Advances and Challenges in Machine Learning-Based Cardinality Estimation for Database Query Optimization

Zhibin Guan

School of Software Engineering, Sichuan University, Chengdu, China

Abstract. Accurate cardinality estimation is critical for optimizing database queries, yet traditional methods often fail to provide reliable predictions in the face of complex queries, skewed data distributions, and high-dimensional schemas. As data volumes and query complexity grow, more robust and adaptive estimation techniques have become essential for maintaining efficient query performance. This paper surveys recent advancements in machine learning-based cardinality estimation methods, categorizing them into three main types: query-driven, data-driven, and hybrid models. Each approach is analyzed in terms of its model architecture, training strategies, and predictive performance. Notable techniques include PostCENN's integration into PostgreSQL, FACE's use of normalizing flows, and UAE's autoregressive learning across data and query distributions. Comparative evaluations highlight how these models address specific challenges in cardinality estimation. The results show that while machine learning methods significantly reduce estimation errors, they also share limitations such as sensitivity to workload drift, scalability challenges with large schemas, and poor generalization in long-tail query regions. To address these, this paper proposed future directions including Bayesian updating, sparse factor graph modeling, and active query synthesis. These innovations hold promise for building more accurate, scalable, and adaptive estimators that can enhance database system performance under diverse and dynamic workloads.

1 Introduction

Database management systems play a critical role in modern information technology, serving as the backbone for data storage, retrieval, and manipulation across countless applications. One of the most important tasks in database query optimization is cardinality estimation, which refers to the process of predicting how many tuples will satisfy a given query predicate. Accurate cardinality estimation is essential because it directly impacts the efficiency of query execution plans selected by the optimizer. When estimations are inaccurate, query processors may choose suboptimal execution plans, resulting in slow query response times and excessive resource consumption. Therefore, exploring more effective methods for cardinality

2022141461143@stu.scu.edu.cn

estimation has become an urgent research focus in the database community as data volumes continue to grow exponentially.

Traditionally, database systems have relied on Alon, Martias and Szegedy (AMS)-based methods for cardinality estimation. The pioneering work by Flajolet introduced the approach of estimating cardinalities in the range of millions with a comparatively high accuracy using only $log_2 log_2$ of memory [1]. Subsequently, Flajolet proposed more sophisticated techniques such as SuperLogLog [1], and HyperLogLog [2]. Additionally, Heule developed the HyperLogLog++ algorithm that could decrease the space cost and increase the estimation accuracy of the original algorithm [3]. However, these traditional approaches suffer from several limitations, including the curse of dimensionality, inability to capture complex data correlations, and poor performance on skewed data distributions. Furthermore, their effectiveness deteriorates significantly when dealing with complex queries involving multiple joins and predicates.

In recent years, researchers have made significant progress in applying machine learning to cardinality estimation, advancing the field through three main technical pathways: query-driven methods prioritizing inference speed, data-driven approaches focusing on distribution modeling, and hybrid frameworks that balance both objectives [4]. Woltmann et al. developed PostCENN, a model that deeply integrates machine learning into database systems, achieving a 76% reduction in estimation errors on the IMDB dataset through SQL-extended lifecycle management [5]. Davitkova et al. introduced LMKG, a hierarchical dependency graph model designed to stabilize estimation accuracy for RDF data, particularly in knowledge graph scenarios [6]. For text-based approximate queries, researchers proposed DREAM, which employs Long Short-Term Memory (LSTM) networks along with TEDDY/SODDY preprocessing algorithms, boosting training data generation speed nearly tenfold [7]. Wu and Cong designed UAE, a unified training strategy that combines data distribution learning and query supervision, demonstrating superior performance on complex queries [8]. Researchers behind FACE validated the effectiveness of normalizing flow models by introducing a spline dequantization technique, which limited maximum errors to 3.0 on the Power dataset [9]. In addition, the developers of Fauce incorporated uncertainty quantification into their model, enabling automatic detection of data shifts and supporting incremental retraining—a critical advancement for dynamic database environments [10]. Finally, a segmentation-based framework employing PCA-Kmeans partitioning achieved a mean Q-error of 1.31 for similarity queries [11]. These breakthroughs underscore the necessity of synthesizing diverse technical innovations—ranging from system-level integration and architectural specialization to adaptive learning methodologies—to shape the next generation of cardinality estimation techniques.

