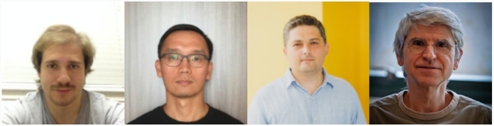
Boolean Tensor Decomposition for Conjunctive Queries with Negation

Mahmoud Abo Khamis   Hung Q. Ngo   **Dan Olteanu**   Dan Suciu
RelationalAI (USA) & **U. Oxford (UK)** & U. Washington (USA)

International Conference on Database Theory
Lisbon, March 2019



Conjunctive Queries with Negated Bounded-Degree Relations

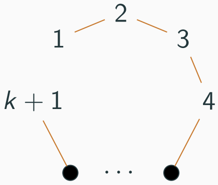
$$Q(\boldsymbol{X}_F) \leftarrow \texttt{body} \wedge \bigwedge_{S \in \overline{\mathcal{E}}} \neg R_S(\boldsymbol{X}_S),$$

- $\texttt{body}$ is the body of an arbitrary (positive) conjunctive query
- $\boldsymbol{X}_F = (X_i)_{i \in F}$ denotes a tuple of variables indexed by $F \subset \mathbb{N}$
- $\overline{\mathcal{E}}$ is the set of hyperedges of a multi-hypergraph $\overline{\mathcal{H}} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$
    - Each $S \in \overline{\mathcal{E}}$ corresponds to a **bounded-degree relation** R_S

Query Example 1/3: k-walk

Directed graph $G = ([n], E)$ with n nodes and $N = |E|$ edges.

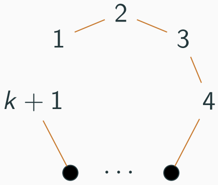
$$W() \leftarrow E(X_1, X_2) \wedge E(X_2, X_3) \wedge \cdots \wedge E(X_k, X_{k+1})$$



Hypergraph $\overline{\mathcal{H}}$ is empty since W has no negated relations.

Query Example 1/3: k-walk

Directed graph $G = ([n], E)$ with n nodes and $N = |E|$ edges.



$$W() \leftarrow E(X_1, X_2) \wedge E(X_2, X_3) \wedge \cdots \wedge E(X_k, X_{k+1})$$

Hypergraph $\overline{\mathcal{H}}$ is empty since W has no negated relations.

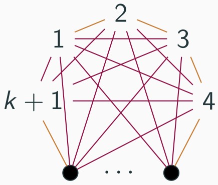
Time complexity:

- $\mathcal{O}(kN \log N)$ [Yannakakis'81]

Query Example 2/3: k-path

Directed graph $G = ([n], E)$ with n nodes and $N = |E|$ edges.

$$P() \leftarrow E(X_1, X_2) \wedge E(X_2, X_3) \wedge \cdots \wedge E(X_k, X_{k+1}) \wedge \bigwedge_{\substack{i,j \in [k+1] \\ i+1<j}} X_i \neq X_j$$



Disequality is negation of bounded-degree equality relation:

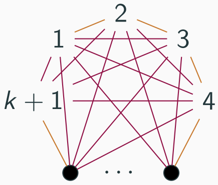
$$X_i \neq X_j \equiv \neg(X_i = X_j)$$

Hypergraph $\overline{\mathcal{H}} = ([k+1], \{(i,j) \mid i,j \in [k+1], i+1 < j\})$

Query Example 2/3: k-path

Directed graph $G = ([n], E)$ with n nodes and $N = |E|$ edges.

$$P() \leftarrow E(X_1, X_2) \wedge E(X_2, X_3) \wedge \cdots \wedge E(X_k, X_{k+1}) \wedge \bigwedge_{\substack{i,j \in [k+1] \\ i+1<j}} X_i \neq X_j$$



Disequality is negation of bounded-degree equality relation:

$$X_i \neq X_j \equiv \neg(X_i = X_j)$$

Hypergraph $\overline{\mathcal{H}} = ([k+1], \{(i,j) \mid i,j \in [k+1], i+1 < j\})$

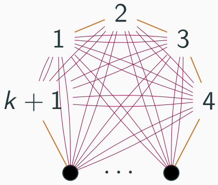
Time complexity:

