基于增量局部加权学习的查询模板自适应基数估计

冯杰明 李战怀 陈 群 陈肇强

(西北工业大学计算机学院 西安 710072)

(西北工业大学大数据存储与管理工业和信息化部重点实验室 西安 710072)

摘 要 基数估计是基于代价查询优化的关键步骤,已经被研究了近 40 年.传统方法如基于直方图的方法在一些假设如属性相互独立、相交的表满足包含原则等成立时能基本满足准确性要求.然而,在真实运行环境中这些假设往往不再成立,可能导致基数估计严重错误进而造成查询延迟.近年来,随着数据的增多和新硬件的发展,使用机器学习方法来提高基数估计的质量成为了可能.由于基于代价的查询优化主要根据查询中子执行计划的估计代价来选择最优的查询执行计划,因此,有一些最近的工作针对一些关键的子执行计划模板建立相应的局部学习模型,取得了不错的进展.但是,这些局部模型主要用于查询(查询空间)分布和数据(数据库数据)分布不变的场景,而在真实运行环境中,它们往往不断地发生变化,限制了这些估计技术的有效性.在本文中,我们针对子执行计划模板在查询分布和数据分布不断变化的环境下提出了一种使用增量的局部加权学习进行自适应基数估计的方法.具体地说,首先抽取子执行计划的语义和统计特征使之能代表当前查询和数据的特性,然后使用增量的局部加权学习模型根据查询分布和数据分布的变化进行自适应的学习,实现基数估计.最后,通过对比实验验证了本文方法的有效性.

关键词 基数估计;查询优化;执行计划;自适应学习;增量学习;局部加权学习

中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2022.00017

Incremental Locally Weighted Learning for Adaptive Cardinality Estimation of Query Template

FENG Jie-Ming LI Zhan-Huai CHEN Qun CHEN Zhao-Qiang

(*School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an* 710072)

(*Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology, Xi'an* 710072)

Abstract Cardinality estimation is crucial for cost-based query optimization and has been studied for nearly forty years. The traditional approach based on histograms can achieve desired accuracy when some assumptions, e.g. independence of attributes and the principle of inclusion between join tables, are valid. However, these assumptions usually become invalid in the real running environment. It may lead to serious estimation errors and thus increase query latency. In recently years, with the increase of data and the development of new hardware, it is possible to use machine learning to improve the accuracy of cardinality estimation. Because cost-based query optimization chooses the optimal execution plan according to cost estimation of sub execution plan in a

收稿日期:2020-03-19;在线发布日期:2020-12-07.本课题得到国家重点研发计划(2018YFB1003400)、国家自然科学基金重点项目(61732014)、国家自然科学基金面上项目(61672432)、中央高校基本科研业务费专项资金资助(3102019DX1004)、陕西省自然科学基础研究计划(2018JM6086)资助.**冯杰明**,博士研究生,主要研究领域为数据库查询优化,E-mail:fengjm@mail.nwpu.edu.cn.**李战怀**(**通信作者**),博士,教授,博士生导师,主要研究领域为数据库理论与技术,E-mail:lizhh@nwpu.edu.cn.**陈 群**,博士,教授,博士生导师,主要研究领域为风险分析、渐进机器学习等人工智能技术.**陈肇强**,博士研究生,主要研究领域为数据质量管理、人工智能.

given query. Therefore, there are some recent works which studied how to build local learning models for sub execution plan templates. However, these models are supposed to be used in the scenarios where query (query space) distribution and data (database data) distribution remain static. However, in real scenarios, where query and data distributions are usually continuously shifting, they may have limited efficacy. To address the said shortcoming, in this paper we propose an approach based on incremental locally weighted learning to perform adaptive cardinality estimation. Specifically, it first extracts both semantic and statistic features of a sub execution plan, which represent the current characteristics of both query and data. The reasonableness behind that is, the semantic features solely represent variables of the query template and are independent with the database. In some cases, we even cannot collect semantic features, e. g. there is no variable for two tables joining without predicate. Therefore, other features are needed. On the other hand, the statistic features based on current data in the database, e. g. the statistics of attributes in a table, are very effective for predicting the cardinality. However, they are not constantly reliable. Combining with the semantic features can relieve this uncertainty. Based on collected features, the proposed approach uses Receptive Field Weighted Regression(RFWR) to adaptively perform cardinality estimation. Generally, the RFWR works well. However, there have two challenges for RFWR to do cardinality estimation. The first one is, with the increase of query feature vector's dimension, RFWR becomes inefficient and inaccuracy. To tackle this challenge, we use Locally weighted projection regression(LWPR) instead of RFWR. LWPR is based on RFWR and uses the incremental partial least square(PLS) to improve the efficiency. The second is, using these two methods to predict the cardinality will be inaccurate when the training data is not enough. To improve the accuracy in this situation, we propose to combine k-nearest neighbor(KNN) to do the prediction. In addition, to further improve the efficiency and accuracy in dynamic environments, the proposed approach respectively uses local model optimization of RFWR to rectify the model parameters when there are only a few training data or the dimension of query feature vector is higher, and prunes the receptive fields when the number of receptive fields becomes larger or the query distribution and data distribution shifts. Our empirical comparative study has verified the effectiveness of the proposed approach.

Keywords cardinality estimation; query optimization; execution plan; adaptive learning; incremental learning; locally weighted learning.

1 引 言

基数估计是数据库领域中一个很老却仍很具挑战的问题.传统方法如基于直方图的方法在一些假设成立的情况下就能基本满足准确性要求.但是,数据库在真实运行环境中存储的数据巨大,同属性下数据偏斜严重,数据的不同属性之间相互关联性高,相交的表之间也经常不满足包含原则.在这些情况下,采用基于直方图的方法进行基数估计经常会非常不准确.特别当一个查询涉及多表的时候,估计错误将会成倍增长,最终导致数据库选择一个很差的执行计划[1].

近年来,随着深度学习技术的迅速发展,基于机器学习的基数估计成为一个研究热点[2].基于机器学习的基数估计方法主要分为两类:数据驱动方法[3-9]和查询驱动方法[10-25].数据驱动方法通过使用数据库数据来构建单表或中间表属性的联合密度分布,以此来估计不同查询的基数.查询驱动方法主要利用过去的查询历史以及对应的真实基数构建监督学习模型.查询驱动的大部分工作[10-15]只针对单表范围查询建立学习模型,而极大地限制了其应用范围.也有工作[16-17]构建一个复杂的神经网络用来预测所有查询的基数.然而,全局模型不但没有解释性,而且要达到预测的准确率,需要大量的训练数据.近年来,有一些工作[18-22]使用一些小的局部模型

代替一个大的全局模型,取得了不错的进展.局部模型主要用于学习和预测某个特定查询类型的基数,如特定的查询模板.由于基于代价的查询优化主要根据查询中子执行计划的估计代价来选择最优的查询执行计划,因此 CARDLEARNER[19] 和 AQO[20] 根据当前查询负载针对一些关键的子执行计划模板构建学习模型.我们注意到,通过这些小的局部模型,不仅能在保持准确性的前提下减少了所需的训练数据,也增加了模型的可解释性.但是,这些局部模型主要用于查询分布和数据分布不变的场景,而在真实运行环境中,它们往往不断地发生变化,限制了这些估计技术的有效性.

对于给定的子执行计划模板,若采用离线的方式构建深度学习模型,要使模型对多个查询的预测较为准确,则需要整个查询空间的训练数据.一方面,构建整个查询空间的模型,既需要大量的训练数据,也增加了存储的代价;另一方面,当查询分布和数据分布发生变化时,模型通常需要重新进行训练,代价巨大.我们认为在大数据量和动态负载环境下,不需要建立整个知识,而是要根据用户感兴趣的知识和知识本身变化进行自适应的调整和积累.因此,本文致力于研究在查询分布和数据分布不断变化的环境下,如何通过在线模型的自适应调整和更新,提高子执行计划基数估计的准确性.在查询分布和数据分布不断变化的环境下,使用在线模型学习和预测基数主要有如下两个挑战:第一,如何在冷启动或少量训练数据下,模型具有较高的准确性.第二,如何探测查询分布和数据分布的变化,且当其变化时,模型如何快速调整自身以适应新的环境.

针对上述挑战,本文使用增量的局部加权学习方法,感受野加权回归[26](Receptive Field Weighted Regression,简称 RFWR)进行在线学习.针对该模型,我们提出一种关于子执行计划的特征化方法,使得其可以根据查询分布和数据分布的变化自适应地更新、增加和剪枝其自身局部模型,使之快速调整以适应新的环境.但是,当查询中谓词或相交的表较多时,子执行计划的特征向量维数急剧增长,因此 RFWR 训练过程变的低效,预测的准确性也降低.而局部加权投影回归[27](Locally weighted projection regression,简称 LWPR)在 RFWR 的基础上,使用增量的偏最小二乘降维方法提高 RFWR 模型的鲁棒性和高效性.因此,针对高维查询,本文采用 LWPR 进行学习.而直接使用这两个模型进行查询预测时,结果等于加权所有其局部模型的预测值,当局部模型数量较多且训练数据不足时,预测既不准确也不高效.因此,本文采用核距离的阈值控制方式使当前的查询只涉及一些相关的局部模型以提高效率,并利用最近邻算法(KNN)在局部模型不可靠时(训练数据方差大或数目少)进行预测以提高准确性.

总结本文的主要贡献为:

(1)我们提出一种利用 RFWR 来进行自适应基数估计的方法.另外,针对高维查询,预测时不准确不高效的问题,分别提出使用 LWPR 和 KNN 等进行处理,使之适用于真实的数据库场景.

(2)针对上述模型,提出一种关于子执行计划的特征化方法.该特征化方法既可以捕捉查询本身的语义特征,又可以捕捉数据库自身的内部特征,因此可以有效反映当前查询和数据的特性;

(3)通过实验验证了所提出的方法在动态环境下的有效性.跟现有的其它算法相比,本文方法获得了更高的基数估计准确性.

本文其余的内容由以下部分组成:第 2 节回顾相关工作;第 3 节介绍预备知识和问题描述;第 4 节介绍基于增量局部加权学习模型的自适应学习和预测过程;第 5 节进行实验验证;第 6 节总结全文并讨论未来的研究.

2　相关工作

2.1　基数估计

有大量的工作致力于基数估计的研究,这些方法根据构建模型的数据来源可以分为数据驱动方法和查询驱动方法,查询驱动方法又可分为查询驱动直方图和机器学习方法.

