良好，但在某些数据集上，它可能无法达到最佳效果。因此，本文考虑使用AdaBoost算法来改进随机森林。具体地，可以利用AdaBoost算法来调整随机森林中每个决策树的权重，使表现较差的决策树在整体模型中占据较小的权重，而表现较好的决策树则占据较大的权重。这样可以进一步提高随机森林的性能，即提高查询执行代价预测的准确率，将查询语句更准确地发送给执行效率比较高的云中心数据仓库或者边缘数据仓库，以实现多条结构化查询语言（structured query language, SQL）语句整体更高的执行效率。

1 相关工作

国内外在云边多数数据仓库、数据仓库查询代价预测和云边多数数据仓库查询调度等方面的研究已经取得了一系列成果。在云边多数数据仓库方面，国内研究主要集中在架构设计、数据处理与存储、安全等方面。在数据仓库查询代价预测方面，研究者通过分析查询语句特征来预测其执行代价，以便进行查询优化。国内的研究者提出了基于模板匹配和深度学习的方

法^[14-16]，而国外的研究者探索了基于随机森林和卷积神经网络等的代价预测方法。这些方法在提高查询性能和优化查询执行计划方面都取得了一定的成果^[17-24]。在云边多数据仓库查询调度方面，研究者探索了静态查询调度、动态查询调度、基于成本和预测的查询调度等多种方法。云中心数据仓库是建立在云计算环境中的大型数据仓库，用于集中存储和管理各种类型的数据。它利用分布式存储和计算，并通过虚拟化技术快速扩展其能力。

查询执行代价（位置）预测常用的算法包括逻辑回归模型、 K 近邻模型、朴素贝叶斯模型、支持向量机模型、人工神经网络和随机森林模型^[25-29]。逻辑回归适用于二元分类， K 近邻处理非线性分类，朴素贝叶斯适用于数据集分类和文本分类问题，支持向量机可处理线性和非线性场景，人工神经网络通过学习来解决复杂问题，随机森林通过建立多棵决策树提高分类精度和泛化能力。

机器学习在数据仓库中有以下应用。

- 自动化查询优化：通过历史查询数据和模型训练来改进查询执行计划和优化器参数，提升查询性能和数据质量^[30-32]。
- 数据库性能监控和故障预测：利用机器学习监控性能瓶颈、检测异常并预测故障、增强数据仓库的可靠性和可用性^[33]。
- 数据存储优化：机器学习用于改善数据压缩、存储方式和分区，降低成本、提升访问速度和查询效率^[34-35]。
- 数据清洗和去重：利用机器学习分析数据特征并建模，自动检测和纠正异常值和重复值，提高数据质量并为数据查询提供服务。

然而，现有研究仍存在一些亟待解决的问题。首先，现有研究尚未完全解决云边多数据仓库的高效调度框架问题，

虽然已有一些工作在这方面进行了探索和尝试，但整体框架还未完全搭建完成；其次，现有的机器学习算法在不同场景下的通用性有待提高；另外，查询代价预测还可考虑其他查询特征，以提高准确性；最后，更精准的代价预测模型仍需通过组合多种机器学习算法和筛选特征来进一步提升。云边多数据仓库、查询代价预测和查询调度领域的研究仍在继续，未来可以进一步提高整体的性能和适用性。

2 基于云边多数仓和代价预测模型的查询调度框架

2.1 调度框架整体分析

为了实现业务的高效部署，企业开始在云端和边缘计算节点上构建多个数据仓库，形成了云边多数仓架构。该架构包括云中心数据仓库、边缘数据仓库客户端和调度器。该架构包括云中心数据仓库、边缘数据仓库客户端和调度器，本文搭建的云边多数仓架构如图1所示。

调度框架主要包含4个部分：云中心数据仓库、边缘数据仓库、客户端和调度器。云中心数据仓库部署在云端，采用存算分离架构，用于存储所有数据，并处理来自边缘用户的查询请求。边缘数据仓库部署在本地，采用存算紧耦合架构，仅存储本地业务所需的数据，并处理本地用户的查询请求。客户端位于用户和边缘数据仓库之间，起到从属关系的作用。局部查询请求在从属的本地数据仓库上执行，而跨数仓查询请求直接发送到云中心数据仓库执行。调度器位于云中心，在接收到用户的查询请求后，通过查询语句解析和集群状态动态感知等步骤，最终确定SQL的执行位置，并将请求转发到相应的位置。