- $\mathcal{O}(k^k N \log N)$ [Plehn, Voigt'90]
- $2^{\mathcal{O}(k)} N \log N$ using color-coding [Alon, Yuster, Zwick'95]

Query Example 3/3: induced (chordless) k-path

Directed graph $G = ([n], E)$ with n nodes and $N = |E|$ edges.

$$I() \leftarrow E(X_1, X_2) \wedge E(X_2, X_3) \wedge \cdots \wedge E(X_k, X_{k+1}) \wedge \bigwedge_{\substack{i,j \in [k+1] \\ i+1<j}} (\neg E(X_i, X_j) \wedge X_i \neq X_j))$$

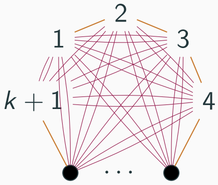


Each edge twice in $\overline{\mathcal{H}}$ due to negated edge relation and disequality

Query Example 3/3: induced (chordless) k-path

Directed graph $G = ([n], E)$ with n nodes and $N = |E|$ edges.

$$I() \leftarrow E(X_1, X_2) \wedge E(X_2, X_3) \wedge \cdots \wedge E(X_k, X_{k+1}) \wedge \bigwedge_{\substack{i,j \in [k+1] \\ i+1<j}} (\neg E(X_i, X_j) \wedge X_i \neq X_j))$$



Each edge twice in $\overline{\mathcal{H}}$ due to negated edge relation and disequality

Time complexity:

- W[2]-hard [Chen, Flum'07]
- $\mathcal{O}(f(k,d) N \log N)$ if G has *maximum degree* d;
  f depends exponentially on k and d [Plehn, Voigt'90]

Main Result: Time Complexity for Query Evaluation

Database with relations of size $\mathcal{O}(N)$

Query Q with positive `body` and negation hypergraph $\overline{\mathcal{H}}$

Main Result: Time Complexity for Query Evaluation

Database with relations of size $\mathcal{O}(N)$

Query Q with positive $\texttt{body}$ and negation hypergraph $\overline{\mathcal{H}}$

Using a reduction to InsideOut [Abo Khamis et al'16]

$$\mathcal{O}\Big(\underbrace{F_{\text{InsideOut}}(Q)}_{\substack{\text{depends on structure of } \overline{\mathcal{H}} \\ \text{degree of relations} \\ \text{and InsideOut}}} \cdot \underbrace{\log N \cdot (N^{\text{fhtw}_F(\texttt{body})} + |\text{output}|)}_{\text{same as for } \texttt{body}}\Big)$$

Main Result: Time Complexity for Query Evaluation

Database with relations of size $\mathcal{O}(N)$

Query Q with positive `body` and negation hypergraph $\overline{\mathcal{H}}$

Using a reduction to InsideOut [Abo Khamis et al'16]

$$\mathcal{O}(\underbrace{F_{\mathsf{InsideOut}}(Q)}_{\substack{\text{depends on structure of } \overline{\mathcal{H}}\\ \text{degree of relations}\\ \text{and InsideOut}}} \cdot \underbrace{\log N \cdot (N^{\mathsf{fhtw}_F(\mathtt{body})} + |\mathsf{output}|)}_{\text{same as for } \mathtt{body}})$$

Using a reduction to PANDA [Abo Khamis et al'17]

$$\mathcal{O}(\underbrace{F_{\mathsf{PANDA}}(Q)}_{\substack{\text{depends on structure of } \overline{\mathcal{H}}\\ \text{degree of relations}\\ \text{and PANDA}}} \cdot \underbrace{(\mathsf{poly}(\log N) \cdot N^{\mathsf{subw}_F(\mathtt{body})} + \log N \cdot |\mathsf{output}|)}_{\text{same as for } \mathtt{body}})$$

Our Query Evaluation Approach

1. Untangling negated bounded-degree relations

   Rewrite negated subquery into not-all-equal conjunction

   Not-all-equal (NAE) is multi-dimensional analog of $\neq$

2. Boolean tensor decomposition for NAE conjunction

   Probabilistic construction with efficient derandomization

   Generalization of color-coding from cliques of $\neq$ to NAE conjunctions

3. Use existing algorithms InsideOut and PANDA

   Decomposition preserves fhtw and subw of positive body

Untangling Bounded-Degree Relations

The Untangling Step via an Example

Given: Database with relations R, S, T with sizes $\mathcal{O}(N)$

Task: Compute the Boolean query

$$Q() \leftarrow R(A, B) \wedge S(B, C) \wedge \neg T(A, C)$$

What is the time complexity for computing Q?

