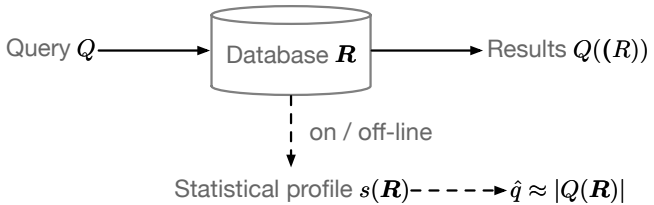
Recent Advances in Pessimistic Cardinality Estimation

wbai@seu.edu.cn

July 29, 2025

Outline

Cardinality Estimation at First Glance



Key question: How to come up with an **accurate** estimate $\hat{q}$ of $|Q(\boldsymbol{R})|$ from the profile $s(\boldsymbol{R})$, as **quickly** as possible?

The Achilles Heel

Quoted from Guy Lohman [13]: Is Query Optimization a "Solved" Problem?

Everything in cost estimation depends upon how many rows will be processed, so the entire cost model is predicated upon the cardinality model.

...

In my experience, the cost model may introduce errors of at most 30% for a given cardinality, but the cardinality model can quite easily introduce errors of many orders of magnitude!

A Wish List

Following [2], the good cardinality estimator (*i.e.*, CE) should have:

- **Accuracy/Speed/Memory**: Good accuracy and efficiency.
- **Locality**: Use statistics computed separately on each input relation.
- **Composition**: Estimate from subquery estimates (*i.e.*, optimal substructure).
- **Combination**: Combine multiple statistics sources for better estimates.
- **Incremental Updates**: Update statistics incrementally when data changes.
- **Guarantees**: Provide theoretical guarantees for reasoning about decisions.

Review: Selective Estimation

Ngo [14] summarized System-R-style selective estimation approaches as follows:

Predicate p	**Estimated size** $s(p)$	**Note**
$\neg p'$	$1 - s(p')$	-
$p_1 \wedge p_2$	$s(p_1) \cdot s(p_2)$	-
$A = c$	$1/d_A$	$d_A = \#$ of dist. vals
$A > c$	$(\max_A - c)/(\max_A - \min_A)$	if known
$c_1 < A < c_2$	$(c_2 - c_1)/(\max_A - \min_A)$	if known
$R(A, B) \bowtie S(B, C)$	$1/\max(d_B^R, d_B^S)$	*i.e.*, $\lvert R \bowtie S\rvert \approx \lvert R\rvert \cdot \lvert S\rvert \cdot s(\bowtie)$
A `IN` L	$\min\{1/2, s(A = c)\lvert L\rvert\}$	$L :=$ literal set
A `IN` Q	$\lvert Q\rvert/\lvert X\rvert$	$Q :=$ subquery; X is cross-prod

Review: Sampling-based Estimation

- **Offline sampling**
  - Pre-computes a uniform sample $R_{\text{sample}} \subseteq R$ of each relation, then estimates the size of the query output from the size of the query output over the sample [1].
  - **Pros**: Accurate (on single-relation query); Good compatibility with SQL operators.
  - **Cons**: Degrades to guessing (when no sampled tuple matches the query).
- **Online sampling**
  - Only sample tuples that join with already sampled tuples [12].
  - **Pros**: Accurate [15] (by resolving sampling collapse of offline sampling).
  - **Cons**: High latency (requires index accessing on every join column).

[1] Estimated using Horvitz-Thompson's formula.

Review: Learned Estimation

- **Data-driven estimators**
  - Train a *generative* ML model to learn the joint distribution $Pr(X, Y, \ldots)$ over all attributes in the database [17, 18].
  - **Pros**: A (lossy) compression of the full outer join of the database relations.
- **Query-driven estimators**
  - Train a *discriminative* model to directly predict cardinality $\text{Est}(Q)$ from a workload of past queries and their true results.
  - **Pros**: Accurate on seen queries, simpler than full generative models.

In general, Learned CE aims to capture the query-data correlation empirically [11, 16].

However, it suffers from (1) distribution shift, (2) intensive memory footprint, (3) limited support for query types and predicates [9].

Prelim: Query Class

We focus on full conjunctive queries (FCQs) [1].

Definition: conjunctive query (CQ) and full conjunctive query (FCQ).

A conjunctive (or join) query $Q(\boldsymbol{X})$ is defined by:

$$Q(\boldsymbol{X}) := \bigwedge_{j \in [m]} R_j(\boldsymbol{Y_j}), \tag{1}$$

where (1) $[m] := \{1, 2, \ldots, m\}$; (2) $\boldsymbol{Y_j}$ is the tuple of variables in R_j, and $\boldsymbol{Y_j}$ must have the same arity as R_j; (3) each variable occuring in $\boldsymbol{X}$ must appear in at least one R_j.

