|  |  |  |
| --- | --- | --- |
| 技术交底书  案号：必填  日期：必填 | 申请人：华为技术有限公司  深圳市龙岗区坂田华为总部办公楼  邮编：518129，总机：0755-28782828 | **华为商业秘密**  请勿泄漏 |

|  |  |
| --- | --- |
| 交底书名称 | 基于机器学习的索引推荐技术 |
| 发明人 | 王宏志、朱锐（必须与Anaqua中一致） |
| 第一发明人国籍及居民身份证件号码 | 210602197810111535 |
| 交底书撰写人 | 朱锐 |
| 撰写人办公电话 | 选填 |
| 撰写人手机号码 （急案必须填写） | 13069887146 |
| 撰写人 E-mail | wangzh@hit.edu.cn |



**华为技术有限公司**

版权所有 侵权必究

|  |  |  |  |
| --- | --- | --- | --- |
| **可选项：**产品相关性信息（请交底书撰写人填写） （请在勾选的项中填☒） | | | |
| □ | 我司产品可能采用的多个可选方案之一 | 预计该方案在我司产品中实现的时间 |  |
| ☐ | 我司产品确定将采用的方案 | 首款使用该技术的产品型号 | *（包括市场型号，如Ascend P1，及内部型号，如UXXX）* |
| 该产品的预计上市时间 |  |
| 该产品的参展信息 | *若有相关的展会信息，请填写预计的参展时间，及展会名称，如巴展，北展等。* |
| 展会上是否会公开本发明 | *请填写“是”或“否”* |
| ☐ | 暂无我司产品采用该方案，也无相关计划 | | |

|  |  |
| --- | --- |
| 撰写指导意见  （IPR填写） | *交底书的整体要求：能让代理人快速、清楚地理解本发明要解决的技术问题及相应的技术方案，不会因为逻辑不清晰或无关内容过多造成理解障碍，并在该基础上提供基本足够的技术方案实现细节而无需后续做大量补充/修改。*  *代理人注意事项：*   1. *在与申请文件不冲突的情况下，交底书中的内容都应在申请文件中体现；* 2. *在阅读技术交底书之后，撰写申请文件之前，给IPR打电话，沟通技术方案；* 3. *给发明人审核的专利申请文件，在实施例中增加批注（批注内容是：请发明人重点审核这部分），在权利要求中增加批注（批注内容是：发明人可以不审核）* |

**必选项：**请交底书撰写人针对如下交底书Checklist中各项内容进行确认，确认无误后，在“确认结果”中填写“OK”。

|  |  |  |
| --- | --- | --- |
| **交底书Checklist** | | |
| **检查项目** | **确认内容** | **确认结果** |
| 1 | 描述了现有技术存在的技术缺点，且分析了该技术缺点存在的原因 | OK |
| 2 | 描述了实施例相比于现有技术的改进之处 | OK |
| 3 | 实施例提供了必要的、可编辑的附图，且结合附图进行了描述 | OK |
| 4 | 描述了本发明应用的系统架构或应用场景以及涉及的物理器件 | OK |
| 5 | 正确、完整、充分详细地描述了每个实施例技术方案 | OK |
| 6 | 描述了实施例的技术效果，且详细分析了该技术效果产生的原因 | OK |
| 7 | 当存在多个实施例时，描述了多个实施例之间的区别 | OK |
| 8 | 清楚写明发明点和次要发明点 | OK |
| 9 | 对“缩略语和关键术语”按要求进行了描述；非通用术语在交底书正文部分进行明确定义 | OK |
| 10 | 全文对同一事物的名称描述一致 | OK |

**缩略语**（必选）

|  |  |  |
| --- | --- | --- |
| **英文缩略语** | **完整的英文表述/英文标准用语** | **中文表述/中文术语** |
| CNN | Convolutional Neural Network | 卷积神经网络 |
| RNN | Recurrent Neural Network | 循环神经网络 |
| LSTM | Long Short-Term Memory | 长短期记忆网络 |

**关键术语定义**（必选）

最佳索引：一种索引方案，以该种索引方案将数据存储到数据库中时，该数据集上的工作负载执行效率最高。索引推荐任务就是要寻找最佳索引，从而提高工作负载的执行效率。

# 相关技术背景以及最接近的现有技术

## 背景技术：用于方便读者理解本发明的技术背景（必选）

（参考文字描述）

本发明涉及通过学习模型替换数据库传统索引达到索引优化的效果。现有的索引结构的优化大多是基于数据最差的情况，并且是对读写相对均匀进行优化，对于一些特殊的情况，例如数据集就是1-100M顺序排列，那么这时候如果使用 B -Tree其实不是最优解，因为key值本身就可以作为偏移量使用，索引的时间复杂度和空间复杂度反而会因为B-Tree构建而增加。事实上索引可以视作为模型：B/B+ tree索引可以看作是将键映射到已排好序的数据中记录位置的模型，hash可以视作将键映射到无序数据中记录位置的模型，位图索引可以视作是判断数据是否存在的模型。由此假设我们能对现有的传统模型进行优化，那么就能够提高数据集上索引的执行效率。通常情况下，业界大部分采用的是B-Tree或其变种B+ Tree作为索引结构，传统索引采用通用的数据结构，没有利用已知数据的分布特点。虽然在现实生活中，并没有一种已知的数据模式分布可以遵循，且尝试去为这些分布不定的实际数据构建解决方案，求出分布模型的成本开销太大，但是考虑到利用机器学习的方法来反映数据模式，就可以以低成本自动构建针对已知数据集的模式索引。

17年时Jeff Dean在SystemML会议上就提出了学习索引优化问题，并从理论上分析了在哪些条件下，学习索引优于传统索引结构，以及设计一个好的学习索引的主要挑战，根据Jeff提出的学习索引框架 （LIF），递归模型索引（RMI）和基于标准误差的搜索策略，我们主要关注简单的全连接神经网络，因为它们的简单性，此外其他许多类型模型也是可能的，我们将在后续的工作中不断改进模型。虽然机器学习在索引优化方面具有巨大潜力，但还存在诸多待解决的问题（如：过拟合、索引的增删改等问题）。对于过拟合，若新索引的key依然满足累计分布函数则并不需要重新训练，直接插入到预测结果的位置即可。若数据分布会发生变化，则需要尝试在线学习（Online learning）的方法。对于数据更新频繁的系统，可采用delta-index技术增量更新learned index。

