Graph Based Cardinality Estimation from Query Constraints

Chun Wang
University of Michigan - Ann Arbor
Ann Arbor, Michigan
chunwc@umich.edu

Wanning He
University of Michigan - Ann Arbor
Ann Arbor, Michigan
hwanning@umich.edu

Changwen Xu
University of Michigan - Ann Arbor
Ann Arbor, Michigan
changwex@umich.edu

1 INTRODUCTION

Database query optimization aims to select the most efficient execution plan to minimize resource usage and improve performance. Cardinality estimation is essential in this process, as it predicts the number of records that will match a query's conditions. These estimates directly impact the optimizer's choice of execution plans.

Given a database schema along with a set of queries and their associated cardinality constraints, the objective of cardinality estimation is to construct a database instance that adheres to the schema while satisfying all constraints. Simultaneously, the system must deliver accurate cardinality estimates for future query executions. These estimates assist the optimizer in selecting the best execution strategies, optimizing resource utilization, and enhancing query efficiency. In this project, we aim to learn the joint distribution of the origin database from the workload by Graph Neural Network (GNN) according to the cardinality of queries in the workload.

2 RELATED WORK

Previous research on the cardinality estimation problem has predominantly employed symbolic methods and Integer Linear Programming (ILP) to address Constraint Satisfaction Problems (CSP).

The symbolic CSP-based approach [2, 7] starts with a symbolic representation of the database and translates Approximate Query Processing (AQP) inputs into constraints with symbolic variables. However, this method faces two significant drawbacks: (1) it does not always guarantee the generation of a unique database instance, and (2) the time required to solve the CSP increases exponentially with the size of the database.

Conversely, the ILP-based approach [1, 3] formulates the cardinality constraints as a system of linear equations and employs standard integer linear solvers to resolve them. While this approach provides exact solutions, it incurs a significant time overhead, with resource demands growing exponentially as the number of attributes increases.

Recent deep learning-based work [10, 12] utilizes a supervised deep autoregressive model to obtain unbiased samples from the probability density function (PDF), representing the neural network capable of efficiently handling large-scale query constraints. Nonetheless, the inherent sparsity of the PDF in high-dimensional spaces still leads to notable errors in both the sampling process and model accuracy.

3 DATASETS

In our experiments, each data would be a SQL query that is randomly generated. Similar to SAM's [12], we plan to test our algorithm on 1) two single-relation datasets and 2) one multi-relation dataset. The DMV dataset [13] comprising 11.6M tuples and 11 columns, and the Census dataset [8] includes 48K tuples and 14 columns. Each dataset is queried with a workload of 20K queries, with an additional test of 100K queries to assess scalability. For the multi-relation database, we use the IMDB dataset [5], which provides film and television-related data. Its schema follows the JOB-light benchmark [6], involving six joined relations, with a full outer join resulting in approximately 2×10^{12} tuples. The query workload comprises 100K queries, featuring 0 to 2 joins, where filters are randomly applied to a subset of columns in each base relation. The number of filters per relation is also selected at random.

The selected datasets are well-suited for our task as they provide diverse relational structures and query workloads, allowing us to thoroughly evaluate the performance and scalability of our model. The DMV and Census datasets offer large-scale single-relation databases, while the IMDB dataset, with its multi-relation schema and complex join operations, is ideal for testing the algorithm's efficiency in handling intricate queries involving joins and filters.

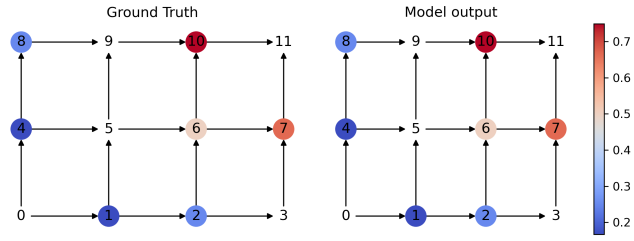

Figure 1: 2D demonstration of modeling the query space with GNN, with node $(x_i, y_i) = P(X \leq x_i, Y \leq y_i)$ is monotonically increasing along each dimension.

4 METHODOLOGY

In this work, we propose employing Graph Neural Networks (GNNs) to model the cumulative distribution function (CDF) within a multi-dimensional query space. Each node in the graph will represent a cumulative probability $F_X(x) = P(X_1 \leq x_1, X_2 \leq x_2, \cdots, X_d \leq x_d)$ where d is the dimension of query space and $x = [x_1, x_2, \cdots, x_d]$ is a point of multi-dimensional space D, and the graph is structured in a directed manner to ensure that node activations reflect monotonic increases in selectivity. The model will be trained in a semi-supervised manner, thus mapping the graph to the label of the node given the query, as is shown in Figure 1. The cardinality estimation task involves predicting the selectivity of queries over database tables, typically constrained by ranges on different attributes. By learning the joint CDF of the data through a GNN, we aim to capture the column dependency between query attributes in a scalable way.