We say $Q(\boldsymbol{X})$ is a **full conjunctive query** when $\boldsymbol{X} = \bigcup_{j \in [m]} \boldsymbol{Y_j}$.

Prelim: Query Class (Example)

source	target
A	B
B	C
C	A

Table: $R(\texttt{source}, \texttt{target})$ represents a directed graph.

- $Q_1(X, Y, Z) = R(X, Y) \wedge R(Y, Z) \wedge R(Z, X)$
- $Q_2(X) = R(X, Y) \wedge R(Y, Z) \wedge R(Z, X)$

In this example, Q_1 (FCQ) list all triangles in the directed graph, while Q_2 (CQ) lists all nodes that are part of a triangle, *i.e.*, $Q_2 = \prod_X Q_1$.

By restricting the query to be FCQ, we can temporarily ignore the projection operations.

Prelim: Statistics

We define statistics over database instance $\boldsymbol{D}$ with schema $\boldsymbol{R}$ following [3].

- **Abstract statistics** $\tau := (f, \boldsymbol{X})$
  - $f : \{\boldsymbol{D_V}\}_{\boldsymbol{V} \subseteq \text{Attr}(\boldsymbol{R})} \to \mathbb{R}$, where $\boldsymbol{D_V}$ is the projection of $\boldsymbol{D}$ on attributes $\boldsymbol{V}$.
  - $\boldsymbol{X}$ is a set of attributes. We say τ is *guarded* by R if $R \in \boldsymbol{R}$ and $\boldsymbol{X} \subseteq \text{Attr}(\boldsymbol{R})$.
  - If τ is guarded by R, then we can evaluate the statistics $\tau_{\boldsymbol{X}} := f(\boldsymbol{D_X})$.

Example: Abstract Statistics

Consider an abstract (cardinality) statistics $\tau_1 = (|\cdot|, \{A\})$, $\tau_2 = (|\cdot|, \{A, B\})$, and $\tau_3 = (|\cdot|, \{C\})$ over database instance $\boldsymbol{D}$ with a single relation $R(A, B)$.

Then τ_1 and τ_2 is guarded by R and we can get $\tau_A = |\prod_A(R)|, \tau_{AB} = |R|$; τ_3 is not guarded by R since $\{C\} \not\subseteq \{A, B\}$.

Prelim: Statistics (Cont.)

- **Concrete statistics** (τ, B).
  - $B \in \mathbb{R}$ is a threshold.
  - We denote by $\boldsymbol{D} \models (\tau, B)$ if τ is guarded by $R \in \boldsymbol{R}$ and $\tau_{\boldsymbol{X}} \leq B$.
- **Statistics set** $(\Sigma, \boldsymbol{B})$.
  - $\Sigma := \{\tau_i\}_{i=1}^{s}$, $\boldsymbol{B} := \{B_i\}_{i=1}^{s}$.
  - We say Σ is *guarded* by the query $\boldsymbol{R}$ if $\forall i \in [s], \exists R \in \boldsymbol{R}, \tau_i$ is guarded by R.
  - We denote by $\boldsymbol{D} \models (\Sigma, \boldsymbol{B})$ if $\forall i \in [s], \boldsymbol{D} \models (\tau_i, B_i)$.

Remark: Cross-relation statistics are not permitted here by definition due to locality concerns (see the wish list mentioned earlier).

Prelim: Pessimistic Cardinality Estimation

Now we can formally define the problem of pessimistic cardinality estimation [3]:

Problem: Pessimistic Cardinality Estimation (PCE)

Given (1) a full conjunctive query $Q(\boldsymbol{X})$ and (2) a set of statistics $(\Sigma, \boldsymbol{B})$ guarded by the (schema of the) query $\boldsymbol{Q}$, find a bound $U \in \mathbb{R}$ such that:

$$\forall \boldsymbol{D} \models (\Sigma, \boldsymbol{B}), \quad |Q(\boldsymbol{D})| \leq U.$$

Tightness: The bound U is *tight* if there exists a database instance $\boldsymbol{D}$ such that $\boldsymbol{D} \models (\Sigma, \boldsymbol{B})$ and $|Q(\boldsymbol{D})| = U$.

PCE Bounds

We now review the intuitions and techniques behind existing PCE bounds.

