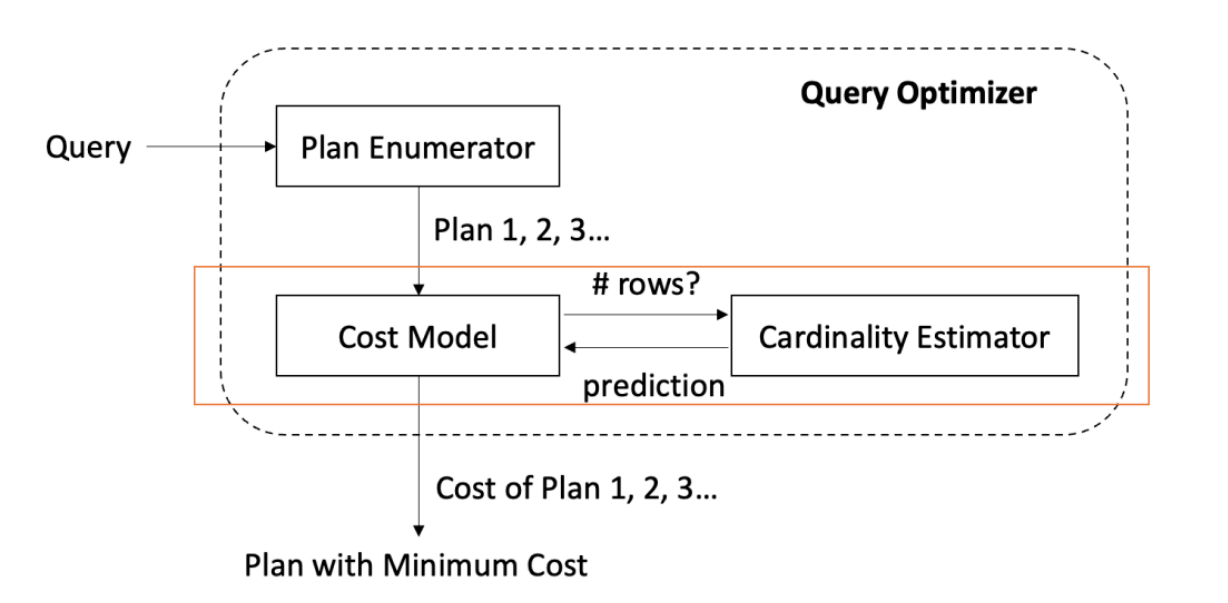
大数据计算基础实验

1. **绪论**

基数估计是查询优化的基础，但是这个问题长时间来未被解决。最近人们发现深度学习模型或许可以替换基数估计器[1, 2, 3, 4, 5, 6]。Are We Ready For Learned Cardinality Estimation?这篇文章由三部分组成：第一部分，在无数据更新的静态环境，数据集配置相同的情况下，比较5个深度学习模型与9个传统估计模型在真实场景下的效果。研究发现深度学习模型精确度高于传统模型，但是会产生高昂的代价；第二部分，研究人员测试这些深度学习模型是否适用于频繁更新的动态环境。实验发现它们反应太慢，而且由于各种原因，它们有很大的偏差。如果数据更新得不那么频繁，这些模型的效果会稍好，但是他们之间没有明显差别；第三部分，文章研究这些模型什么时候会出错，实验发现模型的表现受相关性，数据偏倚和域的大小变化的影响，但是模型的行为很难诠释和预测。文章给出两个研究方向：1.研究深度学习模型代价的控制；2.研究提升深度学习模型的置信度。

文章注重于单表基数估计，即估计表中符合查询谓词的元组个数。文章第一部分介绍研究背景，综述全文内容；第二部分展示了基数估计的研究；第三部分明示通用研究设置；第四部分研究深度学习模型与静态环境；第五部分研究深度学习模型与动态环境；第六部分研究模型出错的情况；第七部分给出未来研究方向与讨论；第八部分给出多表基数估计的想法；第九部分展示相关工作；第十部分进行总结。



1. **现有系统/方法架构**
2. 问题陈述

考虑一个具有n个属性关系和在关系R上具有d个谓词的查询

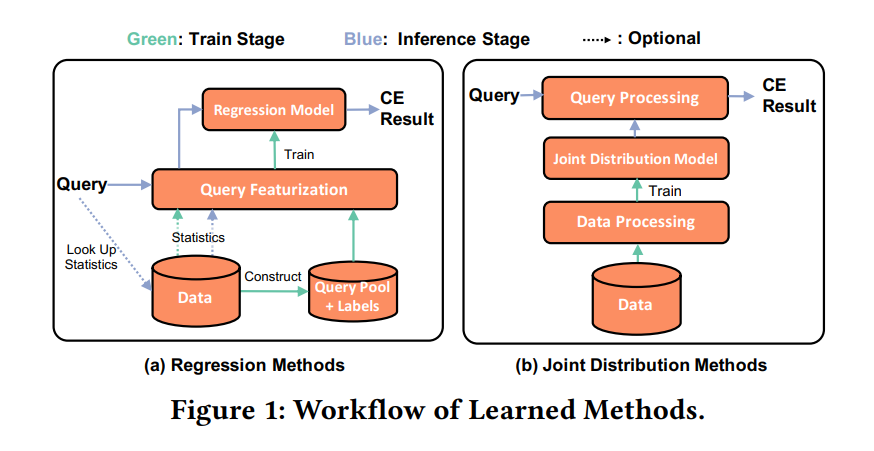
SELECT COUNT(\*) FROM R

WHERE AND … and

其中的d个谓词可以是等值谓词和开放范围谓词或者是封闭范围谓词。

1. 分类方法

ML for CE(最大似然法的基数估计)的思想早已有之。最新进展是采用更先进的最大似然模型如深度神经网络(DNN)，梯度增强树，和积网络和深度自回归网络，这些也被我们称为学习模型方法。相比之下，我们称直方图或者经典的最大似然模型如核密度估计和贝叶斯网络为传统方法。



图一给出了新方法的分类准则：回归方法和联合分布方法。回归方法，也即查询驱动方法，将基数估计问题抽提为回归问题，其中心为将查询和基数估计通过特征向量建立起映射关系，即(query 🡪 feature\_vector 🡪 CE\_result)，其中特征向量充当映射函数作用。同分布方法，亦即数据驱动方法，将基数估计问题转化为同分布问题，从数据表中构建联合分布，从而估计基数。

1. 回归方法

首先构建一个查询池，将查询语句作为特征(query fearturization module)，将初始查询结果作为查询标签，从而构建回归模型。查询时只需将查询语句提取特征后放入回归模型中即可获得预测基数。更新数据时，需要更新查询池和标签集，提取新的特征向量，从而训练更新回归模型。

回归模型有四种：MSCN，LW-NN，LW-XGB，DQM-Q。

A. Kipf, et al.Learned cardinalities: Estimating correlated MSCN[7]

joins with deep learning.CIDR 2019

A.Dutt, et al. Selectivity estimation for range predicates using LW-NN[8]

lightweight models. VLDB 2019 LW-XGB[8]

S.Hasan, et al. Deep learning models for selectivity DQM-Q[9]

estimation of multi-attribute queries.SIGMOD 2020.

