查询规模估算 实验报告

2018202177 官佳薇　2018202183何雨琪

摘要

在本次实验中，我们分别使用机器学习和深度学习方法对查询规模进行预测。首先利用原始查询语句构造各条查询的特征向量，以实际查询条数为y值，使用多种机器学习方法进行回归预测，最终使用 Gradient Boosting Regressor 得到 $RMSLE$ 分值为 1.33。然后针对查询计划进行特征抽取，提取其中的 Execution Time、Filter rows、Cost rows三个特征加入向量表示，得到 $RMSLE$ 分值 0.87。我们尝试了参考论文 "*Learned Cardinalities: Estimating Correlated Joins with Deep Learning*" 使用深度学习方法进行预测，但效果不如机器学习理想。通过调整参数，最终得到 $RMSLE$ 分值 0.86。

Phase 1 查询语句+机器学习

一、 构造查询语句特征向量

数据集包含查询语句、各属性最大最小值，以及查询计划。我们首先不考虑查询计划，单独使用查询语句进行特征抽取和向量表示。查询语句中包含每条查询涉及的表名、表连接和查询条件，训练数据中额外包含查询规模。各类数据间以#分隔，数据内部以逗号分隔。

参照 $AndreasKipf$ 等人的研究[1]，我们将每一条查询 $q \in Q$ 看成表 $Tq \in T$、连接 $Jq \in J$、谓词条件 $Pq \in P$ 的组合(Tq, Jq, Pq)。考虑到连接条件中已暗含表信息，我们在实现中省去了表的独热表示，使组合变为 (Jq, Pq)，从而在维持性能的同时降低了向量维度。举例如下：

title t, movie_companies mc # t.id=mc.movie_id # t.production_year>2010

Join set $\{[0\ 0\ 1\ 0\ 0\ 0]\}$　Predicate set $\{[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0.72]\}$

1. **连接向量表示**

   我们观察到，数据中共 6 种不同的连接，且每条查询最多包含 2 个连接条件。即连接条件数目有限且形式固定，故将每个连接 $j \in J$表示为 6 维的独热向量，并对2个独热向量进行堆叠，对仅有 1 个连接条件的向量用 0 填充。

2. **谓词条件向量表示**

   每条数据包含多个查询条件，而每个查询条件具有固定形式(col, op, val)，且三者彼此之间相互关联，故考虑将三者结合进行特征表示。其中列名 col 个数有限且形式固定， 数据中共 9 种不同的属性名称，故将其表示为 9 维的独热向量。运算符 op 仅有三种取值 $op \in (>, =, <)$，故同样将其表示为 3 维的独热向量。而val 为数值型变量，故将其设置为单一维度，并记录所有谓词条件中的最大值和最小值，对该维数据进行归一化处理。

   综上所述，每个谓词查询条件的特征表示由 9 维属性名向量、3 维运算符向量和 1 维 val 取值拼接而成，共 13 维。查询数据中最多同时出现 6 个谓词查询条件，故设置查询条件向量共 78维，不够 6 个条件的填充对齐。