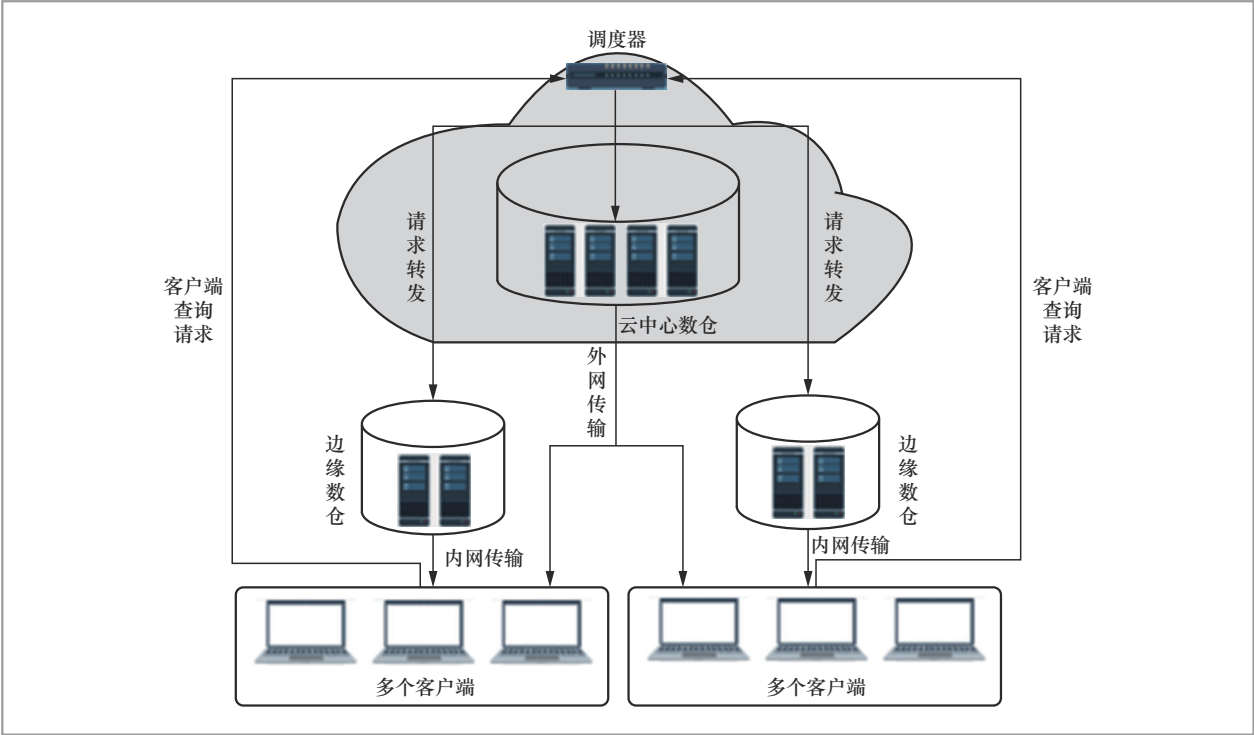


图1 云边多数仓架构

调度器的工作流程包括查询语句解析、动态感知集群状态、位置决策和转发请求等步骤，通过这些步骤的协同配合，实现了

在多SQL查询场景下，对云边数仓的高效利用和资源的优化调配，具体工作流程如图2所示。

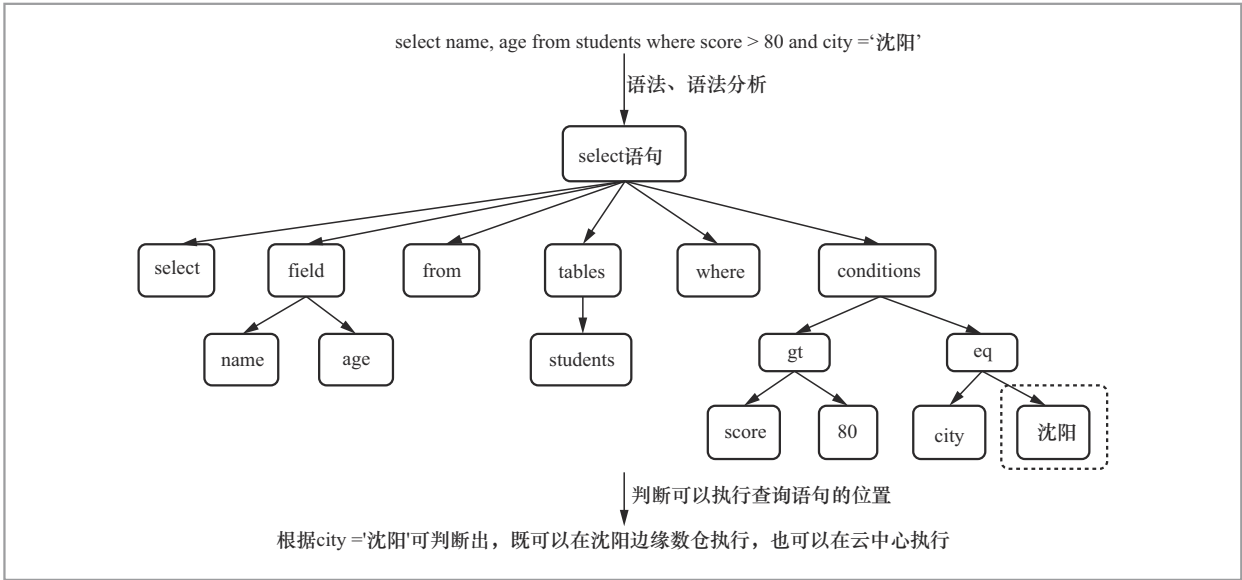


图2 云边多数仓查询语句解析过程

2.2 集成查询代价预测模型的调度框架

面向主题的数据仓库涵盖了计算密集型和输入输出密集型的大数据查询。云中心具有丰富的计算资源，但网络瓶颈和高延迟导致部分SQL在本地边缘数仓执行时间较短。基于云边多数仓架构，查询请求可能会被转发到不同的数据仓库执行，因此需要引入查询代价预测模型，选择最优的数据仓库来执行查询请求。

查询代价预测模型可以预测查询在每个数据仓库上执行的代价，从而选择代价最小的数据仓库执行查询请求。这样可以避免在代价较高的数据仓库上执行查询请求，从而提高查询性能和效率。同时，代价预测模型也可以在数据仓库规模变化时进行动态调整，以适应不同规模的数据仓库，从而实现更好的查询性能。本文基于代价预测模型的调度器架构如图3所示。

调度器接收查询请求后，通过解析SQL语句和查询执行计划获取模型需要的参数，通过代价预测模型预测SQL在云中心数仓和本地数仓执行的代价。调度器综

合考虑SQL能否在边缘执行、边缘执行效率和边缘集群负载情况，从而决定SQL的执行位置，将其发送到目标集群，并获取执行结果。

在查询任务结束后，收集执行日志，解析训练模型需要的输入输出，加入模型训练数据集中，定期以离线方式训练代价预测模型，然后加载新模型以供使用。

3 基于随机森林算法的查询执行代价预测模型

在基于云边多数仓和代价预测模型的查询调度框架中，随机森林算法可以用来构建查询执行代价预测模型。该模型可以根据查询的特征（查询的复杂度、数据规模、网络延迟等）来预测在不同执行环境下的执行代价。这样的预测模型是为了在查询调度框架中做出更加智能的决策，以优化查询的执行效率和资源利用率。随机森林算法可以通过训练来学习查询特征与执行代价之间的关系，从而生成一个预测