通过机器学习可以启发系统设计的优化和思考，探索出之前未曾发现的系统设计思路。在优化原理清晰、场景固定的情况下，显然由人加以解释和重新实现的传统数据结构在效率和稳定性上更胜机器学习方法一筹。但在数据分布等特征动态变化的场景，机器学习方法可以针对性优化和适应数据特征，理论上可以优于通用的算法和数据结构。

## 与本发明相关的现有技术（必选）

学习索引框架LIF（learning index framework）：LIF可以认为是一个索引合成系统， 它可以生成不同的索引配置并对它们进行优化，而且可以自动对它们进行测试。可以对简单的模型(线性回归模型)进行on-the-fly训练，对NN模型on-the-fly时需要Tensorflow，但是并不是直接使用Tensorflow，而是直接从模型中提取权值参数生成C++数据结构。。

递归式模型索引RMI（recursive-model indexes）：一个模型的层次结构，在每个阶段，模型以键作为输入，并基于它选择另一个模型，直到最后阶段预测位置。考虑不同模型的一种方法是，每一个模型都对键key的位置进行一定误差的预测，并使用该预测来选择下一个模型，该模型负责键key空间的某个区域，以较低的误差做出更好的预测。但是，递归模型索引不必是树，即处于同一层级的不同模型预测结果可能均为下一层级的某个模型。

基于标准错误的搜索策略（standard-error-based search strategies）

递归型模型的另一个优势是可以构建混合模型，比如，模型顶层使用一个小的RLU神经网络，因为它们可以模拟出复杂的数据分布。在模型底部可以使用大量的线性回归模型，或者直接使用B-tree。

Search Strategies and Monotonicity（查询策略和单调性）

范围索引一般需要确定一个upper bound(key) 或者[lower bound(key)]的位置来实现，learned范围索引也是如此。learned范围索引对比（Btree）的优势在于，它给出确定的key位置而并不是给出所在region。

两种查询策略：Model Biased Search策略和Biased Quaternary Search策略

Model Biased Search策略：和传统的二分法不同的是：第一个中间点设置为模型预测的值。

Biased Quaternary Search策略：四分法查询是将一个分裂点变为三个分裂点，将分裂点放入cache以提升性能。例如将分裂点初始化为pos - σ、pos和pos + σ

在所有的方法中都使用最大最小误差来确定查询范围。

### 1.2.1 现有技术一的技术方案（必选）

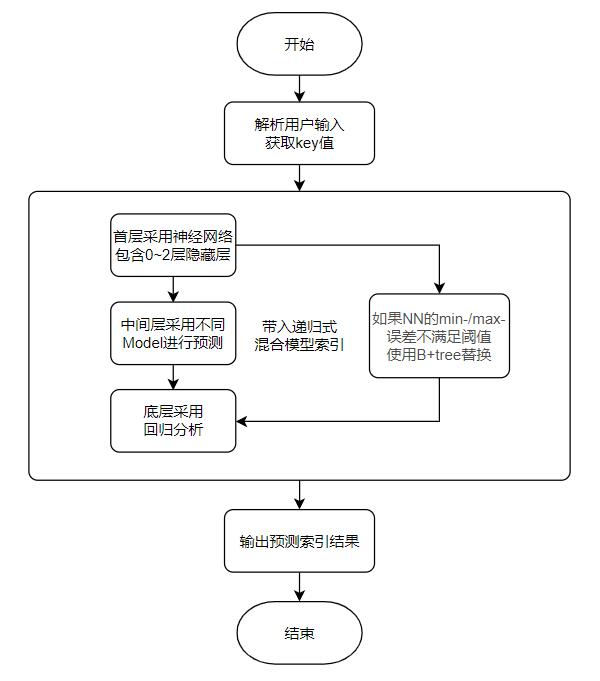
. 

图1 现有技术一的方法流程图（必选）

（参考文字描述）

如上图是现有技术一的方法流程图：

具体步骤如下：

步骤一：解析用户sql查询语句，获取查询的key值；步骤二：将步骤一获取的key值代入递归式模型并逐层计算；步骤三：最底层输出预测范围（pos-σ，pos+σ）作为回归分析输入；步骤四：通过回归分析确定模型结束本次查询。

### 1.2.2 现有技术一的缺点（必选）

（参考文字描述）

递归型模型可以构建混合模型，比如，在模型顶层使用一个小的ReLU神经网络，模拟出复杂的数据分布。在模型底部可以使用大量的线性回归模型，或者直接使用B-tree。以全体数据为数据集训练第一层模型，基于该顶级节点模型的预测，然后从下一阶段中挑选模型并添加落入该模型的所有键。在混合模型中如果NN的min-/max-误差不满足阈值，则使用Btree替换。混合索引允许我们将learned索引的最坏情况性能与B+树的性能联系起来。在极端的情况下，所有的模型会被替换成B+tree而成为一个完整的B+tree。由于学习索引的构建中范围和点索引模型都与CDF模型密切相关，所以估计CDF是非常重要的，如何对CDF进行最有效的建模仍然是一个有待进一步研究的问题。此外，许多简洁的数据结构都侧重于H0熵（对索引中的每个元素进行编码所需的位数），而学习型索引则试图学习底层数据分布以预测每个元素的位置。因此，学习的索引可能获得比H0熵更高的压缩率，但可能会以较慢的操作为代价。比如spline B-tree使用B-tree的每个叶子节点只存一个spline（即key和其位置），两个spline之间的数据用两点之间的直线来预测。这样一个简单的数据结构，很多时候效果相当于复杂的Learned Index，甚至更好。在Point Index 领域，Learned Index通过减少冲突实现的优化可以被bucketized cuckoo hashing轻松解决，该方法只是简单的将每个key同时hash到两个bucket而已。

# 2、本发明技术方案的详细阐述（发明内容）

## 2.1 本发明所要解决的技术问题（必选）

（参考文字描述）

综合现有技术存在的缺点，总结本发明要解决的技术问题如下：

（一）现有的索引推荐技术，需要在推荐出索引之后，利用数据库的查询计划去评估索引性能，然而在利用查询计划的过程中，不可避免地需要额外的时间开销和数据转换，这会在一定程度上降低速度，并且导致一些信息损失，导致索引的适用性降低。