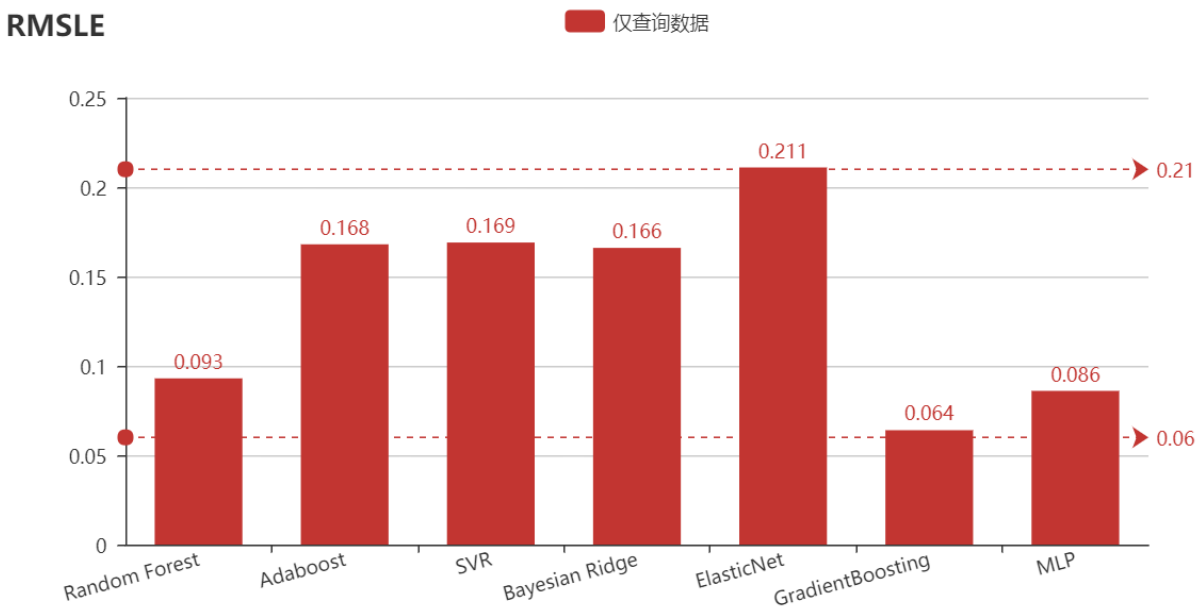
3. **单条查询向量构建**

   根据单条查询组合 (Jq, Pq)，将 12 维连接向量和 78 维谓词条件向量合并，组成 90维特征向量，作为单条查询的特征表示。

二、 机器学习模型

实验将训练集按 9：1 比例划分，使用9000条数据作为训练集进行训练，1000条数据作为validation 进行验证。

我们使用了
Random Forest、Adaboost、SVR、Bayesian Ridge、ElasticNet、Gradient Boosting Regressor、MLP Regressor 共7 种 Machine Learning 方法进行训练，使用 GridSearchCV 选择参数，并利用条数据的预测结果计算 $RMSLE$ 分值，得到预测效果如图：



其中Random Forest、Gradient Boosting Regressor 和 MLP 表现较好，其中 Gradient Boostring Regressor表现最为出色，$RMSLE$ 分值仅为0.064。

使用Gradient Boosting 对测试集进行预测，得到$RMSLE$ 分值为1.33，如图：

结果提交成功！

您的测评得分为 1.3312861844489032

注意：评测结果使用MSLE，评测分数越小代表结果越好。

查看排名

返回首页

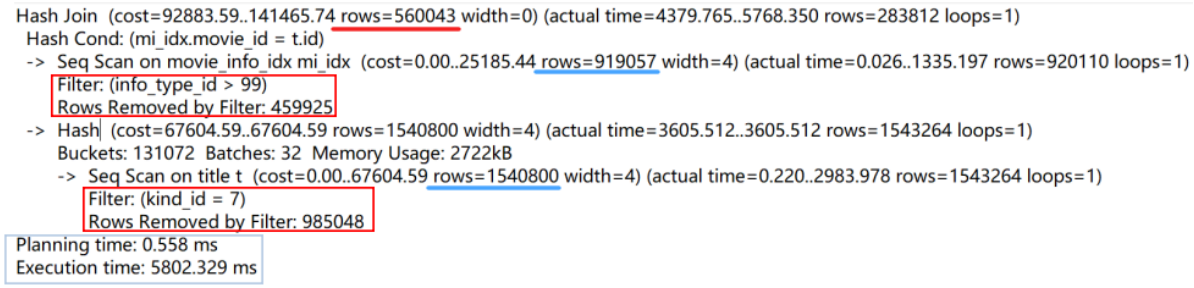
Phase 2 查询语句+查询计划+机器学习

一、构造含查询计划的特征向量

我们观察到，每条查询计划中均包含Planning Time 和 Execution Time，意为查询计划耗时和实际执行耗时。训练集查询计划最外层包含计划输出的总行数 rows 。查询中的每个谓词条件均对应一个过滤器 Filter，以及被过滤掉的行数 Rows Removed by Filter 和执行子计划输出的总行数 rows。这些对于我们的查询规模估算任务来说是极有帮助的，例如执行时间可以在一定程度上反映查询的规模，DBMS 估算的行数会支撑我们