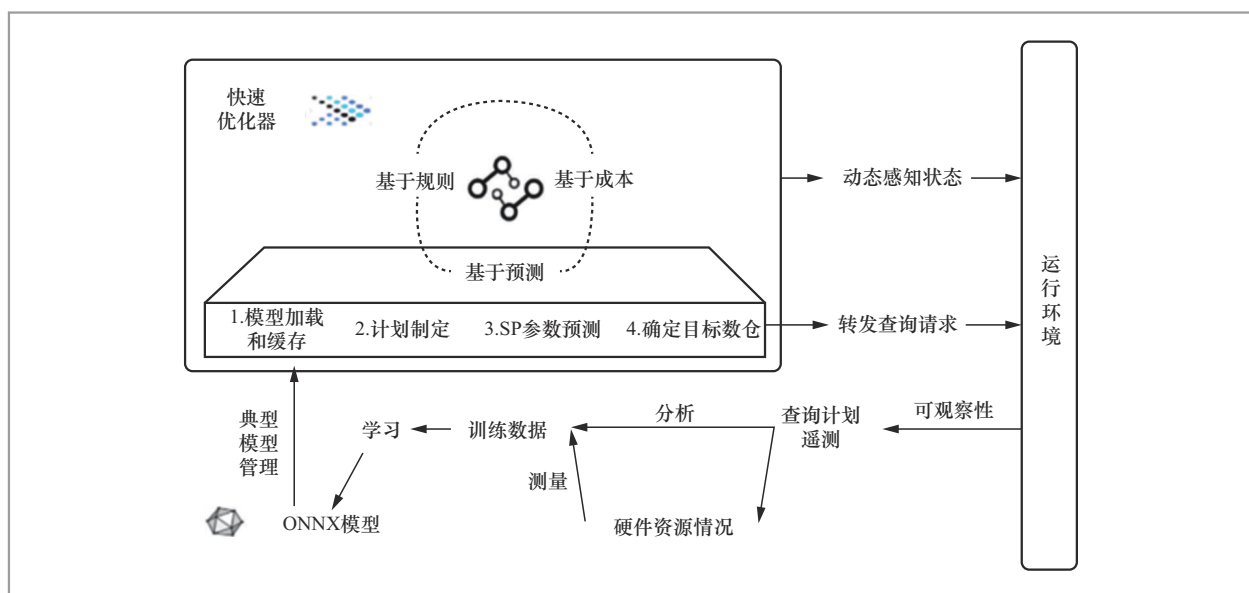


图3 基于代价预测模型的调度器架构

模型。该模型可以根据查询的特征向量,预测在不同环境下执行该查询所需的时间、资源等情况。在查询调度框架中,该预测模型可以帮助调度器选择最佳的执行计划,以最小化整体的查询执行代价。

3.1 查询代价预测模型多特征融合与特征筛选

一个查询语句在数据仓库的执行流程如下:首先,SQL语句被解析优化,生成物理执行计划;然后,以树的形式表示查询计划,其中树上的每个节点都表示数据仓库执行引擎需要执行的一个物理操作,如扫描操作、哈希连接、聚合操作等,物理操作之间的边表示依赖关系;最后,数据仓库会按照查询计划树自下而上地依次执行每个物理算子。

基于云边多数仓的代价预测模型多特征提取主要考虑以下3个方面:查询语句本身特征、查询物理执行计划特征和硬件资源的特征。这些特征被用于构建分类模型的输入变量,用于预测查询语句的代价。特征之间可能存在相关性,需要进行特征工程。这些特征需要在SQL语句执行之前被获取,并且获取代价较小,以满足数据仓库实时查询场景对时间高效性的要求。

这些特征汇集在一起,形成了一个多维的、涉及数据类型复杂的数据集。考虑到单独使用某种特征往往不能完全反映查询语句的代价,并且特征多会提高模型的复杂度和过拟合的风险,本文采用特征融合的方式对多种类型的多个特征进行处理。特征融合方法包括特征组合、特征交叉、特征降维和特征拼接等。特征融合能够减少特征数量,降低过拟合风险,提高模型训练效率,并通过综合考虑查询的多个因素提高查询代价预测的准确性。根据实际情况,本文采取算数平均的方式将作用相

似的特征进行融合,有选择地减少特征数量,提高预测精度。

为了修正基于随机森林的特征重要性排序方法,本文引入随机假特征,将基于随机森林的特征筛选法用来评估特征的重要性,并根据重要性进行特征选择。

算法具体流程如下。

(1) 引入随机假特征

假设有 m 个原始特征,用 $X=(x_1, x_2, \dots, x_m)$ 来表示所有原始特征的集合,其中 x_i 表示第 i 个原始特征。引入随机假特征后,本文得到一个包含 n 个随机假特征的新特征集合 $X'=(x_1, x_2, \dots, x_m, x_{m+1}, \dots, x_{m+n})$,其中 $x_{m+1}, \dots, x_{m+n}$ 是随机生成的假特征。对于每个假特征,它们的取值是在原始特征的值范围内随机生成的。本文用 X'_i 来表示包含前 i 个特征的集合,其中 $1 \leq i \leq m+n$ 。

对于每个原始特征 x_i ($i \leq m$),计算平均特征重要性得分 FI_j 。

(2) 计算特征重要性

对于每个随机森林,在随机抽样的训练数据集上训练一个基学习器,得到一个包含所有特征的特征重要性向量 $\mathbf{FIrf}$ 。本文用 T 表示随机森林中树的数量。

对于每个随机假特征 x_j ($j > m$),本文计算它在随机森林中的平均特征重要性得分 FI_j ,计算式如式(1)所示:

$$FI_j = \frac{1}{T} \sum_{t=1}^T FI_{j,t} \quad (1)$$

其中, $FI_{j,t}$ 表示第 t 棵树中 x_j 的特征重要性得分。

(3) 特征重要性比较

计算每个特征的随机假特征重要性,再根据每个特征的随机假特征重要性对所有特征重新排序,得到新的特征重要性排序,选择排名前 k 个特征作为最终的特征集合。

基于星形架构基准(star schema

benchmark, SSB) 和决策支持系统的基准测试 (transactional processing performance council decision support benchmark, TPC-DS) 数据集的特征重要性得分排序如图4、图5所示。从图4、图5可以看出, SSB数据集最佳特征组合有23个, TPC-DS数据集最佳特征组合有22个。

3.2 代价预测模型实现

在代价预测模型训练阶段, 随机森林是一种引导聚集 (Bootstrap aggregating, Bagging) 算法的优化版本, 使用分类与回归树 (classification and regression trees, CART) 决策树作为基学习器。该方法首先通过自助法 (Bootstrapping) 抽样形成训练集, 然后在每轮生成决策树的过程中, 随机选择一部分特征生成决策树。随机森林中的决策树是相互独立的, 最终的预测结果通过投票方式确定。在代价预测模型参数调优阶段, 本文通过调整随机森林模型的参数, 包括树的数量、树的深度、特征数量、样本数量和随机种子, 达到更好的性能。这些参数通过网格搜索和交叉验证得到最优组合, 具体参数见表1。

3.3 利用 AdaBoost 算法改进基于随机森林的代价预测模型

AdaBoost^[36]是一种常见的迭代算法, 主要用于解决二分类和多分类问题。它可以将多个弱分类器组合成一个强分类器, 提高分类的准确率。在每一轮迭代中, 它会根据前一轮分类器的分类效果调整样本的权重, 使分类器更加关注前一轮分类错误的样本, 然后基于调整后的样本权重训练出一个新的弱分类器。