（二）在现有的研究中，通过机器学习技术来实现索引推荐，这个过程从训练开始所需要的时间难以明显缩短。而从整体性能上考虑，受模型选择的影响，学习索引所需的空间开销有较为庞大，缩小了这一技术的适用范围。

（三）索引推荐面向的数据量比较庞大，传统的B-树如果不加以改造，难以保证以更快的速度进行检索。同时，学习模型的输入范围也需要进行逐层分块削减，以便加快搜索。我们采用决策树的方式进行了改造，显著缩小了在最终叶结点中需要检索的数据范围。

（四）现有的索引推荐模型大多都需要使用者提前设置好当前场景下的可行索引集合，然后根据该集合进行索引推荐。在现实生活中往往需要大量的调研和统计结果才能得到该集合，本发明从数据库本身出发从零开始进行索引推荐

## 2.2 本发明的主要发明要点概述（必选）

（参考文字描述）

结合上述问题，本发明的主要创新技术点总结如下：

（一）本发明将索引推荐问题处理的数据规模逐层降低，减小了由于数据规模较大带来读写工作负载产生误差的可能，同时加快了系统运作效率。

（二）本发明综合时间和空间两个角度进行了优化。时间上通过采用新的数据结构，有着一定程度的缩减；空间上通过缩减每层节点的模型大小，节约了大量的空间开销。

（三）本发明全面的处理策略可以处理多种不同的索引模式，并且保证了即使采用机器学习训练的方式，其性能也一定不低于传统的B+树。

（四）本发明充分利用每层模型得到的参数和数据，进行模型索引选择，逐层传递，用于下一层分枝节点处模型的训练。

## 2.3 本发明技术的技术方案

### 2.3.1 本发明应用的系统架构或场景（必选）

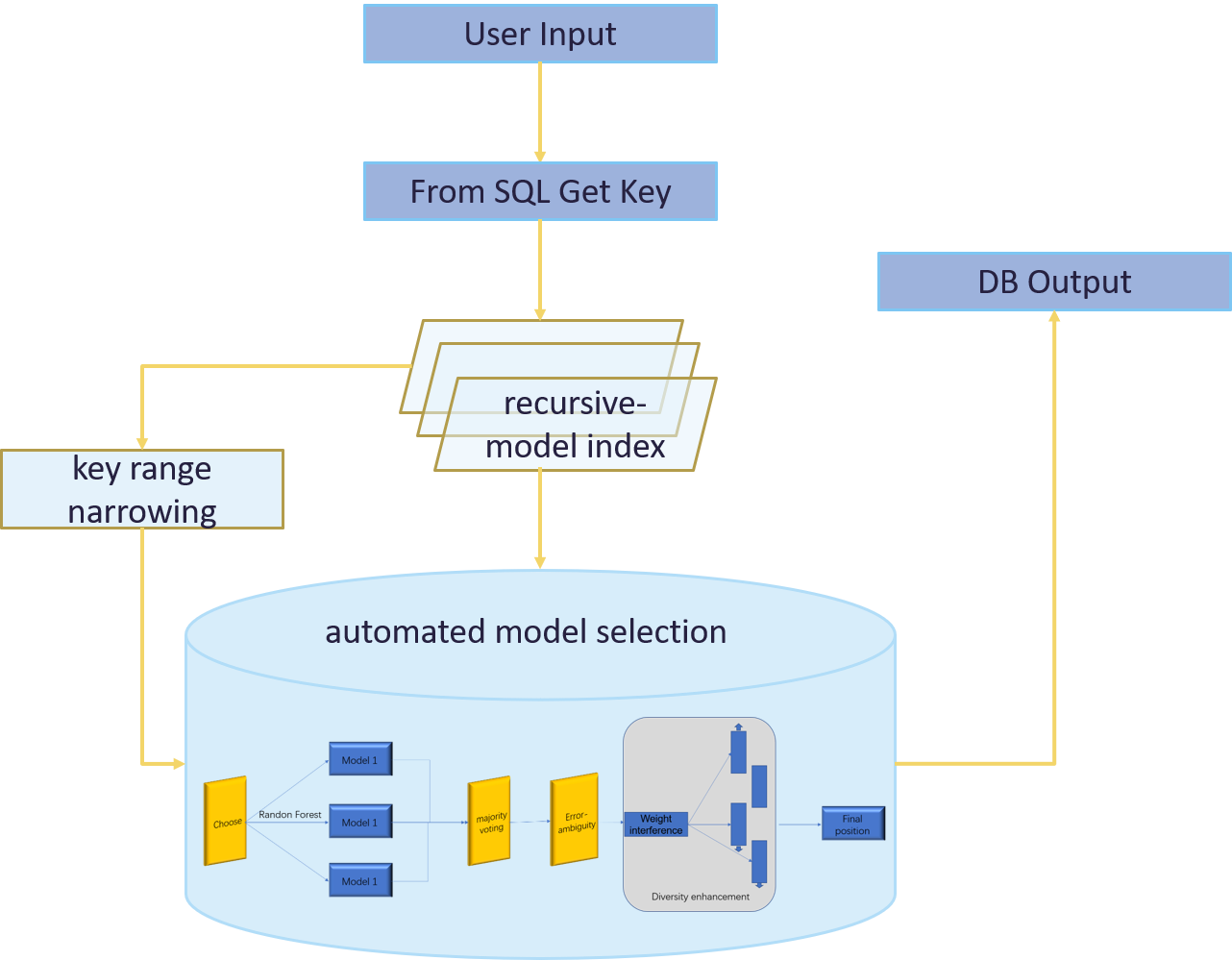


图3 本发明的系统架构图（或应用场景示例图）（必选）

（参考文字描述）

如上图是本发明方案的系统架构图（或应用场景示例图）：

其包含输入模块：输入模块负责将原始数据和sql工作负载读入。

其包含sql解析模块：将sql语句进行截断，得到key范围，用于决策树分枝节点的模型输入。

其包含递归学习索引模块：在每个分枝节点处，层层缩减key范围。

其包含自动选择模块：利用叶结点处得到的最小key范围进行模型自动选择，通过数据和sql工作负载特征得出对应的索引推荐结果。

其包含输出模块：将自动选择得到的结果，经过处理输出为对应的sql查询结果。

他们的功能如下：本系统可以实现单query语句的索引推荐，输入为一条query语句和涉及到推荐字段的数据，输出为使用了这些字段的推荐索引查询得到的数值。系统的核心由递归学习索引模块和自动选择模块组成。学习索引模块嵌入在决策树中，接收sql解析得到的key范围，并将训练结果逐层传递。此模块在叶结点处将处理得到的最小key范围传递给自动选择模块。自动选择模块通过使用随机森林的方式，学习数据特征，最终输出结果到输出模块，得到查询数值。整个系统可以实现自动化的语句解析，模型选择，索引推荐。各个模块之间紧密连接，相辅相成。