- $\mathcal{O}(N^2)$ trivially: First join R and S and then filter with T

The Untangling Step via an Example

Given: Database with relations R, S, T with sizes $\mathcal{O}(N)$

Task: Compute the Boolean query

$$Q() \leftarrow R(A, B) \wedge S(B, C) \wedge \neg T(A, C)$$

What is the time complexity for computing Q?

- $\mathcal{O}(N^2)$ trivially: First join R and S and then filter with T
- Subquadratic if T has degree bounded by a constant

Intermezzo: Bounded-degree Relations

Classical notion of degree $\Delta(T)$ of relation $T(A, C)$:

Maximum number of tuples with the same value for A or C

Intermezzo: Bounded-degree Relations

Classical notion of degree $\Delta(T)$ of relation $T(A, C)$:

Maximum number of tuples with the same value for A or C

Our notion of degree $deg(T)$ accounts for the arity of T:

Smallest number d such that T is a disjoint union of d **matchings**

If T has schema S: $\Delta(T) \leq deg(T) \leq |S| \cdot (\Delta(T) - 1) + 1$

Intermezzo: Bounded-degree Relations

Classical notion of degree $\Delta(T)$ of relation $T(A, C)$:

Maximum number of tuples with the same value for A or C

Our notion of degree $deg(T)$ accounts for the arity of T:

Smallest number d such that T is a disjoint union of d **matchings**

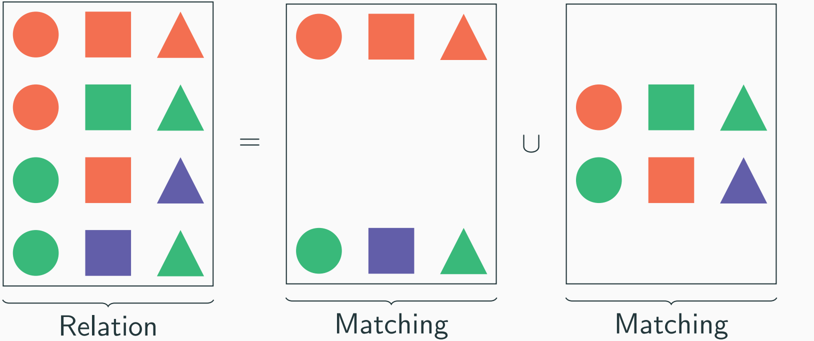
If T has schema S: $\Delta(T) \leq deg(T) \leq |S| \cdot (\Delta(T) - 1) + 1$

Assumption in our example: T has degree 2, that is,

$\exists$ matchings M_1 and M_2: $T(A, C) \equiv M_1(A, C) \vee M_2(A, C)$

Intermezzo: What is a Matching?

$\boxed{M \text{ is matching iff } \forall \boldsymbol{x}_S, \boldsymbol{x}'_S \in M \text{ either } \boldsymbol{x}_S = \boldsymbol{x}'_S \text{ or } \forall i \in S : x_i \neq x'_i}$



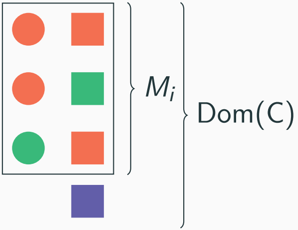
Linear-time decomposition of relation R into $|S| \cdot \Delta(R)$ matchings

Intermezzo: Negating a Binary Matching

Assume matching $M_i(A, C)$. When is (●, ■) $\in \neg M_i$?

1. ■ is in the domain of C but not in M_i

$$W_i(C) = Dom(C) \land \neg(\exists_X M_i(X, C))$$

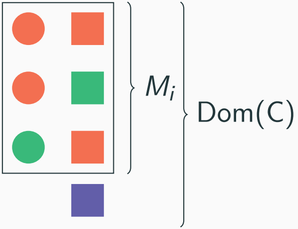


Intermezzo: Negating a Binary Matching

Assume matching $M_i(A, C)$. When is $(,) \in \neg M_i$?