的预测，每个谓词条件过滤掉的行数显然与我们的预测高度相关。因此我们对查询计划中的上述特征进行抽取，加入到特征表示中进行预测。

查询计划特征如图:



1. **执行时间**

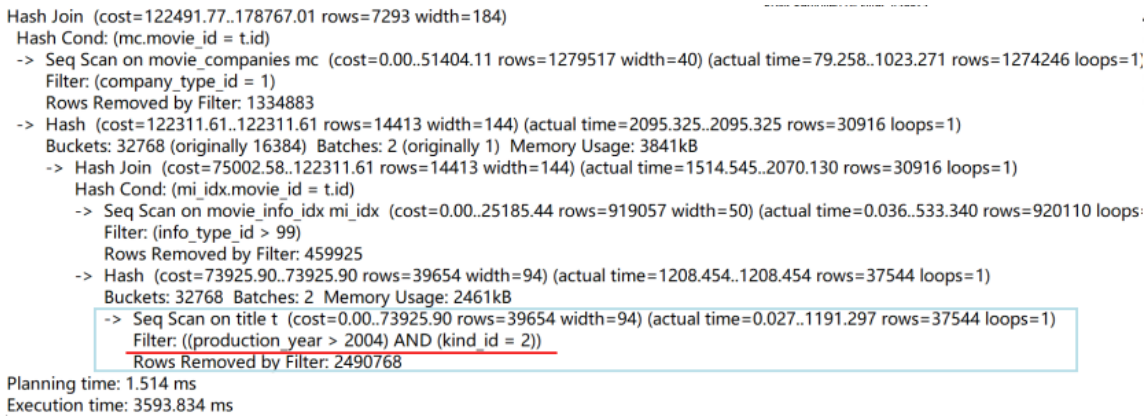
   查询计划底部记载着查询的计划执行时间和实际执行时间，然而计划查询时间和实际执行时间的差距过大，差异时常在10^3及以上数量级，故考虑提取 Execution time 加入特征表示。由于Execution Time 为单条查询的综合表示，且为数值型变量，故直接将其作为 1 维特征加入向量末端，并记录最大和最小执行时间，由于该维度数据值较大且分布极端，故取 log 并归一化处理。

2. **计划输出行数**

   在查询计划首行标识着整个查询计划输出行数。上图为 Training Plan ，包含实际输出行数，显然更有助于我们的预测，但 Testing Plan 中不报哈实际输出行数，故考虑使用计划输出行数加入特征向量。该特征同样为单条查询的综合特征，且为数值型变量，故直接将其作为 1 维特征加入向量末端，并记录最大和最小输出行数，由于该维度数据值较大且分布极端，故取 log 并归一化处理。

3. **谓词过滤行数及输出行数**

   查询计划中每个谓词条件均对应一个Filter 过滤器形式输出其过滤的总行数Rows Removed by Filter，以及计划输出的总行数rows，这两个数值可以表征每个查谓词条件对结果的影响力。然而我们发现，有些查询计划中由于做了部分优化，使得 2 个谓词条件对应同一个Filter，即对应同一个过滤总行数和输出总行数。如图中标注处所示：



   故我们采取了两种解决办法：

   a. 将该数值直接赋给 2 个谓词条件，即 Production_year>2014 和 kind_id=2 两个谓词条件的行数值均取 2490768 和 39654.

