RelationalAI

Logic, Algebra, & Category Theory 2025

Optimization Techniques for a Datalog-Inspired Query Language

Hung Q. Ngo

Outline

Computational Problems in `Rel`

```
def Δ_exists { exists( (a,b,c) | E(a,b) and E(b,c) and E(c,a))}
def num_Δ { // 4,159,226
    count[ [a,b,c] : E(a,b) and E(b,c) and E(a,c) and a<b and b<c]
}

def distance[u, v]: {
    min[ [len] :
        len = 0 and u = source and u = v or
        len = min[(w,z) : z = distance[u, w] + 1 and E(w,v)]
    ]
}

// Number of paths of lenght 4, 63,966,233,925,222
def num_4_paths { count[(a,b,c,d,e) : E(a,b) and E(b,c) and E(c,d) and E(d,e)] }
```

There are customer's `Rel` programs with thousands of mutually recursive rules.

Queries in Rel

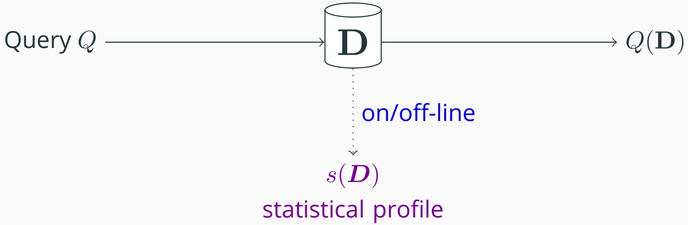
First order logic queries with

- Built-in predicates such as arithmetic ops, hashing, string ops, $\cdots$
- Aggregations (reduce operator such as sum, max, min, $\cdots$)
- Recursion (fixpoint operator) with negation, aggregation

Focus of this talk:

- Conjunctive queries
- Aggregation queries over a conjunctive body

The Main Query Optimization / Evaluation Problem



The Query Optimization Problem

Given Q and $\mathbf{D}$, compute $Q(\mathbf{D})$ in the most efficient (optimal!?) way possible.

Precise Problem Formulation

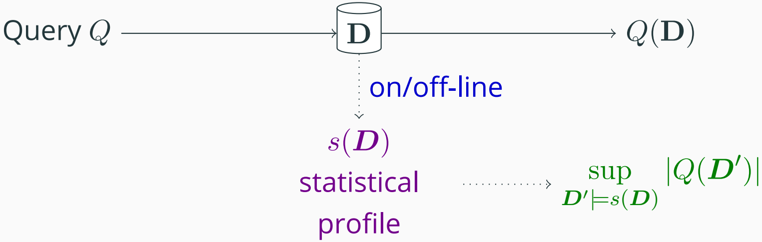
- What do we mean by “query"?
  - Full conjunctive queries
  - Sum-product queries
- What is in the statistical profile $s(\boldsymbol{D})$?
  - Degree constraints
  - $\cdots$ Frequency moment constraints, histograms, samples, ML models
- What do we mean by “optimality"?
  - Worst-Case Optimality
  - Fixed-Parameter Tractability, Fine-Grained Complexity
  - $\cdots$ Instance Optimality

Note: Optimizer designed to work before seeing Q.

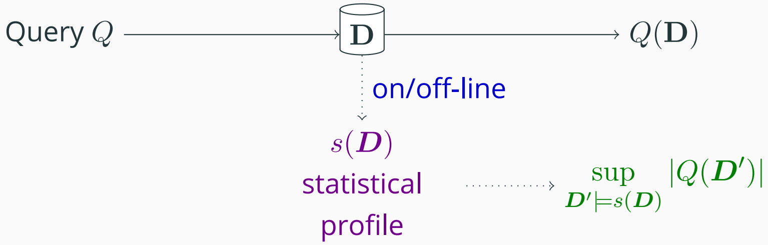
- Optimizer = *Meta-Algorithm* (input: problem, output: algorithm)

Outline

Output Cardinality Bound



Worst-Case Optimal Join (WCOJ) Algorithm



Definition

A "worst-case optimal" join algorithm is an algorithm computing $Q(\boldsymbol{D})$ in time

$$\tilde{O}\left(|\boldsymbol{D}| + \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} |Q(\boldsymbol{D}')|\right)$$

$\tilde{O}$ hides $\log$ and query dependent factors

(For Now) Assume Q is a Full Conjunctive Queries

In a movie database

```
Q(director, actor, movie, actor_age, name) ←
      parent(director, actor)
      ∧ acted_in(actor, movie)
      ∧ director_of(director, movie)
      ∧ age(actor, actor_age) ∧ (20 < actor_age ∨ actor_age != 10)
      ∧ person_name(director, name) ∧ regex_match(".*spiel.*", name)
```

In a graph database with edge relation E,

```
Q(a, b, c) ← E(a, b) ∧ E(a, c) ∧ E(b, c)
```

(For Now) Assume Q is a Full Conjunctive Queries

More generally, $\mathcal{H} = (V, \mathcal{E})$ is the hypergraph of a query:

$$Q(\boldsymbol{X}_V) \leftarrow \bigwedge_{S \in \mathcal{E}} R_S(\boldsymbol{X}_S)$$

For example `Q(a, b, c)` $\leftarrow$ `E(a, b)` $\wedge$ `E(a, c)` $\wedge$ `E(b, c)`

- $V = \{a, b, c\}$
- $\mathcal{H} = (V, \mathcal{E}) = (V, \{ab, ac, bc\})$
- $R_F = E$ for all $F \in \mathcal{E}$.

What is in the Statistical Profile $s(D)$?

$$\text{Query } Q \longrightarrow \mathbf{D} \longrightarrow Q(\mathbf{D})$$

$$\mathbf{D} \xrightarrow{\text{on/off-line}} s(\boldsymbol{D})$$

$s(\boldsymbol{D})$
statistical
profile

$$s(\boldsymbol{D}) \dashrightarrow \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} |Q(\boldsymbol{D}')|$$

Degree constraints

Degree Constraints

[ANS 16, ANS 17]

Relation $R(\text{actor}, \text{movie}, \text{role})$, imagine a frequency vector $\boldsymbol{d}_{\text{actor}}$ (billions of entries)

actor	movie	role
alice		
bob		
bob		
bob		
bob		
carol		
carol		

$$d_{\text{actor}}(\text{alice}) = 1 \qquad d_{\text{actor}}(\text{bob}) = 4$$
$$d_{\text{actor}}(\text{carol}) = 2$$
$$d_{\text{actor}}(v) = 0 \qquad v \notin \{\text{alice}, \text{bob}, \text{carol}\}$$

The profile $s(\boldsymbol{D})$ contains degree constraints:

- $||\boldsymbol{d}_{\text{actor}}||_\infty = 4$ (degree constraint!)
- $||\boldsymbol{d}_\emptyset||_\infty = 7 = |R|$ (cardinality constraint!)
- $||\boldsymbol{d}_{\text{actor},\text{movie}}||_\infty = 1$ (functional dependency)

General DC : (X, Y, N) in relation R means $|\pi_Y \sigma_{X=\boldsymbol{x}} R| \leq N, \; \forall \boldsymbol{x}$

Hierarchy of Output Cardinality Bounds Under Degree Constraints

$$\log \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} |Q(\boldsymbol{D}')| = \text{combinatorial-bound}(Q, s)$$ computable (?) but impractical

$$\leq \text{entropic-bound}(Q, s)$$ not (known to be) computable

$$\leq \boxed{\text{polymatroid-bound}(Q, s)}$$ [ANS 17]

$$\leq \text{flow-bound}(Q, s, \sigma)$$ [IMNP 25]

$$\leq \text{chain-bound}(Q, s, \sigma)$$ [N 18]

$$\leq \text{agm-bound}(Q, s)$$ [AGM 08]

$$\leq \text{integral-edge-cover}(Q, s)$$

- We settle for algorithms running in $\tilde{O}\left(|\boldsymbol{D}| + \text{bound}(\text{Q}, \text{s})\right)$
- Algorithm is a WCOJ algorithm when the bound is tight

Hierarchy of Join Algorithms

(not achievable ?)	$\sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} \lvert Q(\boldsymbol{D}') \rvert$	
(not achievable ?)	$\leq$ entropic-bound(Q, s)	
(IAAT)	$\leq$ polymatroid-bound(Q, s)	PANDA – [ANS 17]
(VAAT)	$\leq$ chain-bound(Q, s, σ)	NPRR, LFTJ, GJ
(VAAT)	$\leq$ agm-bound(Q, s)	NPRR, LFTJ, GJ
(JAAT)	$\leq$ integral-edge-cover(Q, s)	Binary Join Plans

Some bounds are in log-scale, where runtime = $\tilde{O}(|\boldsymbol{D}| + 2^{\text{bound}})$

Hierarchy of Join Algorithms

JAAT	Join at a time
VAAT	Variable at a time
IAAT	Inequality at a time

The Algorithm is in the Pudding

Proof $\Longrightarrow$ Algorithm!