- **AGM Bound** [6] (FOCS'08).
  - Tight. Fractional edge cover.
- **Chain Bound** [4] (PODS'16).
  - Not tight. Conditional edge cover.
- **Polymatroid Bound** [5] (PODS'17).
  - Not tight. Shannon-type inequalities, degree sequences and ℓ_p-norms.
- **Degree Sequence Bound** [7, 8] (ICDT'23, SIGMOD'23).
  - Tight for Berge-acyclic queries. Degree sequence.

The AGM Bound: A Warm-up Example

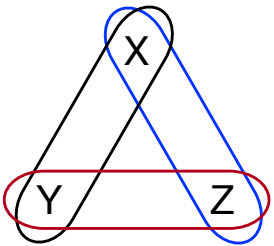


Figure: A hypergraph $\mathcal{H} = (\boldsymbol{V}, \boldsymbol{E})$ for the 3-cycle query: $C_3 = R(X,Y) \wedge S(Y,Z) \wedge T(Z,X)$, where $\boldsymbol{V} = \{R, S, T\}$ and $\boldsymbol{E} = \bigcup \boldsymbol{R} = \{X, Y, Z\}$

It's obvious that $|C_3| \leq |R| \cdot |S| \cdot |T|$.

Then a **key insight** is:
$|R(X,Y) \wedge S(Y,Z) \wedge T(Z,X)| \leq |R(X,Y) \wedge S(Y,Z)|$.

This is because $R(X,Y) \wedge S(Y,Z)$ already **covers** all variables in C_3 and $T(Z,X)$ cannot contribute any new tuples to the result.

The AGM Bound: Integral Edge Cover

We can then reduce PCE to the edge cover problem.

Theorem: Integral Edge Cover for PCE.

For FCQ $Q(\boldsymbol{X}) = \bigwedge_{i=1}^{k} R_i(\boldsymbol{U_i})$, the following inequality holds:

$$|Q| \leq \min_{\mathbf{w} \in \{0,1\}^k} \prod_{i=1}^{k} |R_i|^{w_i} \quad s.t. \quad \forall X \in \boldsymbol{X} : \sum_{i: X \in \boldsymbol{U_i}} w_i \geq 1. \tag{2}$$

In this way, $|C_3| \leq \min(|R| \cdot |S|, |R| \cdot |T|, |S| \cdot |T|)$.

The AGM Bound: Fractional Edge Cover

The AGM bound relaxes the integral edge cover to a **fractional** edge cover.

Theorem: Fractional Edge Cover for PCE (AGM Bound).

For FCQ $Q(\boldsymbol{X}) = \bigwedge_{i=1}^{k} R_i(\boldsymbol{U_i})$, the following inequality holds:

$$|Q| \leq \min_{\mathbf{w} \in \mathbb{R}_+^k} \prod_{i=1}^{k} |R_i|^{w_i} \quad s.t. \quad \forall X \in \boldsymbol{X} : \sum_{i: X \in \boldsymbol{U_i}} w_i \geq 1. \tag{3}$$

In this way, $|C_3| \leq \min(|R| \cdot |S|, |R| \cdot |T|, |S| \cdot |T|, (|R| \cdot |S| \cdot |T|)^{1/2})$.

The AGM Bound: Pros and Cons

- **Pros:**
  - The AGM bound is computable in PTIME in the size of the query Q.
  - The AGM bound is guaranteed to be tight.
- **Cons:**
  - Limited statistics: just the relation cardinalities.
  - Degrades to integral edge cover for acyclic queries.

The Chain Bound: A Warm-up Example

Question: Can we do better by introducing new statistics?

Example: Degree of Attributes

Consider an FCQ $Q(\text{state}, \text{city}, \text{zip}) = R(\text{state}, \text{city}) \wedge S(\text{city}, \text{zip})$.

Assume that it is know that (1) no city contains more than 64 zip codes, and (2) no city is in more than one state.

Then the following assertion holds: $|Q| \leq \min(64 \cdot |R|, 1 \cdot |S|, |R| \cdot |S|)$.

In this example, we improve the AGM bound (*i.e.*, $|R| \cdot |S|$) by introducing new statistics (degree of attributes).

The Chain Bound: Degree Sequence

We now introduce the degree sequence to formalize the concept of *degree of attrs*.

Definition: Degree Sequence

Fix a relation instance R, and two sets of variables $X, Y \subseteq \text{Attrs}(R)$. The degree sequence from X to Y in R is the sequence

$$\deg_R(Y \mid X) := (d_1, d_2, \ldots, d_N),$$

obtained as follows: (1) Compute the domain of X, $\text{Dom}(R.X) = \{x_1, \ldots, x_N\}$, (2) denote by $d_i = |\sigma_{X=x_i}(\prod_{XY}(R))|$ the degree (or frequency) of x_i, and (3) sort the values in the domain $\text{Dom}(R.X)$ such that their degrees are decreasing $d_1 \geq \cdots \geq d_N$.