（参考案例）

**与本发明相关的软件结构，各模块功能如下：**

输入模块：输入模块负责将原始数据和sql工作负载读入。

sql解析模块：sql解析模块将sql语句进行截断，并且进行正则解析，得到输出是一个向量key：表征基于已知误差偏移的，上下界范围的向量，使得其能够作为学习索引模块的输入向量。

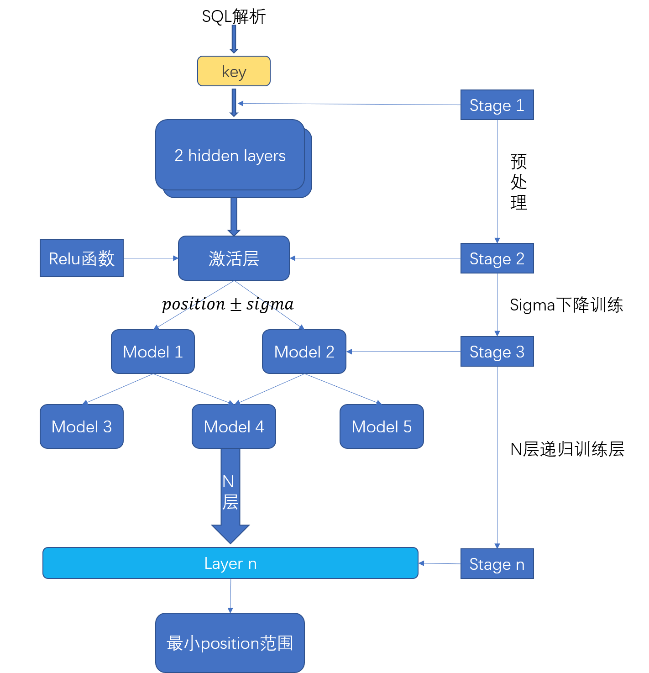
学习索引模块：该模块输入为sql解析模块得到的较大的key范围，输出为其缩减后的范围。该模块嵌在决策树上，每经过一层分枝节点，key的范围就进行了缩减，最终得到一个易于处理的较小范围key。学习索引模块的作用是根据数据特征缩减key范围，节约索引占用的空间资源。

自动选择模块：采用随机森林的方式，根据得到的较小的key范围，选择并组合多个小型的回归模型学习数据的分布，并预测索引推荐查询的位置。并且监测了预测的结果，提供了纠错机制。

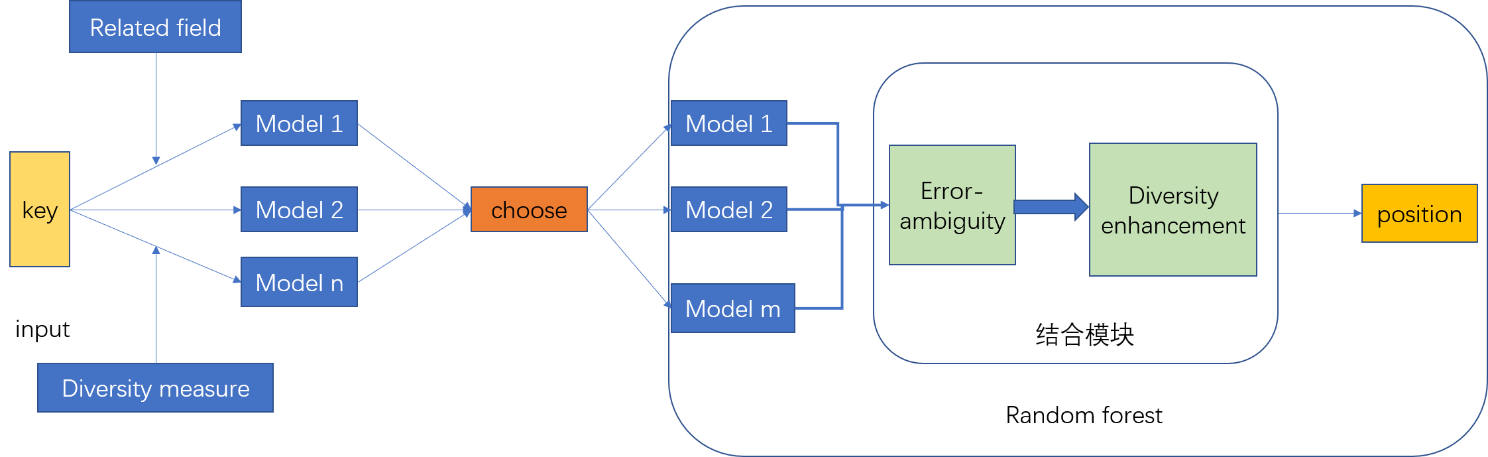
输出模块：输出模块将自动选择模块的输出整理为最终查询得到的结果。。

### 2.3.2 本发明核心装置&产品实现（必选）

（参考案例）



4-1递归学习索引模型



4-2自动选择模块

如图4-1所示，本发明的核心组件之一是递归学习索引模型。此模型与sql解析模块紧紧相连。它包括key输入层，样条插值划分层，n个递归层次训练层，以及输出层。Key输入层接收sql解析得到的key范围。这个key范围很大，并且误差偏差也很大。为了在后面的递归训练中取得好的效果，我们设计了样条插值划分层，将输入进来的所有数据划分为多段，称为模型训练得第一阶段，模型也从此进入了stage1。从此之后，模型进入了递归层次训练模式。往后得每一层，stage都会加一。这个层次结构得组成可能是决策树，B+树等结构。在每个结点上，我们加入了自己训练的模型，其中以redix模型尤为突出，因为它的表现性能在不差于现有研究中好的模型之外，他的内存占用明显下降。这就使得我们可以采用更多的层数进行递归训练，同时降低了空间资源得消耗。这也是本发明的一大明显优势之一。层之间传递的数据是以position-sigma和position+sigma的形式传递的。key的范围如下所示：