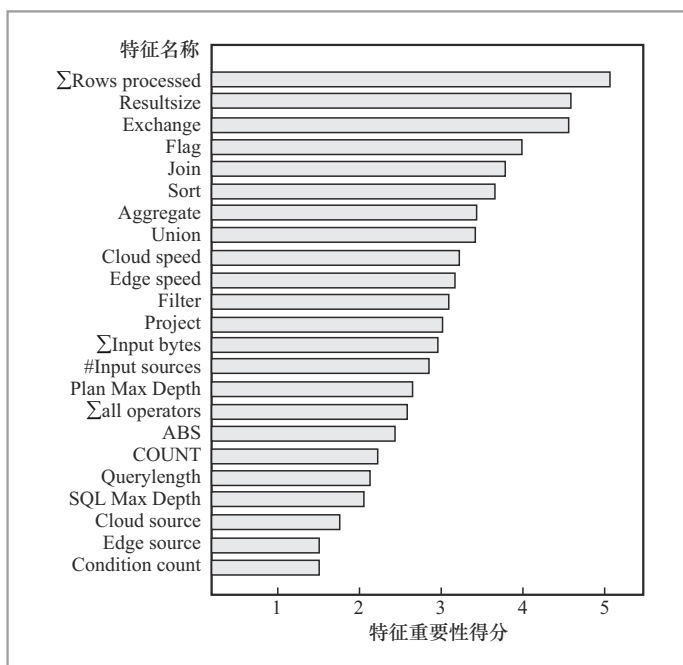


图4 基于SSB数据集的特征重要性得分排序

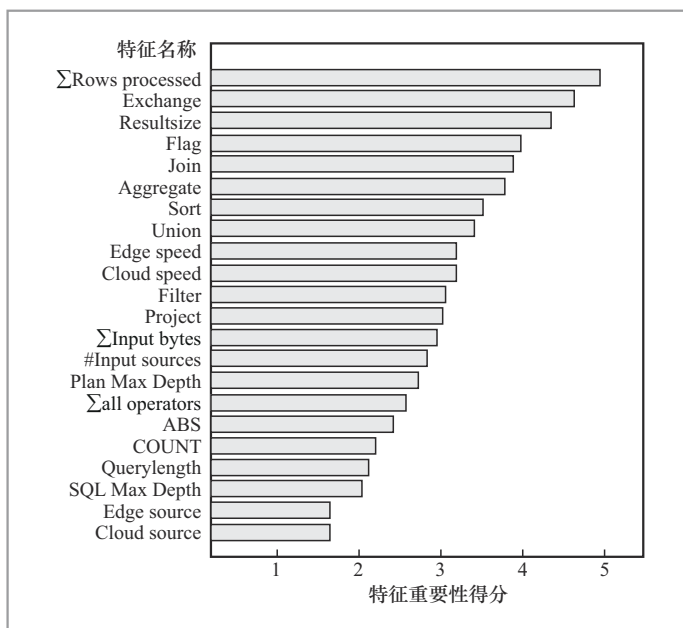


图5 基于TPC-DS数据集的特征重要性得分排序

表1 基于随机森林进行查询代价预测最优参数组合

max_depth	max_features	min_samples_split	min_samples_leaf
20	'sqrt'	2	1

最后,将所有弱分类器组合起来,生成一个强分类器。AdaBoost 算法的原理如图6所示。

本文将随机森林模型与 AdaBoost 算法结合,进一步改进基于随机森林的代价预测模型,也缩短了多条查询语句整体执行的时间。该方法先对训练数据进行特征筛选,然后对多个训练样本进行权重的初始化,随后通过随机森林模型对训练样本进行初步学习,获取每个训练样本的预测值;通过预测值计算获取每个训练样本的相对误差值,进一步通过与设定的误差阈值进行比较,计算获取该弱学习器的预测错误率;基于 AdaBoost 算法的思想,不断更新弱学习器的权重,最终得到一个强预测器。以下是使用 AdaBoost 改进随机森林算法的详细步骤:

(1) 初始化权重 $w_i = \frac{1}{S}$, 其中 S 是样本数量。

(2) 对于 $t=1$ 到 T (T 为决策树的数量):

1) 从样本中有放回地抽取 c 个样本,

并随机选择 p 个特征用于构建决策树;

2) 使用上述样本和特征集构建决策树 h_t ;

3) 计算 h_t 对训练集中每个样本的分类误差率 e_t 。

4) 计算弱分类器 h_t 的权重,如下。

$$\alpha_t = \frac{1}{2} \ln \frac{1-e_t}{e_t} \quad (2)$$

更新样本权重,如式(3)所示。 Z_t 是归一化因子, y_i 是第 i 个样本的真实标签。

$$w_{t+1,i} = \frac{w_{t,i}}{Z_t} \exp(-\alpha_t y_i h_t(x_i)) \quad (3)$$

(3) 最终的分类器如下,如式(4)所示。

$$H(x) = \text{sing}\left(\sum_{t=1}^T \alpha_t H_t(x)\right) \quad (4)$$

在上述算法中, $H_t(x)$ 表示第 t 次迭代后得到的整体分类器, $H(x)$ 是最终的整体分类器,表示将所有弱分类器进行加权组合后得到的结果。样本的权重 w_i 会随着每轮训练的误差分类样本而更新,从而在下

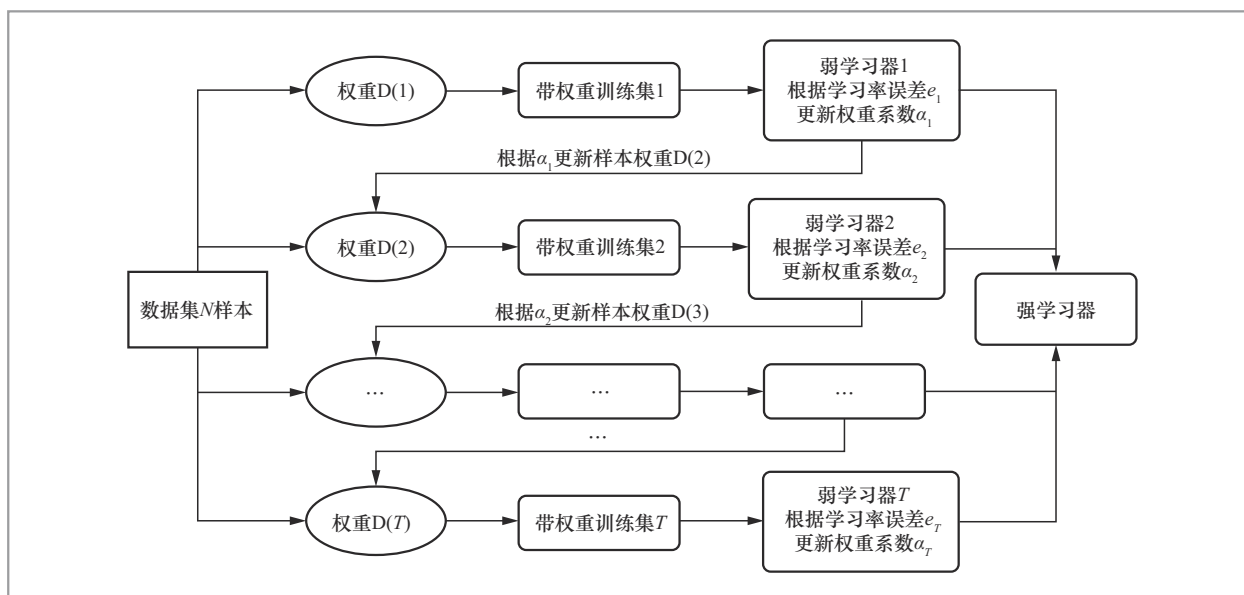


图6 AdaBoost算法的原理

一轮训练中更关注这些样本。而弱分类器的权重 α_t 则取决于其在训练中的准确性，表示其对整体模型的重要性。最终的分类器将所有决策树的输出加权求和，从而得到最终的预测结果。