The Chain Bound: Examples of Degree Sequences

$R =$

X	Y	Z
1	a	...
1	b	...
1	b	...
2	a	...
2	b	...
3	b	...
3	c	...
4	d	...

- $\deg_R(Y \mid X) = (2, 2, 2, 1)$
  - *i.e.,* $(|\{(1, a), (1, b)\}|, |\{(2, a), (2, b)\}|, |\{(3, b), (3, c)\}|, |\{(4, d)\}|)$
- $\deg_R(YZ \mid X) = (3, 2, 2, 1)$
- $\deg_R(YZ \mid XY) = \deg_R(Z \mid XY) = (2, 1, 1, 1, 1, 1, 1)$
- $\deg_R(XYZ \mid \emptyset) = (|R|) = (8)$

ℓ_∞-norms are employed to interprete the max-degree in the sequence.

$$\| \deg_R(Y \mid X) \|_\infty = \left(\sum_{i=1}^{N} d_i^\infty \right)^{1/\infty} = \max_{i \in [N]} d_i = d_1.$$

The Chain Bound: Conditional Edge Cover

Now we can formally define the chain bound.

Theorem: Conditional Edge Cover for PCE (Chain Bound).

For FCQ $Q(\boldsymbol{X}) = \bigwedge_{i=1}^{k} R_i(\boldsymbol{U_i})$, the following inequality holds:

$$|Q| \leq \min_{\mathbf{w} \in \mathbb{R}_+^k} \prod_{i=1}^{k} \| \deg_{R_{j_i}}(\boldsymbol{Y_i} \mid \boldsymbol{X_i}) \|_\infty^{w_i} \quad \text{s.t.} \quad \forall X \in \boldsymbol{X} : \sum_{\substack{i : X \in \boldsymbol{Y_i}, \\ \boldsymbol{X_i} \subseteq \bigcup_{j=0}^{i-1} \boldsymbol{X_j}}} w_i \geq 1. \tag{4}$$

Note that $|R| = \| \deg_R(* \mid \emptyset) \|_\infty$. Therefore, the chain bound strickly generalizes the AGM bound.

The Chain Bound: Revisit the Example

Example: Degree of Attributes

Consider an FCQ $Q(\text{state}, \text{city}, \text{zip}) = R(\text{state}, \text{city}) \wedge S(\text{city}, \text{zip})$.

Assume that it is know that (1) no city contains more than 64 zip codes, and (2) no city is in more than one state.

Then the following assertion holds: $|Q| \leq \min(64 \cdot |R|, 1 \cdot |S|, |R| \cdot |S|)$.

We rewrite the inequality $|Q| \leq 64 \cdot |R|$ with the formal notation.
$|Q| \leq \| \deg_R(\{\text{state,city}\} \mid \emptyset)\|_\infty \cdot \| \deg_S(\{\text{city,zip}\} \mid \{city\})\|_\infty \leq |R| \cdot |S|$.

The existing attributes can serve as conditions for subsequent query operations, thus creating a "**chain**" of dependencies between relations.

The Chain Bound: Pros and Cons

- **Pros:**
  - Introduced max-degree statistics.
  - Outperforms the AGM bound on acyclic queries.
  - When the set of statistics (*i.e.*, max-degrees) is *acyclic*, chain bound is both tight and computable in PTIME.
- **Cons:**
  - The chain introduces the dependencies between relations.
  - Enumerating all possible chains (*i.e.*, query orders) introduce exponential complexity.
  - Chain bound is not tight in general.

The Polymatroid Bound: Intuition & Challenge

- AGM bound and Chain bound only use the ℓ_∞-norms of the degree sequences. Can we do better by using other norms?
- **But**, how to integrate these statistics?

The key is information theory!

The Polymatroid Bound: Background on Information Theory

Let X be a finite random variable, with outcomes $x_1, x_2, \ldots, x_N$ and probability function Pr, its entropy is:

$$h(X) := -\sum_{i=1}^{N} \Pr(x_i) \log \Pr(x_i), \tag{5}$$

where log is in base 2. It always holds that

$$0 \leq h(X) \leq \log N, \quad h(X) = \log N \text{ iff } \forall i \in [N],\ \Pr(x_i) = 1/N. \tag{6}$$

The conditional entropy is defined as:

$$h(U \mid V) := h(UV) - h(V). \tag{7}$$

The Polymatroid Bound: Background on Information Theory (Cont.)