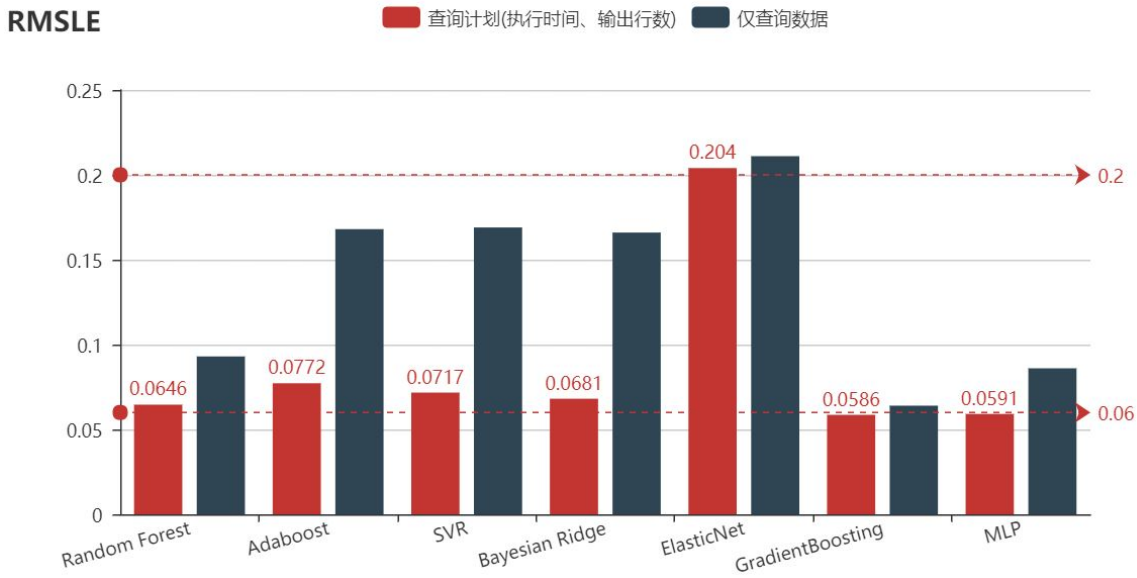
   b. 平均分后赋值给 2 个谓词条件，即两个谓词条件均取 1245384 和 19877.

   提取每条查询中的每个谓词条件对应的 2 个总行数，归一化后加入到谓词条件向量末端，使每个谓词条件向量维度增加到15 维，单一查询中最多 6 个谓词条件，故谓词条件向量变为 90 维。

二、机器学习模型

1. 加入执行时间和计划输出行数

在原来的90 维向量末端叠加执行时间和计划输出行数，构成新的特征向量 92 维。使用机器学习模型得到 $RMSLE$ 分值如图：



图中红色部分为加入查询计划的执行时间和计划输出行数后的分值，可见加入该特征后，各模型的预测效果都有了明显改善，其中Gradient Boosting 模型的预测效果仍为最佳，其$RMSLE$ 分值为 0.586。