所以我们每次训练输出的数据，就是围绕position和sigma为中心。我们的训练目标，就是在明确position的情况下，尽可能减小sigma的数值，也就是缩减了key的范围。

如图4-2所示，本发明的核心组件也包括自动选择模型。此模型由输入层，n层回归模型，模型选择层，随机森林加强层，输出层组成。其中，n层回归模型遵循着“好而不同”的模型选择准则，采用多种类，重叠率低的不同模型，对于输入的key范围分别进行训练，然后把训练得到的参数，比如position的已知偏差参数等，在模型选择层进行模型的选择比较，将表现优秀的模型带入随机森林中进行集体学习，并且进一步强化模型的能力。最后根据随机森林的输出精确定位position的位置，得出用户查询的数值位置。

### 2.3.3 本发明核心方法流程（必选）

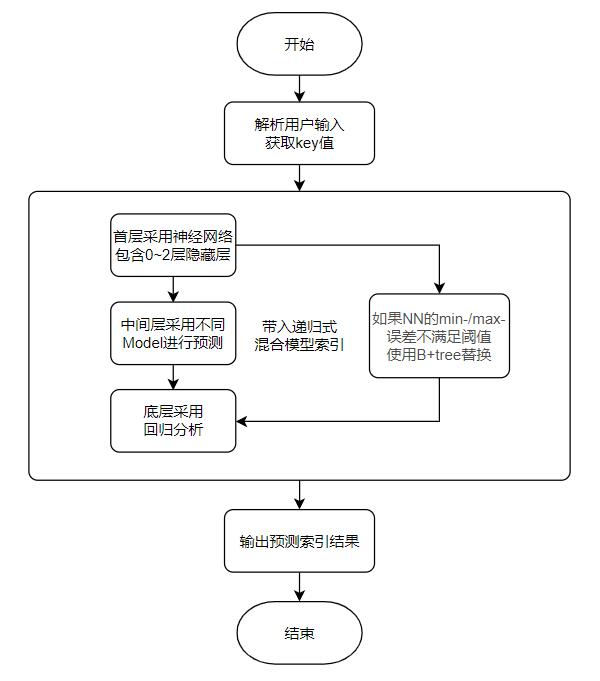


图5 本发明方案的核心方法流程图（必选）

（参考案例）

**结合图5，详细阐述本发明方法的核心执行步骤如下：**

（一）本发明的自动索引功能是基于已知的数据和用户进行query查询的sql语句。首先应将这两项数据输入，等待进行处理。

（二）然后将这两项数据送入sql解析器。通过正则解析技术，把sql语句和涉及的相应字段联系起来，得到一个已知误差上下界的key范围。这个范围可以代替sql语句，和原数据一起进入递归学习索引模块进行训练。

（三）本发明的递归学习索引模块接受原数据和较大的key范围，进过层层训练，逐层确定，缩减key的范围，此过程依赖决策树，RNN等模型。具体情形依照用户是单点查询，范围查询，或者联合查询等决定。本模块处理后得到的是最小的key范围。

（四）最小的key范围进入底层的回归分析，也就是自动模型选择模块，利用随机森林模型进行回归模型选择。此模块中提供了多种类，重叠率小的回归模型以供选择。输出得到精确的待查询数值的位置。如果出错，该方法能够监测发现并自动纠错。

（五）输出最终用户查询的结果，流程结束。

（六）本发明采用学习索引技术，通过递归的方式，逐层减小key的范围，节省了大量的空间资源。同时加速了下一层训练的速度。

## 2.4 本发明技术的实施例（必选）

### 2.4.1 本发明实施例一的技术方案（必选）



图6 实施例一的组件结构图（必选）

（参考案例）

本发明的实施例一共包含三个部分，分别是sql解析，递归学习索引，自动选择模型。图6给出了实施例一的组件结构图。Sql解析模块负责采用正则解析的方式，将用户输入的sql语句解析成对应字段的key范围。并且这个范围是已知其误差的上下界的。递归学习索引负责分析查询情形，如单点查询，范围查询等，并且缩小获取的key范围。自动模型选择负责接收这个已经最小化了的key范围，采用随机森林的方式，选择训练好的回归模型并且进行优化，最后输出查询数据的精确位置。

（参考案例）

本实施例为了实现自动学习索引模型，具体实施步骤如下。

（一）读入用户查询输入的sql语句以及原数据并进行正则解析。

本步骤的主要目的是将用户输入的sql语句转换为数据的key范围，以向量的形式存储，便于后续模块的训练。我们采用正则解析的方式，首先将sql语句转换位一个精确的key。然后将key输入到我们设计的层级模型中进行范围预测，逐层得到某一范围中的精确值key，作为下一层的输入，最终输出得到一个position。我们利用position和min，max数值，来描述key的范围，并且输入给学习索引模块。