Given n jointly distributed random variables $\boldsymbol{X} = \{X_1, \ldots, X_n\}$, we denote *entropic vector* $\boldsymbol{h} \in \mathbb{R}_+^{2^{[n]}}$ by $h_\alpha := h(\boldsymbol{X}_\alpha)$ for $\alpha \in [n]$, where $\boldsymbol{X}_\alpha$ is the joint random variable $(X_i)_{i\in\alpha}$.

A *polymatroid* is a vector $\boldsymbol{h} \in \mathbb{R}_+^{2^{[n]}}$ that satisfies the basic Shannon inequalities:

- $h(\emptyset) = 0$
- **Monotonicity**: $h(U \cup V) \geq h(U)$
- **Submodularity**: $h(U \cup V) + h(U \cap V) \leq h(U) + h(V)$

Remark: Let $\Gamma_n^* \subseteq \mathbb{R}_+^{2^{[n]}}$ is the set of all *entropic vectors* and $\Gamma_n \subseteq \mathbb{R}_+^{2^{[n]}}$ is the set of all *polymatroids*, then

$$\Gamma_n^* \subseteq \Gamma_n. \tag{8}$$

The Polymatroid Bound: Bridging PCE and Information Theory

According to Eq. 6, the cardinality of a relation $R(\boldsymbol{U})$ is bounded by its entropy:

$$|R| \leq 2^{h(\boldsymbol{U})}. \tag{9}$$

Therefore, we can formulate the PCE problem as an optimization problem:

Problem: Identifying maximum entropy.

$$\max_{\boldsymbol{h} \in \Gamma_n} h(\boldsymbol{U}) \quad s.t. \quad \forall U, V \subseteq \boldsymbol{U}, \begin{cases} h(\emptyset) = 0, \\ h(U \cup V) \geq h(U), \\ h(U \cup V) + h(U \cap V) \leq h(U) + h(V). \end{cases} \tag{10}$$

The Polymatroid Bound: Integrating Statistics

Statistics on the query data can be used to constrain the optimization problem 10.

Theorem: ℓ_p-Inequality

Consider n finite random variables $X_1, \ldots, X_n$, and let relation $R(X_1, \ldots, X_n)$ be their set of outcomes. Then, for any subsets of variables $U, V \subseteq \text{Attrs}(R)$ and any $p \in (0, \infty]$, the following hold [3]:

$$\frac{1}{p} h(U) + h(V \mid U) \leq \log \| \deg_R(V \mid U) \|_p. \tag{11}$$

The Polymatroid Bound: Compute the Bound

Using Eq. 11, we can compute the polymatroid bound as follows:

Problem: Compute polymatroid bound.

Assume we have a set of statistics of ℓ_p-norms $(\Sigma, \boldsymbol{B})$ guarded by the query Q. Then we can compute the polymatroid bound as follows:

$$\max_{\boldsymbol{h} \in \Gamma_n} h(\boldsymbol{U}) \quad s.t. \quad \forall U, V \subseteq \boldsymbol{U}, \begin{cases} h(\emptyset) = 0, \\ h(U \cup V) \geq h(U), \\ h(U \cup V) + h(U \cap V) \leq h(U) + h(V), \\ \forall (\sigma, B) \in (\Sigma, \boldsymbol{B}), \text{ Eq. 11 holds.} \end{cases} \tag{12}$$

The Polymatroid Bound: The Tightness

Recall Expr. 8, $\Gamma_n^* \subseteq \Gamma_n$[2], the computed polymatroid bound may not related to a valid entropic vector. Therefore, the polymatroid bound is not tight in general.

Fortunately, the polymatroid bound is guaranteed to be tight when degree sequences in statistics are *simple*, *i.e.*, $\forall \deg_R(V \mid U), |U| \leq 1$.

[2]Refer to Non-Shannon-type inequalities for more details.

The Polymatroid Bound: Pros and Cons

- **Pros:**
  - The polymatroid bound generalizes both the AGM bound and the Chain bound.
  - It's compatible with group-by (and select distinct) clauses.
  - When all statistics are simple degree sequences, it's provably tight.
- **Cons:**
  - The inference time is exponential in the number of query variables.
  - The polymatroid bound is not tight in general.

The Degree Sequence Bound: A Warm-up Example

Example: The Degree Sequence Bound

Consider an FCQ $J_2(X, Y, Z) = R(X, Y) \wedge S(Y, Z)$, and assume that the degree sequence of $R.Y$ and $S.Y$ are:

$$\deg_R(* \mid Y) = (a_1, a_2, \ldots), \quad \deg_S(* \mid Y) = (b_1, b_2, \ldots).$$

Then the following inequality holds:

$$|J_2| \leq \sum_{i=1}^{\min(|R|,|S|)} a_i \cdot b_i. \tag{13}$$

The Degree Sequence Bound: Intuition

Instead of only accessing ℓ_p-norms of the degree sequences, degree sequence bound accesses the *full* degree sequences of the relations in the query.