Example: Output Cardinality Bound for the Triangle Query

$$Q_\triangle(A, B, C) \leftarrow R(A, B) \wedge S(B, C) \wedge T(A, C)$$

AGM-bound for Q: (E.g. num triangles in a graph $\leq |E|^{3/2}$)

$$|Q_\triangle| \leq |R|^{\lambda_R} \cdot |S|^{\lambda_S} \cdot |T|^{\lambda_T}$$

whenever $\boldsymbol{\lambda} = (\lambda_R, \lambda_S, \lambda_T)$ is a fractional edge cover for the triangle:

$$\lambda_R + \lambda_S \geq 1$$
$$\lambda_R + \lambda_T \geq 1 \qquad \boldsymbol{\lambda} \geq \mathbf{0}$$
$$\lambda_S + \lambda_T \geq 1$$

Pick $\boldsymbol{\lambda}$ to minimize the bound.

Example: Proving the AGM Bound for the Triangle Query

Consider a "section" of this query on a given value $a \in \mathrm{Dom}(A)$:

$$Q_\triangle(a, B, C) \leftarrow R(a, B) \wedge S(B, C) \wedge T(a, C)$$

Need (b, c) in the intersection $S \cap (\sigma_{A=a} R \times \sigma_{A=a} T)$, thus

$$\begin{aligned}
|\sigma_{A=a} Q_\triangle| &\leq \min\left\{|S|, |\sigma_{A=a} R| \cdot |\sigma_{A=a} T|\right\} \\
&\leq |S|^{\lambda_S} \cdot (|\sigma_{A=a} R| \cdot |\sigma_{A=a} T|)^{1-\lambda_S} \\
&\leq |S|^{\lambda_S} \cdot |\sigma_{A=a} R|^{\lambda_R} \cdot |\sigma_{A=a} T|^{\lambda_T}
\end{aligned}$$

We used $\lambda_S + \lambda_R \geq 1$ and $\lambda_S + \lambda_T \geq 1$

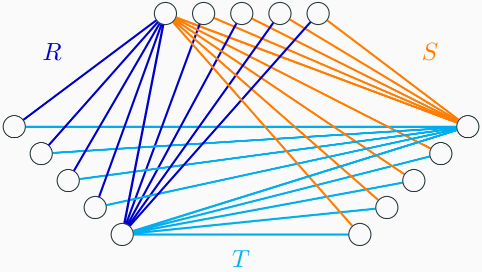
Triangle Query: AGM Bound Based on Hölder Inequality

Iterate over all possible values of a:

$$\begin{aligned}
|Q_\triangle| &= \sum_a |\sigma_{A=a} Q_\triangle| \\
&\le \sum_a |S|^{\lambda_S} \cdot |\sigma_{A=a} R|^{\lambda_R} \cdot |\sigma_{A=a} T|^{\lambda_T} \\
&= |S|^{\lambda_S} \cdot \sum_a |\sigma_{A=a} R|^{\lambda_R} \cdot |\sigma_{A=a} T|^{\lambda_T} \\
\text{(Hölder)} &\le |S|^{\lambda_S} \cdot \Big(\sum_a |\sigma_{A=a} R|\Big)^{\lambda_R} \cdot \Big(\sum_a |\sigma_{A=a} T|\Big)^{\lambda_T} \qquad \text{recall } \lambda_R + \lambda_S \ge 1 \\
&= |S|^{\lambda_S} \cdot |R|^{\lambda_R} \cdot |T|^{\lambda_T}
\end{aligned}$$

Example: $|Q_\triangle| \le \sqrt{|R| \cdot |S| \cdot |T|}$

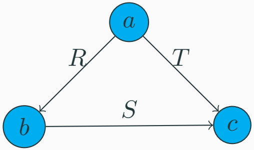
Example: JAAT Query Plans Are Sub-Optimal for Triangle Query

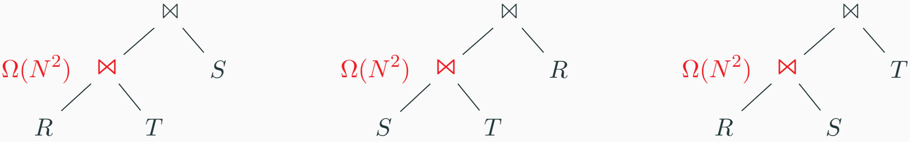


$$|R| = |S| = |T| = 2N - 1$$

$$Q_\Delta(a,b,c) = R(a,b) \wedge S(b,c) \wedge T(a,c)$$

$$\sup_{\mathbf{D}' \models \mathsf{DC}(\boldsymbol{D})} |Q_\Delta(\mathbf{D}')| = O(N^{1.5})$$





$$Q_\triangle(A,B,C) \leftarrow R(A,B), S(B,C), T(A,C)$$

Algorithm 1: based on Hölder's inequality proof

for $a \in \pi_A R \cap \pi_A T$ **do**
  for $b \in \pi_B \sigma_{A=a} R \cap \pi_B S$ **do**
    for $c \in \pi_C \sigma_{B=b} S \cap \pi_C \sigma_{A=a} T$ **do**
      Report (a,b,c);

In words, we followed exactly what the proof did:

- For each $a \in \pi_A R \cap \pi_A T$, enumerate $(a,b,c) \in \sigma_{A=a} Q_\triangle$

Triangle Query: VAAT is in the Pudding

Computing this "section" of the query on a given value $a \in \text{Dom}(A)$

$$Q_\triangle(a, B, C) \leftarrow R(a, B), S(B, C), T(a, C)$$

is to compute the intersection $S \cap (\sigma_{A=a}R \times \sigma_{A=a}T)$, which can be done in time

$$\tilde{O}\left(\min\{|S|, |\sigma_{A=a}R| \cdot |\sigma_{A=a}T|\}\right) \leq \tilde{O}\left(|\sigma_{A=a}R|^{\lambda_R} \cdot |\sigma_{A=a}T|^{\lambda_T} \cdot |S|^{\lambda_S}\right)$$

Overall, the algorithm runs in time (Modulo $\tilde{O}(N)$ pre-processing)

$$\tilde{O}\left(\sum_a |\sigma_{A=a}R|^{\lambda_R} \cdot |\sigma_{A=a}T|^{\lambda_T} \cdot |S|^{\lambda_S}\right) = \tilde{O}\left(|S|^{\lambda_S} \cdot |R|^{\lambda_R} \cdot |T|^{\lambda_T}\right)$$

Full Conjunctive Query

$$Q(V) \leftarrow \bigwedge_{S \in \mathcal{E}} R_S(S)$$

Query hypergraph $\mathcal{H} = (V, \mathcal{E})$

AGM-bound for Q:

Assuming only cardinality constraints $(\emptyset, S, |R_S|)$

$$|Q| \leq \prod_{S \in \mathcal{E}} |R_S|^{\lambda_S}$$

whenever $\boldsymbol{\lambda} = (\lambda_S)_{S \in \mathcal{E}}$ is a fractional edge cover for $\mathcal{H}$:

$$\forall v \in V \ : \sum_{S \in \mathcal{E}, v \in S} \lambda_S \geq 1 \qquad \boldsymbol{\lambda} \geq \mathbf{0}.$$

Pick $\boldsymbol{\lambda}$ to minimize the bound.