The remainder of this review is organized as follows. Section 2 presents a detailed examination of various machine learning methods for cardinality estimation, systematically describing each model's architecture and prediction workflow. This part provides step-by-step explanations of how these models transform query features into cardinality predictions. Section 3 offers a thorough discussion and comparative analysis of the experimental results, highlighting the strengths and limitations of each approach across different database environments and query complexities. This section also identifies existing challenges and areas requiring further improvement. Finally, Section 4 concludes the paper with a summary of key findings and presents the perspective on future research directions, including potential integration with traditional methods and adaptations for emerging database paradigms.

2 Method

2.1 Query-driven models

2.1.1 PostgreSQL with Cardinality Estimation using Neural Network (PostCENN)

Woltmann et al. proposed PostCENN [5], a query-driven cardinality estimator that seamlessly integrates machine learning models into PostgreSQL. In this approach, local ML models are created for specific segments of the database schema, striking a balance between predictive accuracy and storage overhead. By gathering training data through representative queries, PostCENN captures intricate mappings between query predicates and actual result sizes. To reduce the training cost, the system leverages a pre-aggregation strategy that computes partial statistics within the database, thereby expediting data sampling. Once trained, these models are registered in the optimizer as first-class entities: if an incoming query aligns with a particular local model's domain, the query planner employs the model's predictions to enhance execution planning; otherwise, PostgreSQL's native estimators act as a fallback. In essence, PostCENN's design ensures that machine learning benefits can be harnessed selectively, minimizing disruption to PostgreSQL's traditional architecture. This hybrid mechanism not only improves cardinality estimates but also preserves the reliability and versatility of established query optimization workflows.

2.1.2 Deep Cardinality Estimation of Approximate substring M-queries (Dream)

Kwon et al. proposed Dream [7], a query-driven cardinality estimation model tailored for approximate substring queries in text databases. Dream first builds a comprehensive training set by running representative queries and recording their actual result sizes at multiple edit-distance thresholds. This data is then augmented using "prefix-aug" techniques, which capture cardinalities of all prefixes of each query string. To effectively model sequences of varying lengths, Dream adopts an LSTM encoder that processes each character step by step, sharing parameters across positions and thus learning nuanced correlations within the query string. A Feedforward Neural Network (FNN) decoder then predicts the final cardinalities for the full query. To optimize training, Dream employs the "packed learning" method: it processes all prefixes of a query during a single reducing training overhead while preserving accuracy gains. Once deployed, Dream receives a new query string and distance threshold, encodes it via the LSTM layer, and outputs a cardinality estimate through the FNN decoder. By focusing on the queries themselves and thoroughly analyzing their string-based constraints, Dream offers a powerful solution for query optimization in text-centric or bioinformatic databases, especially where direct substring matching under edit-distance conditions is critical.

2.1.3 Fast and Uncertainty-aware Cardinality Estimator (FAUCE)

Liu et al. proposed Fauce [10], a query-driven cardinality estimation framework that employs deep ensembles to incorporate uncertainty into its predictions. First, Fauce generates rich query representations by analyzing table-level join relationships and fine-grained column dependencies, yielding embeddings that capture both local and global correlations. Next, an ensemble of neural networks is trained to estimate query cardinalities, allowing the model to quantify uncertainty in two dimensions: model uncertainty (stemming from parameters) and data uncertainty (due to possible shifts in distribution). By integrating these uncertainty signals, Fauce can dynamically adjust its estimations, especially for queries in complex or under-represented regions of the workload. Such an approach proves advantageous in dynamic database environments, where data and queries evolve continuously and significant misestimation can lead to costly execution plans. Moreover, Fauce's ensemble structure

supports incremental training, letting the system incorporate new training samples without retraining the entire model from scratch. Ultimately, Fauce aims to stabilize query optimization outcomes through robust cardinality predictions, thus minimizing performance regressions and resource wastage when the optimizer encounters out-of-distribution or highly correlated queries.