1. **MSCN**(multi-set convolutional network)，多集卷积网络，支持连接基数估计。MSCN将查询语句表示为包含表，连接关系，谓词模组这三个模块的特征向量，每一模块均由两个神经网络构成，不同模块输出串联并投入最终输出层（两层）。MSCN用具体样本充实训练数据。MSCN对样本计算谓词，并将位图加入特征向量，位图每一位指示样本中的元组是否满足谓词。
2. **LW-XGB/NN**，它的特征向量由范维特征和基数估计(CE)特征组成。范围特征代表一系列谓词范围:<a1, b1, a2, b2, …, an, bn>，CE特征表示启发式估计器(认为每一列数据相互独立)。CE特征可以简单地从数据库系统中统计出来。LW-NN(LW-XGB)用产生的特征训练神经网络(梯度增强树)。相比MSCN使用q均差评价模型，它采用标签对数函数的均方误差相当于q差的几何平均数。
3. **DQM-Q**采用独热向量处理分类列，并通过自动离散化将数字属性视为分类属性。当现实世界的查询工作负载可用时，DQM-Q能够扩充训练集并使用扩充集训练模型。
4. 同分布方法

同分布方法将数据变换成易训练同分布模型的状态。当给出查询语句时，模型生成一个或多个结果，并在比较之下给出最终基数估计。

同分布模型有三种：Naru, DeepDB, DQM-D

S.Hasan, et al. Deep learning models for selectivity

estimation of multi-attribute queries.SIGMOD 2020. DQM-D[9]

z.Yang, et al. Deep unsupervised cardinality estimation.

VLDB 2019 Naru[10]

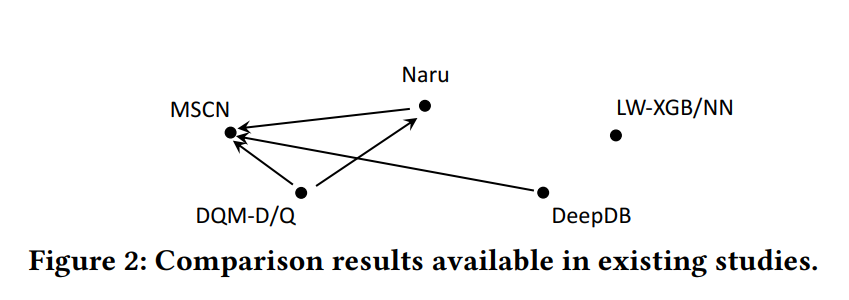
B.Hilprecht, et al. Deepdb: Learn from data, not from queries! DeepDB[11]

VLDB 2020

1. **Autoregressive Model**，它将联合分布通过公式

转换为条件概率分布，而用最新的深度自回归模型近似看作联合分布。联合分布可以直接返回点查询的结果，对于范围查询则需要新的技术。Naru使用了一种称为渐进采样的新技术，该技术根据条件概率分布的每个内部输出逐列采样值。DQM-D采用最初为蒙特卡罗(Monte-Carlo)多维积分设计的算法，该算法进行多阶段采样。在每个阶段，它根据前一阶段的结果，按照样本点对查询基数的贡献比例选择样本点。

1. **Sum-Product Network**，其关键技术是递归地将表划分为不同的行列簇。其中，使用Kmeans算法划分行簇，使用随机依赖系数(Ramdomized Dependency Coefficients)划分列簇。

这些方法存在它们的局限。一，它们没能够相互比较，给出最佳评测。文章将有关联的方法相连接。以此绘制了一个图：

可以看到这张图很稀疏。在这张图中，LW-XGB/NN是独立发展的，DeepDB和Naru同是新方法，却没有相关联性。二，这些方法没有给出规范的数据集使用和生成工作负载。三,现存工作主要聚焦于静态环境下的工作效果，而动态环境也是实际存在的。

1. **实验装置**

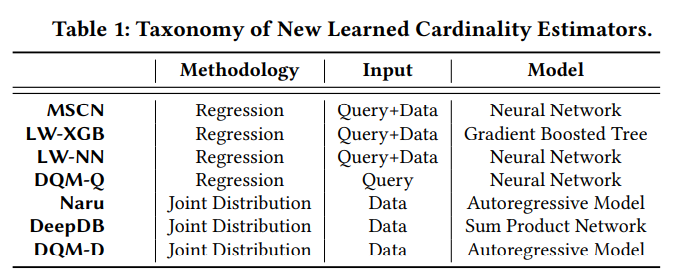
实验研究评估了不同设置下的学习基数估计量。在这一部分将描述所有实验中使用的一般设置。

1. 评估指标

使用q-error作为衡量评估结果质量的准确性指标。q-error是一个对称度量它计算估计值与实际基数不同的因素。

例如，如果一个查询的真实基数是10而估计基数是100，那么。

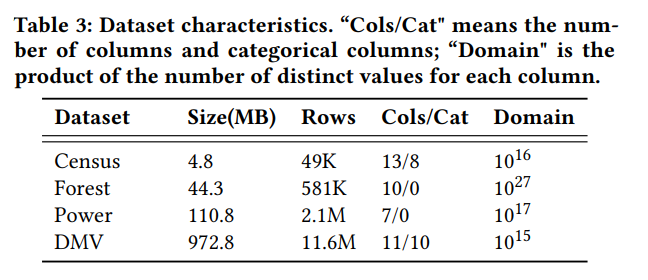
1. 学习方法和实施



如表1所示，最近发表的关于学习方法的论文有5篇:Naru 、MSCN 、LW-XGB/NN 、DeepDB 和DQM。文章将DQM排除在研究之外，因为它的数据驱动模型与Naru具有相似的性能，并且它的查询驱动模型不支持我们的工作负载(DQM的作者证实了这一点)。