Full Conjunctive Query: AGM Bound from Hölder Inequality

Consider a "section" of this query on a given value $a \in \mathrm{Dom}(A)$:

$$\sigma_{A=a} Q(V) \leftarrow \bigwedge_{S \in \mathcal{E}, A \notin S} R_S(S) \wedge \bigwedge_{S \in \mathcal{E}, A \in S} \sigma_{A=a} R_S(S)$$

Now iterate over all possible values of a:

$$
\begin{aligned}
|Q| = \sum_a |\sigma_{A=a} Q_{\triangle}| &\leq \sum_a \prod_{S \in \mathcal{E}, A \notin S} |R_S|^{\lambda_S} \cdot \prod_{S \in \mathcal{E}, A \in S} |\sigma_{A=a} R_S|^{\lambda_S} \\
\text{(Hölder's inequality)} &\leq \prod_{S \in \mathcal{E}, A \notin S} |R_S|^{\lambda_S} \cdot \prod_{S \in \mathcal{E}, A \in S} \left(\sum_a |\sigma_{A=a} R_S| \right)^{\lambda_S} \\
&= \prod_{S \in \mathcal{E}} |R_S|^{\lambda_S}.
\end{aligned}
$$

Full Conjunctive Query: VAAT Algorithm

[NPRR 12] [Veldhuizen 14]

$$Q(V) \leftarrow \bigwedge_{S \in \mathcal{E}} R_S(S)$$

Algorithm 2: based on Hölder's inequality proof

for $a \in \bigcap_{S \in \mathcal{E}, A \in S} \pi_A R_S$ **do**

Recursively solve the query section $\sigma_{A=a} Q$ (on variables $V - \{A\}$);

Report $\{a\} \times \sigma_{A=a} Q$

Runtime $\tilde{O}\left(\sum_{S \in \mathcal{E}} |R_S|\right) + \tilde{O}\left(\prod_{S \in \mathcal{E}} |R_S|^{\lambda_S}\right)$.

Proof: straightforward.

Reminder: Hierarchy of Join Algorithms

(not achievable ?) $\sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} |Q(\boldsymbol{D}')|$

(not achievable ?) $\leq$ entropic-bound(Q, s)

(IAAT) $\leq$ polymatroid-bound(Q, s) PANDA – [ANS 17]

(VAAT) $\leq$ chain-bound(Q, s, σ) NPRR, LFTJ, GJ

(VAAT) $\leq$ agm-bound(Q, s) NPRR, LFTJ, GJ

(JAAT) $\leq$ integral-edge-cover(Q, s) Binary Join Plans

Some bounds are in log-scale, where runtime = $\tilde{O}(|\boldsymbol{D}| + 2^{\text{bound}})$

Polymatroids

Polymatroid is a function $h : 2^V \to \mathbb{R}_+$ such that $h(\emptyset) = 0$ and

- $h(X) \leq h(Y)$ if $X \subseteq Y$ — monotonicity
- $h(X) + h(Y) \geq h(X \cup Y) + h(X \cap Y)$ — submodularity

Let Γ_n be the set of all polymatroids on n variables.

The Polymatroid Bound

Theorem (ANS 17)

If $s(\boldsymbol{D})$ contains only degree constraints, then

$$\log \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} |Q(\boldsymbol{D}')| \leq \max_{h \in \Gamma_n \cap DC} h(V)$$

where DC is the set of linear constraints of the form

$$h(X \cup Y) - h(X) \leq \log N, \qquad \textit{for each degree constraint } (X, Y, N),$$

The PANDA algorithm [ANS 17] achieves this bound

Outline

Worst-Case Optimality is Insufficient for Queries with Aggregations

4-cycle detection

- $Q() \leftarrow \exists A, B, C, D\ E(A, B) \wedge E(B, C) \wedge E(C, D) \wedge E(D, A)$
- Worst-case size bound on $|Q|$ is 1, on the conjunction is $|E|^2$
- Answerable in $O(|E|^{\frac{3}{2}})$-time [AYZ 97]

k-cycle detection

- Worst-case size bound on the conjunction is $O(|E|^{\lceil k/2 \rceil})$
- k-cycle detection can be done in $O(|E|^{2-\frac{1}{\lceil k/2 \rceil}})$-time [AYZ 97]

Worst-Case Optimality is Insufficient for Queries with Aggregations

k-walk counting (or any sub-trees in general)

- $Q = \text{count}[X_1, \ldots, X_{k+1} : E_1(X_1, X_2) \wedge \cdots \wedge E(X_k, X_{k+1})]$
- Worst-case size bound on $|Q|$ is 1, on the conjunction is $|E|^{\lceil k/2 \rceil}$
- But, we can answer this in $\tilde{O}(k|E|)$-time.

Conjunctive query with negation

- $Q() \leftarrow R(X, Y) \wedge S(Y, Z) \wedge T(Z, U) \wedge X \neq U$
- Worst-case size bound on $|Q|$ is 1, on the conjunction is $O(N^2)$
- But, we can answer this in $\tilde{O}(N)$-time.

Need Two New Ideas in Our Framework

Idea 1: an abstraction to capture aggregation queries.

- $\max, \min, \sum, \exists$, etc.

Idea 2: a better notion of "optimality"

- Fine-grained complexity, parameterized complexity.

Idea 1: Sum-Product-Queries

Given a semiring $(D, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, a Sum-Product-query is

$$\varphi(\boldsymbol{X}_F) := \bigoplus_{\boldsymbol{X}_{V-F}} \bigotimes_{S \in \mathcal{E}} \psi_S(\boldsymbol{X}_S)$$

Example 1: $(\{\text{true}, \text{false}\}, \vee, \wedge, \text{false}, \text{true})$ is the Boolean semiring.

$$Q() = \bigvee_{A,B,C} E(A,B) \wedge E(A,C) \wedge E(B,C)$$

Example 2: $(\mathbb{R}, +, \times, 0, 1)$ is the sum-product semiring.

$$Q(A) = \sum_{B,C,D} E(A,B) \times E(B,C) \times E(C,D) \times \mathbf{1}_{A \neq D}$$

Idea 2: Better Notion of Optimality

$$\tilde{O}\left(|\boldsymbol{D}| + \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} |Q(\boldsymbol{D}')|\right) \Longrightarrow \tilde{O}\left(|\boldsymbol{D}| + \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} f(Q, \boldsymbol{D}') + |Q(\boldsymbol{D})|\right)$$

Accompanied by a meta algorithm

$Q \in \{$ database, CSP, PGM, logic, graph, linear algebra $\}$ queries

$\mathbf{D} \dashrightarrow Q_h \dashrightarrow Q_h(\mathbf{D}) \dashrightarrow Q_e \dashrightarrow Q(\mathbf{D})$

Query Q $\xrightarrow[\text{"worst-case optimal"}]{\text{Tensor Decomposition}}$ Intermediate Representation $\xrightarrow[\text{"instance optimal"}]{\text{Variable Elimination}}$ $Q(\mathbf{D})$

Variable Elimination

$Q = \textbf{count}[A, B, C, D : R(A, B) \wedge S(B, C) \wedge T(C, D)]$

Overloading notation

$$R(a, b) := \mathbf{1}_{(a,b)\in R} \qquad S(b, c) := \mathbf{1}_{(b,c)\in S} \qquad T(c, d) := \mathbf{1}_{(c,d)\in T}$$

Variable elimination [ZP 94]

Works for any semi-ring!

$$\begin{aligned}
Q() &= \sum_a \sum_b \sum_c \sum_d R(a, b) \cdot S(b, c) \cdot T(c, d) \\
&= \sum_a \sum_b \sum_c R(a, b) \cdot S(b, c) \cdot \sum_d T(c, d) = \sum_a \sum_b \sum_c R(a, b) \cdot S(b, c) \cdot W(c) \\
&= \sum_a \sum_b R(a, b) \cdot \sum_c S(b, c) \cdot W(c) = \sum_a \sum_b R(a, b) \cdot V(b)
\end{aligned}$$

Database jargon: (aggregation | projection | predicate) pushdowns

Indicator Projection, and Fractional Hypertree Width

[GM 06] [ANR 16]

$$Q = \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f)$$

$$= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e) \underbrace{\sum_{f} \boxed{\pi_{de}T(d,e)} W(e,f)V(d,f)}_{\tilde{O}(N^{3/2}),\ \text{WCOJ}}$$

$$= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e)M_1(d,e)$$

$$= \sum_{a,b,c} R(a,b)S(a,c) \sum_{d,e} \boxed{\pi_b R(b) \pi_S(c)} T(b,c,d,e)M_1(d,e)$$

$$= \sum_{a,b,c} R(a,b)S(a,c)M_2(b,c) \qquad = \underbrace{\sum_{a,b,c} R(a,b)S(a,c)M_2(b,c)}_{\tilde{O}(N^{3/2}),\ \text{WCOJ}}$$