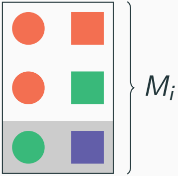
1. is in the domain of C but not in M_i

   $$W_i(C) = Dom(C) \wedge \neg(\exists_X M_i(X, C))$$



2. or is paired with $\neq$ in M_i

   $$\exists_{A_i}(M(A_i, C) \wedge A_i \neq A)$$

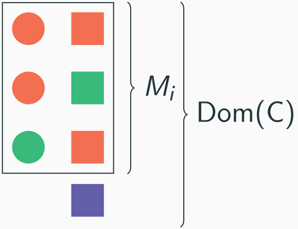


Intermezzo: Negating a Binary Matching

Assume matching $M_i(A, C)$. When is (●, ■) $\in \neg M_i$?

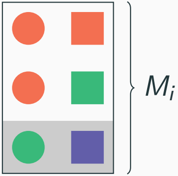
1. ■ is in the domain of C but not in M_i

   $$W_i(C) = Dom(C) \wedge \neg(\exists_X M_i(X, C))$$



2. or ■ is paired with ● $\neq$ ● in M_i

   $$\exists_{A_i}(M(A_i, C) \wedge A_i \neq A)$$



$$\boxed{\neg M_i(A, C) \equiv W_i(C) \vee \exists_{A_i}(M(A_i, C) \wedge A_i \neq A)}$$

Negating a Bounded-degree Relation

Recall: $T(A, C) \equiv M_1(A, C) \vee M_2(A, C)$, M_1 and M_2 matchings

$$\neg T(A, C) \equiv \underbrace{\neg M_1(A, C)}_{W_1(C) \vee \exists_{A_1}(M_1(A_1, C) \wedge A_1 \neq A)} \wedge \underbrace{\neg M_2(A, C)}_{W_2(C) \vee \exists_{A_2}(M_2(A_2, C) \wedge A_2 \neq A)}$$

Flatten out $\neg T(A, C)$ into a disjunction of four conjunctions:

$$W_1(C) \wedge W_2(C)$$
$$W_1(C) \wedge M_2(A_2, C) \wedge A \neq A_2$$
$$W_2(C) \wedge M_1(A_1, C) \wedge A \neq A_1$$
$$M_1(A_1, C) \wedge M_2(A_2, C) \wedge A \neq A_1 \wedge A \neq A_2$$

The negative subqueries are now disequalities on variables

The Untangling Step

The query Q becomes $Q_1 \vee Q_2 \vee Q_3 \vee Q_4$:

$$Q_1() \leftarrow R(A,B) \wedge S(B,C) \wedge \underline{W_1(C) \wedge W_2(C)}$$

$$Q_2() \leftarrow R(A,B) \wedge S(B,C) \wedge \underline{W_1(C) \wedge M_2(A_2,C)} \wedge A \neq A_2$$

$$Q_3() \leftarrow R(A,B) \wedge S(B,C) \wedge \underline{W_2(C) \wedge M_1(A_1,C)} \wedge A \neq A_1$$

$$Q_4() \leftarrow R(A,B) \wedge S(B,C) \wedge \underline{M_1(A_1,C) \wedge M_2(A_2,C)} \wedge A \neq A_1 \wedge A \neq A_2$$

Our rewriting

- **extends** the positive body of Q
  - Replaced T by (conjunctions of some of) its matchings
  - Added unary relations
- **preserves** the data complexity (fhtw and subw) of body
- **blows up** the query size exponentially in the degree

Boolean Tensor Decomposition

How to Evaluate Conjunctions of Disequalities Efficiently?

$\forall i \in [\log N], f_i : \mathrm{Dom}(A) \to \{0,1\}$ gives the i-th bit of A

$$A \neq A_2 \equiv \bigvee_{x \in \{0,1\}} \bigvee_{i \in [\log N]} f_i(A) = x \wedge f_i(A_2) \neq x$$

This is a Boolean decomposition of $A \neq A_2$:

- **Rank** r is the number $2 \log N$ of conjuncts
- Each conjunct is a conjunction of positive unary relations

Analogy: Each function f_i is a "coloring":

It assigns a $\{0,1\}$ color to each element of $\mathrm{Dom}(A)$

How to Evaluate Conjunctions of Disequalities Efficiently?

