Probabilistic Query Evaluation: Towards Tractable Combined Complexity

Mikaël Monet[1,2], supervised by Pierre Senellart[2,3] and Antoine Amarilli[1]

May 31th, 2017

[1]LTCI, Télécom ParisTech, Université Paris-Saclay; Paris, France

[2]Inria Paris; Paris, France

[3]École normale supérieure, PSL Research University; Paris, France

Introduction

- Uncertainty in data

→ Untrustworthy sources, automated information extraction, imperfect sensor precision in experimental sciences, etc.

- Need framework to model this uncertainty and reason about it

Introduction

- Uncertainty in data

→ Untrustworthy sources, automated information extraction, imperfect sensor precision in experimental sciences, etc.

- Need framework to model this uncertainty and reason about it

→ Probabilistic Databases!

Plan

1) Define TID model and probabilistic query evaluation (PQE)

Plan

1) Define TID model and probabilistic query evaluation (PQE)

2) Existing approaches (efficient PQE in the data)

Plan

1) Define TID model and probabilistic query evaluation (PQE)

2) Existing approaches (efficient PQE in the data)

3) Efficient PQE in the query and the data

Plan

1) Define TID model and probabilistic query evaluation (PQE)

2) Existing approaches (efficient PQE in the data)

3) Efficient PQE in the query and the data

4) Efficient PQE in the data, reasonable complexity in the query

Tuple-independent databases (TID)

- Probabilistic databases: model uncertainty about data
- Simplest model: tuple-independent databases (TID)
  - A relational database I
  - A probability valuation π mapping each fact of I to $[0, 1]$
- Semantics of a TID (I, π): a probability distribution on $I' \subseteq I$:
  - Each fact $F \in I$ is either present or absent with probability $\pi(F)$
  - Assume independence across facts

Example: TID

S		
a	b	.5
a	c	.2

Example: TID

S		
a	b	.5
a	c	.2

This TID (I, π) represents the following probability distribution:

Example: TID

S		
a	b	.5
a	c	.2

This TID (I, π) represents the following probability distribution:

$.5 \times .2$

S	
a	b
a	c

Example: TID

S		
a	b	.5
a	c	.2

This TID (I, π) represents the following probability distribution:

$.5 \times .2$

S	
a	b
a	c

$.5 \times (1 - .2)$

S	
a	b

Example: TID

S		
a	b	.5
a	c	.2

This TID (I, π) represents the following probability distribution:

$.5 \times .2$

S	
a	b
a	c

$.5 \times (1 - .2)$

S	
a	b

$(1 - .5) \times .2$

S	
a	c

Example: TID

S		
a	b	.5
a	c	.2

This TID (I, π) represents the following probability distribution:

$.5 \times .2$

S	
a	b
a	c

$.5 \times (1 - .2)$

S	
a	b

$(1 - .5) \times .2$

S	
a	c

$(1 - .5) \times (1 - .2)$

S

Probabilistic query evaluation (PQE)

Let us fix:

- Relational signature σ
- Class $\mathcal{I}$ of relational instances on σ (e.g., acyclic, treelike)
- Class $\mathcal{Q}$ of Boolean queries (e.g., paths, trees)

Probabilistic query evaluation (PQE)

Let us fix:

- Relational signature σ
- Class $\mathcal{I}$ of relational instances on σ (e.g., acyclic, treelike)
- Class $\mathcal{Q}$ of Boolean queries (e.g., paths, trees)

Probabilistic query evaluation (PQE) problem for $\mathcal{Q}$ and $\mathcal{I}$:

- Given a query $q \in \mathcal{Q}$
- Given an instance $I \in \mathcal{I}$ and a probability valuation π
- Compute the probability that (I, π) satisfies q

Probabilistic query evaluation (PQE)

Let us fix:

- Relational signature σ
- Class $\mathcal{I}$ of relational instances on σ (e.g., acyclic, treelike)
- Class $\mathcal{Q}$ of Boolean queries (e.g., paths, trees)

Probabilistic query evaluation (PQE) problem for $\mathcal{Q}$ and $\mathcal{I}$:

- Given a query $q \in \mathcal{Q}$
- Given an instance $I \in \mathcal{I}$ and a probability valuation π
- Compute the probability that (I, π) satisfies q

$\rightarrow$ $\Pr((I, \pi) \models q) = \sum_{J \subseteq I,\ J \models q} \Pr(J)$

Complexity of probabilistic query evaluation (PQE)

Question: what is the (data, combined) complexity of PQE depending on the class $\mathcal{Q}$ of queries and class $\mathcal{I}$ of instances?