数据驱动方法主要分为直方图[28-30]、采样[31-34]和机器学习[3-9]三类.直方图的主要思想是通过扫描大量的数据将数据分布信息根据一定的规则划分到互不相交的桶中,而每个桶的分布通过桶里元素及其相应频率进行刻画.虽然直方图实现较为简单,但是很难确定在哪些属性上建立多维直方图更为高效;采样通过从原数据中抽取部分数据来估测多属性查询和部分 join 查询的基数[31-32].在最近的研究中,也有将采样的方法和其它统计方法[33-34]混用来提高基数估计的准确性.采样方法相对直方图比较准确,但采样代价很昂贵;基于机器学习的方法[3-4,6-7]主要思想类似于直方图,使用概率图、核密度预测、自回归等不同的密度估测模型利用数据构建属性的联

合密度函数,这些方法主要用于单表多属性查询的基数预测.近几年,一些工作[5,8-9]将这些密度估测模型进行扩展用于较为复杂的 join 查询预测,但模型构建过程很复杂且随着数据库数据量的增多而变得低效.数据驱动的方法一般使用数据库数据构建离线模型,而本文使用查询的历史记录构建在线回归模型,使得数据库可以根据负载自适应学习基数知识.另外,数据驱动方法的预测结果可用于本文模型的输入,如本文使用的统计特征(详见 4.1 节),它既可以提升本文模型的准确性也能提升鲁棒性.

查询驱动直方图方法[35-37]根据查询负载的执行历史自适应地建立桶,并估算各个桶的密度分布.最大熵直方图[38-40]在查询驱动直方图的基础上,提出在保证所构建的直方图和过去查询一致的条件下,寻求分布的熵最大化.查询驱动直方图相对于扫描数据直方图代价较小,但是,它会随着查询历史数目和维数的增大而变得低效.本文的方法也是基于查询的历史构建学习模型,但和查询驱动直方图不同,本文通过构建在线监督学习模型累计知识,因此不会随着查询历史个数和谓词个数的增多而低效,并且适用的查询范围更加广泛.

查询驱动的机器学习方法主要思想是利用历史查询和相应的真实基数来构建监督回归模型[23],近期也有工作[24]探索利用两个查询之间的包含比例用于提升基数估计模型的准确性,还有方法[25]在构建基数模型时考虑了其对查询执行性能的影响.查询驱动的学习方法就单个模型所针对的查询类型来说,大体可以分为三类:单表查询[10-15]、某一类查询[18-22]以及整个查询负载[16-17].对于单表查询的方法,因为不能估测 join 查询而限制了它的应用范围.与之相反,为整个查询负载建立一个模型进行处理,要想估测的准确不但需要大量的高质量训练数据,而且模型非常复杂,可解释性也受到极大的限制.LEO[18]针对单个操作建立原基数估计值与真实基数的线性模型,用于修正原基数估计的错误.文献[21]在数据不可见的环境下构建单个查询模板的学习模型.CARDLEARNER[19]在云环境下为一些经常重复出现的关键子执行计划模板建立学习模型,采用离线的方式进行训练,当负载或数据发生变化时,模型需要重新训练.AQO[20]针对单个查询模板建立一些子执行计划的学习模型,每个模型采用在线 KNN 算法进行学习.但是,因为它对子执行计划进行特征化时只使用了不可靠的数据库内部特征,加之使用的在线 KNN 算法不能根据环境变化快速调整,因此其主要适用于查询和数据分布不变的场景.文献[10-12]和本文研究最为接近,也针对查询或数据分布不断变化的环境进行自适应的基数估计,但主要用于单表的范围查询.本文在以上研究的基础上,针对子执行计划模板在查询和数据分布不断变化的环境下进行了深入研究.

2.2 查询优化

基于代价的查询优化在生成执行计划过程中主要依赖于不可靠的基数估计和代价模型,加之它的搜索空间巨大而使用一些启发式的方法进行搜索,由此最终所生成的查询执行计划可能会严重偏离最优的查询执行计划.针对上述问题,近几年一些工作使用强化学习生成查询执行计划,包括如何表示数据库的状态[41]、生成表的连接顺序[42-44]、生成完整查询计划[45-46]、引导执行计划的选择[47]等.虽然有一定的进展,但该项研究还有很多问题有待解决,如怎么初始化状态,如何设计既准确又轻量化的值函数等.除此之外,查询优化也有其它几个方向较为热门,包括自适应查询处理[48]、执行时再优化[49]、健壮查询处理[50]等.

2.3 在线学习

本文使用在线学习来提高基数的估计准确性,当一个查询执行结束,我们可以获取它的真实基数,然后将该查询和相应的基数用于模型的学习,因此,它是一种监督在线学习[51].在真实的数据库应用场景,查询负载和数据往往不断地发生变化,采用离线学习往往代价巨大且不准确,而在线学习提供了自适应的可能性[52].本文使用增量的局部加权回归模型[26-27],通过各个简单局部模型的组合能够表达任意复杂的函数,而且在分布发生变化时,能通过更新、增加和剪枝其局部模型使之快速适应新的环境.

3 预备知识及问题定义

本节先介绍一个相关的预备知识:查询的基数和选择率,然后对本文中所提及的子执行计划模板进行说明,最后定义本文所要解决的问题.

3.1 查询的基数和选择率

一个数据库由多个关系 R 组成,每个 R 由不同的属性组成.不妨设 $R=(A_1,A_2\cdots,A_k)$,R 的元组个数为 $|R|$.设每一个属性 A_i 的值域为 D_i,那么 R 的值域就可表示为 $D(R)=D_1\times D_2\times\cdots\times D_k$.

对于单表范围查询 Q,设它在表 R 上的选择谓词为 P,那么 P 就为所有属性选择条件的合取式,如:$P=(25<A_1<50)\wedge\cdots\wedge(10<A_k<70)$,显然 P 的值域 $D(P)$ 应该满足 $D(P)\subseteq D(R)$. 那么定义查询 Q 的基数为 $C(Q)=|\sigma_P(R)|$,其中 $\sigma_P(R)$ 为 R 上满足选择谓词 P 的元组集合. 相应的 Q 的选择率定义为 $S(Q)=\dfrac{C(Q)}{|R|}$. 对于涉及多表的查询,基数和选择率的定义方式类似,这里省去.

3.2　子执行计划模板

查询的执行计划指它在执行时的单表扫描方式、两表连接算法、多表连接顺序等,其通常可以表示为一棵二叉树,而子执行计划为该二叉树的一个子树,如图 1(左)所示. 当预测一个子执行计划的基数时,需要知道子执行计划涉及哪些基表、每个表中属性的选择谓词和表之间的 join 谓词,而不依赖于它的单表扫描方式、两表连接算法、多表连接顺序. 因此,本文把具有相同基表、除常数不同之外具有相同选择谓词和相同 join 谓词的所有子执行计划归为一个子执行计划模板.

在查询优化过程中,当估测各个子执行计划的基数时,首先需要判断它相应的子执行计划模板模型是否存在,如果存在则使用该模型进行预测,否则使用原基数估计进行预测. 对于一个查询负载,需要建立哪些子执行计划模板模型是另一个重要的问题[19-20],本文主要关注每个模板模型的建立.

3.3　问题定义

本文研究的场景如下:对于一个子执行计划模板 Q,简称查询模板,在查询空间 $p(q)$ 和数据库数据 B 变化的环境下,查询以流的形式进行执行,执行结束后收集并学习历史记录 (q_i,y_i),其中 $q_i\in Q$ 表示第 i 个查询,y_i 为其相应基数的自然对数,后文简称基数. 本文研究如何让数据库在线学习历史记录以提高其对未来查询基数估计的准确性.

本文解决的问题定义如下:对于查询模板 Q,给定来自未知联合分布 $p(q,y)$ 的历史查询流 (q_i,y_i),$i=1,2,\cdots,n,\cdots$,如何训练一个在线回归模型 M 表示查询 q 和基数 y 的映射关系,以及如何通过 M 预测当前查询的基数 $\hat{y}$,使预测基数 $\hat{y}$ 接近于真实基数 y.

4　基数的学习和预测

本节首先介绍如何对查询进行特征化处理,使得所提取的特征能够体现当前查询和数据的特性;接着详细地介绍增量局部加权学习模型 RFWR 的定义、训练过程、预测过程和算法复杂度分析;最后提出模型的一些优化方法.

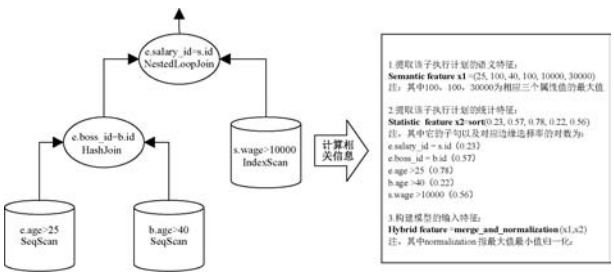


图 1　子执行计划特征提取

4.1　特征提取

对于查询 select * from employee e, boss b, salary s where e.boss_id = b.id and e.salary_id = s.id and e.age>25 and b.age>40 and s.wage > 10000,一个可能的子执行计划形状类似为一棵二叉树,如图 1 所示. 树上包含三类节点,第一类为根节点,为需要预测的节点;第二类为内节点,是两个表或中间表相交之后的中间表;第三类是叶子节点,表示该子计划所涉及的表. 对于给定的子执行计划,可以根据它涉及的基表和谓词子句匹配其学习模型[20],然后根据其子句来推断其基数. 因为各个子句除常数部分外,其它均相同,故本文抽取两种特征来唯一地表示子句:查询语义特征和统计特征. 其中,查询语义特征表示查询本身的含义,本文使用选择谓词的查询范围. 该特征不依赖于数据库,不仅不能反映数据库内部的特性,而且当查询无参数时(如只有 join 谓词),将收集不到该特征,因此还需要别的特征来表示. 借鉴 LEO[18]、CARDLEARNER[19] 和 AQO[20] 等以往工作提取计划节点或子计划树特征的思想,我们使用各个子句的边缘选择率来唯一的表示它们. 该边缘选择率可通过传统基于直方图的规则方法进行估算,虽然该统计值可能不准确,但是对于给定的谓词,由于估算它的规则固定,故其估测值在数据库内部是保持不变的(除非数据发生变化). 另外,由于该统计值的不准确性,不同谓词子句(如常数不同)具有相同(或相近)的边缘选择率可能会导致模型的不可靠,而查询语义特征能唯一的表示当前查询,故可以降低这种不可靠性. 因此,我们融合这两种特征,并分别对每个特征进行相应的最大值最小值归一化处理,实验 5.2 显示了本文所提取特征的有效性. 下面具体讨论这两种特征:

查询语义特征:由于本文针对查询模板构建模型,故不需要为查询中每个关系的每个属性谓词、表之间的相交关系进行特征化,只需要提取查询中出现的选择谓词.一个选择谓词包括三个部分:属性、操作符、语义值.其中,语义值是需要抽取的对象,而语义值的类型随着操作符的不同而不同,因此该问题具有一定的挑战性.在本文中,对于连续数值的属性,将操作符 $=,\neq,<,\leqslant,>,\geqslant,[l_i,r_i]$ 均转化为 $a<(\leqslant a),>b(\geqslant b)$ 的形式;对于分类属性,采用 1-hot 的编码方式.对于图 1 的子执行计划树,它的语义特征为 Semantic_feature.