4 实验和评价

4.1 实验环境与设置

本文的实验环境见表2。

特别地，由于实验条件的限制，本文将云中心数据仓库和边缘数据仓库都部署在阿里云不同地域、相同类型的服务器上，只是搭建数据仓库使用的服务器数量不同。本文搭建云中心数仓1个，部署在成都可用区B，云中心数仓存储全部数据；搭建2个边缘数据仓库，分别部署在北京可用区K和上海可用区B，根据数仓面向主题的特征，边缘数仓只存储涉及边缘查询的数据。同时，本文将客户端部署在和边缘数仓服务器所在的可用区内，以实现客户端和边缘服务器的内网传输。

针对基于云边多数仓架构的查询代价

预测模型，评价指标包括生产者精度 (producer’s accuracy, PA)，总体分类精度和Kappa系数等。

在云边多数仓架构中，研究基于代价预测模型的多数仓架构，通过配置文件设置每个数仓能够同时并行执行的查询语句条数，评价指标包括在相同条件下多条SQL运行的时间。

最后，本文还使用 AdaBoost 算法对基于随机森林的代价预测模型进行改进，并比较了改进后的模型与普通随机森林模型的查询执行位置预测精度、Kappa 系数、预测正确的查询语句数量以及执行时间等指标。

4.2 查询执行代价(位置)预测算法结果

在云边多数仓场景下，本文详细介绍了逻辑回归、K近邻、朴素贝叶斯、支持向量机、人工神经网络和随机森林等机器学习算法，通过网格搜索，利用 GridSearchCV 配置（包含核函数类型、错误项的惩罚系数和核函数系数的参数列表），在参数组合中自动寻找最优模型，最

表2 实验环境

环境及对应参数	云中心对应值	边缘对应值
集群服务器个数	13(12个计算节点)	4(3个计算节点)
数仓个数	1	2
CPU	2.5 GHz主频、3.2 GHz 睿频 Intel®Xeon®Platinum 8269CY	2.5 GHz主频、3.2 GHz 睿频 Intel®Xeon®Platinum 8269CY
内存	16 GB	16 GB
操作系统	centos_7_8_x64_20G_alibase_20211130	centos_7_8_x64_20G_alibase_20211130
带宽	100 Mbit/s	100 Mbit/s
组件	Presto + Hive	Presto + Hive
数据集	SSB、TPC-DS	
机器学习框架	Scikit-Learn	

终得到的最优参数组合及评分见表3。对逻辑回归、K近邻、朴素贝叶斯、支持向量机、人工神经网络和随机森林6种分类算法的精度进行对比，见表4。

由表4可知，随机森林算法在两个数据集上的分类效果均为最佳。对于SSB数据集，随机森林算法的总体分类精度为82.32%，Kappa系数为0.7726；对于TPC-DS数据集，随机森林算法的总体分类精度为81.16%，Kappa系数为0.7433。与逻辑回归算法、K近邻算法、朴素贝叶斯算法、支持向量机算法和人工神经网络算法相比，在SSB数据集上，随机森林算法的分类精度分别提高了6.6%、4.24%、6.01%、4.06%和1.77%，在TPC-DS中，随机森林算法的分类精度分别提高了2.92%、4.37%、4.02%、5.47%和2.14%，该结果可证明随机森林算法在两个数据集的查询执行位置预测具有更高的准确性。

根据实验结果，随机森林算法表现最佳。因为随机森林算法具有处理高维数据和复杂特征交互的优势，它通过构建多棵决策树并对其结果进行综合，减少过拟合的风险，提高预测的准确性。而其他算法（如逻辑回归和支持向量机）可能在处理非线性关系和高维特征空间时不如随机森林算法有效。

深入分析随机森林算法可以揭示哪些特征对预测查询执行代价有较大影响。例如，查询的大小、复杂性、所需数据量、并发查询数等因素可能会显著影响查询的执行代价。识别这些关键特征，可以更好地理解查询代价的主要驱动因素，在未来研究中进行优化。

考察不同算法在不同数据集（如SSB和TPC-DS）上的性能，可以评估它们的稳定性和泛化能力。如果某一个算法在多个数据集上均表现良好，则表明它具有较

表3 支持向量机最优参数组合及得分

数据集	核函数类型	错误项的惩罚系数	核函数的系数	模型得分
SSB数据集	poly	0.9	0.046	0.8674
TPC-DS数据集	poly	0.9	0.009	0.8563

表4 6种分类算法精度对比

数据集	分类算法	Kappa系数	查询执行位置预测的总体分类精度
SSB数据集	逻辑回归	0.7412	77.22%
	K近邻	0.7531	78.97%
	朴素贝叶斯	0.7477	77.65%
	支持向量机	0.7554	79.11%
	人工神经网络	0.7692	80.89%
	随机森林	0.7726	82.32%
TPC-DS数据集	逻辑回归	0.7162	78.86%
	K近邻	0.7098	77.76%
	朴素贝叶斯	0.7101	78.02%
	支持向量机	0.7032	76.95%
	人工神经网络	0.7206	79.46%
	随机森林	0.7433	81.16%

好的泛化能力。此外,分析算法在不同查询类型和不同系统负载条件下的表现,可以进一步验证其稳定性。

除了准确性外,响应时间和执行效率也是评价查询执行代价预测算法的重要指标。算法的预测速度应该足够快,以满足实时查询优化的需求。因此,对算法的响应时间和资源消耗进行分析,对于在生产环境中选择和部署最合适的算法至关重要。

4.3 基于云边多数仓和代价预测模型的查询调度实验结果与分析

对于多个客户端连续发送的多个查询请求,本文分别在纯云中心、云边多数仓架构和基于代价预测模型的云边多数仓架构上执行,具体如图7~图10所示。

从单个实验来看,云边多数仓架构性能好于纯云中心架构,因为边缘数据仓库能够降低数据传输延迟。在整体计算资源相同和尽可能充分并行的情况下,两种架构都能充分利用计算资源,但由于边缘数仓和客户端通过内网通信,部分查询语句在边缘数仓执行的效率高于在云中心执行的效率。基于代价预测模型的云边多数仓架构性能好于云边多数仓架构性能,这是因为代价预测模型能够将每条查询语句都发送到执行效率最高的数仓集群执行,所以整体执行时间更短。