使用Gradient Boosting 模型对测试集进行预测，得到$RMSLE$ 分值为1.15，如图：

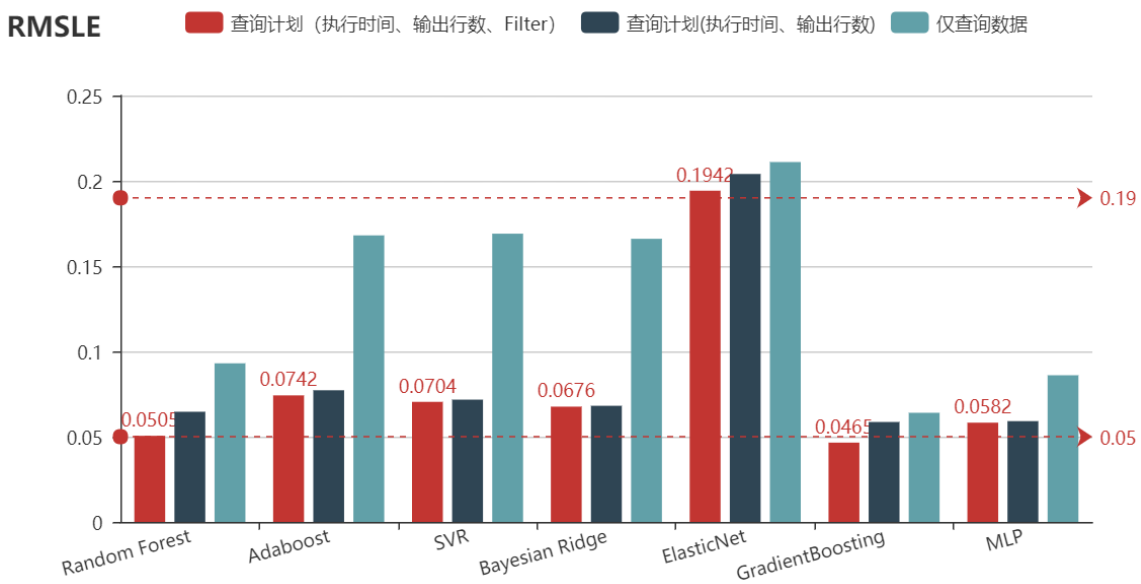
提交记录	MSLE
naive	1.0222304456037665
test	1.1526044830764641
naive_1	1.1889218755743727

2. 加入谓词过滤行数及计划输出行数

在每个谓词条件末端加入谓词过滤行数和计划输出行数，使得每个谓词条件向量维度增至15维，谓词向量共 90 维。得到每个查询的特征向量共 104 维。如前述，查询计划汇中可能出现 2 个谓词条件对应同一个 Filter 的情况，故有两种方法对其进行赋值：

a. 数值直接分配给各谓词条件

使用各机器学习模型对数据进行回归训练，得到$RMSLE$ 分值如下图红色柱形图所示。与前期训练结果相比结果又有提升，最优模型仍为 Gradient Boosting Regressor，在验证集上得到 $RMSLE$ 结果为 0.0465。