2.2 Data-driven models

2.2.1 Flow-based Accurate Cardinality Estimator (FACE)

Wang et al. proposed FACE [9], a data-driven cardinality estimator that leverages normalizing flows to capture the joint distribution of attributes. By mapping complex data distributions into simpler (e.g., Gaussian) spaces, FACE sidesteps high-dimensional embeddings and naturally learns column correlations. For discrete or string-valued attributes, it applies a dequantization step, allowing them to fit seamlessly into the flow-based transformation. During inference, FACE integrates the learned density over the relevant attribute ranges, delivering precise selectivity estimates for both simple and complex queries. Since it is trained only on the underlying data, FACE stays flexible to unseen queries and remains robust in high-dimensional or large-domain scenarios, making it especially useful for data-driven optimization in diverse database applications.

2.2.2 PCA-Kmeans

Sun et al. proposed a data-driven segmentation-based estimator that combines PCA and K-means clustering to handle cardinality estimation for similarity queries [11]. First, the method applies Principal Component Analysis (PCA) to the underlying dataset, projecting high-dimensional objects onto a smaller latent space while retaining crucial variance. Next, it employs K-means clustering in this reduced-dimensional space to segment data into clusters with more homogeneous distributions. Each cluster is then modeled independently to predict how many points fall within a given similarity threshold. By tailoring local models—often simpler or more specialized—this approach can capture fine-grained structures that a single global estimator might miss. Empirically, the PCA-Kmeans framework demonstrates strong performance, achieving near-constant Q-errors across varied similarity thresholds. Because it relies purely on data and unsupervised segmentation, it adapts well to diverse workloads without needing explicit query examples.

2.3 Hybrid cardinality estimation models

2.3.1 Learned Models for Cardinality Estimation in Knowledge Graphs (LMKG)

Davitkova et al. proposed LMKG [6], a hybrid approach that combines data-driven and query-driven techniques for cardinality estimation in knowledge graphs. LMKG first encodes SPARQL queries as subgraph patterns, capturing structural relationships among predicates and objects. Then, it applies both supervised neural networks and unsupervised autoregressive models to learn how these subgraph patterns map to actual cardinalities. In particular, local-models are trained on smaller graph components, which are subsequently combined by a global model to handle more complex query structures. This hierarchical design allows LMKG to capture subtle correlations across different parts of the RDF graph while stabilizing performance for a variety of query types. During query optimization, LMKG leverages learned embeddings and probability distributions to provide accurate

cardinality predictions, especially for star- and chain-shaped queries. Owing to its dual reliance on both data statistics and workload samples, LMKG is better at dealing with sparse or highly correlated subgraph patterns than purely data-driven estimators.

2.3.2 Unified Auto-regressive Estimator (UAE)

Wu and Cong proposed UAE [8], a hybrid cardinality estimator that learns from both data distributions and query workloads. Within an autoregressive framework, UAE models each attribute's probability conditioned on preceding attributes, thus capturing detailed inter-column dependencies. Meanwhile, it embeds query supervision through a differentiable sampling step (via Gumbel-Softmax), handling discrete predicates without disrupting gradient flows. As a result, UAE can focus more on difficult query regions, refining its estimates accordingly. Once trained, the model supports incremental updates for new data or new query patterns, making it suitable for evolving workloads. By coupling data-driven and query-driven signals in a single model, UAE achieves greater accuracy and robustness than purely one-sided approaches, delivering more reliable selectivity predictions across diverse query scenarios.