Data complexity results: related work

- Existing data dichotomy result on queries [Dalvi & Suciu, 2012]
  - $\mathcal{I}$ is all instances
  - There is a class $\mathcal{S} \subseteq$ UCQs of safe queries

Data complexity results: related work

- Existing data dichotomy result on queries [Dalvi & Suciu, 2012]
  - $\mathcal{I}$ is all instances
  - There is a class $\mathcal{S} \subseteq$ UCQs of safe queries
- $\rightarrow$ PQE is PTIME for any $q \in \mathcal{S}$

Data complexity results: related work

- Existing data dichotomy result on queries [Dalvi & Suciu, 2012]
  - $\mathcal{I}$ is all instances
  - There is a class $\mathcal{S} \subseteq$ UCQs of safe queries

$\rightarrow$ PQE is PTIME for any $q \in \mathcal{S}$

$\rightarrow$ PQE is **#P-hard** for any $q \in$ UCQs $\setminus \mathcal{S}$

Data complexity results: related work

- Existing data dichotomy result on queries [Dalvi & Suciu, 2012]
  - $\mathcal{I}$ is all instances
  - There is a class $\mathcal{S} \subseteq$ UCQs of safe queries
  - $\rightarrow$ PQE is PTIME for any $q \in \mathcal{S}$
  - $\rightarrow$ PQE is **#P-hard** for any $q \in$ UCQs $\setminus \mathcal{S}$

- Existing data dichotomy result on instances

Data complexity results: related work

- Existing data dichotomy result on queries [Dalvi & Suciu, 2012]
  - $\mathcal{I}$ is all instances
  - There is a class $\mathcal{S} \subseteq$ UCQs of safe queries
  - $\rightarrow$ PQE is PTIME for any $q \in \mathcal{S}$
  - $\rightarrow$ PQE is **#P-hard** for any $q \in$ UCQs $\setminus \mathcal{S}$

- Existing data dichotomy result on instances
  - $\rightarrow$ PQE for MSO on bounded-treewidth instances has linear data complexity [Amarilli, Bourhis, & Senellart, 2015]

Data complexity results: related work

- Existing data dichotomy result on queries [Dalvi & Suciu, 2012]
  - $\mathcal{I}$ is all instances
  - There is a class $\mathcal{S} \subseteq$ UCQs of safe queries

  $\rightarrow$ PQE is PTIME for any $q \in \mathcal{S}$

  $\rightarrow$ PQE is **#P-hard** for any $q \in$ UCQs $\setminus \mathcal{S}$

- Existing data dichotomy result on instances

  $\rightarrow$ PQE for MSO on bounded-treewidth instances has linear data complexity [Amarilli, Bourhis, & Senellart, 2015]

  $\rightarrow$ There is an FO query for which PQE is **#P-hard** on any unbounded-treewidth graph family $\mathcal{I}$ (under some assumptions) [Amarilli, Bourhis, & Senellart, 2016]

Data complexity results: related work

- Existing **data dichotomy result** on queries [Dalvi & Suciu, 2012]
  - $\mathcal{I}$ is all instances
  - There is a class $\mathcal{S} \subseteq$ UCQs of **safe queries**

  $\rightarrow$ PQE is **PTIME** for any $q \in \mathcal{S}$

  $\rightarrow$ PQE is **#P-hard** for any $q \in$ UCQs $\setminus \mathcal{S}$

- Existing **data dichotomy result** on instances

  $\rightarrow$ PQE for **MSO** on **bounded-treewidth** instances has **linear** data complexity [Amarilli, Bourhis, & Senellart, 2015]

  $\rightarrow$ There is an FO query for which PQE is **#P-hard** on **any** unbounded-treewidth graph family $\mathcal{I}$ (under some assumptions) [Amarilli, Bourhis, & Senellart, 2016]

What about **combined** complexity?