For Berge-acyclic queries[3], the degree sequence bound is guaranteed to be tight.

In this case, the degree sequence bound is also better than the chain bound since $\sum_i a_i b_i \leq a_1 \sum_i b_i = a_1|S|$, and similarly $\sum_i a_i b_i \leq |R|b_1$.

[3] Refer to Berge cycles for more details.

The Degree Sequence Bound: Pros and Cons

- **Pros:**
  - The degree sequence bound can be computed in linear time in the size of the (compressed) degree sequences.
  - It's compositional, *i.e.*, can be applied to subqueries.
  - More accurate than density-based estimators.
  - It's provably tight for Berge-acyclic queries.
- **Cons:**
  - It's limited to Berge-acyclic queries.
  - More memory footprint.

Evaluation of SOTA PCE

We report the evaluation results of the state-of-the-art LpBound [19], which builds upon the polymatroid bound [3] and chain bound [4].

System Configuration:

- **Hardware**: Intel Xeon Silver 4214 (48 cores) with 193GB memory, running Debian GNU/Linux 10 (buster).
- **PostgreSQL**: 4GB shared memory, 2GB work memory, 32GB implicit OS cache, and 6 max parallel workers. Indices enabled on primary/foreign keys.
- **LpBound**: Uses DuckDB for statistics computation and HiGHS 1.7.2 [10] for linear program solving.

LpBound: Effectiveness

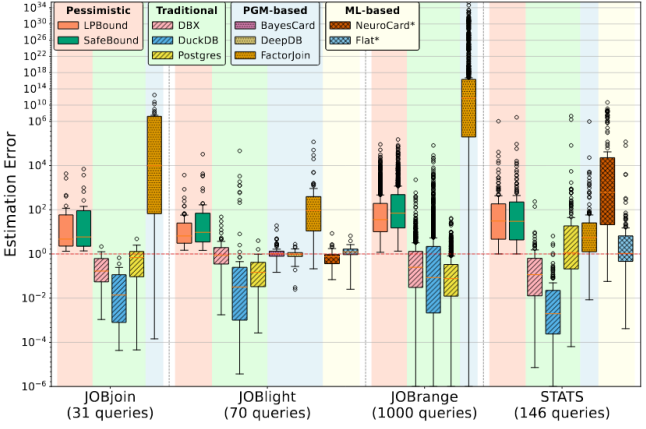


Fig. 5. Estimation errors for JOBJoin, JOBLight, JOBRange, and STATS. For the starred ML-based estimators, we use the errors for JOBLight and STATS reported in the literature [15].

LpBound: Effectiveness (Cont.)

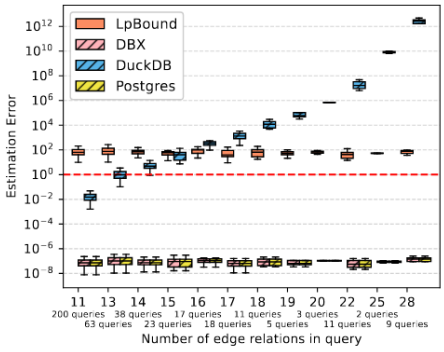


Fig. 6. Estimation errors for the SM cyclic queries.

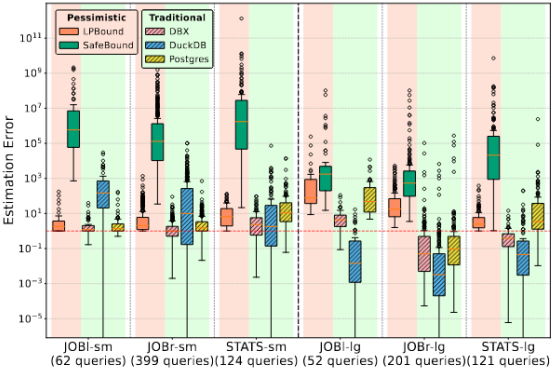


Fig. 7. Estimation errors for group-by queries.