3 Discussion

3.1 Shared limitations and challenges

3.1.1 Workload and Data Drift

Most learned estimators rely on the assumption that future queries and underlying data distributions will closely resemble those seen during training. Unfortunately, this is rarely the case in real-world environments, where databases are dynamic: new data is constantly added, attribute meanings evolve, and analysts introduce previously unseen query patterns. Under such conditions, the estimators often fail silently, resulting in drastic increases in Q-errors—sometimes by factors of 100 or more—leading to extremely inefficient query plans. While FAUCE uses model ensemble variance to hint at uncertainty, none of the current models offer a comprehensive, low-cost mechanism to continuously monitor and adapt to drift. This lack of robustness against change remains a major obstacle to deployment in volatile environments.

3.1.2 High Dimensionality and Strong Correlation

As enterprise databases scale, schemas often include hundreds of columns, many of which participate in multi-table joins. This high dimensionality creates significant challenges for models like UAE, FACE, and LMKG. Their parameter counts typically grow quadratically with the number of features, quickly exhausting GPU memory and leading to longer training and inference times. Additionally, complex cross-table correlations often span distant attributes that aren't easily captured through fixed ordering or simple embedding techniques. When memory or computational constraints force models to ignore or simplify such dependencies, critical correlation patterns—especially those that influence rare but impactful queries—are lost, reducing the effectiveness of the estimators in high-stakes scenarios.

3.1.3 Sparce or Long-Tail Query Regions

Query-driven models tend to perform best on queries similar to those seen during training, but struggle to generalize to uncommon conditions—such as rarely used values, tight similarity thresholds, or deep predicate ranges. Conversely, data-driven models rely on learned distributions to estimate cardinalities for novel queries, but their performance deteriorates when dealing with very small data subspaces or abrupt changes in distribution boundaries. As a result, both families of models share a vulnerability: they often produce the least accurate estimates precisely in the most critical cases—highly selective or skewed queries—where poor predictions can lead to substantial resource overuse and query slowdowns.

3.2 Future Prospects and Theoretical Remedies

3.2.1 Continual-Bayesian Updating for Drift

One promising direction is the integration of lightweight Bayesian adapters into the final layers of the prediction model. These adapters can be updated incrementally using small, recent batches of queries and their true result sizes. Through variational Bayes, they adjust the model's predictions without overwriting the original training knowledge. This strategy allows the model to "bend" towards new data while retaining stability. A simple statistical test, such as a one-sided CUSUM on recent prediction errors, can trigger updates when significant changes are detected. Because the adapters are compact, they can be retrained quickly—within milliseconds—making this approach suitable for online transactional processing (OLTP) systems with strict latency requirements. Furthermore, online Bayesian learning theory provides error bounds, ensuring that the model remains reliable even under substantial distribution shifts.

3.2.2 Composable Sparse Factor-Graphs for Large Schemas

Another effective strategy for coping with high-dimensional schemas involves breaking down the attribute space into smaller, correlated subsets—or "micro-cliques"—using structure-learning methods such as Chow–Liu trees or score-based DAG algorithms. Each clique can be modeled with a smaller, more efficient density estimation module. These modules are then linked using a sparse attention transformer, which only allows information flow between statistically significant relationships. This architecture reduces the model's parameter count and computational cost while maintaining accuracy. In fact, theoretical guarantees from junction-tree factorization suggest that estimation errors decay exponentially with clique size. Preliminary results from synthetic datasets with hundreds of attributes show that this method can reduce memory usage by more than 85%, with no loss in estimation accuracy—indicating substantial potential for scalability.