Variable Elimination, Message Passing, Belief Propagation, Yannakakis

$$Q = \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f)$$

$$\text{Output} \xleftarrow{Q()} (a, b, c) \xleftarrow{M_2(b,c)} (b, c, d, e) \xleftarrow{M_1(d,e)} (d, e, f)$$

- Yannakakis algorithm [Yannakakis 81]
- Belief propagation [Pearl 82]

A Different Variable Elimination Order for the Same Query

$$Q = \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f)$$

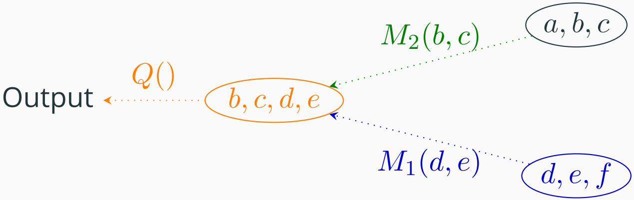
$$= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e) \underbrace{\sum_{f} \boxed{\pi_{de}T(d,e)} W(e,f)V(d,f)}_{M_1(d,e) \text{ in } \tilde{O}(N^{3/2}),\ \text{WCOJ}}$$

$$= \sum_{b,c,d,e} T(b,c,d,e)M_1(d,e) \underbrace{\sum_{a} \pi_{bc}T(b,c)R(a,b)S(a,c)}_{M_2(b,c) \text{ in } O(N^{3/2}),\ \text{WCOJ}}$$

$$= \sum_{b,c,d,e} M_1(d,e)T(b,c,d,e)M_2(b,c)$$

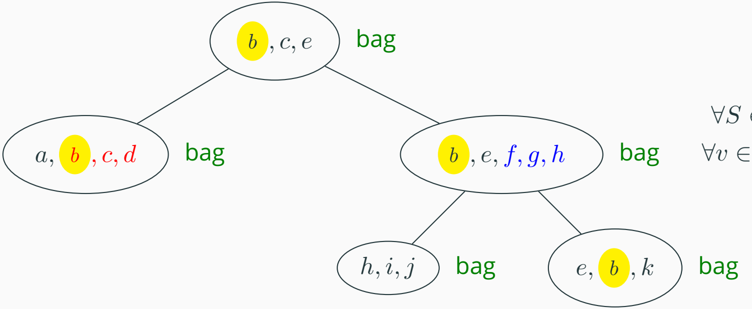
Different Variable Elimination Order $\Rightarrow$ Different Tree Decomposition

$$Q = \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f)$$



Detour: Tree Decompositions

$$S() \leftarrow R(a,b,d) \wedge c < d \wedge T(c,b,d) \wedge U(b,e) \wedge V(c,e)$$
$$\wedge\, b + e = f \wedge W(b,e,g) \wedge g/f = h \wedge X(i,j,h) \wedge e - b = k.$$



Hypergraph $\mathcal{H} = (V, \mathcal{E})$
Tree $\mathcal{T} = (V(\mathcal{T}), E(\mathcal{T}))$
$\forall S \in \mathcal{E}, \exists t \in V(\mathcal{T})$ s.t. $S \subseteq \chi(t)$
$\forall v \in V, \{t \mid v \in \chi(t)\}$ is a subtree

Dynamic Programming over a Single Tree Decomposition

Fix $(\mathcal{T}, \chi)$

Multiple Conjunctive Rules

$$\mathbf{D} \dashrightarrow Q_h \dashrightarrow Q_h(\mathbf{D}) \dashrightarrow Q_e \dashrightarrow Q(\mathbf{D})$$

$$\bigwedge_{F \in \mathcal{E}} R_F \xrightarrow[\text{WCOJ algo}]{\tilde{O}\left(\min_{(\mathcal{T},\chi)} \max_{\mathbf{D}' \models \mathsf{DC}} \max_{t \in V(\mathcal{T})} |P_t(\mathbf{D}')|\right)} \bigwedge_{t \in V(\mathcal{T})} T_{\chi(t)} \xrightarrow[\text{Variable Elimination}]{\tilde{O}(|Q_h(\mathbf{D})| + |Q(\mathbf{D})|)} Q(\mathbf{D})$$

$$P_t \ : \ T_{\chi(t)} \leftarrow \bigwedge_{F \in \mathcal{E}} R_F$$

Recall the polymatroid bound:

$$\log \min_{(\mathcal{T},\chi)} \max_{\mathbf{D}' \models \mathsf{DC}} \max_{t \in V(\mathcal{T})} |P_t(\mathbf{D}')| \leq \min_{(\mathcal{T},\chi)} \max_{t \in V(\mathcal{T})} \max_{h \in \Gamma_n \cap \mathsf{DC}} h(\chi(t))$$

$$\tilde{O}\left(|\boldsymbol{D}| + \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} f(Q, \boldsymbol{D}') + |Q(\boldsymbol{D})|\right)$$

Becomes:

$$\tilde{O}\left(|\boldsymbol{D}| + 2^{\min_{(\mathcal{T},\chi)} \max_{t \in V(\mathcal{T})} \max_{h \in \Gamma_n \cap \mathsf{DC}} h(\chi(t))} + |Q(\boldsymbol{D})|\right)$$

Corrolaries when the input has only cardinality constraints:

- $\tilde{O}\left(|\boldsymbol{D}| + |Q(\boldsymbol{D})|\right)$ if Q is acyclic [Yannakakis 81]
- $\tilde{O}\left(|\boldsymbol{D}| + |\boldsymbol{D}|^{\mathsf{fhtw}(Q)} + |Q(\boldsymbol{D})|\right)$ fhtw $=$ *fractional hypertree width*, [GM 06]

Outline

Multiple Tree Decompositions!

Idea from [Marx 13]

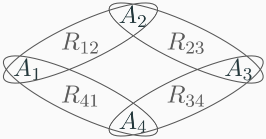
$$\mathbf{D} \dashrightarrow Q_h \dashrightarrow Q_h(\mathbf{D}) \dashrightarrow Q_e \dashrightarrow Q(\mathbf{D})$$

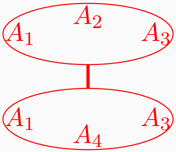
$$\bigwedge_{F\in\mathcal{E}} R_F \xrightarrow[\boxed{???}]{\max_{\mathbf{D}'\models s(\boldsymbol{D})} |Q_h(\mathbf{D}')| \approx \boxed{?}} \bigvee_{(\mathcal{T},\chi)} \bigwedge_{t\in V(\mathcal{T})} T_{\chi(t)} \xrightarrow[\text{Variable Elimination}]{\tilde{O}(|Q_h(\mathbf{D})| + |Q(\mathbf{D})|)} Q(\mathbf{D})$$

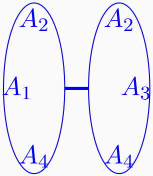
Ranges over tree decompositions

Assuming Boolean conjunctive query for now

Example: $Q \ : \ S() \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$







$$(T_{123} \wedge T_{134}) \vee (T_{124} \wedge T_{234}) \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$$

By distributivity, rewrite the head:

$$(T_{123} \vee T_{124}) \wedge (T_{123} \vee T_{234}) \wedge (T_{134} \vee T_{124}) \wedge (T_{134} \vee T_{234}) \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$$

(Each “clause” has one bag per TD)

Example: $Q \ : \ S() \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$

$$(T_{123} \vee T_{124}) \wedge (T_{123} \vee T_{234}) \wedge (T_{134} \vee T_{124}) \wedge (T_{134} \vee T_{234}) \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$$

Is the same as evaluating 4 disjunctive datalog rules:

$$T_{123} \vee T_{124} \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$$
$$T_{123} \vee T_{234} \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$$
$$T_{134} \vee T_{124} \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$$
$$T_{134} \vee T_{234} \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$$

Semantics of a Disjunctive Datalog Query

$$P: \quad T_{123} \vee T_{124} \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}$$

The model contains two tables $T_{123}(A_1, A_2, A_3)$ and $T_{124}(A_1, A_2, A_4)$, such that:

- if $R_{12}(a_1, a_2)$, $R_{23}(a_2, a_3)$, $R_{34}(a_3, a_4)$, and $R_{41}(a_4, a_1)$
- then $T_{123}(a_1, a_2, a_3)$ OR $T_{124}(a_1, a_2, a_4)$.