统计特征:本文使用基于直方图的规则方法计算各个子句的边缘选择率,并取其对数作为统计特征.子句的边缘选择率指只考虑该子句约束条件下的选择率.比如,对于 PostgreSQL,它主要基于一维直方图的方法计算各个子句的选择率.这类特征反映了数据库内部的状态信息,从而不仅使模型具有更好的鲁棒性和准确性,而且由于计算边缘信息时使用了随数据变化更新较快的统计信息如一维直方图信息,从而有助于处理数据变化的情况.对于图 1 的计划树,它的统计特征为 Statistic_feature.

4.2 模型定义、训练和基数预测

对于一个子执行计划,设它相应的特征向量为 $\boldsymbol{x}=(x_1,x_1,\cdots,x_n)$,其对应的基数为 y,在本文假设 $\boldsymbol{x}$ 和 y 满足如下关系:

$$y=f(\boldsymbol{x})+\varepsilon,E(\varepsilon)=0,$$

其中,f 代表 $\boldsymbol{x}$ 关于 y 的回归方程,ε 为噪音误差.

局部加权学习(LWL)[53]的主要思想是对预测点附近某个邻域内的每个样本点进行加权,并利用加权的最小二乘法得到预测点的估计值.感受野加权回归方法(RFWR)[26]是一种增量的局部加权学习方法,它不需要存储历史数据,而只需保留一些关键的统计信息进行增量训练.其核心思想是随着训练数据不断的到来,根据数据的空间位置,动态将其分成相互可重叠的数据区域,且每个局部区域(感受野)使用一个加权的线性函数来表示.

在 RFWR 中,局部模型的感受野通过一个高斯核函数[26]来限定:

$$w_k=\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{c}_k)^{\mathrm{T}}\boldsymbol{D}_k(\boldsymbol{x}-\boldsymbol{c}_k)\right),$$
$$\boldsymbol{D}_k=\boldsymbol{M}_k{}^{\mathrm{T}}\boldsymbol{M}_k \tag{1}$$

其中,$\boldsymbol{c}_k$ 为局部模型的中心点,$\boldsymbol{D}_k$ 为一个正定矩阵,决定该局部模型感受野的大小和形状.为了计算方便,通常通过 cholesky 分解将 $\boldsymbol{D}_k$ 分解为上三角矩阵 $\boldsymbol{M}_k$ 和它转置的乘积.其对应的局部模型使用线性模型[26]可表示为

$$y_k=(\boldsymbol{x}-\boldsymbol{c}_k)^{\mathrm{T}}\boldsymbol{b}_k+b_{0k}=\widetilde{\boldsymbol{x}}\boldsymbol{\beta}_k,$$
$$\widetilde{\boldsymbol{x}}=((\boldsymbol{x}-\boldsymbol{c}_k)^{\mathrm{T}},1)^{\mathrm{T}} \tag{2}$$

当一个新查询 $\boldsymbol{x}$ 来临时,预测值 $\hat{y}$ 就等于所有局部线性模型预测值的归一化加权之和[26],即

$$\hat{y}=\frac{\sum_{k=1}^{K}w_k\hat{y}_k}{\sum_{k=1}^{K}w_k} \tag{3}$$

其中,K 为局部模型的总数,$\hat{y}_k$ 为第 k 个模型的预测值,w_k 为查询 $\boldsymbol{x}$ 相对于第 k 个模型的权重.计算直观图如图 2 所示.

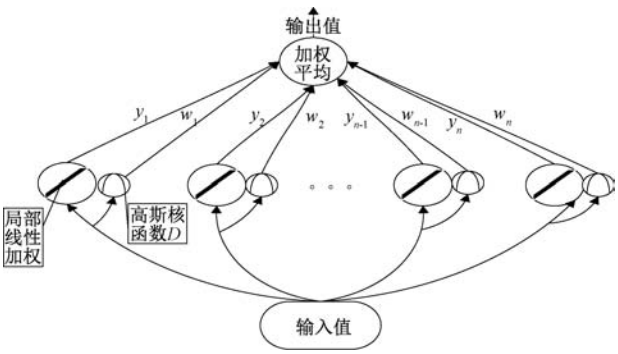


图 2 感受野加权回归

该模型能够根据查询分布和数据分布的变化自适应的增加局部模型.具体地说,当新来的查询 $\boldsymbol{x}$ 与每一个局部模型的中心点的核距离 w 小于某个预先给定阈值 w_{gen} 时,新建一个局部模型,并取该局部模型的中心点为 $\boldsymbol{c}=\boldsymbol{x}$.对于每一个局部模型,在训练的过程中,$\boldsymbol{\beta}$ 通过增量的方法不断的更新,以提高该区域的预测准确率.于此同时,$\boldsymbol{M}$ 也进行自适应的调整,不断的逼近最优的感受野.

4.2.1 学习线性模型

在介绍增量的训练算法之前,我们先说明批量的局部加权线性回归模型中如何计算回归系数 $\boldsymbol{\beta}$.

假设收集了该局部模型的 p 个历史查询数据,表示为矩阵形式 $\boldsymbol{X}=(\widetilde{\boldsymbol{x}}_1,\widetilde{\boldsymbol{x}}_2,\cdots,\widetilde{\boldsymbol{x}}_p)^{\mathrm{T}}$,对应的输出表示为向量 $\boldsymbol{Y}=(y_1,y_2,\cdots,y_p)^{\mathrm{T}}$ 以及相应的权重表示为矩阵 $\boldsymbol{W}=diag(w_1,w_2,\cdots,w_p)^{\mathrm{T}}$.那么,根据局部加权回归可得[53]:

$$\boldsymbol{\beta}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{Y}=\boldsymbol{P}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{Y} \tag{4}$$

而在 RFWR 中,当新建一个局部模型时,则初始化矩阵 $\boldsymbol{P}$ 为一个对角矩阵,然后随着训练数据的到来,通过递归的最小二乘法(RLS)更新相关参数.具体地说,对于一个新的训练数据 $(\boldsymbol{x},y)$,回归系数

的迭代如下[26]:

$$\boldsymbol{\beta}^{n+1}=\boldsymbol{\beta}^{n}+w\boldsymbol{P}^{n+1}\widetilde{\boldsymbol{x}}e_{cv}^{\mathrm{T}},$$

$$\text{where}\quad \boldsymbol{P}^{n+1}=\frac{1}{\lambda}\left(\boldsymbol{P}^{n}-\frac{\boldsymbol{P}^{n}\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{P}^{n}}{\frac{\lambda}{w}+\widetilde{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{P}^{n}\widetilde{\boldsymbol{x}}}\right)\quad\text{and}$$

$$e_{cv}=(y-\boldsymbol{\beta}^{n\mathrm{T}}\widetilde{\boldsymbol{x}}) \tag{5}$$

其中,λ 表示遗忘因子,可以对之前的学到的知识进行适当遗忘以减少初始值的干扰.

4.2.2　学习感受野的形状和大小

训练过程另外一个重要的部分是根据查询的预测值和真实值的误差自适应地调节局部线性模型的感受野.这个调整是通过调节矩阵 $\boldsymbol{M}$ 来实现.同样的,假设收集了 p 个批量历史查询数据,局部模型的交叉验证损失函数定义如下[26]:

$$J=\frac{1}{W}\sum_{i=1}^{p}\frac{w_i\|y_i-\hat{y}_i\|^2}{(1-w_i\widetilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{P}\widetilde{\boldsymbol{x}}_i)^2}+\gamma\sum_{i,j=1}^{n}D_{ij}^2 \tag{6}$$

其中,$W=\sum_{i=1}^{p}w_i$,γ 表示正则项的系数.而在 RFWR 中,当新建局部模型时,则初始化矩阵 $\boldsymbol{M}$,然后基于以上损失函数,通过梯度下降的方法增量式地对矩阵 $\boldsymbol{M}$ 进行学习.具体地说,对于新的训练数据 $(\boldsymbol{x},y)$,计算 J 关于该节点的梯度 $\frac{\partial J}{\partial \boldsymbol{M}}$,$\boldsymbol{M}$ 根据学习率 α 更新如下[26]:

$$\boldsymbol{M}^{n+1}=\boldsymbol{M}^{n}-\alpha\frac{\partial J}{\partial \boldsymbol{M}} \tag{7}$$

4.2.3　训练算法

总之,对于每一个局部模型,RFWR 通过快速的牛顿方法调整局部线性系数 $\boldsymbol{\beta}$,使用梯度下降方法优化感受野参数 $\boldsymbol{M}$,使模型自适应地快速调整.RFWR 也可以根据查询分布和数据分布的变化情况动态地增加局部模型.完整的训练算法见算法 1.

算法 1.　感受野加权回归(RFWR)训练算法.

输入:新来的查询$(\boldsymbol{x},y)$,模型 RFWR,学习阈值 l,学习率 α,遗忘因子 λ,初始值 $\boldsymbol{M}_{def}$,w_{gen} 等

输出:更新后的模型 RFWR

1. For k in K:
2.　根据公式(1)计算查询的权重 w_k
3.　IF $w_k>l$
4.　　根据公式(5)更新回归系数 $\boldsymbol{\beta}$
5.　　根据公式(7)更新感受野参数 $\boldsymbol{M}$
6.　END IF
7. END FOR
8. IF 没有局部模型被激活,即 $w_k<w_{gen},\forall k$
9.　新建局部模型,令 $\boldsymbol{c}=\boldsymbol{x}$,$\boldsymbol{M}=\boldsymbol{M}_{def}$ $K=K+1$
10. END IF
11. RETURN RFWR

4.2.4　预测算法

当一个新查询来临时,根据公式(3)计算查询的基数需要使用每一个局部模型,这可能会降低准确率和估计效率.针对上述问题,对于一个查询,本文选择部分局部模型进行预测,类似于 KNN 算法中取 k 个数据进行预测,这不仅不影响预测的准确性也能加快预测的速度.本文选取部分局部模型的原则如下:依赖于我们预先给定的预测阈值 w_{pred},如果该查询和某个局部模型的核距离 w 大于 w_{pred},则说明该查询属于该局部模型的感受野,从而用于该查询的预测.另外,我们还需要进一步验证这些所选取局部模型的预测是否准确.在本文,使用该局部模型的样本方差和样本个数作为是否有效的依据,如果样本方差小于某个给定的阈值 var 并且样本个数大于某个给定的阈值 $ndata$ 时模型有效,否则无效,而无效时采用最近邻算法 KNN 进行预测.具体地说,在该局部模型有效之前,还需要保留每个感受野中的历史数据用于 KNN 算法预测.对于新查询 $\boldsymbol{x}$,其相应 k 最邻预测值[20]为:

$$\hat{y}=\frac{\sum_{i=1}^{k}y_i\cdot sim(\boldsymbol{x}_i,\boldsymbol{x})}{\sum_{i=1}^{k}sim(\boldsymbol{x}_i,\boldsymbol{x})}, \tag{8}$$

其中,$sim(\boldsymbol{x}_i,\boldsymbol{x})$ 为 $\boldsymbol{x}$ 与 $\boldsymbol{x}_i$ 的相似度,本文中定义 $sim(\boldsymbol{x}_i,\boldsymbol{x})=\frac{1}{0.1+\|\boldsymbol{x}_i-\boldsymbol{x}\|_2}$,$\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_k$ 为距 $\boldsymbol{x}$ 最近的 k 个数据.当局部模型有效时,则利用这些暂存的数据建立局部加权线性模型,此后不再保留数据,只需增量更新模型.如果对于该查询,没有局部模型被选择时,则使用核距离最大的局部模型进行预测.完整预测算法见算法 2.

