[introduction 3 min + framework 5 min + experiment 5 min]

**[5 s]**

Esteemed audience,

Thank you for being here today. It is my great honor to present to you this paper.

**[1 min 30 s]**

Traditional approaches to cardinality and cost estimation rely on histograms, assuming specific data distributions and independent attributes. However, real-world datasets often deviate from these assumptions.

Recent years have witnessed the development of learning models as advanced cardinality and cost estimation tools. Learning models can be divided into data-driven methods and query-driven methods. The former is usually more precise, such as Naru, NeuroCard, QuickSel. The latter typically has faster computational efficiency, less storage space, and can seamlessly integrate with query optimizer. In this work, we will focus on query-driven models.

Query-driven models can be further classified into two categories: query-statement-based models and query-plan-tree-based models. The former rely on the query statement as input, while the latter utilize the query plan tree generated by a DBMS optimizer.

Although plans generated by DBMS are usually suboptimal, we can use the information embedded in them for estimation. Therefore, the learning models based on the query plan tree can utilize more helpful information and generate a more accurate estimation with little extra computational cost.

**[45 s]**

Although query-based learning models have many advantages, some limitations prevent them from becoming silver bullets for query optimization.

Firstly, existing query-plan-tree-based models often suffer from a paucity of extracted information, as models like MSCN and TPool rely on one-hot encoding and cannot leverage the rich information contained in the query plan tree.

Secondly, these models can be computationally expensive during both training and inference stages, which hinders them from applying to more real world scenarios

Lastly, a single model may not be able to fit all kinds of queries, resulting in poor data adaptability.

**[40s]**

In this paper, we propose a cardinality and cost estimation framework based on Bidirectional Compressor-based Ensemble Learning (BICE).

To address limitation one and two, we introduced four sub encoders that efficiently encode various information in the query plan tree, and used a compressor to reduce the learning difficulty of the estimation model.

To address limitation three, we introduced ensemble models and transfer learning to enhance the model's data adaptability and efficiently obtain cost estimation models.

**[1 min]**

Now let's take a detailed look at the framework of this model.

This image shows the architecture of the entire model.

Firstly, the data preparation component. Through a database management system (DBMS), we obtain the query statements based on the dataset and the corresponding query plan trees.

Next is the Feature Extractor. The feature encoder consists of four sub encoders, which encode the node sequence obtained by traversing the query plan tree into a node embedding sequence.

Next is the compressor. We use Long Short Term Memory Networks (LSTM) to compress embedded sequences to handle tree structures of query plan tree.

Finally, there is the training phase. During the training process, we employed Bayesian neural networks, active learning, and transfer learning methods.

**[1 min 30 s]**

Let's skip the data preparation and first see the feature encoder, which aims to encode the query plan tree. It consists of four sub-encoders that capture rich information from the query plan tree to facilitate the learning process.

The first sub-encoder is the Join Encoder. It processes the join operations in the query into an undirected graph, and then uses the node2vec algorithm to encode the columns. By representing the join operation as a graph structure and embedding the columns, we can capture the relevant information in the join operation.

The second sub-encoder is the Type Encoder. It encodes different node types using one-hot encoding. Each node type is represented as a vector, where only one element is 1 and the rest are 0. This encoding method can help the model distinguish different node types.

The third one is the Filter Encoder. It is based on range representation and parallel networks, which can effectively represent the information in the filter and reduce the learning difficulty of the model.

Finally, the Information Encoder leverages the estimated cardinality and cost generated by the optimizer of a DBMS. These estimated values can provide important information about the query plan.

**[30 s]**

The goal of the compressor is to capture rich information and temporal relationships in the node sequence. To achieve this goal, a two-layer bidirectional LSTM model was adopted in the paper.

Through such a compressor model, we can further compress the node embedding sequence obtained during the feature extractor, in order to better handle the characteristics of tree structured data.

**[1 min 30 s]**

Next is the content of Ensemble Learning, including Bayesian neural networks, active learning, and transfer learning.

Firstly, Bayesian neural networks. BNN is a probabilistic model that can handle the randomness and uncertainty of data. It represents the weights and parameters of the neural network by introducing probability distributions, and uses Bayesian inference for model training and inference. This method can better model the uncertainty of data, improve the accuracy and robustness of estimation.

Next is active learning. Active learning is a method of actively selecting meaningful data samples for annotation. In the task of cardinality and cost estimation, data annotation is often expensive and time-consuming. Through active learning, we can select the samples that are most helpful for model training and inference for annotation, thereby improving the model's generalization ability and effectiveness. Active learning can select samples based on different strategies.

