On Functional Aggregate Queries with Additive Inequalities

fdbresearch.github.io relational.ai

Maximilian Schleich
University of Oxford

Mahmoud Abo Khamis, relationalAI
Ryan R. Curtin, relationalAI
Benjamin Moseley, CMU
Hung Q. Ngo, relationalAI
XuanLong Nguyen, University of Michigan
Dan Olteanu, University of Oxford

ACM PODS June, 2019

Example I

Given: Relations R, S of size $O(N)$.

Task: Count number of tuples that satisfy

$$R(a, b) \wedge S(b, c) \wedge a \leq c$$

- Existing approaches take $O(N^2)$ time.
  1. Join R and S
  2. Count number of tuples that satisfy $a \leq c$
- Our approach takes $O(N \log N)$ time.

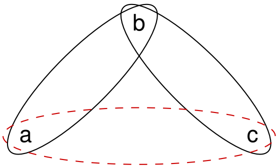
R

a	b
1	1
1	2
2	1
2	2
3	2
3	3

S

b	c
1	1
1	2
2	0
2	2
2	3
2	4

R and S are sorted



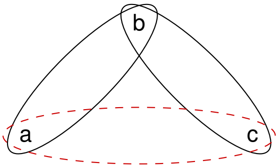
Hypergraph $\mathcal{H}$

Example I

Given: Relations R, S of size $O(N)$.

Task: Count number of tuples that satisfy

$$R(a,b) \wedge S(b,c) \wedge a \leq c$$



Hypergraph $\mathcal{H}$

- Existing approaches take $O(N^2)$ time.
  1. Join R and S
  2. Count number of tuples that satisfy $a \leq c$
- Our approach takes $O(N \log N)$ time.

R

a	**b**
1	1
1	2
2	1
2	2
3	2
3	3

S

b	**c**	**#**
1	1	2
1	2	1
2	0	4
2	2	3
2	3	2
2	4	1

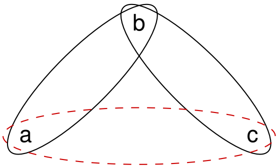
Step 1: Pre-aggregate S

Example I

Given: Relations R, S of size $O(N)$.

Task: Count number of tuples that satisfy

$$R(a,b) \wedge S(b,c) \wedge a \leq c$$



Hypergraph $\mathcal{H}$

- Existing approaches take $O(N^2)$ time.
  1. Join R and S
  2. Count number of tuples that satisfy $a \leq c$
- Our approach takes $O(N \log N)$ time.

R

a	b
1	1
1	2
2	1
2	2
3	2
3	3

lookup $b = 2$

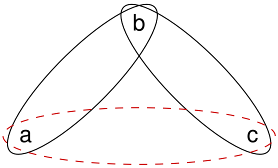
S

b	c	#
1	1	2
1	2	1
2	0	4
2	2	3
2	3	2
2	4	1

find smallest $c \geq 1$

Step 2: For each $R(a,b)$, locate $S(b,c)$ with $c \geq a$

Example I



Hypergraph $\mathcal{H}$

Given: Relations R, S of size $O(N)$.

Task: Count number of tuples that satisfy

$$R(a, b) \wedge S(b, c) \wedge a \leq c$$

- Existing approaches take $O(N^2)$ time.
  1. Join R and S
  2. Count number of tuples that satisfy $a \leq c$
- Our approach takes $O(N \log N)$ time.

R

a	b
1	1
1	2
2	1
2	2
3	2
3	3

lookup $b = 2$

count = 3

S

b	c	#
1	1	2
1	2	1
2	0	4
2	2	3
2	3	2
2	4	1

find smallest $c \geq 1$

Step 4: Return pre-aggregated count

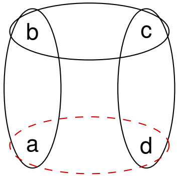
Example II

Given: Relations R, S, T of size $O(N)$.

Task: Count number of tuples that satisfy

$$R(a, b) \wedge S(b, c) \wedge T(c, d) \wedge a \leq d$$

- Existing approaches take $O(N^2)$ time.
- Our approach takes $O(N^{1.5} \log N)$ time.



Hypergraph $\mathcal{H}$

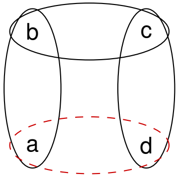
Example II

Given: Relations R, S, T of size $O(N)$.