We plan to try different GNN architectures, including Graph Convolutional Networks (GCN) [4], Graph Isomorphism Network (GIN) [11], and Graph Attention Networks (GAT) [9]. We will utilize the trained GNN to sample attribute values that respect the learned joint distribution to generate database instances conforming to the schema and satisfying all constraints. By iteratively sampling attribute values based on the GNN's learned distribution, we generate a database instance that adheres to the schema constraints. The sampling process ensures consistency between generated values and the cardinality constraints derived from the query workload.

5 EVALUATION

We assess the similarity between the synthetic and original database instances using two key metrics:

(1) Q-Error, which measures how well the synthetic database satisfies the given cardinality constraints and generalizes to queries beyond the input query workload.

$$Q\text{-}Error = \frac{\max(\hat{C}, C)}{\min(\hat{C}, C)} \in [1, +\infty) \tag{1}$$

Where:

- C is the actual cardinality (the true number of records returned by the query);
- $\hat{C}$ is the estimated cardinality;
- $Q\text{-}Error = 1$ if the estimate is exactly correct.

(2) Cross-Entropy, which quantifies the statistical divergence between the generated instance and the original. A lower cross entropy indicates a closer alignment between the two database instances.

$$H(P, Q) = -\sum_{i} P(x_i) \log Q(x_i) \tag{2}$$

Where:

- $P(x_i)$ is the true probability distribution of event x_i,
- $Q(x_i)$ is the predicted probability distribution of event x_i.

REFERENCES

[1] Arvind Arasu, Raghav Kaushik, and Jian Li. 2011. Data generation using declarative constraints. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data* (Athens, Greece) *(SIGMOD '11)*. Association for Computing Machinery, New York, NY, USA, 685–696. https://doi.org/10.1145/1989323.1989395

[2] Carsten Binnig, Donald Kossmann, Eric Lo, and M. Tamer Özsu. 2007. QAGen: generating query-aware test databases. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data* (Beijing, China) *(SIGMOD '07)*. Association for Computing Machinery, New York, NY, USA, 341–352. https://doi.org/10.1145/1247480.1247520

[3] Amir Gilad, Shweta Patwa, and Ashwin Machanavajjhala. 2021. Synthesizing Linked Data Under Cardinality and Integrity Constraints. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, New York, NY, USA, 619–631. https://doi.org/10.1145/3448016.3457242

[4] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[5] Viktor Leis, Andrey Gubichev, Atanas Mirchev, Peter Boncz, Alfons Kemper, and Thomas Neumann. 2015. How good are query optimizers, really? *Proceedings of the VLDB Endowment* 9, 3 (2015), 204–215.

[6] Viktor Leis, Bernhard Radke, Andrey Gubichev, Atanas Mirchev, Peter Boncz, Alfons Kemper, and Thomas Neumann. 2018. Query optimization through the looking glass, and what we found running the join order benchmark. *The VLDB Journal* 27 (2018), 643–668.

[7] Eric Lo, Nick Cheng, and Wing-Kai Hon. 2010. Generating databases for query workloads. *Proc. VLDB Endow.* 3, 1–2 (Sept. 2010), 848–859. https://doi.org/10.14778/1920841.1920950

[8] Kolby Nottingham Markelle Kelly, Rachel Longjohn. [n.d.]. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/index.php Accessed: 2024-09-26.

[9] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[10] Jiayi Wang, Chengliang Chai, Jiabin Liu, and Guoliang Li. 2021. FACE: a normalizing flow based cardinality estimator. *Proc. VLDB Endow.* 15, 1 (Sept. 2021), 72–84. https://doi.org/10.14778/3485450.3485458

[11] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[12] Jingyi Yang, Peizhi Wu, Gao Cong, Tieying Zhang, and Xiao He. 2022. SAM: Database Generation from Query Workloads with Supervised Autoregressive Models. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 1542–1555. https://doi.org/10.1145/3514221.3526168

[13] Federico Zanettin. 2019. State of New York. Vehicle, snowmobile, and boat registrations. https://catalog.data.gov/dataset/vehicle-snowmobile-and-boat-registrations Accessed: 2024-09-26.