3.2.3 Active Query Synthesis and Importance-Weighted Training for Long-Tail Coverage

To better handle long-tail and rare query scenarios, active learning strategies can be employed. Specifically, the model can generate synthetic queries in areas of high uncertainty—where predictive variance is high or ensemble disagreement is greatest. These queries are run on a stratified sample of the database, and the resulting cardinalities are used to fine-tune the model using importance-weighted loss functions. This targeted retraining

emphasizes accuracy in the most challenging regions of the query space. Additionally, the synthetic query workload can serve as an efficient and automatic benchmark to validate the model's performance after system updates or hardware changes. Theoretical results show that such active sampling minimizes the worst-case estimation error and approaches the optimal bound for a given budget of training queries.

4 Conclusion

In conclusion, as underscored in the introduction, accurate cardinality estimation remains a cornerstone of query optimization, and the diverse machine learning approaches surveyed in this paper demonstrate substantial progress in tackling this core, long-standing database challenge. This review presented a comprehensive survey of state-of-the-art ML-based cardinality estimators, spanning query-driven, data-driven, and hybrid paradigms, and offered comparative insights into their model architectures and performance across different database environments and query complexities. These techniques significantly advance the field by capturing complex data correlations and reducing estimation error, yet important limitations remain: current learned estimators can be sensitive to data drift, face scalability issues in high-dimensional schemas, and underperform on long-tail or highly selective queries. To address these challenges, this review highlighted several promising directions for future research, including continual Bayesian updating to adapt models to evolving data distributions, composable sparse factor graph models to manage large schema complexities, and active query synthesis to improve coverage and accuracy for rare query patterns. By pursuing these avenues, future cardinality estimators can become even more robust and adaptive, enabling query optimizers to consistently choose efficient execution plans even in large-scale, dynamic, and complex database environments.

References

1. Durand, M., Flajolet, P.: 'LogLog counting of large cardinalities', Eur. Symp. Algorithms, 2003, pp. 605–617
2. Flajolet, P., Fusy, É., Gandouet, O., et al.: 'HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm', Discrete Math. Theor. Comput. Sci., 2008, AH(1), pp. 127–146
3. Heule, S., Nunkesser, M., Hall, A.: 'HyperLogLog in practice: algorithmic engineering of a state of the art cardinality estimation algorithm'. Proc. Int. Conf. Extending Database Technol., 2013, pp. 683–692
4. Kim, K., Jeong, J., Seo, I., et al.: 'Learned Cardinality Estimation: An In-depth Study'. Proc. ACM SIGMOD Int. Conf. Manage. Data, Philadelphia, PA, USA, June 2022, 14 pages.
5. Woltmann, L., Olwig, D., Hartmann, C., et al.: 'PostCENN: PostgreSQL with Machine Learning Models for Cardinality Estimation', PVLDB, 2021, 14(12), pp. 2715–2718.
6. Davitkova, A., Gjurovski, D., Michel, S.: 'LMKG: learned models for cardinality estimation in knowledge graphs'. arXiv Prepr., 2021, arXiv:2102.10588
7. Kwon, S., Jung, W., Shim, K.: 'Cardinality Estimation of Approximate Substring Queries using Deep Learning', PVLDB, 2022, 15(11), pp. 3145–3157
8. Wu, P., Cong, G.: 'A Unified Deep Model of Learning from both Data and Queries for Cardinality Estimation'. Proc. ACM SIGMOD Int. Conf. Manage. Data, Virtual Event, China, June 2021, 14 pages.

9. Wang, J., Chai, C., Liu, J., et al.: 'FACE: A Normalizing Flow based Cardinality Estimator', PVLDB, 2022, 15(1), pp. 72–84.
10. Liu, J., Dong, W., Zhou, Q., et al.: 'Fauce: Fast and Accurate Deep Ensembles with Uncertainty for Cardinality Estimation', PVLDB, 2021, 14(11), pp. 1950–1963.
11. Sun, J., Li, G., Tang, N.: 'Learned Cardinality Estimation for Similarity Queries'. Proc. ACM SIGMOD Int. Conf. Manage. Data, Virtual Event, China, June 2021, 13 pages.