Task: Count number of tuples that satisfy

$$R(a,b) \land S(b,c) \land T(c,d) \land a \le d$$

- Existing approaches take $O(N^2)$ time.
- Our approach takes $O(N^{1.5} \log N)$ time.



Hypergraph $\mathcal{H}$

Partition S

$$R(a,b) \land (S^{\text{light}}(b,c) \lor S^{\text{heavy}}(b,c)) \land T(c,d) \land a \le d$$

Turn into DNF

$$\underbrace{R(a,b) \land S^{\text{light}}(b,c)}_{U(a,b,c)} \land T(c,d) \land a \le d \quad \bigvee \quad R(a,b) \land \underbrace{S^{\text{heavy}}(b,c) \land T(c,d)}_{W(b,c,d)} \land a \le d$$

Compute U,W

$$U(a,b,c) \land T(c,d) \land a \le d \quad \bigvee \quad R(a,b) \land W(b,c,d) \land a \le d$$

Like Example I

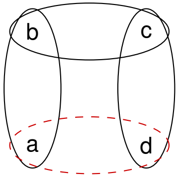
Example II

Given: Relations R, S, T of size $O(N)$.

Task: Count number of tuples that satisfy

$$R(a,b) \wedge S(b,c) \wedge T(c,d) \wedge a \leq d$$

- Existing approaches take $O(N^2)$ time.
- Our approach takes $O(N^{1.5} \log N)$ time.



Hypergraph $\mathcal{H}$

Partition S

$$R(a,b) \wedge (S^{\text{light}}(b,c) \vee S^{\text{heavy}}(b,c)) \wedge T(c,d) \wedge a \leq d$$

$\boxed{S^{\text{light}} = \{t \in S : |\sigma_{b=t.b} S| \leq \sqrt{N}\}}$

$\boxed{S^{\text{heavy}} = S \setminus S^{\text{light}}}$

Turn into DNF

$$\underbrace{R(a,b) \wedge S^{\text{light}}(b,c)}_{U(a,b,c)} \wedge T(c,d) \wedge a \leq d \quad \bigvee \quad R(a,b) \wedge \underbrace{S^{\text{heavy}}(b,c) \wedge T(c,d)}_{W(b,c,d)} \wedge a \leq d$$

$\boxed{O(N^{1.5})}$ $\boxed{O(N^{1.5})}$

Compute U,W

$$U(a,b,c) \wedge T(c,d) \wedge a \leq d \quad \bigvee \quad R(a,b) \wedge W(b,c,d) \wedge a \leq d$$

Like Example I

Example III: Linear SVM over Databases

Task: Compute $J(\beta)$ over dataset **D** defined by query Q over database $\mathcal{I}$.

$$J(\boldsymbol{\beta}) = \sum_{(\mathbf{x},y)\in\mathbf{D}} \underbrace{\max\{0, 1 - y \cdot f_{\boldsymbol{\beta}}(x)\}}_{\text{Hinge Loss}}$$

$$= \sum_{(\mathbf{x},y)\in\mathbf{D}} \begin{cases} 1 & \text{if } y \cdot f_{\boldsymbol{\beta}}(x) < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$= \underbrace{\sum_{(\mathbf{x},y)\in\mathbf{D}} (1 - y \cdot f_{\boldsymbol{\beta}}(x)) \cdot \mathbf{1}_{y \cdot f_{\boldsymbol{\beta}}(x) \leq 1}}_{\text{Aggregate with inequality}}$$

Existing approaches:

1. materialize **D**
2. learn model in favorite ML tool

Our Approach:

1. avoid materialization of **D**
2. learn model using aggregates with inequalities

We can learn the SVM model in time **sublinear** in the size of **D**.