算法 2.　预测基数.

输入:新来的查询 $\boldsymbol{x}$,模型 RFWR,核距离阈值 w_{pred},方差阈值 var,数据阈值 $ndata$,最近邻 k

输出:查询 $\boldsymbol{x}$ 的基数 y

1. For k in K:
2.　根据公式(1)计算查询的权重 w_k
3.　IF $w_k>w_{pred}$ THEN
4.　　IF 方差<var and 训练数据> $ndata$ THEN
5.　　根据公式(2)计算 $\hat{y}_k$
6.　　ELSE
7.　　根据公式(8)计算 $\hat{y}_k$
8.　　END IF
9.　ELSE

10. 令 $w_k=0,\hat{y}_k=0$
11. END IF
12. END FOR
13. IF 公式(3)分母为 0 THEN
14. y 等于最大 w_k 对应的 $\hat{y}_k$
15. ELSE
16. y 等于公式(3)的值
17. END IF
18. RETURN y

4.2.5 时间复杂度分析

设当前模型的感受野个数为 K,查询 $\boldsymbol{x}$ 的特征向量维数为 n,它的预测时间复杂度为 T_1,训练复杂度为 T_2,下面计算在最坏情况下的 T_1 和 T_2.在最坏情况下,查询 $\boldsymbol{x}$ 的预测和训练过程涉及所有 K 个局部模型.对于给定的一个局部模型,它的预测根据公式(2)进行时,时间复杂度为 $O(1)$,根据公式(8)预测时,由于此时历史数据很少,因此时间复杂度也为 $O(1)$;它的训练过程根据公式(5)和(7)进行,主要涉及一些矩阵的计算,不难证明它的复杂度为 $O(n^3)$.因为各个局部模型的预测和训练相互独立,所以 $T_1=O(K)$,$T_2=O(Kn^3)$.

从上面的时间复杂度来看,查询的预测复杂度较小,而训练的复杂度随着 K 和 n 增长而变大.一方面,随着新硬件、多核、多线程等技术的发展和应用,每个局部模型的预测和训练过程完全可以并行执行,这样可以减少复杂度至 $T_1=O(1)$,$T_2=O(n^3)$.另一方面,本文探索剪枝方法在不影响准确性的前提下减少 K,详见 4.3.2 节,并且在 n 较大(高维查询)时,利用偏最小二乘法减少训练复杂度至 $T_2=O(Kn)$,详见 4.3.3 节.

4.3 模型训练优化

4.3.1 局部模型优化

在 RFWR 中,每一个局部模型都用一个线性函数表示.在批量数据的情形下,使用公式(4)求取回归系数.但是,当输入的特征向量维数很高或数据少时,可能发生 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X}$ 秩亏损,从而无法求取它的逆矩阵 $\boldsymbol{P}$.虽然训练的时候使用迭代公式(5)求取 $\boldsymbol{P}$,但是,如果当秩亏损发生时,求取的 $\boldsymbol{P}$ 误差很大,从而严重影响回归系数 $\boldsymbol{\beta}$ 的估计.因此,类似于岭回归,引入另一组参数 $\boldsymbol{r}=(r_1,r_2,\cdots,r_n)$ 表示 $\boldsymbol{\beta}$ 的偏差.在学习的时候,类似地通过梯度下降方法进行更新.通过求取 $\frac{\partial J}{\partial \boldsymbol{r}}$,$\boldsymbol{r}$ 更新如下[26]:

$$\boldsymbol{r}^{n+1}=\boldsymbol{r}^{n}-\alpha_r\frac{\partial J}{\partial \boldsymbol{r}},$$

其中 α_r 为学习率.

4.3.2 模型剪枝

在 RFWR 中,随着查询的不断到来,局部模型将会逐渐增多,这样均会导致模型的学习过程和查询预测过程逐渐地变慢,间接增加查询的延迟时间.事实上,当两个局部模型靠的很近时,它们俩的贡献就会很相似.因此,我们给定一个剪枝的阈值 w_{prune},设一个新来的查询 $\boldsymbol{x}$ 和所有局部模型的最大和次大核距离分别为 w_{max},w_{min},当满足 $w_{max}\geqslant w_{prune}$,$w_{min}\geqslant w_{prune}$ 时,就将其中感受野矩阵 $\boldsymbol{D}$ 较大者剪枝($\boldsymbol{D}$ 越大,感受野区域越小).

另外,在处理查询分布和数据分布变化的时候,无论是查询区域的变化还是数据发生变化,本文所提取的特征向量都能有效地捕捉这两方面的变化信息.因此,*RFWR* 会自适应地建立新的局部模型来适应分布的变化.那些曾经活跃的局部模型如果很久没被访问,则说明该局部模型体现的是数据变化前的知识或用户之前感兴趣的数据区域,也可以将其剪枝,以适应分布的变化.

4.3.3 降维处理

对于一个子执行计划树,当单表上不同属性的选择条件或查询所涉及的表增多时,它对应特征向量的维数将会急剧增长,特征向量元素之间可能相互关联.这激发了我们探究如何让模型处理这种具有相互关联元素的高维特征向量.

局部加权投影回归[27](*LWPR*)通过使用偏最小二乘法(*PLS*)对 *RFWR* 中的每个局部模型进行降维处理以提高模型的鲁棒性和准确性.其思路主要为:使用增量的 *PLS* 替换增量的最小二乘法.在新建局部模型的时候,初始化 *PLS* 的各个统计量参数为 0,然后随着查询的到来对其不断的更新.具体地说,对于查询的特征空间,假定有 r 个成分 $\boldsymbol{p}_i$,其对应的回归系数为 β_i,其中,$1\leqslant i\leqslant r$.给定去中心化的第 $n+1$ 训练数据 $(\boldsymbol{x},y)$,初始化 $\boldsymbol{z}=\boldsymbol{x}$,$res=y$,PLS 参数更新如下[27]:

对于 $i=1:r$,

$\boldsymbol{u}_i^{n+1}=\lambda\boldsymbol{u}_i^{n}+w\boldsymbol{z}res$,$s=\boldsymbol{z}^{\mathrm{T}}\boldsymbol{u}_i^{n+1}$,

$SS_i^{n+1}=\lambda SS_i^{n}+ws^2$,$SR_i^{n+1}=\lambda SR_i^{n}+wsres$,

$SZ_i^{n+1}=\lambda SZ_i^{n}+w\boldsymbol{z}s$,$\beta_i^{n+1}=SR_i^{n+1}/SS_i^{n+1}$,

$\boldsymbol{P}_i^{n+1}=SZ_i^{n+1}/SS_i^{n+1}$,$\boldsymbol{z}=\boldsymbol{z}-s\boldsymbol{p}_i^{n+1}$,

$res=res-s\beta_i^{n+1}$,$SSE_i^{n+1}=\lambda SSE_i^{n}+wres^2$.

其中,λ 表示遗忘因子,SSE_i 表示第 i 维的预测误差.在本文实验中,设定 r 初始值为 $r=2$,随着训练的进行根据需要自增长 r,具体地说,判断最后两维

方向的预测误差比值 SSE_r/SSE_{r-1} 是否大于给定的参数 φ(本文取 $\varphi=0.5$).如果大于 φ,则说明最后一维对回归的结果有一定的影响,于是令 $r=r+1$,否则,r 保持不变.在预测阶段,给定 $\boldsymbol{x}$,y 可以由下式递归的算出[27]:

对于 $i=1:r$,

$$s=\boldsymbol{z}^{\mathrm{T}}\boldsymbol{u}_i,y=y+s\beta_i,\boldsymbol{z}=\boldsymbol{z}-s\boldsymbol{p}_i^{\mathrm{T}}.$$

通过以上的替换,首先可以减少计算的复杂度.具体地说,当 r 较小时,对比 4.2.5 节的训练时间复杂度 $O(n^3)$,使用 LWPR 训练时间复杂度为 $O(n)$,n 表示查询向量的维数.其次,由于 PLS 自身的特性,模型有更强的鲁棒性和准确性.

5　实验验证

5.1　实验设置

5.1.1　对比方法

针对基数估计的准确性,本实验对比了三类方法:数据驱动的方法、局部模型组合方法、全局模型方法.

数据驱动的方法:

(1) PostgreSql 基数估计方法(图中记作 RULE):通过一些假设规则进行估计,比如要估测单表多属性的查询,将假设表不同属性之间相互独立.

(2)Naru[7]:该方法使用自回归模型将多属性数据的联合分布问题转化为求多个条件分布的问题.使用抽样数据进行非监督训练,此过程不需要做任何数据分布假设.预测的过程通过结合渐进采样进行,采样数据越多,预测越准确,但延迟也更大.我们使用其开源代码[54]进行实验,Naru-num 表示使用 num 个样本进行预测.

局部模型组合方法:

(3)动态 KNN 方法(KNNDynamic)[20]:它是最近邻方法的在线算法.通过 AQO[20]的研究可知,动态 KNN 方法基本和 KNN 方法的水平相当,且不需要保存全部数据,也优于固定样本的 KNN 方法.实验中设定参数数目 $K=300$,相邻个数 $k=3$.

(4)QueryModel1[11]:该方法随查询的到来,根据查询空间变化自适应的增加局部模型,每个局部模型使用一个线性模型来表示.该方法不支持根据查询的误差自动调节局部模型的有效区间,预测的时候只使用和当前查询最近的局部模型.

(5)QueryModel2[10]:该方法随查询的到来,根据查询空间的变化,通过 SOM 神经网络将查询映射到一个预先给定的网格上.其中,网格上每个点为一个用中位数表示的局部模型.做预测时,使用加权平均各局部模型输出进行预测.但是,该方法类似于动态 KNN,需要预先给定局部模型的数量,这在动态环境下很难确定.实验设定局部模型为 100(实验发现 300 不如 100 好).

(6)XGBOOST[13,55]:该方法为树(或线性)模型的集成.在学习的过程中,将输入空间使用不同的方式划分成各个子空间,预测时,通过加权各个局部模型的预测值得到.本实验中,局部模型采用 CART 回归树,使用增量(batch=50)的方式进行训练.

全局模型方法:

(7)线性回归(Linear):该方法简单的假设输入和输出成线性关系,学习速度较快,但当假设不成立时预测误差较大.本文使用随机梯度下降的方法进行在线训练.

(8)深度神经网络(DNN)[13]:保持和文献[13]一致,我们使用 keras 实现一个全连接的深度神经网络,隐藏层均使用 RELU 激活函数,输出层使用线性激活函数.在本实验中,尝试了不同的层,不同隐藏层节点数目,调节了不同的参数,使用 adam 优化方法进行在线训练.最终权衡学习的平稳性和准确性,选取层为 3,隐节点数目为 300,学习率为 0.001,衰减因子为 0.001 进行学习曲线对比.