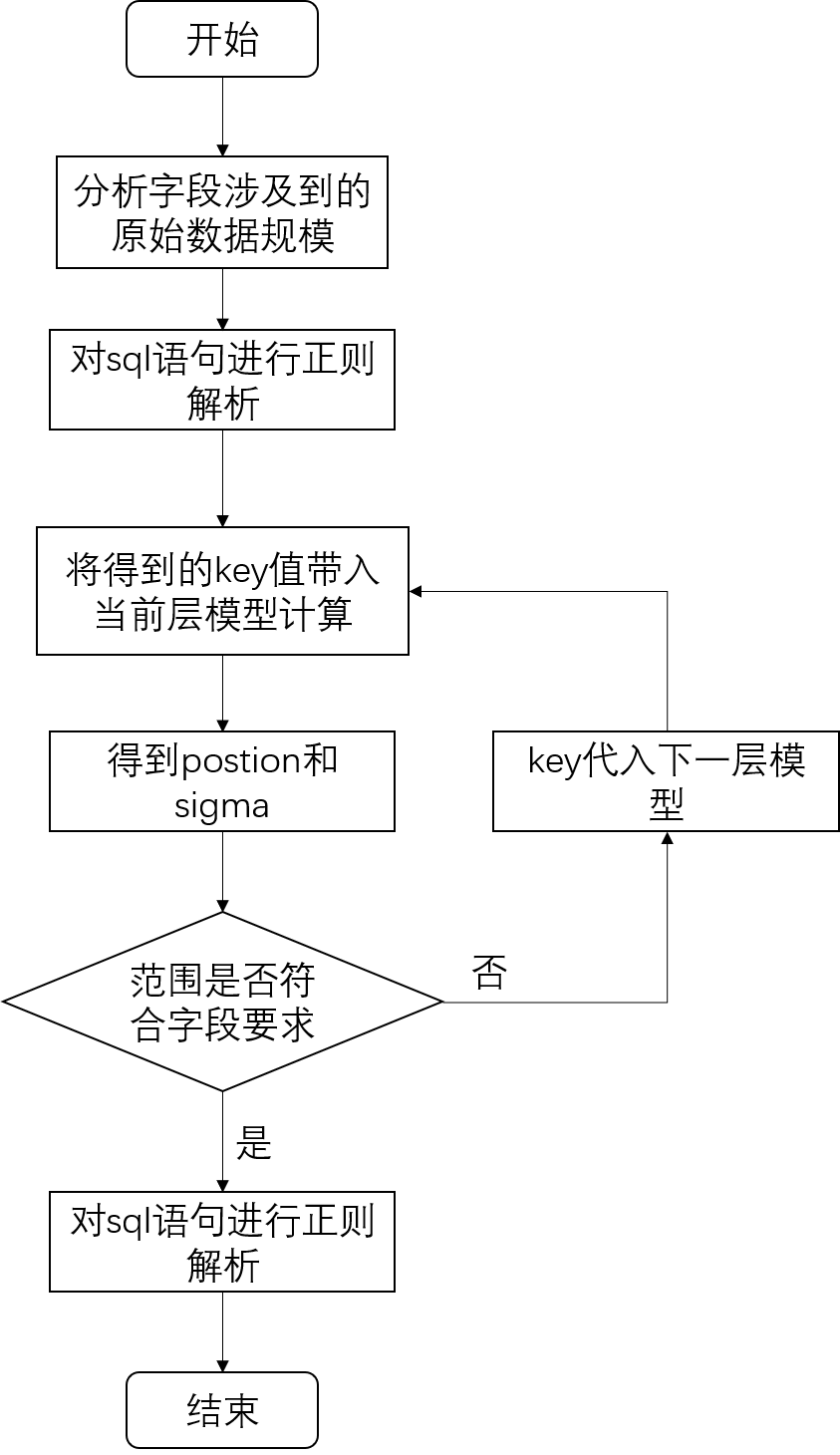


图7.1. 步骤（一）的实施流程图（必选）

（二）根据不同查询类型使用不同策略，制定递归学习索引模型。

本模块继sql解析模块之后，利用上层模块得到的key范围作为输入。我们会根据上一步得到范围，首先进行顶层的预训练，顶层我们采用的是三次样条插值模型，对整体数据进行粗略拟合和划分。随后我们推测用户查询的类型，提供相应的框架，进行索引学习，并进一步缩小key的范围，查询类型包括：单点查询（采用模型与hash相结合的方式处理），范围查询，联合查询等。提供的框架包括有：决策树，B+树等。在此基础上将切分后的数据分别迭代进入底层模型选择最优模型以及最高预测准确率，并且，对于范围索引等需要大量时间进行搜索查询的行为，我们会监视模型运行时间。如果时间超过了经调参设定的阈值，我们就认为用此种架构训练效果差于B+树。那么我们就采用基本的B+树。从而保证我们的模型时间复杂度介于O(1)和O(logn)之间。同时，此模块使得key的范围大量缩减，这为后续的自动选择处理提供了便利。

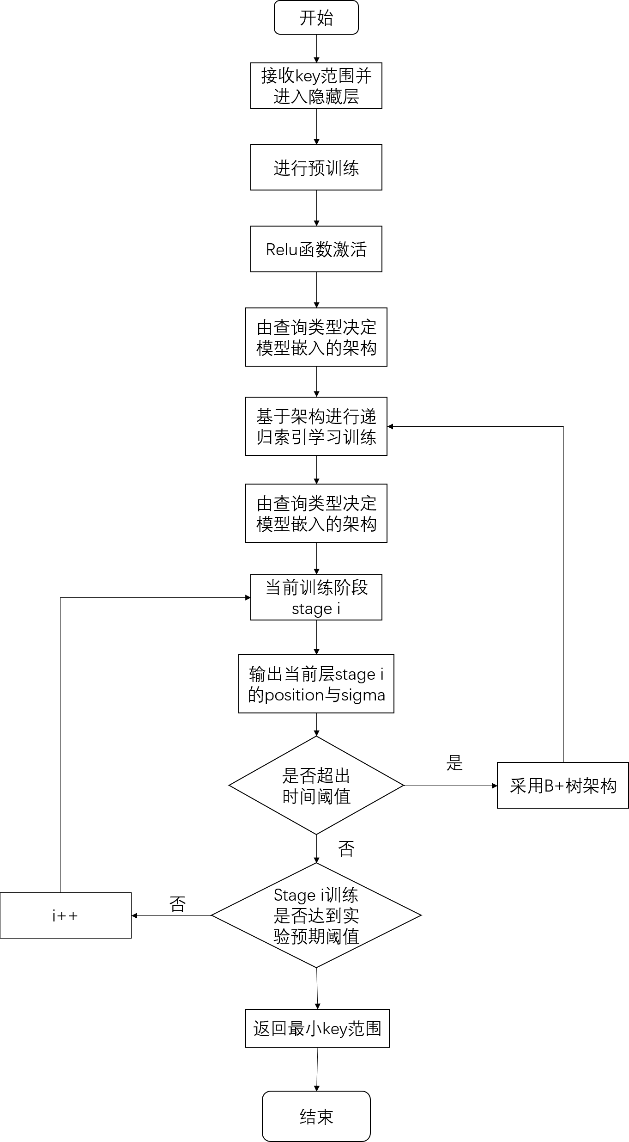


图7.2. 步骤（二）的实施流程图（必选）

（三）自动选择模块选择合适模型，获取数据准确位置。

此模块以多种类，重叠率小的多个回归模型作为个体学习器，采用随机森林方式进行模型的优化与加强。学习索引模块提供了这部分所需要的数据输入。基于这个数据，首先训练多个个体学习器，比如线性回归，多项式回归，弹性回归等，记录模型的训练参数。然后通过设计打分函数，比较模型的训练效果，选择出表现较好的多个回归模型和当前训练的数据建立对应关系。通过随机森林的方式，对选择的模型进行优化与加强，包括多样性增强，误差-分歧分解等操作。对于预测过程，则无需再用所有的底层回归模型进行学习训练，可以预测出带查询数据对应的推荐回归模型，利用随机森林进行优化加强，最终输出数据精确预测位置。

