Technical Perspective: Join Size Bounds using ℓ_p-Norms on Degree Sequences

Hung Q. Ngo
RelationalAI Inc.

Cardinality estimation is one of the most, if not the most, important components of the query optimization pipeline: these estimates are the main parameters in the cost-estimators of query plans, parallel query processing, and in computing budgets for in-memory query processing. After more than half a century of theory and implementation of relational database systems, whose global market size is on the order of 100 billions USD, commercial database systems still routinely misestimate cardinalities by a factor of 1000 or more. Two major reasons for the misestimation are: (1) relational RDBMSs employ estimators that make distributional assumptions about the data (such as uniformity) which may not hold in real workloads or in standard benchmarks, and (2) traditional estimators treat selection predicates independently, leading to error accumulation on large queries. Hence, estimation errors grow exponentially as the number of joins increases.

Slowly emerging in the past 15 years or so from the database theory community, there is a new, information-theoretically sound approach to cardinality estimation that is designed to address both of the weaknesses mentioned above. Briefly, the problem can be formulated as follows. We have a database $\boldsymbol{D}$, and some statistics $s(\boldsymbol{D})$ collected on $\boldsymbol{D}$ such as histograms, functional dependencies, cardinalities, unique value counts per column, and so forth. For an arbitrary query Q, we would like to estimate $|Q(\boldsymbol{D})|$, the number of rows in the output of the query Q on the database $\boldsymbol{D}$ as quickly as possible, without actually executing the query. The cardinality estimator is invoked repeatedly during query planning, for different candidate sub-queries of Q; thus, the estimation has to be extremely fast, and as accurate as possible.

We write $\boldsymbol{D}' \models s(\boldsymbol{D})$ to mean "database $\boldsymbol{D}'$ that has the same statistical profile $s(\boldsymbol{D})$" as $\boldsymbol{D}$, and $\boldsymbol{D}' \sim \mathcal{D}$ to mean "$\boldsymbol{D}'$ drawn from some database distribution $\mathcal{D}$". There are two types of cardinality estimators:

$$\hat{q} \approx \mathop{\mathbb{E}}_{\boldsymbol{D}' \sim \mathcal{D}}[|Q(\boldsymbol{D}')| \mid \boldsymbol{D}' \models s(\boldsymbol{D})] \qquad \text{model-based} \qquad (1)$$

$$\hat{q} \approx \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} |Q(\boldsymbol{D}')| \qquad \text{model-free} \qquad (2)$$

The *average-case estimator* (1), which represents traditional estimators, assumes that we know the distribution $\mathcal{D}$ of the data, often manisfested in practice with uniformity, independence, and hard-coded selectivity constants. The *pessimistic estimator* (2) upper bounds the output size of query Q over all databases $\boldsymbol{D}'$ having the same statistical profile $s(\boldsymbol{D})$. The upper bound (2) is then modeled by an optimization problem over entropic functions and polymatroids. Pessimistic estimators are robust to outliers. Heavy skews or corner cases do not affect the estimator and thus they help avoid query plans which explode in runtime under bad inputs. The main drawback of pessimistic estimators is that they can be too pessimistic. The way to deal with that is to collect more statistics, enriching $s(\boldsymbol{D})$, which will push the upper-bound down, closer to the real output size.

The paper by Abo Khamis, Nakos, Olteanu, and Suciu made a significant advance in making the pessimistic estimators more practical, *and* making them a better guide for worst-case optimal join algorithms. On the cardinality estimation front, while staying within the same information-theoretic framework carved by AGM and PANDA, they showed that we can enrich $s(\boldsymbol{D})$ by collecting ℓ_p-norms of degree sequences for arbitrary values of p, while previous works can only deal with $p \in \{0, 1, \infty\}$. Since having the first $O(1)$-moments often suffices to approximate a typical data distribution, the ability to support ℓ_p-norms for arbitrary p resolves a major question of "which statsitics should we put in $\boldsymbol{s}(\boldsymbol{D})$?" Even for a simple two-relation join query, the new statistics already give rise to non-trivial estimates. On the join-algorithm front, the output size bounds are lowered as we have more constraints in $s(\boldsymbol{D})$; for some classes of inputs, the bounds can be asymptotically lowered. Satisfyingly, the paper showed how to design a query evaluation algorithm meeting the new bound by using PANDA as a black-box. Last but not least, both in this paper and in a followup paper to appear in SIGMOD 2025, the authors validate their work by implementing the new cardinality estimator and showed that it does not only improve the cardinality estimates in real datasets, but also leads to better query plans in a wide variety of practical settings.

I am very excited about this work for several reasons. First, as mentioned above it showed that the information-theoretic framework can deal with a very rich class of input statistics, and retains the matching worst-case optimal join algorithms. Second, experimental results are very promising, showing that the new estimator can be practical in real-world settings. Third, the work opens up interesting and fundamental questions for future research: (1) what other classes of statistics can we put in $s(\boldsymbol{D})$? (2) how do we compute the new bounds efficiently? (3) how do we design query evaluation algorithms that meet the new bounds?