本文方法:

(9)增量的局部加权学习:模型的训练和预测分别根据算法 1,2 进行,记作 RFWR_MODIFY.对 RFWR_MODIFY 降维优化后的方法记作 RFWR_MODIFY_PLS.

5.1.2　数据集

本文使用四个真实数据集和一个合成数据集.

(1)Forest[13]:原有的 Forest 表含有 54 个属性和 581012 条数据,我们保持和以前的研究一致使用前 10 个数值属性,且属性之间基本相互独立.

(2)StrongCor[15,20]:该数据集为合成数据集,可以根据需要生成任意多个属性和数据,它的属性之间相互关联.设 $randInt(a,b)$ 表示从 a 和 b 之间取随机数,x_{ic} 表示第 i 个数据第 c 个属性的值,那么数据集生成如下:

$$x_{i,1}=randInt(0,100),$$

$$x_{i,n}=(x_{i,n-1}+randInt(2,3))\mathrm{bmod}100.$$

(3)Power[13]:该数据集为一个家庭 4 年内的用电量信息.其包括 9 个属性,前两个分别为日期和时

间,后 7 个均为数值类型.该数据集包含 200 多万条数据.

(4)TPC-H:该数据集有 8 张表,表的数据量由 SF 进行描述,本文选取 SF=1(1GB).

(5)IMDB[1,56]:该数据集有 21 张表,主要包含电影、演员等方面的真实数据.不同于 TPC-H,它的数据分布不均匀,属性之间相互关联,对基数估计方法来说,具有更高的挑战性.

5.1.3 查询负载

本实验中,我们使用数据集 Forest 和 Strongcor 验证本文方法较其它监督学习模型和 PostgreSQL 估计器具有更好的优势,也用来验证提取特征的合理性和高效性、模型的自适应性等;使用 Power 和基于数据驱动的神经网络方法 Naru 进行对比;使用 IMDB 验证本文方法适用于一般含 Join 的查询并验证本文所提取特征的合理性;最后,使用 TPC-H 验证本文方法对查询性能的影响.

针对 IMDB 数据集,在特征和方法对比实验中,我们对第 4 条查询模板的两个选择谓词 mi_idx.info >'const1'、t.production_year >'const2'中的 const1/2 在数据域内进行随机均匀生成,以此生成一个计算在 4 个选择条件下 info_type、keyword、movie_info_idx、movie_keyword 和 title 五表 join 后基数的负载,简称 Imdb_q4;针对 TPC-H,在性能验证实验中,我们将本文方法在 PostgreSQL 数据库内核中进行实现,然后使用 TPC-H 的 20 条查询语句验证其对查询性能的影响.

针对其它三个单表数据集,对于给定的表和查询维数 d,可以生成如下两类范围查询负载:依赖表中数据的查询和依赖属性值域的查询.其中,依赖表中数据的查询主要分布在数据空间之上,本文生成的查询集中在一组数据区域中;依赖属性值域的查询只关注属性值域,而不关注数据本身.对于后一类负载,本文使用两种方式生成:第一种,每个查询的中心点均匀分布在各个属性值域上,查询范围小于该属性相应的值域;第二种,每个查询来自于值域内 K 个不同高斯分布 (c_k,v_k,l_k),$k=1,2\cdots,K$,其中,c_k 表示分布的均值,v_k 表示方差,l_k 为查询区间的半长.具体地说,从 (c_k,v_k,l_k) 分布中抽取查询 q 的过程如下:对于查询 q 的每一个属性 i,根据高斯分布 $N(c_{ki},v_{ki})$ 随机抽取查询的中心点 x_{ki},从而得到查询的下界为 $a=x_{ki}-l_{ki}$,上界为 $b=x_{ki}+l_{ki}$.

本文对不同数据量、不同维度以及不同生成方式的查询负载进行了充分的实验验证.在实验中,除了特殊说明外,每一组查询负载均随机生成 5000 个用于迭代在线训练模型,再随机生成 200 个查询作为测试数据用于验证迭代学习的效果,最终得到学习曲线.本文所用的查询负载如下:为 StrongCor 数据集生成三组查询负载,第一组,设定 StrongCor 的数据量为 40 万,通过依赖表中数据的生成方式获取 3 维查询负载 1:StrongCor_workload1;第二组设定 StrongCor 数据量为 4 万,使用依赖属性值域的方式根据均匀分布获取 3 维查询负载 2:StrongCor_workload2;第三组和第二组生成方式一致,只是生成 6 维的查询负载 3:StrongCor_workload3.对于数据集 Forest 生成维数为 3 的三组查询负载,通过依赖属性值域的均匀分布、高斯分布以及依赖表中数据的生成方式分别生成负载:Forest_workload1、Forest_workload2、orest_workload3.对于数据集 Power,通过依赖属性值域的均匀分布生成 7 维负载 Power_workload.各个负载在实验中的作用见表 1.

表 1 数据集和相应查询负载的作用

数据集	查询负载	作用
Forest	Forest_workload1,2,3	特征对比:1 方法对比:1,2,3 降维验证:2 参数分析:2
StrongCor	StrongCor_workload1,2,3	特征对比:1 方法对比:1,2,3 降维验证:2 自适应学习:1
Power	Power_workload	与 Naru 对比
TPC-H	20 条 Select 语句	性能验证
IMDB	Imdb_q4	特征对比 方法对比

5.1.4 参数设置和实验环境

本文中,训练和预测算法中的参数设置如表 2 所示,其中,初始矩阵 $\boldsymbol{P}$ 由初始值组成的对角矩阵来表示[26].我们取矩阵 $\boldsymbol{M}$ 为对角矩阵,因为这使得

表 2 实验参数设置

参数	值	参数	值
感受野阈值 w_{gen}	0.1	遗忘因子 λ	1(自适应实验取 0.997)
剪枝阈值 w_{prune}	0.9	矩阵 $\boldsymbol{P}$ 初始生成值	$1.e^{+3}$
学习阈值 l	0.001	矩阵 $\boldsymbol{M}$ 初始生成值	5
预测阈值 w_{pred}	0.001	矩阵 $\boldsymbol{M}$ 学习率初始值	0.05
正则项系数 γ	$1.e^{-4}$	预测算法中 KNN 的 k 值	2

算法更加简单轻量化,并通过实验验证,其可以取得和非对角矩阵相似的效果.参数的不同设置对模型的影响将在第 5.7 节详细讨论.

数据库使用 PostgreSQL10,在查询完成的时候,通过改写内核,将查询和其真实基数进行输出作为训练、测试数据.除性能验证实验外,所有的对比实验均在内存为 16GB,处理器为 Intel(R) Core(TM) i5－8400 CPU@2.80GHZ 的 win10 系统上完成.性能验证实验在内存为 2.3GB,Ubuntu 16.04 的 VMware 虚拟机上进行.

5.1.5　评测标准

为了和之前的工作保持一致[20],本文关于准确性的度量均采用基数自然对数的均方误差(MSE),设测试的查询集 W 个数为 T ,那么均方误差为

$$MSE(W)=\frac{1}{T}\sum_{t=1}^{T}(true\_card_t-pred\_card_t)^2$$

其中,$true\_card_t$,$pred\_card_t$ 分别表示第 t 个查询的真实基数和预测基数的自然对数值.

5.2　特征提取

本节主要验证本文所提取特征的合理性和高效性.我们使用 RFWR_MODIFY 分别在三个数据集上验证三类特征:查询语义特征(semantic_feature)、统计特征(statistic_feature)和混合特征(hybrid_feature).以基于直方图的方法为参考,实验结果如图 3 所示.横坐标是训练数据不断迭代训练的过程,每增量学习 200 查询,都会对测试数据计算基数自然对数的均方误差.

对于数据集 Forest,实验(a)表明混合特征表现最好,统计特征次之,语义特征最差,但都不如直方图的方法准确率高,原因在于直方图的假设正好成立,而本文的方法起初需要学习这些知识,从图看出,它也在不断逼近直方图的准确性.语义特征不高效的原因在于它不能体现数据库内部的数据分布信息,虽然可能语义很相似,但在数据库内部差距较大,因此加入统计特征能够提高模型的准确性和鲁棒性;对于数据集 Strongcor,图(b)表明在假设不成立时基于直方图方法最差,也表明混合特征和语义特征基本一致,统计特征较差.通过观察实验发现,造成统计特征不高效的主要原因为:该数据集属性之间相互关联,PostgreSQL 对大部分选择谓词的边缘选择率估计为 0.333,这导致每个查询的统计特征都很相似,从而引起学习的偏差较大.因此,结合语义特征可以降低统计特征的不可靠性.最后,验证本文特征方法在 join 查询上的合理性.我们选取具有挑战性的数据集 IMDB 和它上的 Imdb_q4 负载进行实验.不同于 TPC-H,基于直方图的规则方法在计算该类查询的 join 谓词边缘选择率时误差较大,因此可以探索使用不准确的统计特征对本文模型的影响.实验结果如图(c)所示,通过分析和观察,我们得出如下结论:首先,相对于本文方法,基于直方图的方法误差较大;其次,使用混合特征,模型收敛时的误差约为 0.02,相对于单独使用语义特征和统计特征,准确率分别提高了 2.1 和 0.03;最后,加入统计特征可极大提高模型的准确率.这些结论也证明了本文所提取特征的高效性和合理性.后文实验中,对于每种机器学习方法,均采用混合特征.

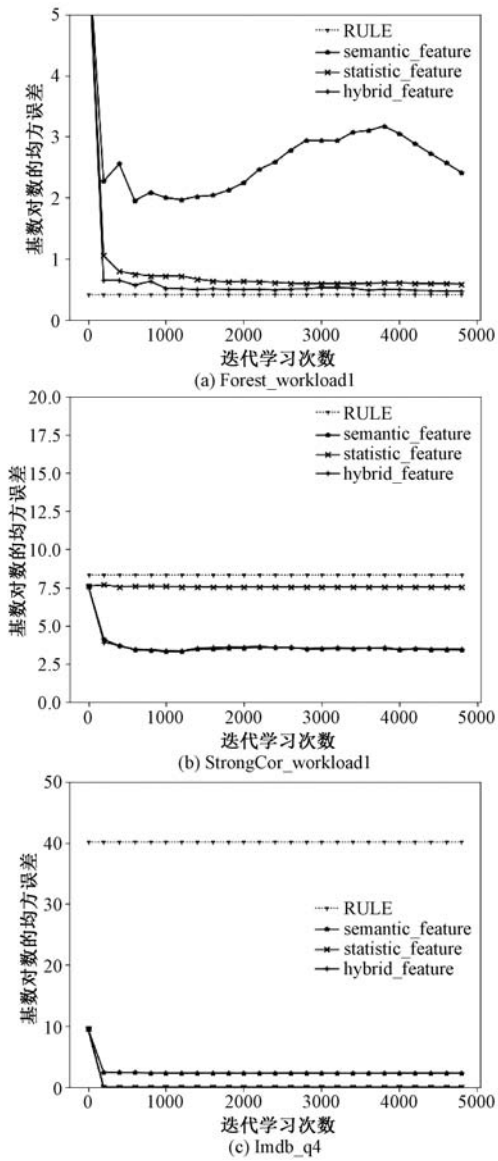


图 3　特征化方法对比

5.3　方法对比

本节使用不同的查询负载对本文方法和其它方

法进行实验对比.我们首先对比了本文方法和除XGBOOST(简称XGB)外其它监督学习模型在迭代学习过程中的优劣;然后再和XGB方法进行对比;最后,和数据驱动的自回归模型Naru[7]进行对比.