LpBound: Efficiency

Estimator	JOBjoin		JOBlight		STATS	
	Time	Space	Time	Space	Time	Space
LpBound	0.48 / 10.5	0.04	0.36 / 1.5	1.25	0.49 / 1.6	4.76
SafeBound	0.85 / 147.9	0.07	1.28 / 13.0	1.75	1.89 / 5.6	5.94
DbX+	- / 371.7	-	- / 35.3	-	- / 13.3	-
DuckDB+	- / 99.4	-	- / 535.2	-	- / 30.3	-
Postgres+	- / 19.8	<0.001	- / 3.4	0.001	- / 18.7	0.011
FactorJoin	0.66 / 202	31.6	16.7 / 166.5	22.8	35.3 / 626	8.2
BayesCard	- / -	-	3.0 / 21.7	1.6	- / -	-
DeepDB	- / -	-	4.3 / 28.6	34.0	- / -	-
NeuroCard*	- / -	-	18.0 / -	6.9	23.0 / -	337.0
Flat*	- / -	-	8.6 / -	3.4	175.0 / -	310.0

Table 2. Time (ms): average wall-clock times to compute estimates for (i) a sub-query of a query, averaged over all sub-queries of queries / (ii) a query and all its connected sub-queries, averaged over all queries. Space (MB): extra space for data statistics and models. The times (+) are for the entire query optimization task. The numbers (*) are from prior work [15] and only available for JOBlight and STATS. (-) means unavailable data or unsupported workload.

LpBound: Efficiency (Cont.)

Estimator	JOBjoin	JOBlight	JOBrange	STATS	SM
LpBound-ℓ_1	4.57	12.88	44.38	27.84	0.65
LpBound	15.19	20.24	57.9	31.58	1.56
SafeBound	88.56	162.09	209.23	32.04	-
FactorJoin	10068.8	4990.9	5042.7	360.92	-
BayesCard	-	493.36	-	-	-
DeepDB	-	1191.17	-	-	-
NeuroCard *	-	3600	-	-	-
Flat *	-	3060	-	-	-

Table 3. Time (sec) to compute the required statistics for the pessimistic and PGM-based estimators. LpBound-ℓ_1 is LpBound with ℓ_1-norms only. (-) means the system cannot estimate for the respective workload. (*) means the times are from prior work [15], as the code is not available.

LpBound: Efficiency (Cont.)

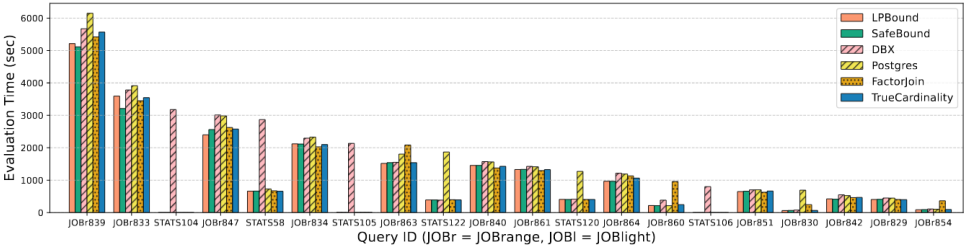


Fig. 8. PostgreS (wall-clock) evaluation time for the 20 most expensive queries in JOBlight, JOBrange, and STATS, when injected the estimates of LpBound, SafeBound, DbX, and PostgreS or the trues cardinalities for all subqueries of the query. The runtimes for STATS 104, 105, 106 are very small when using the estimates of all systems but DbX and therefore not visible.

LpBound: How Many ℓ_p-Norms?

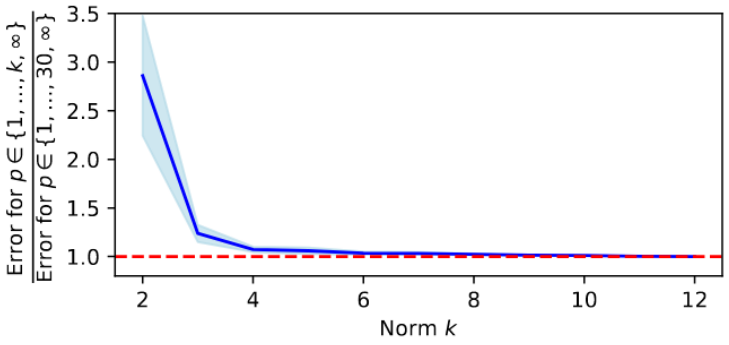


Figure: The amount of useful norms follows the law of diminishing returns: Plotting the division of estimation errors for the norms $\{1, \ldots, k, \infty\}$ and $\{1, \ldots, 30, \infty\}$, averaged over the 70 JOBlight queries.