对于naru3和DeepDB4，研究人员采用了作者发布的实现，并做了一些小的修改，以支持我们的实验。文章选择ResMADE作为Naru的基本自回归构造块，因为它既高效又准确。对于MSCN，由于原始模型支持连接查询，它需要额外的输入特性来指示不同表上的不同连接和谓词。为了确保单表基数估计的公平比较，研究人员修改了原始代码5，只保留了表示谓词和限定样本的特性。根据LW-XGB/NN的原始文献中的描述，研究人员实现了LW-XGB/NN的神经网络(LW-NN，在PyTorch 上)和梯度增强树(LW-XGB，在XGBoost 上)方法，并使用Postgres在单个列上的估计结果来计算CE特征。

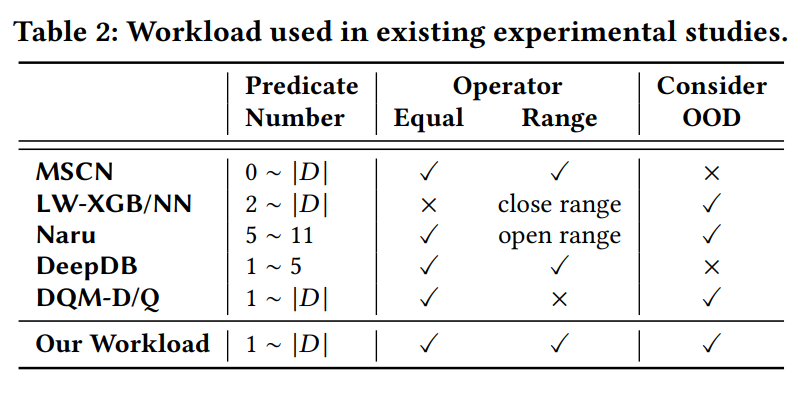
1. **静态环境测试**
2. 数据集



文章采用上述四个数据集，原因是：①这些数据集规模不同，分类列和数字列之间的比率不同 ②都是相关领域已使用过的数据集

1. 工作负载

使用统一工作负载生成器，目的是能够覆盖现有学习方法中使用的所有工作负载设置(表2)

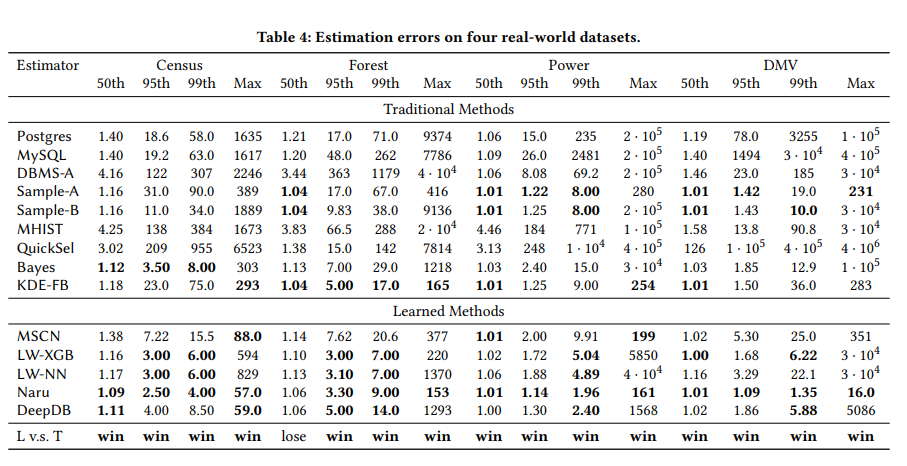


1. 传统技术

文章比较了各种传统的技术，这些技术要么已被数据库系统使用，要么被报道最近达到了最先进的性能。

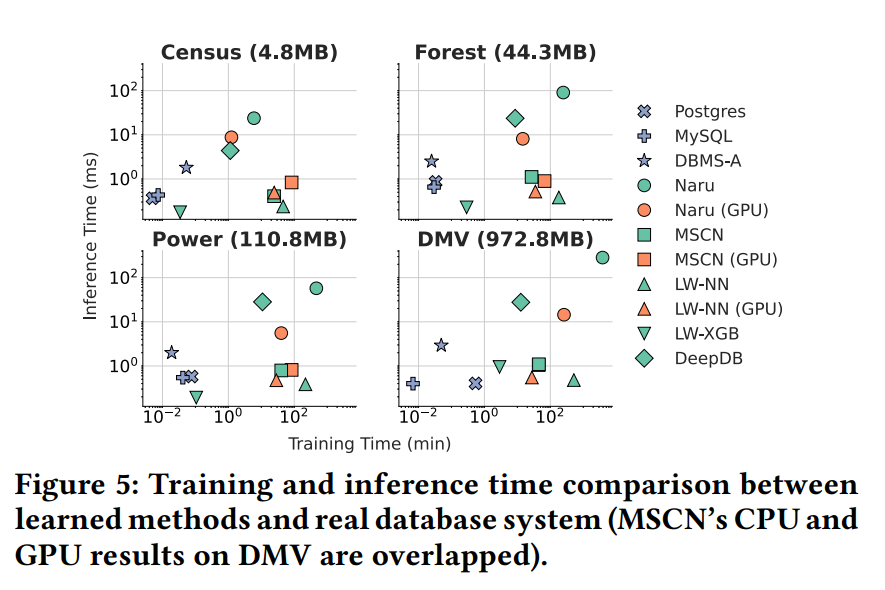
1. Postgres、MySQL和DBMS-A用于表示真实数据库系统的性能。DBMS-A是领先的商业数据库系统，它用简单的统计和假设快速估计基数。为了让模型达到最佳的准确率，研究人员将直方图桶的数量设置为上限(Postgres为10, 000，MySQL为1024)。对于DBMS-A，文章创建了几个多列统计，以便用直方图覆盖所有列。
2. 样本A、样本B显示估计量采用抽样。样本A使用统一的随机样本，当样本中没有元组满足所有谓词时，它会导致很大的错误。因此，样本B假设在零元组情况下每个谓词之间是独立的。对于这两种方法，文章从每个数据集中抽取1.5%的元组。
3. MHIST 在整个数据集上构建多维直方图。研究人员选择Maxdiff作为分区约束，值和面积作为排序和源参数，迭代运行MHIST-2算法，直到它达到数据大小的1.5%。
4. QickSel 代表了查询驱动的多维概要方法的性能。它利用查询反馈，用均匀混合模型对数据分布进行建模。
5. 贝叶斯方法显示了概率图形模型方法的估计结果，它使用渐进采样来估计范围查询，并显示出非常有希望的准确性。
6. KDE-FB 代表了用核密度模型模拟数据分布的性能。它通过查询反馈优化带宽来改善朴素KDE。
7. 实验所得的精确性