对于数据集Forest,图4(a)、(b)、(c)为不同方法在不同负载上学习曲线的对比.实验结果表明,本文的方法在学习起初阶段误差较小而且收敛较快,约在1000个查询后收敛,满足我们提出的在少量数据下预测准确性高的要求,也表明本文的方法在整个学习过程中准确率较高.DNN在起初阶段误差较大且不平滑,但随着训练数据的增多误差逐渐变小,甚至在有些负载上略高于本文方法.但是,在真实场景下,特别在数据和查询不断变化环境下,较难有大量训练数据.动态KNN跟本文方法类似,但是由于模型较简单拟合能力有限,因此准确率较低,且它在数据变化时自适应能力较差,详见实验5.4.另外,线性回归模型因其不能拟合较复杂的函数,故预测误差较大,而QueryModel1较线性回归准确率高,但学习抖动较大,QueryModel2学习较为平稳,准确率较低.对于数据集StrongCor,通过对图5(a)、(b)、(c)分析,得出和Forest数据集相似的结论.另外,我们计算了5000次学习过程中其它方法与本文方法平均测试误差的差值,结果如表3所示,除去黑体部分显示本文方法在少数负载上平均误差微大于相应方法,大部分结果表明本文方法在不同负载下都优于其它的对比方法.对于Imdb-q4,通过图6可知,在单表数据集上得出的结论同样适用于Join查询.具体地说,本文方法、动态KNN以及NN在该负载上表现较好,但本文方法相对于动态KNN和NN收敛的更快,鲁棒性更强.其它方法相对本文方法,准确率较低.

表3 学习过程中其它算法和本文方法平均MSE的差值

数据集	算法	负载1	负载2	负载3
Forest	PostgreSql	−0.12	0.02	−0.02
	KnnDynamic	0.16	0.04	0.12
	QueryModel1	0.94	0.23	2.40
	QueryModel2	1.51	0.44	0.35
	Linear	2.02	0.46	0.18
	DNN	0.81	0.16	0.18
Strong cor	PostgreSql	4.77	2.04	5.43
	*K*NNDynamic	1.08	−0.03	1.50
	QueryModel1	3.31	2.20	6.07
	QueryModel2	3.68	3.51	6.87
	Linear	2.29	1.77	4.70
	DNN	1.06	0.32	1.07

对于XGB,由于其不能进行单个训练数据的增量学习,故在本实验中,我们使用批量增量学习的方式(batch=50)对模型进行迭代更新.在XGB每次学习完新的4个batch数据后,我们使用测试集进行测试,最终在三个数据集上得到各个学习曲线,如图4,5,6所示.在数据集strongcor的负载1和2上,XGB在学习过程中准确率较高,且最高准确率分别高出本文方法1.96和0.17,但是,它增量学习过程抖动较大;在数据集forest的负载3上,起初学习时准确率高于本文,但在学习1000样本后,本文的方法准确率更高,收敛时准确率高于XGB约为0.007;在IMDB数据集上,本文的方法较XGB收敛的更快,且在收敛后的准确率较XGB高0.006;在剩余的负载上,相对本文的方法,XGB的学习准确率较低且学习抖动较大.总结来说,一方面,XGB方

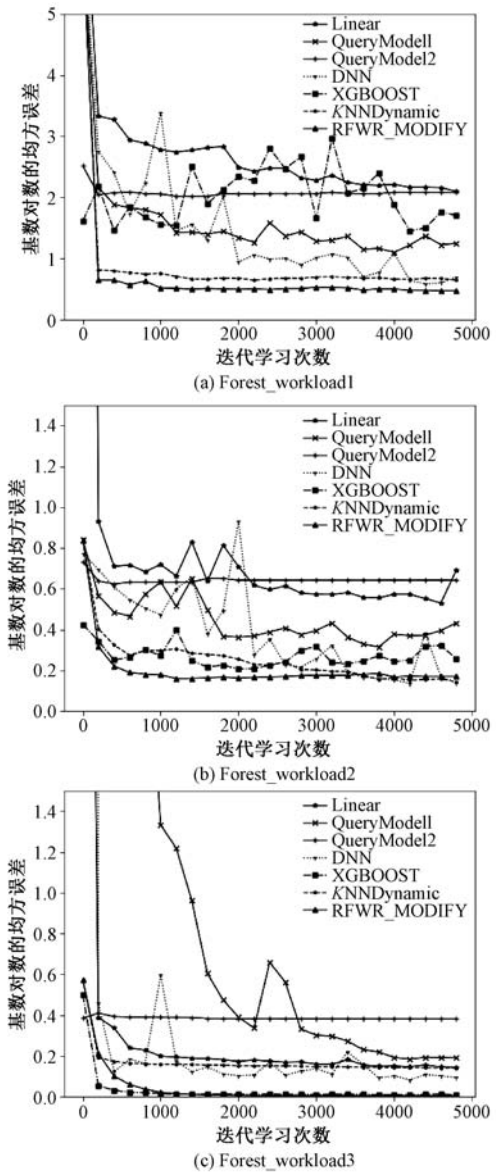
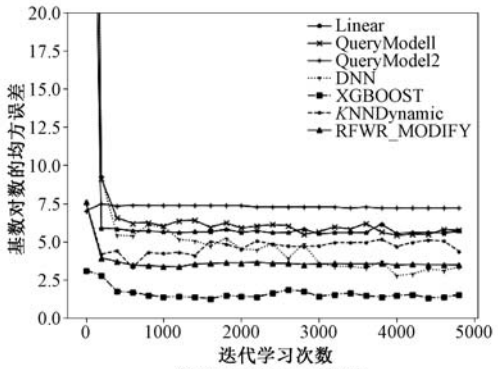
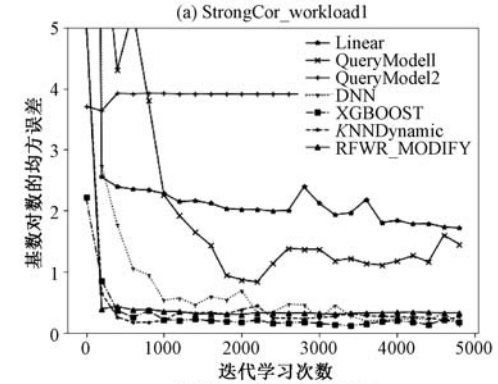


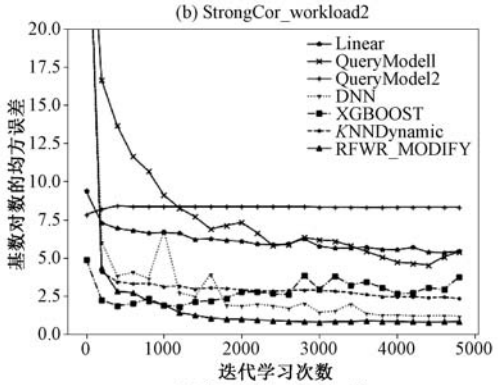
图4 Forest数据集



(a) StrongCor_workload1



(b) StrongCor_workload2



(c) StrongCor_workload3

图 5　StrongCor 数据集

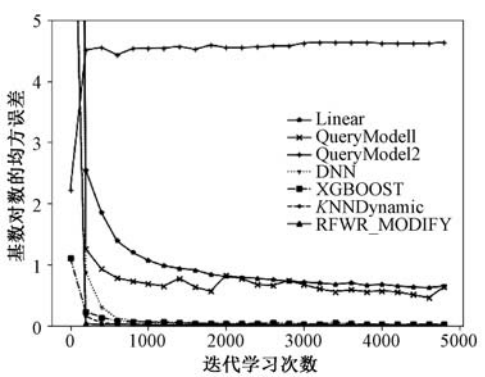


图 6　IMDB 数据集(Imdb_q4)

法在多个负载上准确率低于本文方法;另一方面,采用增量训练的方式,XGB 学习过程普遍抖动较大. 这是因为 XGB 的增量学习过程,不允许改变之前树的结构,从而导致学习的鲁棒性和自适应性较低. 不同于 XGB,本文方法可进行单个数据的增量学习,能真实的用于流学习环境,且学习过程可根据查询和数据的特性,对之前学习的局部模型进行更新,剪枝.

对于 Naru,我们分别尝试了原文[7]中的三个自回归模型:MADE、ResMADE 以及 Transformer,并对每个模型试验了不同的神经网络结构和训练参数. 在预测的过程中使用相应的模型对随机抽取的 2000 条查询进行预测,并在预测查询的基数时尝试了不同的样本数目(2000/4000/10000). 在本实验中,我们在负载 Power_workload 上进行实验,选取了最高的 Naru 预测准确率(MSE=102.32)作为参考值,和本文的方法进行对比. 对本文方法,使用 900 个查询对模型进行增量训练,且在每次迭代学习 100 个查询后,对同样的 2000 查询进行预测并求取 MSE,实验结果如图 7 所示. 从图中可以看出,本文的方法在学习起初阶段,相对于 Naru 准确率较高,且在学习 900 查询后,基数对数的均方误差可降至 10 以下,远远的优于数据驱动方法 Naru. 通过分析,相对于本文方法,Naru 在该负载上准确率较低的原因有二:其一,Naru 在预测点查询的基数时,本身存在误差;其二,在预测查询空间较大的范围查询时,需要结合各采样数据的点查询结果进行估算,而此过程产生了很大的误差.

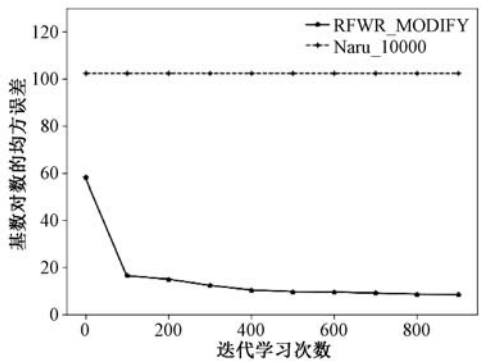


图 7　与 Naru 对比(Power_workload)

5.4　自适应学习

本节验证本文所提出的方法能随查询分布和数据分布的变化进行自适应学习的能力. 本节只验证其中之一变化时模型的自适应能力,并和动态 KNN 方法进行实验对比,结果如图 8 所示. 对于图(a),我们首先使用固定的查询分布生成训练数据对两个模型进行训练至收敛. 然后改变查询分布,生成一组查询流对模型迭代测试并学习. 具体地说,分别在开始和靠近第 290 个查询的位置对分布进行变化,从图中可以看出两个方法对不同位置的分布变化敏

感度不同,但均可在学习 50 个查询左右后收敛.对于图(b),先从固定的查询分布中生成训练数据,对模型进行训练至收敛.在测试阶段,再使用同一查询分布生成一组查询流在不同数据环境下进行迭代测试并学习.具体地说,在第 70 个查询的位置对 strongcor 表增加 0.5 倍数据.从图中可以看出,当数据发生变化时,二者预测误差均增大,动态 KNN 方法并不能像在查询分布变化后进行自适应调整,而本文的方法随着学习 80 查询左右后,误差逐渐减小,这说明本文的方法对数据和查询分布的变化均有自适应性.