Back to the Wish List

As summarized by [2]:

- **Accuracy/Speed/Memory**: PCE improves the performance of expensive SQL, but suffers from a regression for cheap queries.
- **Locality**: Only use statistics collected on each relation independently.
- **Composition**: Degree sequence bound is compositional, while polymatroid bound are not (*i.e.*, needs to estimate from scratch).
- **Combination**: PCE can be combined: simply taking the minima.
- **Incremental Updates**: No RDBMS updates its statistics incrementally (because it would slow down OLTP workload, especially because it requires acquiring a lock).
- **Guarantees**: PCE provides one-side guarantee.

References I


Serge Abiteboul, Richard Hull, and Victor Vianu.
Foundations of Databases.
Addison-Wesley, 1995.


Mahmoud Abo Khamis, Kyle Deeds, Dan Olteanu, and Dan Suciu.
Pessimistic cardinality estimation.
ACM SIGMOD Record, 53(4):1–17, 2025.


Mahmoud Abo Khamis, Vasileios Nakos, Dan Olteanu, and Dan Suciu.
Join size bounds using lp-norms on degree sequences.
Proceedings of the ACM on Management of Data, 2(2):1–24, 2024.

References II


Mahmoud Abo Khamis, Hung Q Ngo, and Dan Suciu.
Computing join queries with functional dependencies.
In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 327–342, 2016.


Mahmoud Abo Khamis, Hung Q Ngo, and Dan Suciu.
What do shannon-type inequalities, submodular width, and disjunctive datalog have to do with one another?
In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 429–444, 2017.

References III


Albert Atserias, Martin Grohe, and Dániel Marx.
Size bounds and query plans for relational joins.
SIAM Journal on Computing, 42(4):1737–1767, 2013.


Kyle Deeds, Dan Suciu, Magda Balazinska, and Walter Cai.
Degree sequence bound for join cardinality estimation.
arXiv preprint arXiv:2201.04166, 2022.


Kyle B Deeds, Dan Suciu, and Magdalena Balazinska.
Safebound: A practical system for generating cardinality bounds.
Proceedings of the ACM on Management of Data, 1(1):1–26, 2023.

References IV

Yuxing Han, Ziniu Wu, Peizhi Wu, Rong Zhu, Jingyi Yang, Liang Wei Tan, Kai Zeng, Gao Cong, Yanzhao Qin, Andreas Pfadler, et al.
Cardinality estimation in dbms: A comprehensive benchmark evaluation.
arXiv preprint arXiv:2109.05877, 2021.

Qi Huangfu and JA Julian Hall.
Parallelizing the dual revised simplex method.
Mathematical Programming Computation, 10(1):119–142, 2018.

References V


Kyoungmin Kim, Jisung Jung, In Seo, Wook-Shin Han, Kangwoo Choi, and Jaehyok Chong.
Learned cardinality estimation: An in-depth study.
In *Proceedings of the 2022 international conference on management of data*, pages 1214–1227, 2022.


Feifei Li, Bin Wu, Ke Yi, and Zhuoyue Zhao.
Wander join: Online aggregation via random walks.
In *Proceedings of the 2016 International Conference on Management of Data*, pages 615–629, 2016.

References VI


Guy Lohman.
Is query optimization a "solved" problem.
In *Proc. Workshop on Database Query Optimization*, volume 13, page 10. Oregon Graduate Center Comp. Sci. Tech. Rep, 2014.


Hung Q Ngo.
On an information theoretic approach to cardinality estimation (invited talk).
In *25th International Conference on Database Theory (ICDT 2022)*, pages 1–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022.

References VII

Yeonsu Park, Seongyun Ko, Sourav S Bhowmick, Kyoungmin Kim, Kijae Hong, and Wook-Shin Han.
G-care: A framework for performance benchmarking of cardinality estimation techniques for subgraph matching.
In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1099–1114, 2020.

Xiaoying Wang, Changbo Qu, Weiyuan Wu, Jiannan Wang, and Qingqing Zhou.
Are we ready for learned cardinality estimation?
arXiv preprint arXiv:2012.06743, 2020.

References VIII


Ziniu Wu and Amir Shaikhha.
Bayescard: A unified bayesian framework for cardinality estimation. corr (2020).
arXiv preprint arXiv:2012.14743, 2020.


Zongheng Yang, Amog Kamsetty, Sifei Luan, Eric Liang, Yan Duan, Xi Chen, and Ion Stoica.
Neurocard: one cardinality estimator for all tables.
arXiv preprint arXiv:2006.08109, 2020.


Haozhe Zhang, Christoph Mayer, Mahmoud Abo Khamis, Dan Olteanu, and Dan Suciu.
Lpbound: Pessimistic cardinality estimation using lp-norms of degree sequences.
Proceedings of the ACM on Management of Data, 3(3):1–27, 2025.