在其他条件相同的条件下,基于TPC-DS数据集执行查询语句整体的时间比基于SSB数据集的更长,这是因为TPC-DS的数据表、表之间的关系,以及查询语句更复杂。这些发现强调了数据本地化在提高数据处理效率中的作用,代价预测模型在系统负载平衡中的重要性,以及云边多数仓架构在处理复杂查询时的巨大潜力,为优化大数据查询处理提供了重要的指导。

4.4 基于AdaBoost算法的改进随机森林模型实验结果与分析

实验表明,基于AdaBoost算法的改进随机森林模型,在基于SSB和TPC-DS数据的多种实验场景下,表现出更高的随机森林模型的总体精度和Kappa系数,具体见表5。对于SSB和TPC-DS数据集,优化后的模型在仅物理执行计划特征、多特

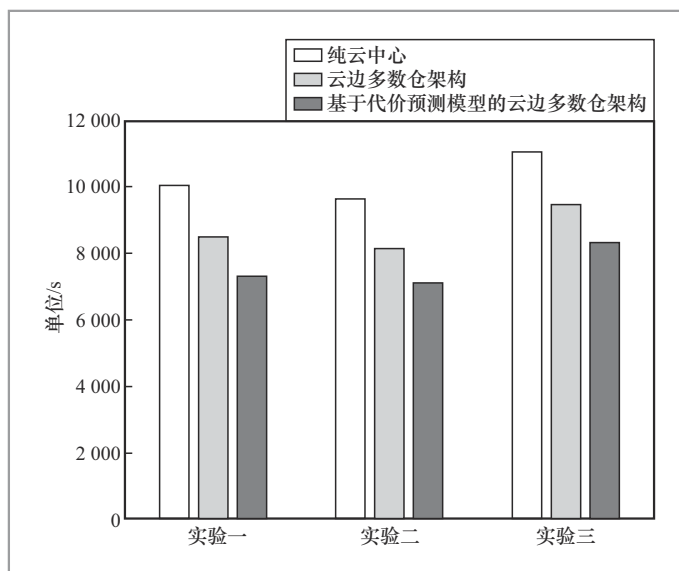


图7 基于30 GB SSB数据的400条查询语句的执行时间

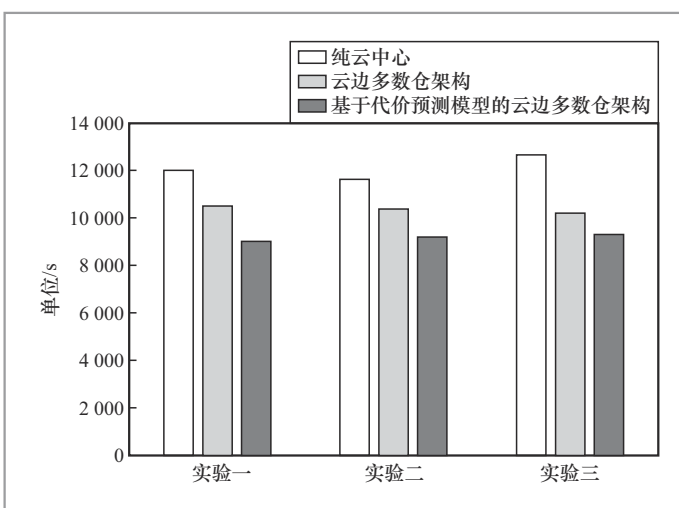


图8 基于30 GB TPC-DS数据的400条查询语句的执行时间

征+特征融合和基于融合后的特征进行特征筛选这3种情况下总体精度均有提升。

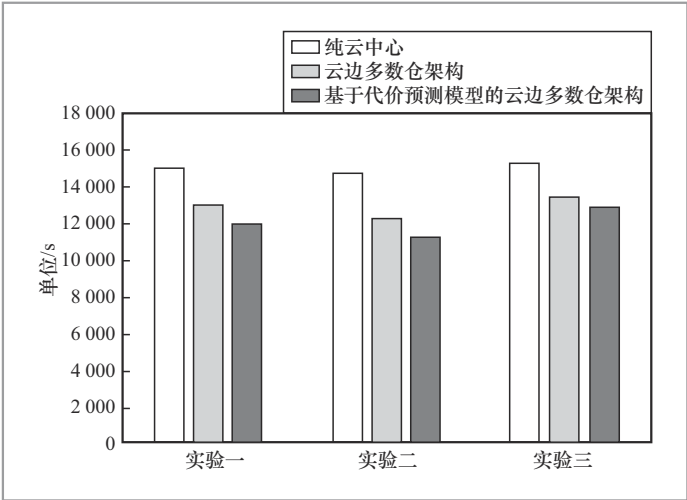


图9 基于50 GB SSB数据的400条查询语句的执行时间

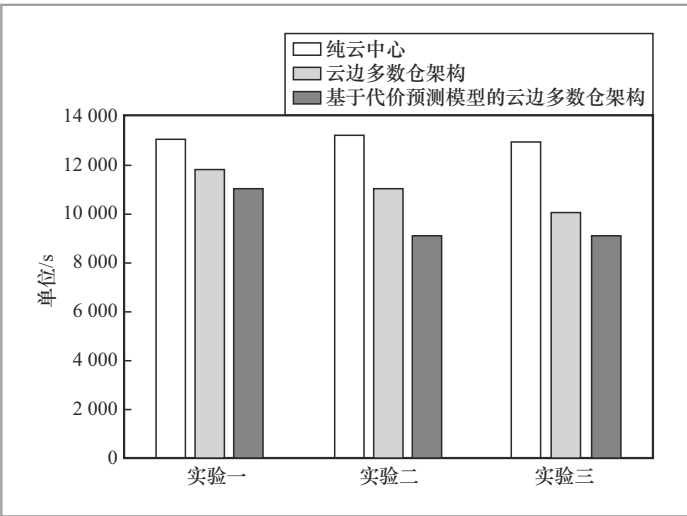


图10 基于50 GB TPC-DS数据的400条查询语句的执行时间

另外，模型精度的提升意味着能够将更多的查询语句准确地发送到执行代价最低的数据仓库执行，减少这些语句执行的时间，进一步减少查询语句整体的执行时间，具体结果如图11、图12所示。