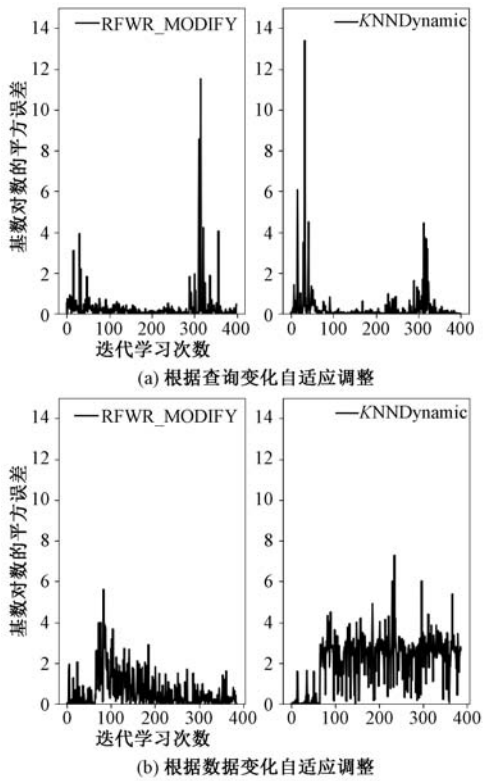


图 8 查询和数据变化自适应调整

5.5 降 维

本节验证降维技术的有效性,实验结果如图 9 所示,我们从二个不同的数据集进行了验证,所选取的两个相应负载均为三维查询,按照 4.1 节所提出的特征方法,其对应的特征向量维数均为 12.图 9(a)、(b)表明降维的方法在起初的时候因为选用的成分 r 较少而误差比没有降维时大,但随着 r 的增加,其具有更好的鲁棒性和准确性.

5.6 预测时间对比及性能验证

本节首先对比了各个监督学习算法的平均预测时间,结果如表 4 所示,RFWR 的预测时间随局部模型的增多而变长,如 1 个局部模型时只需 0.1ms,而 130 个局部模型时需要 2.6ms.在局部模型较多时,较神经网络和线性回归等高,但其可通过并行

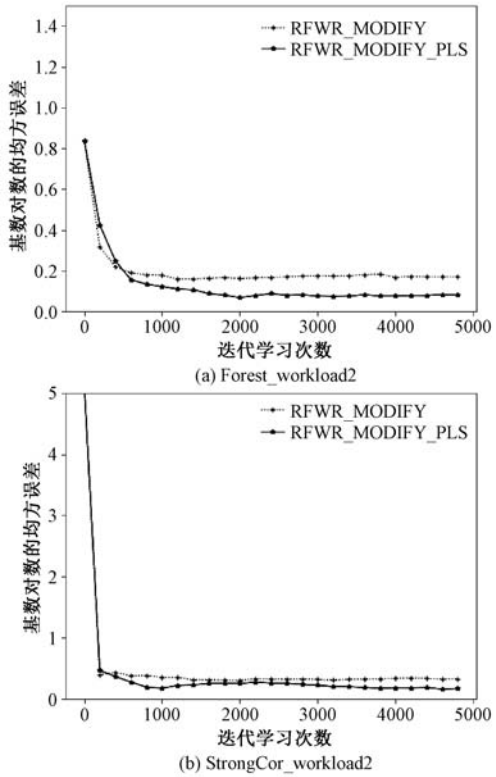


图 9 降维对比

进行加速,而且相对于减少的执行时间,这些代价基本可忽略不计.本节也验证使用本文方法是否可以减少查询执行时间,以及验证其查询优化的开销时间.我们和 PostgreSQL 自身优化时间和执行时间做了对比,结果如图 10 所示,第 17、20 条查询因其执行时间较长且结果类似,为了作图方便将其省去.图(a)表明使用本文的方法查询优化时间稍有延迟,较原优化时间平均延迟 1.23445ms,基本可忽略.图(b)表明,使用本文方法有些查询的执行时间变短(如 18),也有些执行时间反而变长(如 4),通过统计计算,执行时间平均提高 5.224ms.提高不多的原因,我们通过分析发现其不在于基数估计的错误,而在于执行计划生成算法本身的缺陷,使得有些子计划有很大基数估计错误但永远不被执行,从而使模型无法学习该错误[15],探索新的执行计划生成算法是我们的未来工作.

表 4 单个查询平均预测时间对比(ms)

算法	KnnDynamic	Query Mode1	Query Mode2	Linear	DNN	RFWR_MODIFY
预测时间(ms)	3.75	0.44	2.35	0.01	1.77	0.1(1) 0.5(50) 1.8(100) 2.6(130)

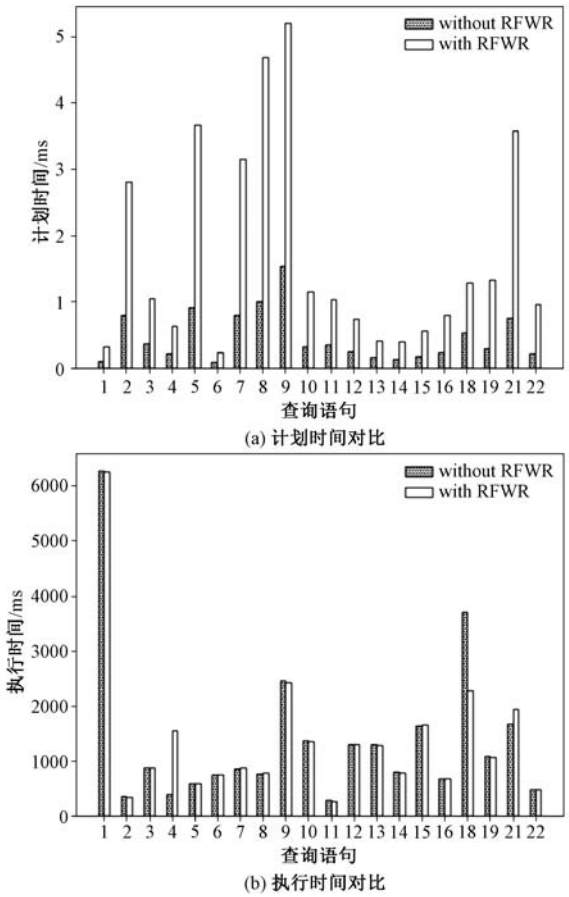
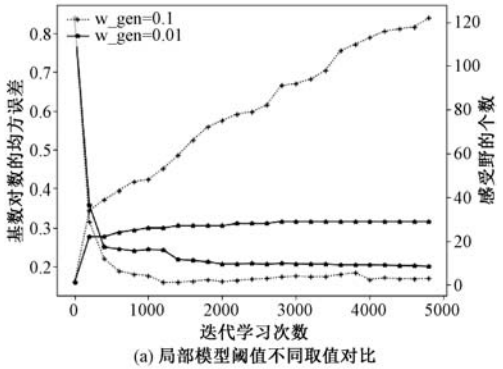


图 10　性能对比

5.7　参数分析

本节讨论和分析模型中参数不同取值对模型的影响,结果如图 11 所示.其中,(a)表明当生成阈值越大时,它的准确性越高,但相应感受野就越多;(b)表明通过剪枝,可以在保证准确率的前提下,减少感受野的数目,从而加快训练和预测过程;(c)表明感受野矩阵越大,则准确率越高,但感受野也就越多.因此,本文权衡准确率和高效性选取各参数.另外,(d)、(e)、(f)表明,该模型对于其它参数敏感性不高.



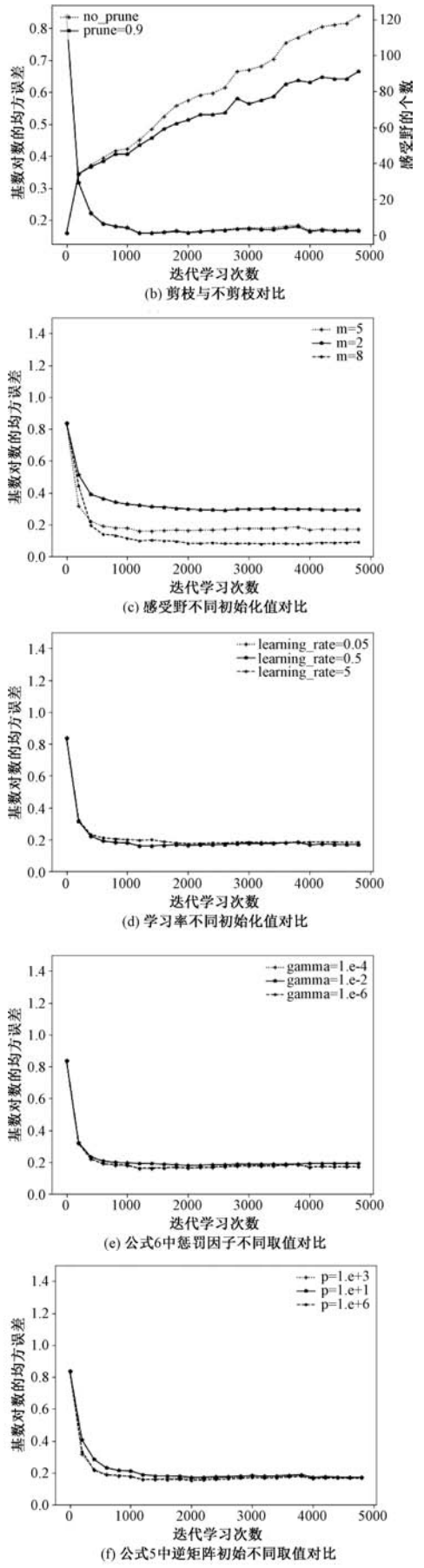


图 11　参数分析

6 总结和未来工作

本文针对子执行计划模板在数据分布和查询分布不断变化的环境中基数估计不准确的问题,提出一种利用在线学习来提高准确性的方法.该方法首先提取关于子执行计划的语义和统计特征,使之能够代表当前查询和数据的特性;然后引入增量的局部加权学习方法根据查询分布和数据分布的变化进行自适应的学习,实现基数估计.通过实验验证了该方法的有效性.

虽然本文的方法可以较为准确的修正数据库基数估计的错误,但是,因为执行计划生成算法的自身缺陷,使得数据库可能依然选择不好的执行计划,从而性能提升不高[19-20].因此,结合本文工作探索其它执行计划搜索算法是我们的未来工作.