在每个数据集测试10k个查询语句，表四给出q-error比较结果。



在传统方法中粗体字表示可达最小误差范围，在学习方法中则表示强调学习方法。通过”win”与”lose”来记录比较两种方法的效果。

对于这些相比传统方法更精准的学习方法来说，它们也有自己的问题。图五给出上述方法在训练时间和推理时间的比较



1. 训练时间

对于学习的方法，文章记录用于训练表4中报告的模型的时间。对于数据库系统，则记录运行统计信息收集命令的时间。

数据库系统可以在几秒钟内完成对所有数据集的统计，而学习的方法一般需要几分钟甚至几个小时，这取决于底层的机器学习模型。基于梯度增强树的LW-XGB是学习速度最快的方法。当使用的树较少时，如在人口普查和电力数据集中，它可以像一些数据库管理系统一样快。DeepDB是第二快的，构建SPN模型仅需要几分钟，不过这也受输入样本量和停止条件的影响。采用神经网络的方法一般需要较长的时间，而Naru的训练时间高度依赖于数据大小和平台。有了GPU，在普查上只需要1分钟，而在车管所上需要4个多小时，在CPU上这个时间会慢5倍到15倍。GPU加速也会影响LW-NN，在所有数据集上完成训练大约需要30分钟，但在CPU上的时间可能会延长20倍。另一方面，MSCN在两台设备上表现出相似的训练时间，在小数据集上GPU甚至比CPU慢3.5倍。这是因为MSCN需要处理条件工作流来最小化其损失(平均q误差)，这在GPU上变得更慢，并且当模型更小时，影响变得更加明显。

在训练时间和模型精度之间有一个折衷。以迭代方式训练的神经网络方法(Naru、MSCN和LW-NN)会以较少的训练迭代产生较大的误差。一些模型可以用很少的迭代获得类似的性能，例如，仅仅使用原来的20%时间，便可以在DMV数据集上训练一个Naru模型，而性能仅略有下降。但是，即使模型只在GPU上运行1 epoch，它仍然会比数据库系统慢得多。

1. 推断时间

文章通过逐个发出查询来计算10K测试查询的平均推理时间。图5给出了运行结果。对于数据库系统，文章通过它们返回执行计划(不执行)的延迟来近似拟合表示时间，但由于解析和绑定等其他开销，该时间应比实际基数估计时间长。尽管如此，所有三个数据库管理系统都可以在1或2毫秒内完成整个过程。学习方法的推理时间取决于底层模型。查询驱动的方法(MSCN和LW-XGB/NN)很好，可以实现与DBMS相似或更好的延迟(但是DBMS的结果包括其他开销)。这是因为它们采用了通用回归模型，直接对查询空间建模，并且在实现方面也进行了很好的优化。另一方面，剩下的方法采用更专业的模型，速度要慢得多。DeepDB中的SPN模型在三个较大的数据集上需要大约25毫秒，在人口普查中平均需要5毫秒。Naru的推理过程包括一个渐进采样机制，需要运行模型上千次才能得到准确的结果。它的总时间对运行的设备比较敏感，在GPU上需要5毫秒到15毫秒，CPU则会慢20倍。

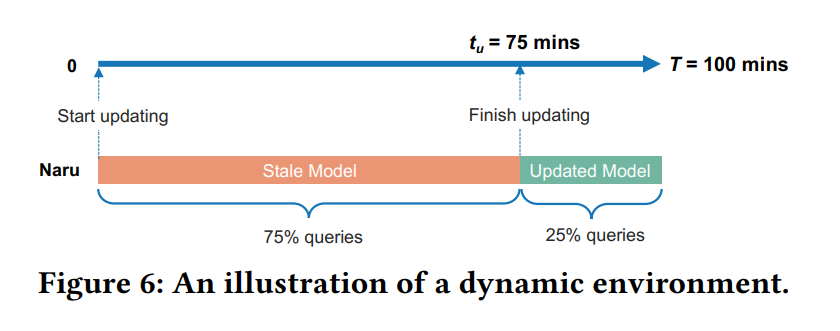
1. 超参数的调整

超参数调整是学习方法的另一个成本。表4中所示的学习模型代表具有最佳超参数的模型。然而，如果没有超参数调整，学习模型的表现可能会非常糟糕。在文章的实验中，研究人员发现对于相同的方法，在具有不同超参数的模型中，最大和最小最大q误差之间的比率可以高达105。

尽管超参数调谐对于高精度至关重要，但它是一个非常昂贵的过程，因为它需要训练多个模型来找到最佳的超参数。例如，如图5所示，Naru花了4个多小时用GPU在DMV上训练单个模型。如果训练五个模型，那么Naru需要花费20多个小时(几乎一天)进行超参数调整。

1. 总结
2. 在研究的实验中，新的学习估计器可以提供比传统方法更准确的预测，在学习方法中，Naru表现出最稳健的性能。
3. 就训练时间而言，除了LW-XGB之外，新的学习方法在数量上可能比DBMS产品慢。
4. 基于回归模型(MSCN和LW-XGB/NN)的新学习估计器在推理时间上可以与数据库系统竞争，而直接对数据进行联合分布建模的方法(Naru和DeepDB)需要更长的时间。
5. GPU可以加快一些新学习的估计器的训练和推理时间，但它不能使它们像DBMS产品一样快，有时会引入开销。
6. 对于采用基于神经网络的估计器来说，超参数调整是一个不可忽视的额外成本。
7. **动态环境测试**
8. 动态环境

动态环境注重于模型精度和更新速度。图6给出了一个模型在动态环境下的更新过程。模型从0时刻开始更新，到完成更新，若假设有n个查询独立均匀分布地进行，则前个查询语句是使用未更新的模型，后面剩余的查询语句使用的是更新过后的语句。