Q_2 becomes the disjunction of $2 \log N$ acyclic queries

$Q_2^{x,i} \leftarrow R(A, B) \wedge S(B, C) \wedge W_1(C) \wedge M_2(A_2, C) \wedge f_i(A) = x \wedge f_i(A_2) \neq x$

Time complexity:

- $Q_2^{x,i}$ can be answered in time $\mathcal{O}(N \log N)$
- Q_2 can be answered in time $\mathcal{O}(N \log^2 N)$
- Further shave off a $\log N$ factor (see paper)

Boolean semiring $\rightarrow$ Bit-vector semiring

How to Evaluate Conjunctions of Disequalities Efficiently?

Q_2 becomes the disjunction of $2\log N$ acyclic queries

$Q_2^{x,i} \leftarrow R(A,B) \wedge S(B,C) \wedge W_1(C) \wedge M_2(A_2,C) \wedge f_i(A) = x \wedge f_i(A_2) \neq x$

Time complexity:

- $Q_2^{x,i}$ can be answered in time $\mathcal{O}(N \log N)$
- Q_2 can be answered in time $\mathcal{O}(N \log^2 N)$
- Further shave off a $\log N$ factor (see paper)

Boolean semiring $\rightarrow$ Bit-vector semiring

Q_4 is more involved: $A \neq A_1 \wedge A \neq A_2$

- Three-dimensional tensor of Boolean rank $log^2 N$
- **We can reduce the rank to** $\log N$

Boolean Tensor Decomposition for $A \neq A_1 \wedge A \neq A_2$

$$A \neq A_1 \wedge A \neq A_2 \equiv \bigvee_{\substack{(c,c_1,c_2)\in\{0,1\}^3 \\ c\neq c_1 \wedge c \neq c_2}} \bigvee_{f\in\mathcal{F}} f(A) = c \wedge f(A_1) = c_1 \wedge f(A_2) = c_2$$

There exists a family $\mathcal{F}$ of functions $f : Dom(A) \to \{0,1\}$:

- $\forall (a, a_1, a_2) \in \mathrm{Dom}(A)^3$ st $\quad a \neq a_1 \quad \wedge \quad a \neq a_2$:

  $\exists f \in \mathcal{F}$ st $f(a) \neq f(a_1) \wedge f(a) \neq f(a_2)$

- $|\mathcal{F}| = \mathcal{O}(\log N)$

- $\mathcal{F}$ can be constructed in time $\mathcal{O}(N \log N)$

Intermezzo: Disjunct Matrices

k-disjunct $t \times N$ matrix X:

$\forall j \in [N], S \subseteq [N]$ st $|S| \leq k, j \notin S$:

$\exists i \in [t]$ st $X_{i,j} = 1, (X_{i,j'})_{j' \in S} = \mathbf{0}$

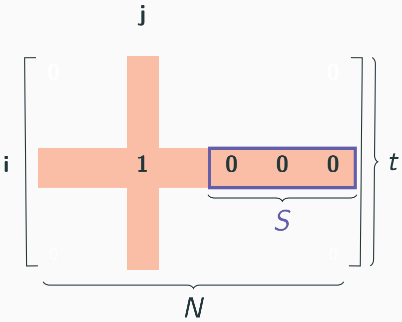
$$\begin{array}{cc}
 & \mathbf{j} \\
\mathbf{i} & \underbrace{\left.\left[\begin{array}{ccc} 0 & & 0 \\ & \mathbf{1} & \underbrace{\boxed{\mathbf{0} \quad \mathbf{0} \quad \mathbf{0}}}_{S} \\ 0 & & 0 \end{array}\right]\right\} t}_{N}
\end{array}$$

Intermezzo: Disjunct Matrices

k-disjunct $t \times N$ matrix X:

$\forall j \in [N], S \subseteq [N]$ st $|S| \leq k, j \notin S$:

$\exists i \in [t]$ st $X_{i,j} = 1, (X_{i,j'})_{j' \in S} = \mathbf{0}$



We can construct a k-disjunct matrix X [Porat, Rothschild'11]

- with $t = \mathcal{O}(k^2 \log N)$
- in time $\mathcal{O}(k^2 N \log N)$

How to Use Disjunct Matrices for Our Problem?