Relational Machine Learning over Databases

We can express range of models as aggregates with inequalities, including:

- Support Vector Machines (SVM)
- k-Means Clustering
- Robust Regression with Huber Loss
- Boolean Principle Component Analysis (PCA)
- Low Rank Matrix Factorization
- ... and several other models trained with Non-Polynomial Loss functions

Functional Aggregate Queries with Additive Inequalities

FAQ-AIs encode:

1. Relations as factors $R_K : \prod_{i \in K} Dom(X_i) \to \mathbf{D}$
2. Additive Inequality E as Kronecker delta $\mathbf{1}_E$
3. Sum-Product Operations over Semiring

Examples for Sum-Product Semiring $(\mathbb{R}, +, \times)$:

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot \mathbf{1}_{a \leq c} \qquad \text{(Example I)}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot \mathbf{1}_{a \leq c} \cdot \mathbf{1}_{\frac{b}{2} \leq c} \cdot \mathbf{1}_{a^2 + \frac{b}{2} + 5c \leq 0} \qquad \text{(more AIs)}$$

$$Q(a,b) = \sum_{c} R(a,b) \cdot S(b,c) \cdot \mathbf{1}_{a \leq c} \cdot \mathbf{1}_{\frac{b}{2} \leq c} \cdot \mathbf{1}_{a^2 + \frac{b}{2} + 5c \leq 0} \qquad \text{(with free variables)}$$

Functional Aggregate Queries with Additive Inequalities

FAQ-AIs encode:

1. Relations as factors $R_K : \prod_{i \in K} Dom(X_i) \to \mathbf{D}$
2. Additive Inequality E as Kronecker delta $\mathbf{1}_E$
3. Sum-Product Operations over Semiring

Examples for Sum-Product Semiring $(\mathbb{R}, +, \times)$:

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot \mathbf{1}_{a \leq c} \quad \text{(Example I)}$$

$$Q() = \sum_{a,b,c} R(a,b) \cdot S(b,c) \cdot \mathbf{1}_{a \leq c} \cdot \mathbf{1}_{\frac{b}{2} \leq c} \cdot \mathbf{1}_{a^2 + \frac{b}{2} + 5c \leq 0} \quad \text{(more AIs)}$$

$$Q(a,b) = \sum_{c} R(a,b) \cdot S(b,c) \cdot \mathbf{1}_{a \leq c} \cdot \mathbf{1}_{\frac{b}{2} \leq c} \cdot \mathbf{1}_{a^2 + \frac{b}{2} + 5c \leq 0} \quad \text{(with free variables)}$$

Change the semiring to get different aggregates:

$$Q(a,b) = \bigvee_{c} R(a,b) \wedge S(b,c) \wedge \mathbf{1}_{a \leq c} \wedge \mathbf{1}_{\frac{b}{2} \leq c} \wedge \mathbf{1}_{a^2 + \frac{b}{2} + 5c \leq 0} \quad \text{Boolean: } (\{1, 0\}, \vee, \wedge)$$

$$Q(a,b) = \max_{c} R(a,b) \cdot S(b,c) \cdot \mathbf{1}_{a \leq c} \cdot \mathbf{1}_{\frac{b}{2} \leq c} \cdot \mathbf{1}_{a^2 + \frac{b}{2} + 5c \leq 0} \quad \text{Max-Product: } (\mathbb{R}, \max, \times)$$

$$Q(a,b) = \bigoplus_{c} R(a,b) \otimes S(b,c) \otimes \mathbf{1}_{a \leq c} \otimes \mathbf{1}_{\frac{b}{2} \leq c} \otimes \mathbf{1}_{a^2 + \frac{b}{2} + 5c \leq 0} \quad \text{Arbitrary: } (\mathbf{D}, \oplus, \otimes)$$

Functional Aggregate Queries with Additive Inequalities

$$Q(\mathbf{x}_F) = \bigoplus_{\mathbf{x}_{\mathcal{V} \setminus F}} \left(\bigotimes_{K \in \mathcal{E}_s} R_K(\mathbf{x}_K) \right) \otimes \left(\bigotimes_{K \in \mathcal{E}_\ell} \mathbf{1}_{\sum_{v \in K} \theta_v^K(x_v) \leq 0} \right)$$

Query Hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E} = \mathcal{E}_s \cup \mathcal{E}_\ell)$

- Set of variables $\mathcal{V} = \{X_1, \ldots, X_n\}$
- Set of "skeleton" hyperedges $\mathcal{E}_s$
  - Each hyperedge is defined by a factor $R_K(\mathbf{x}_K)$
- Set of "ligament" hyperedges $\mathcal{E}_\ell$
  - Each hyperedge is defined by sum of univariate functions

X_i - variable $\quad$ x_i - value in $Dom(X_i)$ $\quad$ $\mathbf{x}_K$ - tuple of values in $\prod_{i \in K} Dom(X_i)$

For this talk:

Queries without free variables.