1. 数据集、工作负载和度量标准

数据集与静态环境相同。为了体现更新，又将20%的新数据附加到原始数据集中，并将工作负载生成方法应用到一般10K测试查询的更新数据中。也就是说，测试工作负载包含10K查询。这些查询将均匀分布在[0, T]中。这里，T是动态环境中的一个参数。直观地说，它表示数据更新的“频率”。例如，如果数据每100分钟定期更新一次，那么我们可以设置T=100分钟。我们报告了10k个查询的第99个百分位q-error。表4中显示了各种错误度量（50%、95%、99%和最大错误）。根据表4的结果可得之，与传统方法相比，学习方法在较大误差（99%和最大误差）上的改善更大。因为最大误差对异常值很敏感，所以文章选择了99%的误差。为了进一步减轻异常值的影响，在实验设置中使用了大量查询（10k个查询）进行测试。这意味着99%的错误是第100个最大的错误，因此它不是由一些异常查询控制的。

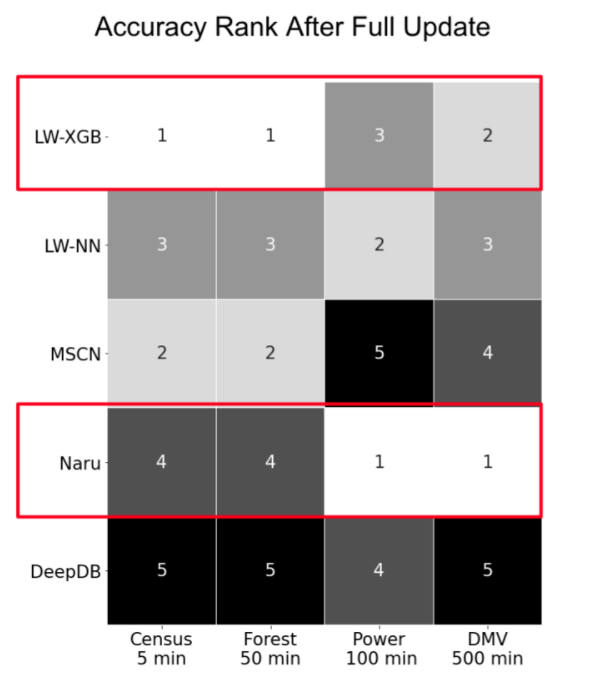
1. 数据更新

文章确保附加的20%新数据与原始数据集具有不同的相关性特征。否则，未更新的模型可能依然会表现良好，并因此不需要更新模型。为了实现这一点，研究人员创建了原始数据集的一个副本，并按升序分别对每一列进行排序，从而使每一对列之间的Spearman秩相关性达到最大，然后从复制的数据集中随机选取20%的元组，并将它们附加到原始数据集中。

1. 模型更新

Naru和DeepDB接受数据培训。正如他们在论文中所描述的，Naru更新了一个历元，而DeepDB则通过在树模型中插入一个小样本(1%)的附加数据来更新。MSCN和LW-XGB/NN使用查询结果作为训练数据。由于原始MSCN文件中未讨论更新过程，因此采用LW-XGB/NN的MSCN更新过程。生成培训工作负载后，文章使用一个样本(原始数据集的5%)来更新查询标签。LW-XGB和LW-NN最初使用2K和16K查询进行相应更新。研究实验为MSCN分配10K查询，作为相当大的训练数据。

1. 方法比较



1. 结论
2. 学习方法模型赶不上快速的数据更新。在动态环境中，由于不同的原因，MSCN、LW-NN、Naru和DeepDB会返回较大的误差。
3. 在所学的方法中，没有明确好使的模型。Naru在数据更新不频繁的情况下表现最好，而LW-XGB在更动态的环境下表现最好。
4. 在更新时间上，DeepDB是最快的数据驱动方法，LW-XGB是最快的查询驱动方法。
5. 学习方法的更新时间和准确性之间存在权衡。在实践中权衡利弊并不容易，需要在这个问题上进行更多的研究。
6. GPU能够提高性能，但不一定。为了从GPU中受益，设计一个好的策略来处理模型更新是很重要的。
7. **学习方法什么时候出错**
8. 数据集

文章通过改变三个关键因素来生成具有两列的数据集：分布（第一列）、相关性（两列之间）和域大小（两列的）。每个数据集包含100万行。第一列由scipy中的genpareto函数生成，它可以生成从均匀分布到非常倾斜的随机数。第二列基于第一列生成，以控制两列之间的相关性。域的大小取决于数据集的规模，

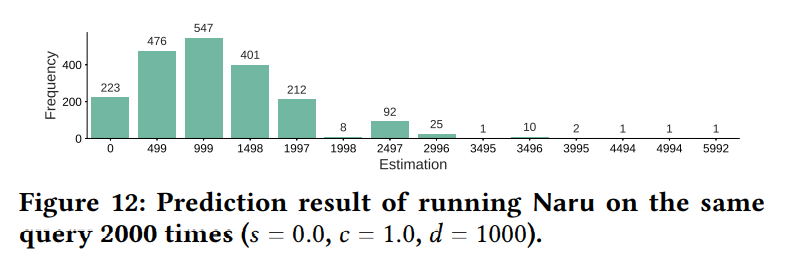
1. 工作负载

由于实验的目标是研究学习方法出错时的情况，因此文章为所有查询独立地从每个列的域（OOD）中生成中心值，以便探索整个查询空间并找到尽可能多的硬查询。其他工作负载生成设置与上文相同

1. 超参数的调整

采用推荐默认超参数（RDC阈值=0.3，最小实例切片=0.01），用于DeepDB，并将LW-XGB的树大小固定为128。对于神经网络模型，随机选取三个超参数设置，大小为1%，并选择一个始终报告良好结果的设置。