Output size

- Write $\boldsymbol{T} \models P$ if $\boldsymbol{T}$ is a model $\qquad \boldsymbol{T} = (T_{123}, T_{124})$.
- Define $|\boldsymbol{T}| = \max\{|T_{123}|, |T_{124}|\}$

Multiple Tree Decompositions

Multiple Disjunctive Datalog Rules!

$$\mathbf{D} \dashrightarrow Q_h \dashrightarrow Q_h(\mathbf{D}) \dashrightarrow Q_e \dashrightarrow Q(\mathbf{D})$$

$$\bigwedge_{F\in\mathcal{E}} R_F \xrightarrow[\text{how?}]{\max\limits_{\mathbf{D}'\models s(\boldsymbol{D})}\ \max\limits_{\mathcal{B}} |P_{\mathcal{B}}(\mathbf{D}')|} \bigwedge_{\mathcal{B}\in\boldsymbol{B}}\ \bigvee_{B\in\mathcal{B}} T_B \xrightarrow[\text{Variable elimination}]{\tilde{O}(|Q_h(\mathbf{D})| + |Q(\mathbf{D})|)} Q(\mathbf{D})$$

$$P_{\mathcal{B}}\ :\ \bigvee_{B\in\mathcal{B}} T_B \leftarrow \bigwedge_{F\in\mathcal{E}} R_F$$

$$|P_{\mathcal{B}}| \stackrel{\text{def}}{=} \min_{\boldsymbol{T}:\boldsymbol{T}\models P_{\mathcal{B}}}\ \max_{B\in\mathcal{B}} |T_B|$$

Each $\mathcal{B} \in \boldsymbol{B}$ is a collection of bags, one per TD.

New Problem: Polymatroid Bound for Disjunctive Datalog

$$P_{\mathcal{B}} \;:\; \bigvee_{B \in \mathcal{B}} T_B \leftarrow \bigwedge_{F \in \mathcal{E}} R_F$$

what is $\max_{\mathbf{D}' \models s(\boldsymbol{D})} |P_{\mathcal{B}}(\mathbf{D}')|$?

Theorem (ANS 17)

Define DC $\stackrel{\text{def}}{=} \{h \mid \quad h(Y|X) \leq \log N \qquad \forall (X, Y, N) \in s(\boldsymbol{D})\}$, *then*

$$\log \sup_{\mathbf{D}' \models s(\boldsymbol{D})} |P_{\mathcal{B}}(\boldsymbol{D}')| \leq \max_{h \in \overline{\Gamma}^*_n \cap DC} \min_{B \in \mathcal{B}} h(B)$$

Entropic Bound

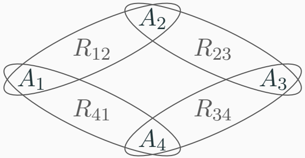
$$\leq \max_{h \in \Gamma_n \cap DC} \min_{B \in \mathcal{B}} h(B)$$

Polymatroid Bound

Generalize the entropy argument for conjunctive datalog (fun exercise!)

4-Cycle Example

$$P: \quad \bigvee_{B\in\mathcal{B}} T_B \leftarrow \bigwedge_{F\in\mathcal{E}} R_F \qquad \boxed{|P(\boldsymbol{D})| \stackrel{\text{def}}{=} \min_{\boldsymbol{T}:\boldsymbol{T}\models P}\ \max_{B\in\mathcal{B}} |T_B|}$$



$$\text{DC}: \quad |R_{12}| \le N, \quad |R_{23}| \le N, \quad |R_{34}| \le N, \quad |R_{41}| \le N.$$

$$P_{123,234}: \quad {\color{red}T_{123}} \vee {\color{blue}T_{234}} \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41} \qquad \mathcal{B} = \{{\color{red}123}, {\color{blue}234}\}$$

$$\begin{aligned}
\sup_{\mathbf{D}'\models\text{DC}} \log|P_{123,234}(\boldsymbol{D}')| &\le \max_{h\in\Gamma_n\cap\text{DC}} \min\{h(A_1A_2A_3), h(A_2A_3A_4)\} \\
&\le \max_{h\in\Gamma_n\cap\text{DC}} \frac{1}{2}[h(A_1A_2A_3) + h(A_2A_3A_4)] \\
\text{(Shannon-inequality)} &\boxed{\le} \max_{h\in\Gamma_n\cap\text{DC}} \frac{1}{2}[h(A_1A_2) + h(A_2A_3) + h(A_3A_4)] \\
&\le \frac{3}{2}\log N.
\end{aligned}$$

Realization of Idea 2

$$\tilde{O}\left(|\boldsymbol{D}| + \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} f(Q, \boldsymbol{D}') + |Q(\boldsymbol{D})|\right)$$

Becomes:

$$\tilde{O}\left(|\boldsymbol{D}| + 2^{\max_{\mathcal{B} \in \boldsymbol{B}} \max_{h \in \Gamma_n \cap \mathsf{DC}} \min_{B \in \mathcal{B}} h(B)} + |Q(\boldsymbol{D})|\right)$$

Another problem: WCOJ for disjunctive datalog!

Another Problem : WCOJ for Disjunctive Datalog

$$P_{\mathcal{B}} \;:\; \bigvee_{B \in \mathcal{B}} T_B \leftarrow \bigwedge_{F \in \mathcal{E}} R_F$$

compute a model in worst-case optimal time

Theorem (ANS 17)

Given a collection $s(\boldsymbol{D})$ of degree constraints, and a disjunctive datalog program $P_{\mathcal{B}}$, PANDA computes a model of $P_{\mathcal{B}}$ in time

$$\left(|\boldsymbol{D}| + 2^{\max\limits_{h \in \Gamma_n \cap DC} \min\limits_{B \in \mathcal{B}} h(B)} \right)$$

$$\tilde{O}\left(|\boldsymbol{D}| + \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} f(Q, \boldsymbol{D}') + |Q(\boldsymbol{D})|\right)$$

Becomes:

$$\tilde{O}\left(|\boldsymbol{D}| + 2^{\max_{\mathcal{B} \in \boldsymbol{B}} \max_{h \in \Gamma_n \cap \mathsf{DC}} \min_{B \in \mathcal{B}} h(B)} + |Q(\boldsymbol{D})|\right)$$

Corrolaries when the input has only cardinality constraints:

- $\tilde{O}\left(|\boldsymbol{D}| + |\boldsymbol{D}|^{\mathsf{subw}(Q)} + |Q(\boldsymbol{D})|\right)$ subw $=$ *submodular-width*, [Marx 13]
- Detecting k-cycle can be done in $O(N^{2-\frac{1}{\lceil k/2 \rceil}})$-time [AYZ 97]

Outline

References (and references thereof)

Yannakakis 81 Mihalis Yannakakis: Algorithms for Acyclic Database Schemes. VLDB 1981: 82-94

Pearl 82 Pearl, Judea. “Reverend Bayes on inference engines: A distributed hierarchical approach”). AAAI-82

CGFS 1986 Fan R. K. Chung, Ronald L. Graham, Peter Frankl, James B. Shearer: Some intersection theorems for ordered sets and graphs. J. Comb. Theory, Ser. A 43(1): 23-37 (1986)

ZP 1994 Zhang, N.L., Poole, D.:A Simple Approach to Bayesian Network Computations.In: 7th Canadian Conference on Artificial Intelligence,pp. 171–178

•

AYZ 97 Noga Alon, Raphael Yuster, Uri Zwick: Finding and Counting Given Length Cycles. Algorithmica 17(3): 209-223 (1997)

GM 06 Martin Grohe, Dániel Marx: Constraint solving via fractional edge covers. SODA 2006: 289-298

AGM 08 Albert Atserias, Martin Grohe, Dániel Marx: Size Bounds and Query Plans for Relational Joins. FOCS 2008: 739-748

GLVV 12 Georg Gottlob, Stephanie Tien Lee, Gregory Valiant, Paul Valiant: Size and Treewidth Bounds for Conjunctive Queries. J. ACM 59(3): 16:1-16:35 (2012)

NPRR 12 Hung Q. Ngo, Ely Porat, Christopher Ré, Atri Rudra: Worst-case optimal join algorithms. PODS 2012.