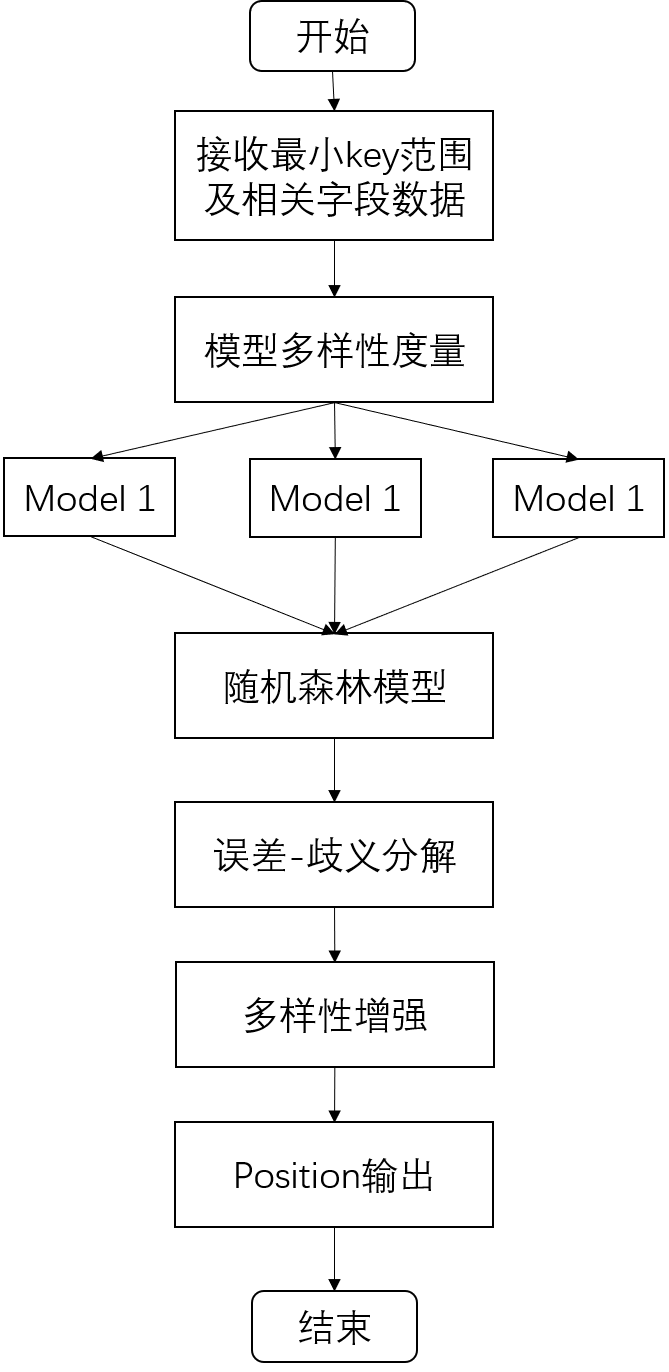


图7.3. 步骤（三）的实施流程图（必选）

2.4.3 本发明实施例一区别于现有技术的改进之处（必选）

（参考文字描述）

现有技术采用强化学习方法，无法实现从零开始的索引推荐，需要人为首先定义好当前数据集和工作负载下的可选索引配置集；并且，现有的模型内存占用较大，尤其是为了达到好的准确度，需要多层模型进行训练，占用了大量的空间资源。

本发明与它不同之处在于：本发明将采用了学习索引，配合模型自动选择的功能模块，可以实现从零开始的索引推荐，提高工作运行效率。并且本发明的模型在维持运行时间稳定的基础上，在空间上比现有研究中的模型节约了大量的空间资源。这使得我们的发明和现有研究成果相比，具有更广泛的适用性。

# 3、本发明关键技术点概括及对应的有益效果（必选）

### 3.1 本发明的关键技术点概括（重要：必选）

（参考文字描述）

本专利保护的关键技术创新技术点包括：

1. 采用了层次化学习索引技术逐渐缩小索引推荐问题处理数据的规模。

该创新点包括如下特征：1、数据规模的减小，使得模型空间占用消耗显著减小，使得索引推荐技术具有更高的适用性；2、模型训练时，每一个模型都对键key的位置进行一定误差的预测，并使用该预测来选择下一个模型，该模型负责键key空间的某个区域，以较低的误差做出更好的预测更快的进行训练； 3、缩减处理的数据规模可以降低由于读写大量数据所带来的效率降低问题；4、顶层模型采用简单样条插值模型，对于数据分布进行粗略划分，计算开销相较于传统采用的神经网络显著降低；

1. 设计了sql解析器，把query语句转成易于模型处理的key范围。

该创新点包括如下特征：1、query语句的结构，决定了它可以采用正则解析的方式加以处理； 2、query查询的sql语句难以直接提取特征进行索引学习。采用sql解析的方式把sql语句对应字段的key范围，处理成向量形式，便于模型的训练；

1. 本发明设计了自动选择模块，利用随机森林模型增强面向小规模数据的模型效果。

该创新点包括如下特征：1、自动选择模块在决策树的叶节点部分，需要处理的数据经过递归学习索引模型，规模大量减小； 2、本模块采用了现有研究中较为先进的回归模型作为个体学习器，包括多项式回归，岭回归等，同时采用了一些树结构回归模型，如极限树回归，梯度提升树等模型，尽可能地做到“好而不同”，得到了性能较好的模型。3、实现了从零开始的索引推荐。用户不用定义索引类型，我们根据回归模型自动选择并且进行优化，最终直接输出用户查询的数值。并且，通过实验发现，利用随机森林模型进行增强后，即使训练时预测出错，预测位置偏离正确位置极小，便于纠错处理。