1. 发现
2. 估计结果不单调，即在集合存在逻辑包含关系，查询范围增大的情况下，查询结果可能会变小。
3. 估计结果不稳定

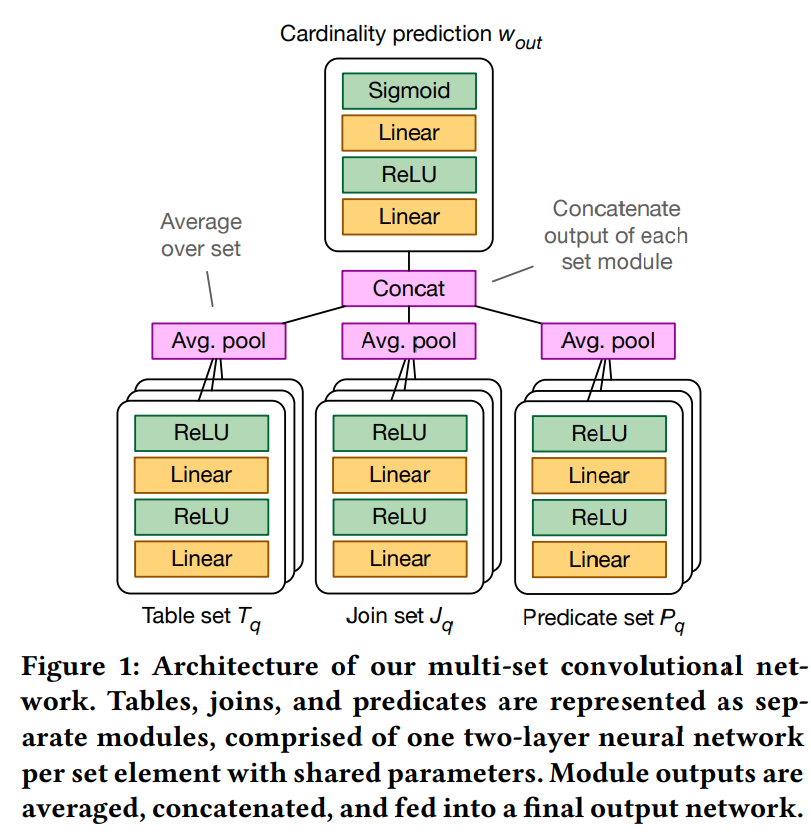


同一个查询的估计结果范围从 0 到 6000，这样的估计结果会导致系统的表现不稳定，从而影响人们对系统的信任。

1. 结论
2. 所有新学习的估计量倾向于在更相关的数据上输出更大的误差，并且当两列是函数相关的时，最大q-error跳跃相当显著。
3. 对于更偏斜的数据或具有更大域大小的数据，不同的方法有不同的反应。这可能是由于模型选择、输入特征和损失函数的不同。
4. 文章为基数估计器提出了五条规则，并发现除了DeepDB之外，所有新学习的模型都违反了这些规则。
5. 当在生产中部署时，在新学习的估计器中使用的模型的不透明性在可调试性、可解释性、可预测性和可再现性方面可能很麻烦。
6. **MSCN方法复现**

该方法学习预测数据中的（连接交叉）相关性，并解决了基于抽样技术的一些弱点。该方法允许使用集合（例如，（A B） C和A （B C）表示为{A，B，C}）来表示查询特征，因此，模型不会浪费任何存储查询特征的不同排列（所有排列都具有相同的基数，但成本不同）的能力，这将生成更小的模型和更好的预测。连接枚举和成本模型特意留给查询优化器处理。

使用真实世界的IMDb数据集评估MSCN，其通过使用大约3mib的（可配置的）低占用空间大小实现的（而基于采样的技术可以访问覆盖整个数据库的索引）。



模型简介

标准的深层神经网络结构，如卷积神经网络（CNN）、递归神经网络（RNN）或简单多层感知器（MLP），不直接适用于数据集中数据结构，需要序列化，即将数据结构转换为有序的元素序列，这是有显限制的，因为模型必须花费一定的能力来学习发现原始表示的对称性和结构。例如，它必须学会在由多个大小不同的集合组成的数据结构中发现不同集合之间的边界，并且集合序列化中元素的顺序是任意的。假设我们事先知道数据的底层结构，我们可以将这些信息记录到我们的深度学习模型中，同时有效地为其提供一种归纳偏差，有助于将其推广到相同结构的不可见实例，例如，具有不同数量的元素的集合的组合在训练过程中不可见。

但是MSCN不同。MSCN的思路源于Deep Sets, 在Deep Sets上，集合S中元素对具有置换不变性的任何函数f(S)都可以分解为的形式，这里和是自己选择的函数。使用简单的全连接多层神经网络（MLP）对函数和进行参数化，并依靠其函数逼近特性学习任意集合S的灵活映射f(S)。对每个集合元素单独应用可学习映射（使用共享参数）类似于1×1的概念卷积，而该方法常用于CNN中的图像分类。

模型的查询表示由多个集合组成，这使我们对MSCN模型架构进行以下选择：对于每个集合，我们学习一个集合中每个元素特定的神经网络。然后，该集合的最终表示由其元素的各个变换表示的平均值给出，即。 我们选择一个平均值（而不是一个简单的和）来简化对集合S中不同数量元素的概括，否则信号的整体幅度将根据S中元素的数量而变化。

最后，我们通过串联合并各个集合表示，然后将它们传递给最终输出MLP: 。MLP模块使用两次全连接的激活层ReLU，输出层使用sigmoid函数。我们对隐藏层使用ReLU激活函数，因为它们显示出很强的经验性能，并且评估速度很快。MLP的所有其他表示向量和隐藏层激活被选择为维度d的向量，其中d是超参数，通过网格搜索在单独的验证集上进行优化。

我们将目标基数规范化如下：首先取对数以更均匀地分布目标值，然后使用从训练集获得的对数化后的最小值和最大值将其规范化为区间[0, 1]内的值。规范化是可逆的，因此我们可以从预测结果中恢复非标准化的基数。

我们训练我们的模型以最小化平均q误差（q≥ 1). q误差是估计值与真基数之间的因子（反之亦然）。我们使用Adam优化器进行训练。

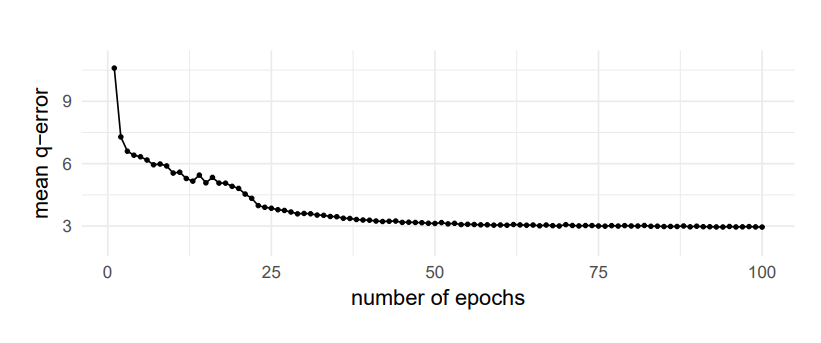
模型性能评价

使用三种不同的查询工作负载：

1. 由同一个查询生成器生成的合成工作负载作为训练数据（使用不同的随机种子），5000个唯一的查询包含（合取）相等谓词和范围谓词（非键列上有零到两个连接），
2. 包含500个查询的合成工作负载量表，旨在显示模型如何推广到更多连接；
3. JOB-light，一个从连接顺序基准(JOB)衍生的工作负载，包含原始113个查询中的70个。与JOB不同，JOB-light不包含任何字符串或析取谓词，只包含具有一到四个联接的查询。JOB light中的大多数查询都在维度表属性上具有相等谓词。唯一的范围谓词是生产年。表1显示了关于三个查询工作负载中联接数量的查询分布。合成工作负载中的非均匀分布是由于我们消除了重复查询造成的。