GJ 13 Hung Q. Ngo, Christopher Ré, Atri Rudra: Skew Strikes Back: New Developments in the Theory of Join Algorithms. SIGMOD Records, 2013.

Marx 13 D. Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. J. ACM, 60(6):Art. 42, 51, 2013.

LFTJ 14 Todd L. Veldhuizen: Triejoin: A Simple, Worst-Case Optimal Join Algorithm. ICDT 2014: 96-106

References (and references thereof)

ANR 16 Mahmoud Abo Khamis, Hung Q. Ngo, Atri Rudra: FAQ: Questions Asked Frequently. PODS 2016: 13-28

ANS 17 Mahmoud Abo Khamis, Hung Q. Ngo, Dan Suciu: What Do Shannon-type Inequalities, Submodular Width, and Disjunctive Datalog Have to Do with One Another? PODS 2017: 429-444

ANOS 19 Mahmoud Abo Khamis, Hung Q. Ngo, Dan Olteanu, Dan Suciu: Boolean Tensor Decomposition for Conjunctive Queries with Negation. ICDT 2019: 21:1-21:19

N 18 Hung Q. Ngo: Worst-case Optimal Join Algorithms: Techniques, Results, and Open Problems. PODS 2018.

N 22 Hung Q. Ngo: On an Information Theoretic Approach to Cardinality Estimation (Invited Talk). ICDT 2022: 1:1-1:21

IMNP 25 Sungjin Im, Ben Moseley, Hung Q. Ngo, Kirk Pruhs, and Alireza Samadian: Efficient Algorithms for Cardinality Estimation and Conjunctive Query Evaluation With Simple Degree Constraints. PODS 2025.

Thank You!

Outline

Appendix: PANDA Algorithm

Outline

The Polymatroid Bound and Its Dual $\max\{h(V) \mid h \in \Gamma_n \cap \mathsf{DC}\}$

More explicitly,

$$
\begin{array}{llrr}
\max & h(V) & & \text{dual vars} \\
\text{s.t.} & h(Y) - h(X) \le \log N, & (X, Y, N) \in \mathsf{DC} & \delta_{Y|X} \\
& h(I \cup J|J) - h(I|I \cap J) \le 0, & I \perp J & \sigma_{I,J} \\
& h(X) - h(Y) \le 0, & \emptyset \neq X \subset Y \subseteq V & \mu_{Y|X} \\
& h(Z) \ge 0, & \emptyset \neq Z \subseteq V. &
\end{array}
$$

$I \perp J$ means $I \not\subseteq J$ and $J \not\subseteq I$.

The Polymatroid Bound and Its Dual

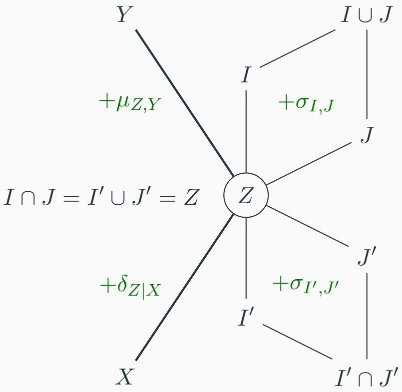
$\max\{h(V) \mid h \in \Gamma_n \cap \mathsf{DC}\}$

$$\begin{aligned} \min \quad & \sum_{(X,Y,N)\in\mathsf{DC}} \log N \cdot \delta_{Y|X} \\ \text{s.t.} \quad & \mathsf{excess}(V) \geq 1, \\ & \mathsf{excess}(Z) \geq 0, & \emptyset \neq Z \subseteq V. \\ & (\boldsymbol{\delta}, \boldsymbol{\sigma}, \boldsymbol{\mu}) \geq \mathbf{0}. \end{aligned}$$

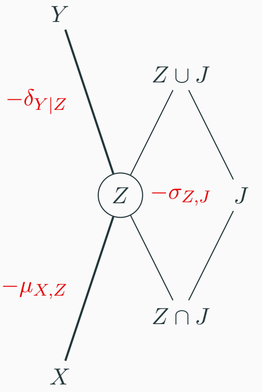
where, for any $\emptyset \neq Z \in 2^V$, the quantity $\mathsf{excess}(Z)$ is defined by

$$\begin{aligned} \mathsf{excess}(Z) := & \sum_{X:(X,Z)\in\mathsf{DC}} \delta_{Z|X} - \sum_{Y:(Z,Y)\in\mathsf{DC}} \delta_{Y|Z} + \sum_{\substack{I\perp J \\ I\cap J = Z}} \sigma_{I,J} \\ & + \sum_{\substack{I\perp J \\ I\cup J = Z}} \sigma_{I,J} - \sum_{J:J\perp Z} \sigma_{Z,J} - \sum_{X:X\subset Z} \mu_{X,Z} + \sum_{Y:Z\subset Y} \mu_{Z,Y}. \end{aligned}$$

Contributions of coefficients to $\mathsf{excess}(Z)$



$X \subset Z \subset Y$



Shannon-Flow Inequalities

[ANS 17]

Definition

Given $\boldsymbol{\delta} \geq 0$, the following is a Shannon-flow inequality if it holds for all $h \in \Gamma_n$:

$$h(V) \leq \sum_{(X,Y,N)\in\text{DC}} \delta_{Y|X} \cdot (h(Y) - h(X))$$

- $\boldsymbol{\delta}$ defines a Shannon-flow inequality iff $\exists(\boldsymbol{\sigma}, \boldsymbol{\mu})$ s.t. $(\boldsymbol{\delta}, \boldsymbol{\sigma}, \boldsymbol{\mu})$ is dual-feasible.
- If DC contains only cardinality constraints $(X = \emptyset, Y, N)$, then $\boldsymbol{\delta}$ defines a Shannon-flow inequality iff it is a fractional edge cover of the query hypergraph.

Shearer's Lemma!

Example: Shannon-Flow Inequality for Triangle Query

$$h(A,B,C) \leq \frac{1}{2}\left(h(A,B) + h(B,C) + h(A,C)\right)$$

A step-by-step proof: [Radhakrishnan 2003]

$$\begin{aligned}
& h(A,B) + h(A,C) + h(B,C) \\
\text{(decomposition)} &= h(A) + h(B|A) + h(B,C) + h(A,C) \\
\text{(sub-modularity)} &\geq (h(A|B,C) + h(B,C)) + (h(B|A) + h(A,C)) \\
\text{(composition)} &= h(A,B,C) + (h(B|A) + h(A,C)) \\
\text{(sub-modularity)} &\geq h(A,B,C) + (h(B|A,C) + h(A,C)) \\
\text{(composition)} &= h(A,B,C) + h(A,B,C)
\end{aligned}$$

Example: Another Shannon-Flow Inequality

$$h(ABCD) \leq \frac{1}{2}[h(AB) + h(BC) + h(CD) + h(D|AC) + h(A|BD)],$$

$$\begin{aligned}
& h(AB) + h(BC) + h(CD) + h(D|AC) + h(A|BD) \\
\text{(decomposition)} &= h(AB) + h(B) + h(C|B) + h(CD) + h(D|AC) + h(A|BD) \\
\text{(sub-modularity)} &\geq h(AB) + h(B) + h(C|B) + h(CD|B) + h(D|AC) + h(A|BD) \\
\text{(composition)} &= h(AB) + h(C|B) + h(BCD) + h(D|AC) + h(A|BD) \\
\text{(sub-modularity)} &\geq h(AB) + h(C|B) + h(BCD) + h(D|AC) + h(A|BCD) \\
\text{(composition)} &= h(AB) + h(C|B) + h(D|AC) + h(ABCD) \\
\text{(sub-modularity)} &\geq h(AB) + h(C|AB) + h(D|AC) + h(ABCD) \\
\text{(composition)} &= h(ABC) + h(D|AC) + h(ABCD) \\
\text{(sub-modularity)} &\geq h(ABC) + h(D|ABC) + h(ABCD) \\
\text{(composition)} &= h(ABCD) + h(ABCD).
\end{aligned}$$

Shannon-flow Inequalities – Summary

[ANS 17]

From LP-duality, there exists $\boldsymbol{\delta} \geq \mathbf{0}$ s.t.