Transfer learning utilizes existing models' knowledge to quickly train and adapt to target tasks. In cardinality and cost estimation, a prediction model can be initially trained for cardinality estimation, and then the learned knowledge can be transferred to the cost prediction model. This approach reduces the data requirements for the target task, speeds up model convergence, and enhances estimation accuracy.

**[40 s]**

Next is the specific content of the experiment, including the dataset used, evaluation methods, and metrics.

Firstly, in the experiment, we used multiple datasets for evaluation, including:

The JOB light, Scale, Synthesis, TPC-H, which had different numbers of queries and joins

Next is Evaluation Methods. To evaluate the performance of base and cost estimation, we used various other methods as references, including:

PostgreSQL, MSCN, QPPNet, TPool, QueryFormer，which have different characteristics.

Finally, We use q-error to evaluate the accuracy of cardinality and cost estimation. We also evaluate the efficiency of the models, including the training and testing time.

**[30 s]**

In terms of precision, based on the provided table, we can draw the following conclusions:

1. The first result is the cardinality. From the table above, it can be seen that the average and maximum metrics of BICE on the four workloads are significantly higher than the optimal model (i.e., QueryFormer)

2. The second result is in terms of cost. From the table in the lower part, it can be seen that although BICE is slightly inferior to QueryFormer in Synthesis, it is much higher than QueryFormer in the other three workloads

**[40 s]**

In terms of time, based on the provided table and charts, we can draw the following conclusions:

The aggregated results, shown in Table II, indicate that BICE exhibits the most significant improvement and the shortest plan time, except for TrueCard.

In BICE, the encoding is made as simple and efficient as possible by the feature extractor, the compressor reduces the learning difficulty of the estimation model, and the bidirectional LSTMs can compute in parallel. Furthermore, only lightweight networks need to be updated when building the ensemble model. Thus the computational efficiency of BICE is better than some models.

**[20 s]**

The following is the specific content of ablation study, aiming to analyze and evaluate the contribution and importance of each component in the model, and observe changes in performance by gradually removing or modifying a part of the model. The charts here show the results of ablation studies on all workloads.

**[45 s]**

Firstly, the content of ablation research on feature extractors. Including three contents

NoJoin: This feature extractor uses one hot encoding, which does not consider the relationships between columns.

NoPre: This feature extractor uses the filter encoding method in Tpool, and it has sparsity problem.

NoInfo: This feature extractor does not have an information encoder.

From the charts below, it can be observed that the performance of the three indicators is much worse than that of BICE, indicating that the join encoder, filter encoder and information encoder used by BICE in the feature extractor, play an important role in prediction.

**[40 s]**

Next is the content about Compressor. There are two different configurations.

BICE-LSTM: This configuration uses a single-layer LSTM and does not use a compressor.

BICE-LSTM2: This configuration uses a two-layer unidirectional LSTM, but also does not use a compressor.

From the charts below, it can be observed that the performance of the two indicators is much worse than that of BICE, indicating that the Compressor based bidirectional LSTM, play an important role in prediction.

**[35 s]**

Next is the content about batch training. There are two different configurations: BICE-SeqBatch and BICE.

In the BICE-SeqBatch method, we classify sequences based on their length and perform batch training on sequences of the same length.

In the method used in the final model, we obtain a fixed length sequence by filling in unequal sequences.

From the charts below, it can be observed that the performance of the SeqBatch method is worse than that of BICE, indicating that the pdding-based batch training method is better in estimation.

**[20 s]**

Next is the specific content about Ensemble Model,

For the BICE NoAl term, the data was not sampled through active learning and no integrated model was constructed. From the chart, it can be observed that BICE NoAl is significantly worse than BICE. This indicates that integrated models are crucial in prediction

**[40 s]**

Next is the specific content about Analysis of Data Sampling.

Optimal k value: Based on the provided information, the optimal k value for achieving the best performance lies between 20% and 40%. This means that retaining approximately the top 20% to 40% of the data during sampling yields the best results.

Different data sampling strategies: Different data sampling strategies were attempted in the study. Table III presents the impact of using various strategies on estimation results. For BICE\_strategy12 using strategies 1 and 2, and for BICE\_strategy13 using strategies 1 and 3. The results indicate that the sampling strategy designed in this paper positively impacts estimation accuracy.

**[1 min]**

This study proposes an ensemble learning framework based on bidirectional compressors for cardinality and cost estimation tasks. The following are the key points of this study:

Feature Encoder: Encoding query plan trees using a feature encoder.

Compressor: Obtaining compressed embeddings using a bidirectional LSTM based compressor.

Ensemble model: Ensemble model incorporating Bayesian neural networks, active learning, and transfer learning.

Empirical research: this study showcasing enhanced accuracy and computational efficiency compared to the state of the art.

This indicates that the framework has better performance in cardinality and cost estimation tasks.