整体来看，优化后的模型预测更加准确，能够节省更多的执行时间。特别地，本文分别进行了基于SSB数据集30 GB数据、TPC-DS数据集30 GB数据、SSB数据集50 GB数据和TPC-DS数据集50 GB数据4种类别的实验，每组实验都是用400条查询语句做3次查询，并取平均值得到最终结果。具体来看，由图11可知，这4次实验分别比原来多正确预测了6、5、7、5条查询语句的执行位置，由图12可知，多正确预测的查询语句分别比原来节省了108 s、125 s、196 s和183 s的执行时间。

深入分析基用AdaBoost算法改进的随机森林模型的实验结果，可以看出结合AdaBoost算法可以有效提高随机森林模型对查询执行代价预测的性能，尤其是准确性AdaBoost算法通过迭代调整分类器对训练样本的权重，重点关注在前一轮迭代中被错误分类的样本，从而在每一轮中提升模型的性能。这种机制使模型能够更加精细地捕捉到数据中的复杂模式和变化，提高模型对新数据的泛化能力。

此外，模型精度的提升表明模型能够更准确地预测查询的执行代价，做出更合理的调度决策。这不仅能够缩短查询语句的整体执行时间，还能提高系统资源的利

表5 优化后模型精度评价

优化后的模型精度评价		仅物理执行计划特征	多特征+特征融合	特征筛选结果
SSB	总体精度	84.83%	91.07%	93.51%
	Kappa 系数	78.63%	86.35%	89.12%
TPC-DS	总体精度	83.11%	89.42%	92.08%
	Kappa 系数	76.95%	84.85%	87.56%

用效率,降低系统的运行成本。

因此,引入AdaBoost算法改进随机森林模型,不仅增强了模型的预测性能,还为优化数据仓库的查询调度策略提供了新的思路。未来的研究可以进一步探索这种方法在更广泛数据环境下的应用潜力,以及如何将其与其他优化技术相结合以达到更好的调度效果。

5 结束语

随着云计算和大数据技术的飞速发展,云边协同架构成为处理海量数据、实现高效计算的关键技术。本文提出的云边多数仓架构和代价预测模型不仅解决了传统数据仓库在伸缩性和效率方面的问题,而且在数据安全、隐私保护和跨区域数据共享等方面表现出显著优势。通过精准地查询代价预测和智能地查询调度策略,本文不仅提高了查询效率和系统性能,还为数据仓库系统在云边计算环境下的应用提供了创新的解决方案。

根据相关研究^[37]所提出的挑战和研究路线,本文的研究成果不局限于现有的数据仓库和云边计算场景,还能扩展到元宇宙这一新兴领域。在元宇宙复杂多样化的应用背景下,如虚拟现实、远程协作、在线教育等场景,本文提出的云边多数仓架构和代价预测模型能够为数据管理、查询优化和资源调度提供更加高效、灵活和安全的技术支撑。

参考文献:

[1] THUSOO A, SARMA J S, JAIN N, et al. Hive - a petabyte scale data warehouse using Hadoop[C]//Proceedings of the

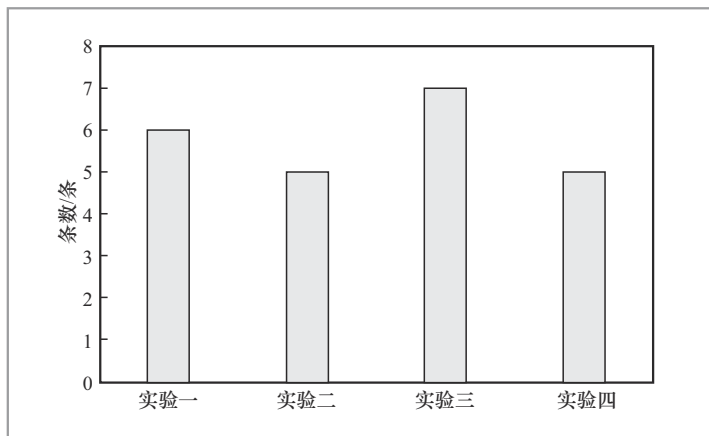


图11 优化后的模型整体比原模型多正确预测的条数

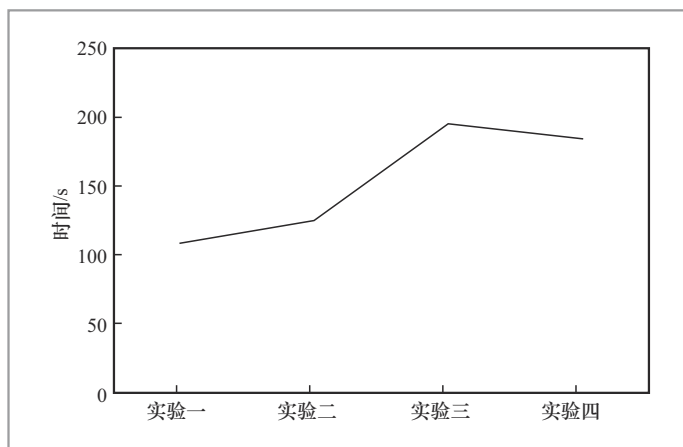


图12 不同类型实验中优化后节省的时间

2010 IEEE 26th International Conference on Data Engineering (ICDE 2010). Piscataway: IEEE Press, 2010: 996-1005.

- [2] VAISMAN A, ZIMÁNYI E. Data warehouse systems: design and Implementation[J]. Data-Centric Systems and Applications, 2014, 24(3): 55-74.
- [3] GOLFARELLI M, RIZZI S. From star schemas to big data: 20 years of data warehouse research[M]//A comprehensive guide through the italian database research over the last 25 years. Cham: Springer, 2017: 93-107.
- [4] PANWAR A, BHATNAGAR V. Data lake