$$\text{polymatroid-bound} := \max\{h(V) \mid h \in C \cap \Gamma_n\} = \sum_{(X,Y,N) \in \mathsf{DC}} \delta_{Y|X} \log N,$$

and for these $\boldsymbol{\delta}$, from Farkas's lemma we have

$$h(V) \leq \sum_{(X,Y,N) \in \mathsf{DC}} \delta_{Y|X} \cdot h(Y|X), \quad \forall h \in \Gamma_n$$

Proof Sequence for a Shannon-Flow Inequality

$$h(V) \le \sum_{(X,Y,N)\in \mathsf{DC}} \delta_{Y|X} \cdot h(Y|X)$$

A proof sequence is a conversion from RHS to LHS using a sequence of steps of the form

(In)equality	Steps ($X \subseteq Y$)
$h(X) + h(Y\|X) = h(Y)$	$h(X) + h(Y\|X) \to h(Y)$
$h(Y) = h(X) + h(Y\|X)$	$h(Y) \to h(X) + h(Y\|X)$
$h(Y) \ge h(X)$	$h(Y) \to h(X)$
$h(Y\|X) \ge h(Y \cup Z\|X \cup Z)$	$h(Y\|X) \to h(Y \cup Z\|X \cup Z)$

Existence of Proof Sequence

Lemma (ANS 2017)

There is a proof sequence for every Shannon-flow inequality. (The length is at most doubly exponential in $|V|$).

The Shannon-flow inequality is a linear combination of dual constraints; the proof sequence is more stringent than that.

Outline

Appendix: PANDA Algorithm

- Shannon-Flow Inequalities
- IAAT Algorithm
- PANDA for Disjunctive Datalog

One Inequality At A Time (IAAT)

There is an algorithm (called PANDA) that converts a proof sequence $\to$ an efficient algorithm to answer the original query

Steps ($X \subseteq Y$)	Relational Operator
$h(X) + h(Y \mid X) \to h(Y)$	(join)
$h(Y) \to h(X) + h(Y \mid X)$	(data partition)
$h(Y) \to h(X)$	(projection)
$h(Y \mid X) \to h(Y \cup Z \mid X \cup Z)$	(NOP)

Theorem

PANDA solves any conjunctive query Q in time $\tilde{O}(N + \text{poly}(\log N) \cdot 2^{\textit{polymatroid bound}})$

Example: PANDA for Triangle Query

$$Q(A,B,C) \leftarrow R(A,B), S(B,C), T(A,C)$$
$$R^{\text{heavy}}(A,B) = \{(a,b) \ : \ |\sigma_{A=a}R| > \sqrt{N}\}$$
$$R^{\text{light}}(A,B) = \{(a,b) \ : \ |\sigma_{A=a}R| \leq \sqrt{N}\}$$

Algorithm is in the pudding!

$$\begin{aligned}
h(A,B) + h(A,C) + h(B,C) & \quad R(A,B), S(B,C), T(A,C) \\
= h(A) + h(B|A) + h(B,C) + h(A,C) & \quad R^{\text{heavy}}(A,B), R^{\text{light}}(A,B), S(B,C), T(A,C) \\
\geq (h(A|B,C) + h(B,C)) + (h(B|A) + h(A,C)) & \quad R^{\text{heavy}}(A,B), R^{\text{light}}(A,B), S(B,C), T(A,C) \\
= h(A,B,C) + (h(B|A) + h(A,C)) & \quad I^{\text{heavy}}(A,B,C), R^{\text{light}}(A,B), T(A,C) \\
\geq h(A,B,C) + (h(B|A,C) + h(A,C)) & \quad I^{\text{heavy}}(A,B,C), R^{\text{light}}(A,B), T(A,C) \\
= h(A,B,C) + h(A,B,C) & \quad I^{\text{heavy}}(A,B,C), I^{\text{light}}(A,B,C).
\end{aligned}$$

Example: PANDA for Triangle Query

The real query plan:

$$
\begin{aligned}
&R(A,B) \wedge S(B,C) \wedge T(A,C) \\
&= (R^{\text{heavy}}(A,B) \vee R^{\text{light}}(A,B)) \wedge S(B,C) \wedge T(A,C) \\
&= (R^{\text{heavy}}(A,B) \wedge S(B,C) \wedge T(A,C)) \vee (R^{\text{light}}(A,B) \wedge S(B,C) \wedge T(A,C)) \\
&= (R^{\text{heavy}}(A,B) \wedge S(B,C) \wedge T(A,C)) \vee (R^{\text{light}}(A,B) \wedge S(B,C) \wedge T(A,C)) \\
&= I^{\text{heavy}}(A,B,C) \wedge T(A,C) \vee I^{\text{light}}(A,B,C) \wedge S(B,C).
\end{aligned}
$$

- Note that $|I^{\text{heavy}}(A,B,C)| \leq N^{3/2}$ and $|I^{\text{light}}(A,B,C)| \leq N^{3/2}$.
- Overall runtime is $\tilde{O}(N^{3/2})$.

Example: PANDA for a More Interesting Example [ANS 17]

$$Q(A,B,C,D) \leftarrow R(A,B) \wedge S(B,C) \wedge T(C,D) \wedge \mathsf{hash}(A,C) = D \wedge \mathsf{hash}(B,D) = A$$

From the Shannon-flow inequality:

$$h(ABCD) \leq \frac{1}{2}[h(AB) + h(BC) + h(CD) + h(D|AC) + h(A|BD)],$$

we know

$$\log_2 |Q| \leq \frac{1}{2}[\log_2 |R| + \log_2 |S| + \log_2 |T| + 0 + 0]$$

or

$$|Q| \leq \sqrt{|R||S||T|}$$

Example: Another Shannon-Flow Inequality

$$h(ABCD) \leq \frac{1}{2}[h(AB) + h(BC) + h(CD) + h(D|AC) + h(A|BD)],$$

$$\begin{aligned}
& h(AB) + h(BC) + h(CD) + h(D|AC) + h(A|BD) \\
\text{(decomposition)} &= h(AB) + h(B) + h(C|B) + h(CD) + h(D|AC) + h(A|BD) \\
\text{(sub-modularity)} &\geq h(AB) + h(B) + h(C|B) + h(CD|B) + h(D|AC) + h(A|BD) \\
\text{(composition)} &= h(AB) + h(C|B) + h(BCD) + h(D|AC) + h(A|BD) \\
\text{(sub-modularity)} &\geq h(AB) + h(C|B) + h(BCD) + h(D|AC) + h(A|BCD) \\
\text{(composition)} &= h(AB) + h(C|B) + h(D|AC) + h(ABCD) \\
\text{(sub-modularity)} &\geq h(AB) + h(C|AB) + h(D|AC) + h(ABCD) \\
\text{(composition)} &= h(ABC) + h(D|AC) + h(ABCD) \\
\text{(sub-modularity)} &\geq h(ABC) + h(D|ABC) + h(ABCD) \\
\text{(composition)} &= h(ABCD) + h(ABCD).
\end{aligned}$$

Example : PANDA for a More Interesting Example

$$
\begin{aligned}
&R(A,B) \wedge S(B,C) \wedge T(C,D) \wedge \mathsf{hash}(A,C)=D \wedge \mathsf{hash}(B,D)=A \\
&= R(A,B) \wedge S^{\text{heavy}}(B,C) \wedge T(C,D) \wedge \mathsf{hash}(A,C)=D \wedge \mathsf{hash}(B,D)=A \\
&\vee R(A,B) \wedge S^{\text{light}}(B,C) \wedge T(C,D) \wedge \mathsf{hash}(A,C)=D \wedge \mathsf{hash}(B,D)=A \\
&= R(A,B) \wedge S^{\text{heavy}}(B,C) \wedge T(C,D) \wedge \mathsf{hash}(A,C)=D \wedge \mathsf{hash}(B,D)=A \\
&\vee R(A,B) \wedge S^{\text{light}}(B,C) \wedge T(C,D) \wedge \mathsf{hash}(A,C)=D \wedge \mathsf{hash}(B,D)=A \\
&= R(A,B) \wedge I^{\text{heavy}}(B,C,D) \wedge \mathsf{hash}(A,C)=D \wedge \mathsf{hash}(B,D)=A \\
&\vee I^{\text{light}}(A,B,C) \wedge T(C,D) \wedge \mathsf{hash}(A,C)=D \wedge \mathsf{hash}(B,D)=A \\
&= R(A,B) \wedge I^{\text{heavy}}(B,C,D) \wedge \mathsf{hash}(A,C)=D \wedge \mathsf{hash}(B,D)=A \\
&\vee I^{\text{light}}(A,B,C) \wedge T(C,D) \wedge \mathsf{hash}(A,C)=D \wedge \mathsf{hash}(B,D)=A \\
&= R(A,B) \wedge J^{\text{heavy}}(A,B,C,D) \wedge \mathsf{hash}(A,C)=D \\
&\vee J^{\text{light}}(A,B,C,D) \wedge T(C,D) \wedge \mathsf{hash}(B,D)=A.
\end{aligned}
$$