参 考 文 献

[1] Leis V, Gubichev A, Mirchev A, et al. How good are query optimizers, really?. Proceedings of the VLDB Endowment, 2015, 9(3): 204-215

[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] Getoor L, Taskar B, Koller D. Selectivity Estimation using Probabilistic Models. ACM SIGMOD Record, 2001, 30(2): 461-472

[4] Heimel M, Kiefer M, Markl V, et al. Self-Tuning, GPU-Accelerated Kernel Density Models for Multidimensional Selectivity Estimation// Proceedings of the International Conference on Management of Data. Melbourne, Australia, 2015: 1477-1492

[5] Kiefer M, Heimel M, Bres S, et al. Estimating join selectivities using bandwidth-optimized kernel density models. Proceedings of the VLDB Endowment, 2017, 10(13): 2085-2096

[6] 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 (6): 1035-1050

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

[8] Yang Z, Kamsetty A, Luan S, et al. NeuroCard: One Cardinality Estimator for All Tables. arXiv: 2006.08109, 2020

[9] Hilprecht B, Schmidt A, Kulessa M, et al. DeepDB: Learn from Data, not from Queries!. arXiv: Databases, 2019

[10] Anagnostopoulos C, Triantafillou P. Query-Driven Learning for Predictive Analytics of Data Subspace Cardinality. ACM Transactions on Knowledge Discovery from Data, 2017, 11 (4):1-46

[11] Anagnostopoulos C, Triantafillou P. Learning to accurately COUNT with query-driven predictive analytics// Proceedings of the IEEE International Conference on Big Data (Big Data). Santa Clara, USA, 2015: 14-23

[12] Anagnostopoulos C, Triantafillou P. Learning Set Cardinality in Distance Nearest Neighbours// Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM). Atlantic City, USA, 2015: 691-696

[13] Dutt A, Wang C, Nazi A, et al. Selectivity estimation for range predicates using lightweight models. Proceedings of the VLDB Endowment, 2019, 12(9): 1044-1057

[14] Park Y, Zhong S, Mozafari B, et al. QuickSel: Quick Selectivity Learning with Mixture Models// Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland, USA, 2020(6): 1017-1033

[15] Liu H, Xu M, Yu Z, et al. Cardinality estimation using neural networks// Proceedings of the International Conference on Computer Science & Software Engineering. Markham, Canada, 2015: 53-59

[16] Kipf A, Kipf T, Radke B, et al. Learned cardinalities: Estimating correlated joins with deep learning. arXiv preprint arXiv:1809.00677, 2018

[17] Sun J, Li G. An end-to-end learning-based cost estimator. Proceedings of the VLDB Endowment, 2019, 13(3): 307-319

[18] Stillger M, Lohman G M, Markl V, et al. LEO-DB2's learning optimizer// Proceedings of the 27th International Conference on Very Large Data Bases. Roma, Italy, 2001, 1: 19-28

[19] 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

[20] Ivanov O, Bartunov S. Adaptive Cardinality Estimation. arXiv: Databases, 2017

[21] Malik T, Burns R C, Chawla N V. A Black-Box Approach to Query Cardinality Estimation// Proceedings of the Third Biennial Conference on Innovative Data Systems Research. Asilomar, USA, 2007: 56-67

[22] Woltmann L, Hartmann C, Thiele M, et al. Cardinality estimation with local deep learning models// Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. Amsterdam, Netherlands, 2019: 1-8

[23] Ortiz J, Balazinska M, Gehrke J, et al. An Empirical Analysis of Deep Learning for Cardinality Estimation. arXiv: Databases, 2019

[24] Hayek R, Shmueli O. Improved Cardinality Estimation by Learning Queries Containment Rates.. arXiv: Databases, 2019

[25] Negi P, Marcus R, Mao H, et al. Cost-Guided Cardinality

Estimation: Focus Where it Matters// Proceedings of the 36th IEEE International Conference on Data Engineering Workshops. Dallas, USA, 2020: 154-157

[26] Schaal S , Atkeson C . Constructive incremental learning from only local information. Neural Computation, 1998, 10(8):2047-2084

[27] Vijayakumar S, D'souza A, Schaal S. Incremental online learning in high dimensions. Neural computation, 2005, 17(12): 2602-2634

[28] Deshpande A, Garofalakis M, Rastogi R. Independence is good: Dependency-based histogram synopses for high-dimensional data. ACM SIGMOD Record, 2001, 30(2): 199-210

[29] Lynch C A. Selectivity Estimation and Query Optimization in Large Databases with Highly Skewed Distribution of Column Values// Proceedings of the Fourteenth International Conference on Very Large Data Bases. Los Angeles, USA, 1988: 240-251

[30] 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. Chicago, USA, 1988: 28-36

[31] Lipton R J, Naughton J F, Schneider D A. Practical selectivity estimation through adaptive sampling// Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data. Atlantic City, USA, 1990: 1-11

[32] Haas P J, Naughton J F, Seshadri S, et al. Selectivity and cost estimation for joins based on random sampling. Journal of Computer and System Sciences, 1996, 52(3): 550-569

[33] Freitag M, Neumann T. Every Row Counts: Combining Sketches and Sampling for Accurate Group-By Result Estimates// Proceedings of the 9th Biennial Conference on Innovative Data Systems Research. Asilomar, USA, 2019, 1: 1-39

[34] Müller M, Moerkotte G, Kolb O. Improved selectivity estimation by combining knowledge from sampling and synopses. Proceedings of the VLDB Endowment, 2018, 11(9): 1016-1028

[35] Aboulnaga A, Chaudhuri S. Self-tuning histograms: Building histograms without looking at data. ACM SIGMOD Record, 1999, 28(2): 181-192

[36] Bruno N, Chaudhuri S, Gravano L, et al. STHoles: a multidimensional workload-aware histogram// Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data. Santa Barbara, USA, 2001: 211-222

[37] Lim L, Wang M, Vitter J S. SASH: A self-adaptive histogram set for dynamically changing workloads// Proceedings of the 29th International Conference on Very Large Data Bases. Berlin, Germany, 2003: 369-380

[38] Markl V, Haas P J, Kutsch M, et al. Consistent selectivity estimation via maximum entropy. The VLDB Journal, 2007, 16(1): 55-76

[39] Markl V, Megiddo N, Kutsch M, et al. Consistently estimating the selectivity of conjuncts of predicates// Proceedings of the 31st International Conference on Very Large Data Bases. Trondheim, Norway, 2005: 373-384

[40] Srivastava U, Haas P J, Markl V, et al. Isomer: Consistent histogram construction using query feedback// 22nd International Conference on Data Engineering. Atlanta, USA, 2006: 39-39

[41] Ortiz J, Balazinska M, Gehrke J, et al. Learning state representations for query optimization with deep reinforcement learning. arXiv preprint arXiv:1803.08604, 2018

[42] Marcus R, Papaemmanouil O. Deep reinforcement learning for join order enumeration// Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. Houston, USA, 2018: 3:1-3:4

[43] Heitz J, Stockinger K. Join Query Optimization with Deep Reinforcement Learning Algorithms. arXiv: Databases, 2019

[44] Krishnan S, Yang Z, Goldberg K, et al. Learning to optimize join queries with deep reinforcement learning. arXiv preprint arXiv:1808.03196, 2018

[45] Marcus R, Papaemmanouil O. Towards a hands-free query optimizer through deep learning. arXiv preprint arXiv: 1809.10212, 2018

[46] Marcus R, Negi P, Mao H, et al. Neo: A learned query optimizer. arXiv preprint arXiv:1904.03711, 2019

[47] Marcus R, Negi P, Mao H, et al. Bao: Learning to Steer Query Optimizers. arXiv: 2004.03814, 2020

[48] Kaftan T, Balazinska M, Cheung A, et al. Cuttlefish: A lightweight primitive for adaptive query processing. arXiv preprint arXiv:1802.09180, 2018

[49] Perron M, Shang Z, Kraska T, et al. How I Learned to Stop Worrying and Love Re-optimization// Proceedings of the 2019 IEEE 35th International Conference on Data Engineering. Macao, China, 2019: 1758-1761

[50] Yin S, Hameurlain A, Morvan F. Robust query optimization methods with respect to estimation errors: A survey. ACM Sigmod Record, 2015, 44(3): 25-36

[51] Hoi S C, Sahoo D, Lu J, et al. Online Learning: A Comprehensive Survey. arXiv: Learning, 2018

[52] Gama J, Žliobait? I, Bifet A, et al. A survey on concept drift adaptation. ACM Computing Surveys, 2014, 46(4): 44:1-44:37

[53] Atkeson C G, Moore A W, Schaal S. Locally weighted learning. Lazy learning. Dordrecht, The Netherlands: Springer, 1997: 11-73

[54] https://github.com/naru-project/naru

[55] Chen T , Guestrin C . XGBoost: A Scalable Tree Boosting System// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, USA, 2016: 785:794

[56] http://homepages.cwi.nl/~boncz/job/imdb.tgz

FENG Jie-Ming, Ph. D. candidate. His research interest is query optimization.

LI Zhan-Huai, Ph. D., professor, Ph. D. supervisor. His research interest is database theory and technology.

CHEN Qun, Ph. D., professor, Ph. D. supervisor. His current research interests include gradual machine learning and risk analysis for AI.

CHEN Zhao-Qiang, Ph. D. candidate. His research interest is data quality management and artifact intelligent.

Background

Cardinality estimation is crucial for the cost-based query optimization, approximate query answering and other tasks. It is a very hard work and has been studied for nearly forty years. The traditional methods, like histogram, sampling, are working well when database has a small amount of data. But with the era of big data coming, these methods either may cause many mistakes by some wrong assumptions, or inefficiency to use, like scanning large amounts of data to do sampling. In recent years, machine learning, particularly deep learning has a great development and is used in many fields, e. g. image recognition, nature language processing. Database has an inherent advantage of handling data-driven applications, therefore it should plays a lead role in support this new wave. So how to use machine learning technology/AI to do cardinality estimation becomes a popular topic and produces many works.

The existing works which using machine learning for cardinality estimation can be classified by two perspectives. The first perspective is what problem to solve: just for single table with filter predicates or more complex query with both filter and join predicates. For single table problem, existing works mostly use the query-driven histogram and neural networks. However, query-driven histogram will become inefficiency and ineffective with the curse of dimension, and it needs many label data to train neural networks that can compete with original estimator. For complex query with join, they construct very complex deep learning models like CNN, which are very time-consuming to train. The second perspective is using one model or more smaller models to do cardinality estimation. There have some works trying to use one deep learning model for all queries, which is very hard to interpret and train. On the contrary, some works use many smaller models for different queries which can get better prediction just using a little training time and can give us a good understanding why it works. However, they are mainly used in the scenarios where query and data distributions are static.

In this paper, we propose using some smaller online learning models to handle complex queries with join predicates. Our method can do better with a little training data and self-adapt to shift of query and data distributions.

The work was supported by the National Key Research and Development Program of China (2018YFB1003400), the National Natural Science Foundation of China under Grant Nos. 61732014, 61672432, the Fundamental Research Funds for the Central Universities (Program No. 3102019DX1004), the Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2018JM6086).