### 3.2 本发明的关键技术点对应的有益效果（必选）

（参考文字描述）

技术问题一：以往查询中利用的的索引推荐模型，都需要用户首先将可行索引集定义出来，然后模型在可行的索引集合中找到最佳的索引组合，进而进行数据查询操作。找到可行索引集合的过程通常需要大量的调研和统计是非常消耗资源的；

技术特征一：本发明的模型可以进行从零开始的索引推荐。从sql解析模块得到的key范围作为输入，模型自动选择合适的索引，进行key范围的缩减；

技术效果一：本发明对于常见的索引模式进行了分类处理，包括单点查询，范围查询，联合查询等。通过采用递归索引学习模型，该模型比现有研究的索引模型，空间占用上更小。采用了三阶样条函数加决策树的方式替代b树，逐层通过模型筛选，缩减key范围。在很大程度上，这样做减少了整个索引部分的空间占用。并且，更小，更精确的key范围为后续回归模型的训练提供了很好的数据质量保证。

技术问题二：query查询的sql语句，由于查询形式多样，并且可能加上很多有用的参数，难以直接提取合适的特征进行索引学习。若直接作为样本，会对后面的学习索引会产生影响；

技术特征二： query查询的sql语句支持进行正则解析，结果输出成key范围的向量形式；

技术效果二：该模块真正实现了从零开始的自动学习索引的查询操作。用户只需要输入query语句，而不用指定索引形式，方便用户使用。并且输出是一个向量key：表征基于已知误差偏移的，上下界范围的向量。

技术问题三：现有技术利用分类模型处理得到结果，由于样本基数大，速度较慢；而此发明将key的范围逐层递减，到底层的数据范围已经很小。现有的研究中是在叶子节点采用多种回归模型，但是训练效果显然差于采用集体学习得到的模型；现有的查询纠错方法是在底层叶子结点的较大范围中采用二分查找的搜索策略，但是时间代价显然差于在小范围内的遍历查找。

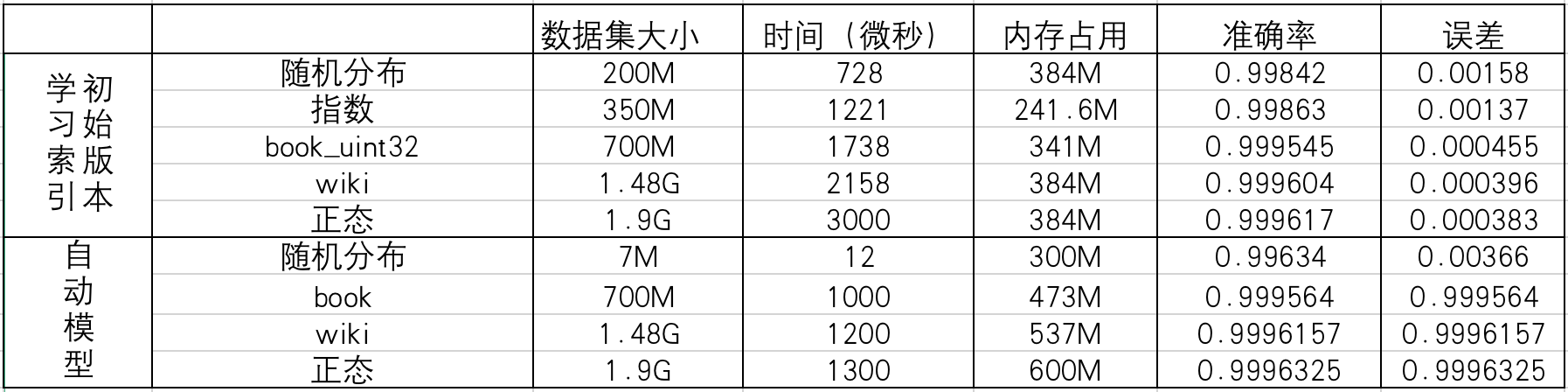
技术特征三：本发明使用随机森林模型，以多种类，但重叠率较小的多个回归模型（如多项式回归，弹性回归等）作为个体学习器，在小范围key输入下，产生对于查询结果的快速精确预测，并对于出错情况进行修复处理。

技术效果三：基于学习索引模块的改进，此处采用回归模型替代现有研究的分类模型，在时间上获得了优势。并且采用随机森林的方式，比一般的回归模型性能更强，预测效果更好。在另一方面，用户不用在查询语句中自己定义索引类型。通过自动选择，我们选择出性能最好的回归模型，并且进行集成学习优化，最终直接输出用户查询的结果。同时，我们对于预测结果进行监测，发现采用这种缩小key范围和采用随机森林模型的方法，可以极大降低出错概率，并使得差错位置在正确位置的3-邻域以内，从而简化了我们的纠错搜索策略，可以用简单的遍历搜索代替二分查找。

# 本发明实验结果

以下两项数据来自网址：[Searching on Sorted Data - Harvard Dataverse](https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/JGVF9A&version=4.0)

注：Mit在kraska中提供的数据集均为二进制文件，这样是为了减少数据读取占用的时间以及内存占用。这些数据集都用了特殊的符号对于样本X与带预测值y进行分割



1. Book\_200M\_uint32：是mit开源提供的一项关于图书的，已经经过排序处理的数据集。内部是用32位模式下的unsigned int 图书信息。这部分数据集在利用rust环境下的cargo命令进行实验时，采用了线性回归层linear与三阶样条函数cubic相结合的方式进行优化测试。
2. Wiki\_ts\_200M\_uint64：是mit开源提供的来自wiki百科的已经排序的数据集，内部是64 位模式下的unsigned int。此数据集在测试时，用采用jeff论文中各种底层的模型的组合优化，其中包括线性回归，逻辑回归，cdf方法，样条函数。

经过实验发现，jeff提出的这一方法在多次训练后，组合形成的这一套底层结构大小是基本确定的，约为384M，他们最终的准确度会随着数据集的不同有着小幅度变化，约为0.0001。

1. 其余数据集：200M, 350M, 7M, 1.9G这四个：利用自己编写的create函数生成，其中加入了适当的噪音，制定了他们的数据分布。

200M：随机分布

350M：指数分布

1.9G：正态分布

注：自动模型选择部分的实验，内存占用主要在数据集处理部分（因为kraska直接用的二进制，所以他们没有这个消耗）。经测试得出，8G内存下测试1.9G数据集，内存占用达到600M。数据处理时间达到了30分钟以上，需要优化。