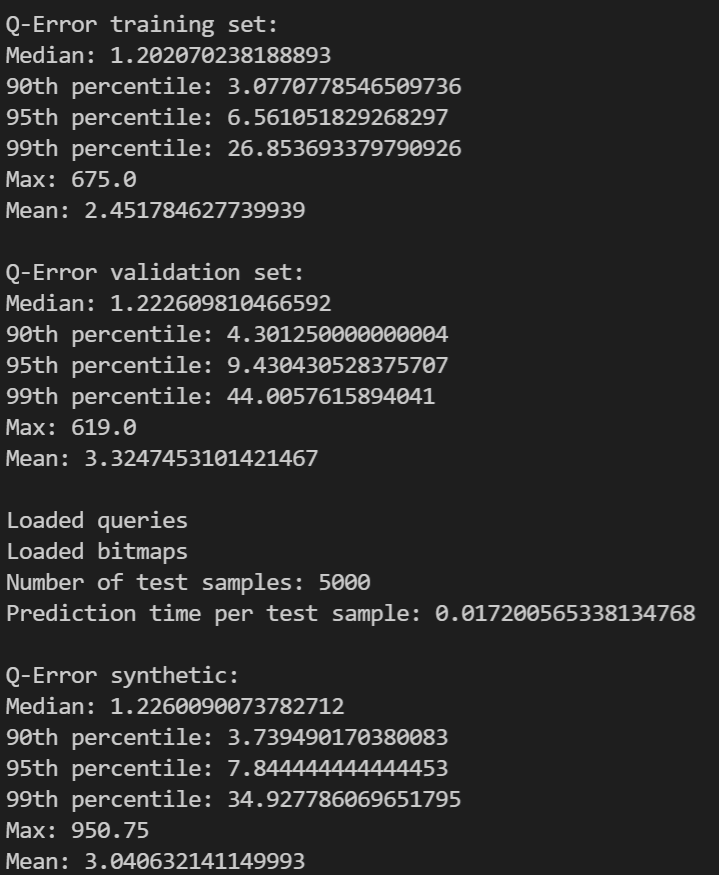
RS在物化样本上执行基表谓词以估计基表基数，并假设独立于估计联接。如果一个连接谓词没有符合条件的样本，它会尝试单独评估连接，并最终返回到使用不同值的数量（具有最选择性连接的列的数量）来估计选择性。

接下来，使用默认的超参数分析MSCN的训练、推理和空间成本。下图显示了验证集误差（验证集中所有查询的平均q误差）是如何随着时间的增加而减小的。该模型需要不到75次的通过（超过90000次训练查询），才能在10000次验证查询中收敛到大约3次的平均q误差。一次平均100次的训练跑（分三次跑）需要将近39分钟。MSCN模型的预测时间大约为几毫秒，包括PyTorch框架引入的开销。理论上（忽略Pyrotch开销），使用深度学习模型（如前所述）的预测主要由矩阵乘法控制，可以使用现代GPU进行加速。因此，模型的性能优化实现能够实现非常低的预测延迟。由于MSCN包含采样信息，因此端到端预测时间将与（每个表）采样技术的数量级相同。对于MSCN（无样本）、MSCN（有样本）和MSCN（位图），我们模型的大小（序列化到磁盘时）分别为1.6 MiB、1.6 MiB和2.6 MiB。

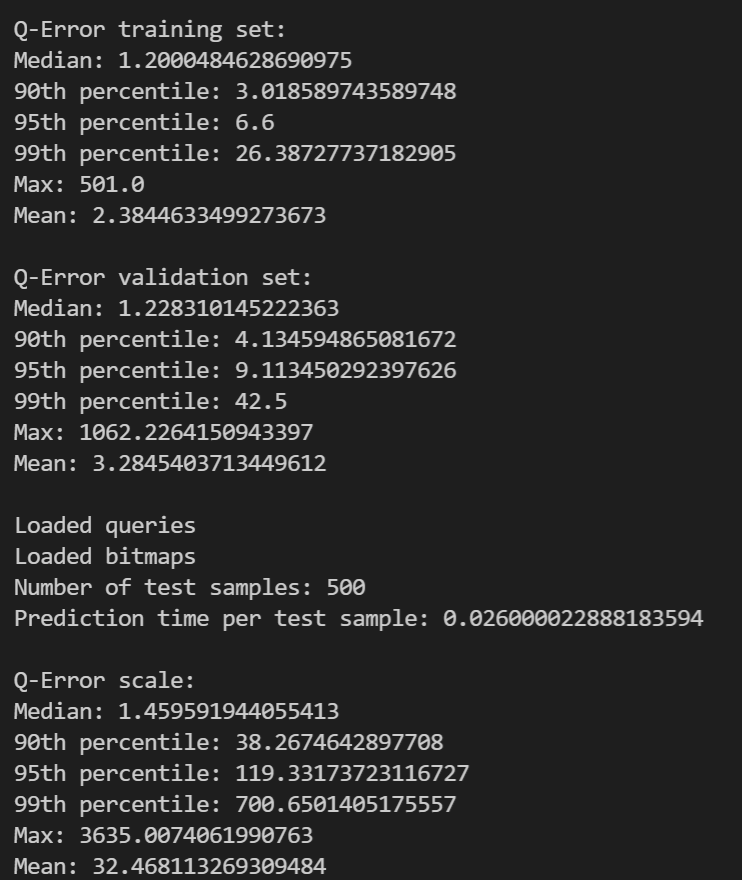


除了优化平均q-误差外，MSCN还探索了使用均方误差和几何平均q-误差作为优化目标。均方误差将优化预测基数和真实基数之间的平方差。由于现阶段研究更感兴趣的是最小化预测基数和真实基数（q-误差）之间的因子，并使用该度量进行评估，因此直接优化q-误差会产生更好的结果。优化q误差的几何平均值会使模型不太重视严重的异常值（这会导致较大的误差）。虽然这种方法一开始看起来很有希望，但结果证明它不如优化平均q误差那么可靠。