The paper:

Generalizes all results via notion of FAQ-width.

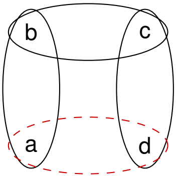
Generalized Hypertree Decompositions

Generalized Hypertree Decomposition (TD) for $\mathcal{H} = (\mathcal{V}, \mathcal{E})$:

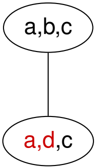
- Tree $T = (V(T), E(T))$
- Bag $\chi(t) \subseteq \mathcal{V}$ for each tree-node $t \in V(T)$

A TD must satisfy:

1. Running intersection property
2. Containment property
   - every hyperedge in $\mathcal{E}$ is covered by some bag $\chi(t)$.



Hypergraph $\mathcal{H}$



TD for $\mathcal{H}$

State-of-the-art for evaluating FAQs and FAQ-AIs

unbounded gap $\longrightarrow$

	Single TD		Data Partitioning Multiple TDs
FAQ	$\texttt{fhtw}$ $\boxed{\text{InsideOut}}$	$\geq$	$\texttt{subw}$ $\boxed{\text{PANDA}}$
FAQ-AI	$\texttt{fhtw}$ $\boxed{\text{InsideOut}}$	$\geq$	$\texttt{subw}$ $\boxed{\text{PANDA}}$
	Arbitrary semiring		Boolean semiring

We can do better for FAQ-AI! Using Relaxed Tree Decompositions.

Relaxed Tree Decompositions

Containment for Tree Decompositions:

1. every hyperedge is covered by some bag

Containment for Relaxed Tree Decompositions:

1. every ‘skeleton’ hyperedge is covered by some bag
2. every ‘ligament’ hyperedge is covered by two adjacent bags

Relaxed Tree Decompositions

Containment for Tree Decompositions:

1. every hyperedge is covered by some bag

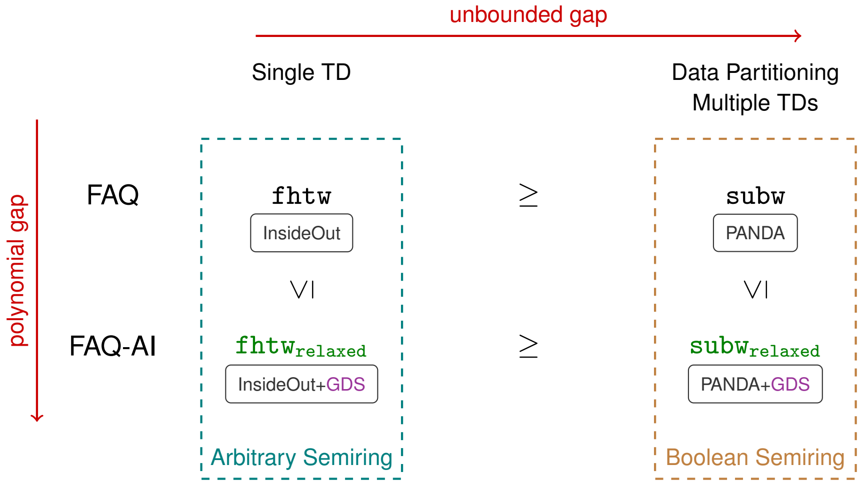
Containment for Relaxed Tree Decompositions:

1. every 'skeleton' hyperedge is covered by some bag
2. every 'ligament' hyperedge is covered by two adjacent bags

Evaluation of ligament hyperedges over two adjacent bags based on

Chazelle's geometric data structure (GDS)