Example : PANDA for a More Interesting Example

Main question

How to define S^{heavy} and S^{light} so that runtime is $\tilde{O}(2^{h^*(A,B,C,D)})$

$$S^{\text{heavy}}(B,C) = \{(b,c) \ : \ |\sigma_{C=c}S| > 2^{h^*(B,C)-h^*(C)}\}$$
$$S^{\text{light}}(B,C) = \{(b,c) \ : \ |\sigma_{C=c}S| \leq 2^{h^*(B,C)-h^*(C)}\}$$

Assuming h^* and $(\boldsymbol{\delta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\mu}^*)$ are primal-dual optimal: $(\delta^*_{CD|\emptyset} > 0$ and $\delta^*_{AB|\emptyset} > 0)$

$$|S^{\text{light}}(B,C) \wedge T(C,D)| \leq 2^{h^*(B,C)-h^*(C)} \cdot 2^{h^*(C,D)} = 2^{h^*(B,C,D)} \leq 2^{h^*(A,B,C,D)}$$
$$|S^{\text{heavy}}(B,C) \wedge R(A,B)| \leq 2^{h^*(C)} \cdot 2^{h^*(A,B)} = 2^{h^*(A,B,C)} \leq 2^{h^*(A,B,C,D)}.$$

$=$ holds because SFI holds with $=$ for h^*.

The Actual PANDA Algorithm (Needs Another Talk)

More complicated because:

- Couldn't prove that every heavy / light copy reaches $h(V)$ eventually.
- Couldn't prove that in the proof sequence we won't ever compose terms which were decomposed in an earlier step

Main ideas to push through:

- A decomposition $h(A, B) \to h(A) + h(B|A)$ corresponds to partitioning R into logarithmically many "uniform" parts.
- Essentially, each each part, both the heavy condition and the light condition are satisfied.
- Induct on logarithmically many subproblems, including constructing a new proof sequence for each of them

The Actual PANDA Algorithm

PANDA runs in Time

$$\tilde{O}(N + \text{poly}(\log N) \cdot 2^{\text{polymatroid bound for } Q}) = \tilde{O}(N + \text{poly}(\log N) \cdot \sup_{\boldsymbol{D}' \models s(\boldsymbol{D})} |Q(\boldsymbol{D}')|)$$

Outline

Evaluating a Disjunctive Datalog Program within $\max_{h \in \Gamma_n \cap \text{HDC}} \min_{B \in \mathcal{B}} h(B)$

There exists non-negative $\boldsymbol{\lambda} = (\lambda_B)_{B \in \mathcal{B}}$, with $\|\boldsymbol{\lambda}\|_1 = 1$, s.t.

$$\max_{h \in \Gamma_n \cap \text{HDC}} \min_{B \in \mathcal{B}} h(B) = \max_{h \in \Gamma_n \cap \text{HDC}} \sum_{B \in \mathcal{B}} \lambda_B \, h(B)$$

Shannon-flow inequality: There exists $\boldsymbol{\delta} \geq \mathbf{0}$ s.t. (Farkas's lemma)

$$2^{\max_{h \in \Gamma_n \cap \text{HDC}} \min_{B \in \mathcal{B}} h(B)} = \prod_{(X,Y,N) \in s(\boldsymbol{D})} N^{\delta_{Y|X}}$$

$$\sum_{B \in \mathcal{B}} \lambda_B \cdot h(B) \leq \sum_{(X,Y,N) \in s(\boldsymbol{D})} \delta_{Y|X} \cdot h(Y|X), \; \forall h \in \Gamma_n$$

Proof sequence

Shannon-flow inequality: $h(Y|X) \stackrel{\text{def}}{=} h(Y) - h(X)$, $X \subseteq Y$

$$\sum_{B \in \mathcal{B}} \lambda_B \cdot h(B) \leq \sum_{(X,Y,N)} \delta_{Y|X} \cdot h(Y|X)$$

Proof sequence, convert RHS to LHS using following steps

(In)equality	Steps ($X \subseteq Y$)
$h(X) + h(Y\|X) = h(Y)$	$h(X) + h(Y\|X) \to h(Y)$
$h(Y) = h(X) + h(Y\|X)$	$h(Y) \to h(X) + h(Y\|X)$
$h(Y) \geq h(X)$	$h(Y) \to h(X)$
$h(Y\|X) \geq h(Y \cup Z\|X \cup Z)$	$h(Y\|X) \to h(Y \cup Z\|X \cup Z)$

Theorem

There is a proof sequence for every Shannon-flow inequality.

Example: $P: \quad T_{123} \vee T_{234} \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}.$

$$|R_{12}|, |R_{23}|, |R_{34}|, |R_{41}| \leq N \quad \Rightarrow \quad \boxed{|P| \leq N^{3/2}}$$

$$\begin{aligned}
\log|P| &\leq \min(h(A_1A_2A_3), h(A_2A_3A_4)) && \text{(polymatroid bound)} \\
&\leq \frac{1}{2}(h(A_1A_2A_3) + h(A_2A_3A_4)) && \text{(linearize)} \\
&\boxed{\leq} \frac{1}{2}(h(A_1A_2) + h(A_2A_3) + h(A_3A_4)) && \text{(Shannon-flow)} \\
&\leq \frac{3}{2}\log N && \text{(Cardinality constraints)}
\end{aligned}$$

Proof sequence	Proof Step
$h(A_1A_2) + h(A_2A_3) + h(A_3A_4)$	$(h(A_3A_4) \to h(A_4 \mid A_3) + h(A_3))$
$h(A_1A_2) + h(A_2A_3) + h(A_4 \mid A_3) + h(A_3)$	$(h(A_4 \mid A_3) \to h(A_4 \mid A_2A_3))$
$h(A_1A_2) + h(A_2A_3) + h(A_4 \mid A_2A_3) + h(A_3)$	$(h(A_2A_3) + h(A_4 \mid A_2A_3) \to h(A_2A_3A_4))$
$h(A_1A_2) + h(A_2A_3A_4) + h(A_3)$	$(h(A_1A_2) \to h(A_1A_2 \mid A_3))$
$h(A_1A_2 \mid A_3) + h(A_2A_3A_4) + h(A_3)$	$(h(A_1A_2 \mid A_3) + h(A_3) \to h(A_1A_2A_3))$
$h(A_1A_2A_3) + h(A_2A_3A_4)$	

Example: $P: \; T_{123} \vee T_{234} \leftarrow R_{12} \wedge R_{23} \wedge R_{34} \wedge R_{41}.$

$$|R_{12}|, |R_{23}|, |R_{34}|, |R_{41}| \leq N \quad \Rightarrow \quad \boxed{|P| \leq N^{3/2}}$$

$h(A_3 A_4) \to h(A_4|A_3) + h(A_3)$

$h(A_4|A_3) \to h(A_4|A_2 A_3)$

$h(A_2 A_3) + h(A_4|A_2 A_3) \to h(A_2 A_3 A_4)$

$h(A_1 A_2) \to h(A_1 A_2|A_3)$

$h(A_1 A_2|A_3) + h(A_3) \to h(A_1 A_2 A_3)$

$R_{34}(A_3, A_4) \to R_{34}^{(\ell)}(A_3, A_4), R_3^{(h)}(A_3)$

$R_{34}^{(\ell)}(A_3, A_4) \to R_{34}^{(\ell)}(A_3, A_4)$

$R_{23}(A_2, A_3) \bowtie R_{34}^{(\ell)}(A_3, A_4) \to T_{234}(A_2, A_3, A_4)$

$R_{12}(A_1, A_2) \to R_{12}(A_1, A_2)$

$R_{12}(A_1, A_2) \bowtie R_3^{(h)}(A_3) \to T_{123}(A_1, A_2, A_3)$