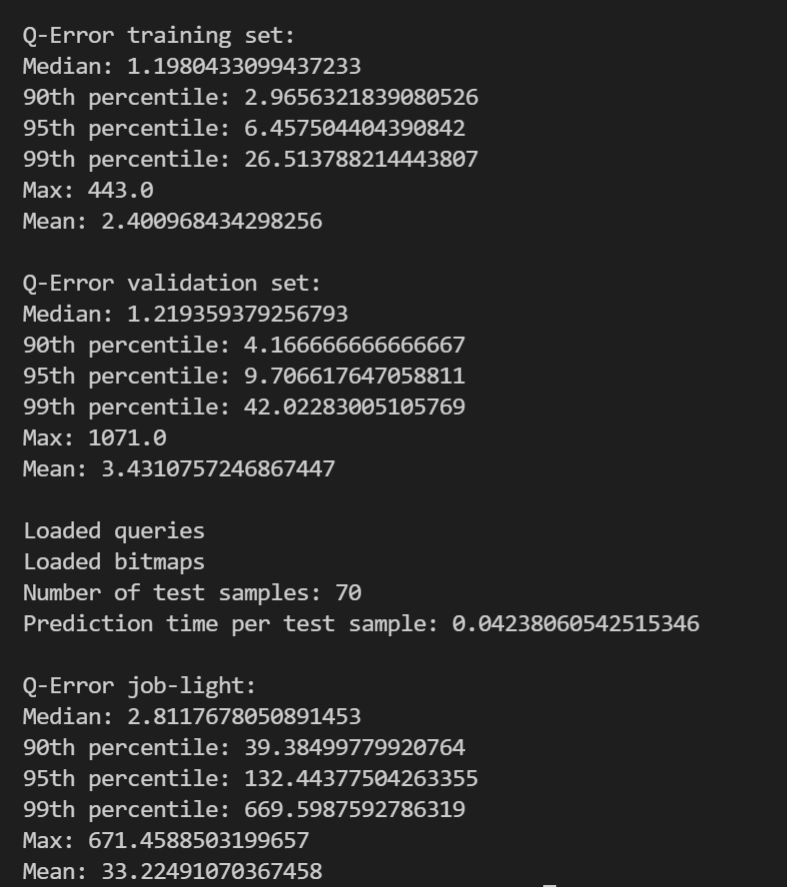
下图是三种工作负载下的运行情况



synthetic



Scale



Job-light

1. **总结**

JOB light中的大多数查询都在维度表属性上具有相等谓词。考虑到MSCN在=、<和>谓词之间的分布是一致的，它的性能相当好。此外，JOB light还包含许多查询，这些查询对生产年份具有封闭范围谓词，而培训数据仅包含开放范围谓词。除此之外，JOB-light还包括五个查询，它们超过了MSCN培训时使用的最大基数。如果没有这些查询，第95百分位q-error为115。总之，该实验表明MSCN可以推广到分布不同于训练数据的工作负载。

**参考文献**

1. A. Jindal, S. Qiao, H. Patel, Z. Yin, J. Di, M. Bag, M. Friedman, Y. Lin, K. Karanasos, and S. Rao. Computation reuse in analytics job service at microsoft. In Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018, pages 191–203. ACM, 2018.
2. T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis. The case for learned index structures. In Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018, pages 489–504. ACM, 2018.
3. A. Kristo, K. Vaidya, U. Çetintemel, S. Misra, and T. Kraska. The case for a learned sorting algorithm. In Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020, pages 1001–1016. ACM, 2020.
4. W. G. Pedrozo, J. C. Nievola, and D. C. Ribeiro. An adaptive approach for index tuning with learning classifier systems on hybrid storage environments. In Hybrid Artificial Intelligent Systems - 13th International Conference, HAIS 2018, Oviedo, Spain, June 20-22, 2018, Proceedings, volume 10870 of Lecture Notes in Computer Science, pages 716–729. Springer, 2018.
5. J. Tan, T. Zhang, F. Li, J. Chen, Q. Zheng, P. Zhang, H. Qiao, Y. Shi, W. Cao, and R. Zhang. ibtune: Individualized buffer tuning for large-scale cloud databases. Proc. VLDB Endow., 12(10):1221–1234, 2019.
6. J. Zhang, Y. Liu, K. Zhou, G. Li, Z. Xiao, B. Cheng, J. Xing, Y. Wang, T. Cheng, L. Liu, M. Ran, and Z. Li. An end-to-end automatic cloud database tuning system using deep reinforcement learning. In Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019, pages 415–432. ACM, 2019.
7. A. Kipf, T. Kipf, B. Radke, V. Leis, P. A. Boncz, and A. Kemper. Learned cardinalities: Estimating correlated joins with deep learning. In CIDR 2019, 9th Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 13-16, 2019, Online Proceedings. www.cidrdb.org, 2019.
8. A. Dutt, C. Wang, A. Nazi, S. Kandula, V. R. Narasayya, and S. Chaudhuri. Selectivity estimation for range predicates using lightweight models. Proc. VLDB Endow., 12(9):1044–1057, 2019.
9. S. Hasan, S. Thirumuruganathan, J. Augustine, N. Koudas, and G. Das. Deep learning models for selectivity estimation of multi-attribute queries. In Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020, pages 1035–1050. ACM, 2020.
10. Z. Yang, E. Liang, A. Kamsetty, C. Wu, Y. Duan, P. Chen, P. Abbeel, J. M. Hellerstein, S. Krishnan, and I. Stoica. Deep unsupervised cardinality estimation. Proc. VLDB Endow., 13(3):279–292, 2019
11. B. Hilprecht, A. Schmidt, M. Kulessa, A. Molina, K. Kersting, and C. Binnig. Deepdb: Learn from data, not from queries! Proc. VLDB Endow., 13(7):992–1005, 2020.
12. M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In Advances in Neural Information Processing Systems, 2017.
13. D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.