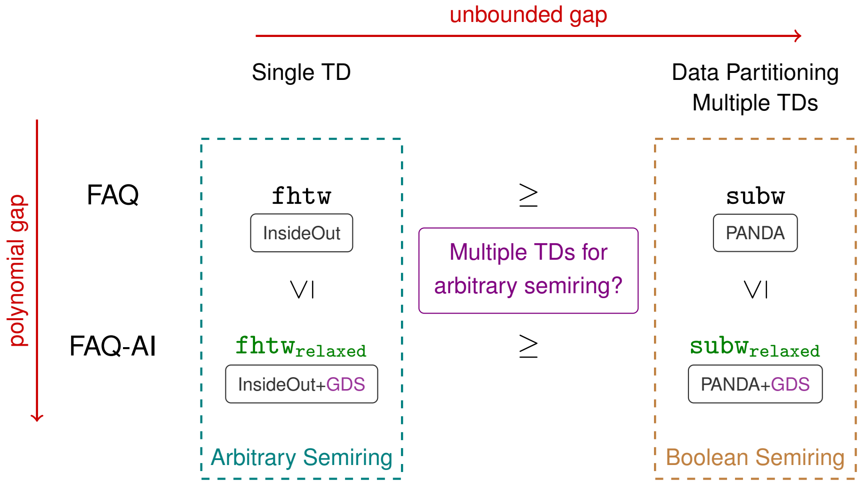
Width Measures for FAQs and FAQ-AIs



GDS = Chazelle's geometric data structure

= new result

Width Measures for FAQs and FAQ-AIs



GDS = Chazelle's geometric data structure

= new result

#PANDA: A PANDA Variant for Arbitrary Semirings

PANDA decomposes the query into several sub-queries

- Based on information theoretic inequalities

Each sub-query is:

- Computed over partitions of factors corresponding to skeleton hyperedges
- Defined by a different tree decomposition

Challenge: The results of sub-queries may overlap

- Boolean semiring: OK (✓)
- Arbitrary semiring: Not OK (✗)

#PANDA ensures that the results of sub-queries are disjoint

- Allows for aggregate computation with Arbitrary Semirings
- Recall Example II

Width Measures for FAQs and FAQ-AIs

unbounded gap →

polynomial gap ↓

	Single TD		Disjoint Partitioning Multiple TDs		General Partitioning Multiple TDs
FAQ	$\mathtt{fhtw}$ (InsideOut)	$\geq$	$\mathtt{\#subw}$ (#PANDA)	$\geq$	$\mathtt{subw}$ (PANDA)
	$\vee\vert$		$\vee\vert$		$\vee\vert$
FAQ-AI	$\mathtt{fhtw}_{\mathtt{relaxed}}$ (InsideOut+GDS)	$\geq$	$\mathtt{\#subw}_{\mathtt{relaxed}}$ (#PANDA+GDS)	$\geq$	$\mathtt{subw}_{\mathtt{relaxed}}$ (PANDA+GDS)
	Arbitrary semiring				Boolean semiring

GDS = Chazelle's geometric data structure

■ = new result

Appendix

Width Measures for FAQs and FAQ-AIs + free variables

unbounded gap $\longrightarrow$

polynomial gap $\downarrow$

	Single TD		Disjoint Partitioning Multiple TDs		General Partitioning Multiple TDs
FAQ	$\texttt{faqw}$ InsideOut	$\geq$	$\texttt{\#smfw}$ #PANDA	$\geq$	$\texttt{smfw}$ PANDA
	$\rotatebox{90}{$\geq$}$		$\rotatebox{90}{$\geq$}$		$\rotatebox{90}{$\geq$}$
FAQ-AI	$\texttt{faqw}_{\texttt{relaxed}}$ InsideOut+GDS	$\geq$	$\texttt{\#smfw}_{\texttt{relaxed}}$ #PANDA+GDS	$\geq$	$\texttt{smfw}_{\texttt{relaxed}}$ PANDA+GDS
	Arbitrary semiring				Boolean semiring

GDS = Chazelle's geometric data structure

■ = new result

What are we hiding?

unbounded gap $\longrightarrow$

polynomial gap $\downarrow$

	Single TD	Disjoint Partitioning Multiple TDs	General Partitioning Multiple TDs
FAQ	$\texttt{fhtw}$ $\log(N)$ InsideOut	$\texttt{\#subw}$ $polylog(N)$ #PANDA	$\texttt{subw}$ $polylog(N)$ PANDA
FAQ-AI	$\texttt{fhtw}_{\texttt{relaxed}}$ $\log^{\max\{k-1,1\}}(N)$ InsideOut+GDS	$\texttt{\#subw}_{\texttt{relaxed}}$ $polylog(N)$ #PANDA+GDS	$\texttt{subw}_{\texttt{relaxed}}$ $polylog(N)$ PANDA+GDS

Arbitrary semiring

Boolean semiring