Each row i = function f_i in $\mathcal{F}$

$X_{i,j} = f_i(A)$

$X_{i,S} = [f_i(A_1), f_i(A_2)] \Rightarrow k = 2$

- X has size $\mathcal{O}(\log N) \times N$
- X constructed in time $\mathcal{O}(N \log N)$

$$\begin{array}{cc} & \mathbf{j} \\ \mathbf{i} & \underbrace{\left[\begin{array}{cccc} 0 & & & 0 \\ & \mathbf{1} & \underbrace{\boxed{\mathbf{0} \quad \mathbf{0}}}_{S} & \\ 0 & & & 0 \end{array}\right]}_{N} \left.\vphantom{\begin{array}{c}0\\0\\0\end{array}}\right\} \mathcal{O}(\log N) \end{array}$$

Generalizing the Example

Negating a Ternary Matching

Matching $M(X_1, X_2, X_3)$. Single out (wlog) X_3.

Tuple $(x_1, x_2, x_3) \in \neg M$ iff

1. At least one of x_1 or x_2 is not in M OR
2. x_1 and x_2 are in M, but at least one is paired with $x_3' \neq x_3$ OR
   they are paired with diff. X_3 values

Negating a Ternary Matching

Matching $M(X_1, X_2, X_3)$. Single out (wlog) X_3.

Tuple $(x_1, x_2, x_3) \in \neg M$ iff

1. At least one of x_1 or x_2 is not in M OR
2. x_1 and x_2 are in M, but at least one is paired with $x_3' \neq x_3$ OR
   they are paired with diff. X_3 values

$$\neg M(X_1, X_2, X_3) \equiv (W_1(X_1) \vee W_2(X_2)) \vee \\ \exists_{Y_1} \exists_{Y_2} [\mathsf{NAE}(Y_1, Y_2, X_3) \wedge M(X_1, \_, Y_1) \wedge M(\_, X_2, Y_2)]$$

$$\mathsf{NAE}(Y_1, Y_2, X_3) \stackrel{def}{=} \neg(Y_1 = Y_2 \wedge Y_1 = X_3 \wedge Y_2 = X_3) \\ = \quad Y_1 \neq Y_2 \vee Y_1 \neq X_3 \vee Y_2 \neq X_3$$

See paper for extension to k-ary matchings.

General Untangling

Query Q rewritten into a disjunction of queries

$$Q_i(\boldsymbol{X}_F) \leftarrow \texttt{body}_i \wedge \bigwedge_{S \in \mathcal{A}_i} \mathsf{NAE}(\boldsymbol{Z}_S).$$

Data complexity (fhtw and subw) of $\texttt{body}_i$ same as for $\texttt{body}$

Number of queries Q_i exponential in the degree

General Boolean Tensor Decomposition

$$\underbrace{\bigwedge_{S} \mathrm{NAE}(\boldsymbol{Z}_S)}_{\substack{\text{rank-r tensor} \\ \text{multivariate function}}} \equiv \bigvee_{j\in[r]} \underbrace{\bigwedge_{i\in\bigcup_S \boldsymbol{Z}_S} \underbrace{f_i^{(j)}(Z_i)}_{\text{univariate function}}}_{\text{rank-1 tensor}}$$

General Boolean Tensor Decomposition

$$\underbrace{\bigwedge_{S} \mathsf{NAE}(\boldsymbol{Z}_S)}_{\substack{\text{rank-r tensor}\\ \text{multivariate function}}} \equiv \bigvee_{j\in[r]} \underbrace{\bigwedge_{i\in\bigcup_S \boldsymbol{Z}_S} \underbrace{f_i^{(j)}(Z_i)}_{\text{univariate function}}}_{\text{rank-1 tensor}}$$

Multi-hypergraph $\mathcal{G} = (\bigcup_S \boldsymbol{Z}_S, \mathcal{A})$ of $\bigwedge_S \mathsf{NAE}(\boldsymbol{Z}_S)$

Boolean rank $r = P(\mathcal{G}, c) \cdot |\mathcal{F}|$ depends on:

- Chromatic polynomial of $\mathcal{G}$ using $c \leq |\bigcup_S \boldsymbol{Z}_S|$ colors

  $c =$ maximum chromatic number of a hypergraph defined by any homomorphic image of $\mathcal{G}$
- Size of a family of hash functions that represent proper c-colorings of homomorphic images of $\mathcal{G}$