- architecture: a new repository for data engineer[J]. *International Journal of Organizational and Collective Intelligence*, 2020, 10(1): 63–75.
- [5] LI Y, WANG L F, JI L, et al. A data warehouse architecture supporting energy management of intelligent electricity system[C]//*Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. Paris: Atlantis Press, 2013: 696–699.
- [6] SKOURLETOPOULOS G, MAVROMOUSTAKIS C X, MASTORAKIS G, et al. Big data and cloud computing: a survey of the state-of-the-art and research challenges[J]. *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, 2017, 27(13): 23–41.
- [7] WINTER C, GICEVA J, NEUMANN T, et al. On-demand state separation for cloud data warehousing[J]. *Proceedings of the VLDB Endowment*, 2022, 15(11): 2966–2979.
- [8] ATTASENA V, HARBI N, DARMONT J. A novel multi-secret sharing approach for secure data warehousing and on-line analysis processing in the cloud[J]. *International Journal of Data Warehousing and Mining*, 2015, 11(2): 22–43.
- [9] CAVICCHIOLI R, MARTOGLIA R, VERUCCHI M. A novel real-time edge-cloud big data management and analytics framework for smart cities[J]. *Journal of Universal Computer Science*, 2022, 28(1): 3–26.
- [10] SINGH R, KISS T. Edge-cloud synergy: unleashing the potential of parallel processing for big data analytics[C]//*Proceedings of the 2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. Piscataway: IEEE Press, 2022: 1–6.
- [11] AMIN J, SHARIF M, RAZA M, et al. Detection of brain tumor based on features fusion and machine learning[J]. *Journal of Ambient Intelligence and Humanized Computing*, 2024, 15(1): 983–999.
- [12] JORDAN M I, MITCHELL T M. Machine learning: trends, perspectives, and prospects[J]. *Science*, 2015, 349(6245): 255–260.
- [13] REDDY G T, REDDY M P K, LAKSHMANNA K, et al. Analysis of dimensionality reduction techniques on big data[J]. *IEEE Access*, 2020, 8: 54776–54788.
- [14] DRESELER M, BOISSIER M, RABL T, et al. Quantifying TPC-H choke points and their optimizations[J]. *Proceedings of the VLDB Endowment*, 2020, 13(8): 1206–1220.
- [15] KATTENBORN T, LEITLOFF J, SCHIEFER F, et al. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing[J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2021, 173: 24–49.
- [16] BANERJEE I, LING Y, CHEN M C, et al. Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification[J]. *Artificial Intelligence in Medicine*, 2019, 97: 79–88.
- [17] ROY A, JINDAL A, GOMATAM P, et al. SparkCruise: workload optimization in managed spark clusters at Microsoft[J]. *Proceedings of the VLDB Endowment*, 2021, 14(12): 3122–3134.
- [18] BURDAKOV A, PROLETARSKAYA V, PLOUTENKO A, et al. Predicting SQL query execution time with a cost model for spark platform[C]//*Proceedings of*

- the 5th International Conference on Internet of Things, Big Data and Security. Setúbal: SCITEPRESS 2020: 279–287.
- [19] TANG C X, WANG B N, LUO Z X, et al. Forecasting SQL query cost at twitter [C]//Proceedings of the 2021 IEEE International Conference on Cloud Engineering (IC2E). Piscataway: IEEE Press, 2021: 154–160.
- [20] ZHOU X H, SUN J, LI G L, et al. Query performance prediction for concurrent queries using graph embedding[J]. Proceedings of the VLDB Endowment, 2020, 13(9): 1416–1428.
- [21] LI G L, ZHOU X H, CAO L. Machine learning for databases[C]//Proceedings of the First International Conference on AI-ML-Systems. New York: ACM, 2021: 1–2.
- [22] MARCUS R, PAPAEMMANOUIL O. Plan-structured deep neural network models for query performance prediction [J]. Proceedings of the VLDB Endowment, 2019, 12(11): 1733–1746.
- [23] POESS M, RABL T, JACOBSEN H A. Analysis of TPC-DS: the first standard benchmark for SQL-based big data systems[C]//Proceedings of the 2017 Symposium on Cloud Computing. New York: ACM, 2017: 573–585.
- [24] O'NEIL P, O'NEIL E, CHEN X, et al. The star schema benchmark and augmented fact table indexing[C]//Proceedings of Performance Evaluation and Benchmarking: First TPC Technology Conference, TPCTC 2009. Berlin: Springer, 2009: 237–252.
- [25] CAO K Y, LIU Y F, MENG G J, et al. An overview on edge computing research [J]. IEEE Access, 2020, 8: 85714–85728.
- [26] ZHANG S C, LI X L, ZONG M, et al. Learning k for KNN classification[J]. ACM Transactions on Intelligent Systems and Technology, 2017, 8(3): 43.
- [27] WEBB G I. Naïve Bayes[M]//Encyclopedia of Machine Learning. Boston: Springer US, 2011: 713–714.
- [28] HUANG S, CAI N, PACHECO P P. Applications of support vector machine (SVM) learning in cancer genomics[J]. Cancer Genomics & Proteomics, 2018, 15(1): 41–51.
- [29] ZOU J, HAN Y, SO S S. Overview of artificial neural networks[J]. Methods Mol Biol, 2008, 458: 15–23.
- [30] SEN R, ROY A, JINDAL A. Predictive price-performance optimization for serverless query processing[EB]. arXiv preprint, 2021, arXiv: 2112.08572.
- [31] PIMPLEY A, LI S, SEN R, et al. Towards optimal resource allocation for big data analytics[C]//Proceedings of International Conference on Extending Database Technology. Edinburgh: Open Proceedings, 2022: 338–350.
- [32] HU Z Y, LIDS, ZHANG D X, et al. ReLoca: optimize resource allocation for data-parallel jobs using deep learning[C]//Proceedings of the IEEE INFOCOM 2020–IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2020: 1163–1171.
- [33] NAWAB F, SADOOGHI M. Consensus in data management: from distributed commit to blockchain[J]. Foundations and Trends in Databases, 2023, 12(4): 221–364.
- [34] ABEBE M, GLASBERGEN B, DAUDJEE K. MorphoSys[J]. Proceedings of the VLDB Endowment, 2020, 13(13): 3573–3587.
- [35] 王润安, 邹兆年. 基于物理操作级模型的查询执行时间预测方法[J]. 计算机科学, 2022, 49(8): 49–55.
- WANG R A, ZOU Z N. Query perfor-

mance prediction based on physical operation-level models[J]. Computer Science, 2022, 49(8): 49-55.

[36] TU C S, LIU H C, XU B. AdaBoost typical Algorithm and its application re - search[J]. MATEC Web of Conferences, 2017, 139: 00222.

[37] 朱锐, 王宏志, 崔双双, 等. 面向元宇宙的云边端协同大数据管理[J]. 大数据, 2023, 9(1): 63-77.

ZHU R, WANG H Z, CUI S S, et al. Cloud-edge-end collaborative big data management for metaverse[J]. Big Data Research, 2023, 9(1): 63-77.

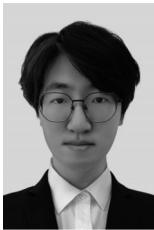
作者简介



高叙宁（2002-），女，东北大学医学与生物信息工程学院在读，主要研究方向为数据库、机器学习。



杨松（1996-），男，就职于沈阳市纪检监察大数据监督保障中心，主要研究方向为数据科学与大数据分析。



李明哲（2003-），男，东北大学医学与生物信息工程学院在读，主要研究方向为机器学习、计算机视觉和人类活动识别等。



张岩峰（1984-），男，东北大学计算机科学与工程学院教授，主要研究方向为分布式系统和大数据处理。

收稿日期: 2024-05-15