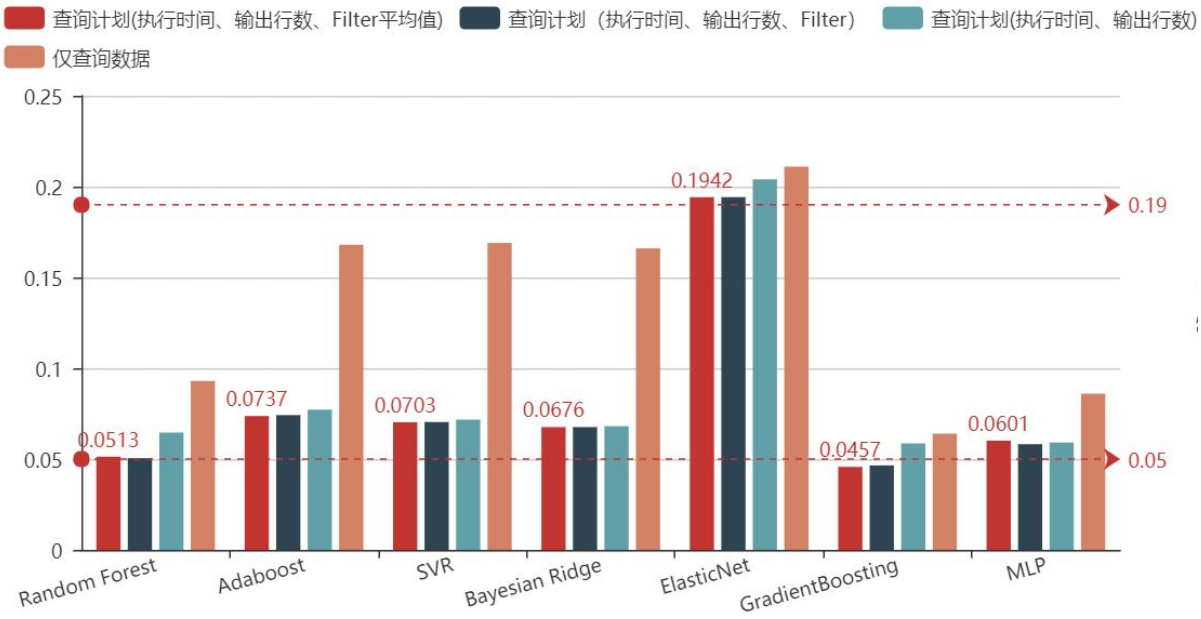


使用Gradient Boosting 模型对测试集进行预测，得到$RMSLE$ 分值为0.879，如图：

提交记录	MSLE
test	0.8795608246075517
naive	1.0222304456037665
ll1	1.0377605783617592

b. 数值平均分给各谓词条件

将 Filter 下的过滤行数和计划输出行数平均分给两个谓词条件，使用各机器学习模型对数据进行回归训练，得到$RMSLE$ 分值如下图红色柱形图所示。与前期训练结果相比结果又有提升，最优模型仍为 Gradient Boosting Regressor，在验证集上得到$RMSLE$结果为 0.0465。



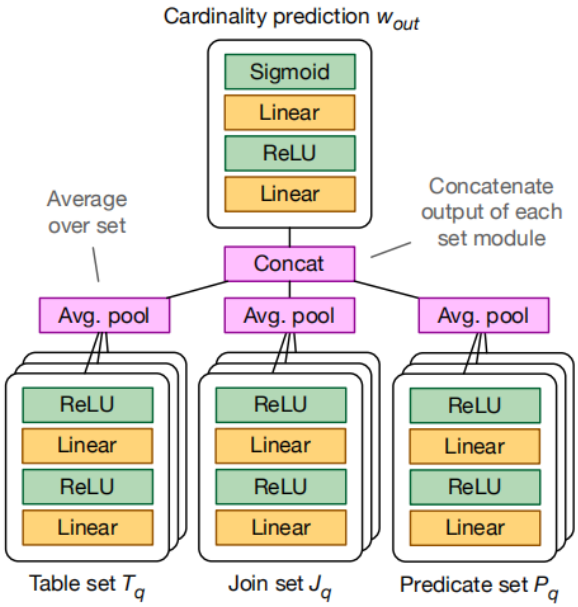
使用Gradient Boosting 模型对测试集进行预测，得到$RMSLE$ 分值为0.861，如图：

提交记录	MSLE
test	0.8612148772240455
naive	1.0222304456037665
ll1	1.0377605783617592

Phase 3 查询语句+深度学习

一、MSCN 多集卷积网络

论文"*Learned Cardinalities: Estimating Correlated Joins with Deep Learning*" 提出了一种基于深度学习进行基数估计的方法，学习预测数据的相关性。查询相关的表、连接和谓词被表示为独立模块，每个集合元素由一个具有共享参数的两层神经网络组成。模块输出经过平均、连接后输入到最终的输出网络。



该模型是建立在基于采样估计的基础之上，采样将基数和位图转换为训练信号。然而我们无法提供采样数据，选择将$MSCN$ 模型中 $Tables$ 模块删除后进行模型训练。经过9个Epoch后，训练结束。如图所示，训练集上的$RMSLE$为2.6614，训练效果没有使用Gradient Boosting 模型好。

train	2.6614290986720186

总结

我们认为机器学习是解决基数估计问题的一个非常有效的技术。基数估计可以表示为一个监督学习问题，其输入为查询特征，输出为估计基数。虽然机器学习的基数估计并不是完美的，但是其产生的估计可以直接利用现有的复杂模型和成本模型。

使用机器学习进行基数估计面临着两个选择：如何将查询特征化以及应该使用哪种监督式学习算法。

- 查询特征化： 我们起初单独使用查询语句进行特征抽取和向量表示，随后加入了查询计划中可以抽取的查询细节特征，对查询特征化进行进一步的改善。
- 监督式学习算法： 我们使用7 种 Machine Learning 方法进行训练，选出最优模型，使用 GridSearchCV 进行最优参数选择。