Wish list

We want:

- PQE tractable in combined complexity

OR

- PQE tractable in the data, reasonable in the query

Restrict to CQs on graph signatures

$\exists x\, y\, z\, t\ R(x, y) \wedge S(y, z) \wedge S(t, z)$

R		
a	b	.1
b	c	.1
c	d	.05
d	a	1.
d	b	.8

S		
b	d	.7

Restrict to CQs on graph signatures

$$\exists x\, y\, z\, t\; R(x,y) \wedge S(y,z) \wedge S(t,z) \quad \rightarrow \quad x \xrightarrow{R} y \xrightarrow{S} z \xleftarrow{S} t$$

R

a	b	.1
b	c	.1
c	d	.05
d	a	1.
d	b	.8

S

b	d	.7

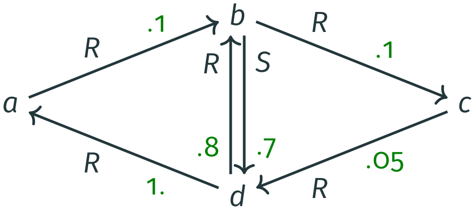
Restrict to CQs on graph signatures

$\exists x\, y\, z\, t\; R(x, y) \wedge S(y, z) \wedge S(t, z) \quad \rightarrow \quad x \xrightarrow{R} y \xrightarrow{S} z \xleftarrow{S} t$

R		
a	*b*	.1
b	*c*	.1
c	*d*	.05
d	*a*	1.
d	*b*	.8

S		
b	*d*	.7

$\rightarrow$



Restrict instances to trees

$\mathcal{Q} =$ one-way paths (1WP), $\mathcal{I} =$ polytrees (PT)

Restrict instances to trees

$\mathcal{Q} =$ one-way paths (1WP), $\mathcal{I} =$ polytrees (PT)

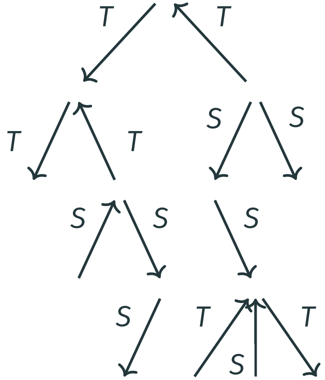
Q: $\xrightarrow{T} \xrightarrow{S} \xrightarrow{S} \xrightarrow{S} \xrightarrow{T}$

Restrict instances to trees

$\mathcal{Q}$ = one-way paths (1WP), $\mathcal{I}$ = polytrees (PT)

Q:

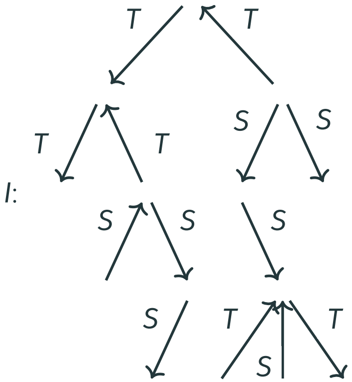
I:



+ prob. for each edge

Restrict instances to trees

$\mathcal{Q}$ = one-way paths (1WP), $\mathcal{I}$ = polytrees (PT)



+ prob. for each edge

Proposition

PQE of 1WP *on* PT *is* ***#P-hard***

Our graph classes

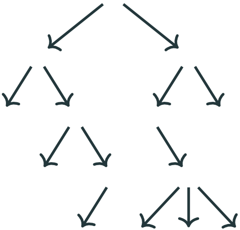
1WP

$\xrightarrow{R}\ \xrightarrow{S}\ \xrightarrow{S}\ \xrightarrow{T}$

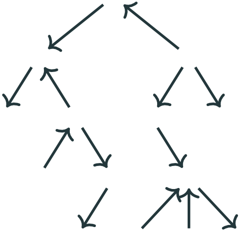
2WP

$\xrightarrow{R}\ \xleftarrow{S}\ \xleftarrow{S}\ \xrightarrow{T}\ \xleftarrow{R}$

DWT



PT



$$\text{1WP} \subseteq \text{2WP} \subseteq \text{PT}, \quad \text{1WP} \subseteq \text{DWT} \subseteq \text{PT}$$

$$\text{PT} \subseteq \text{Connected} \subseteq \text{All}$$

Results

$\downarrow Q$ $\quad I\rightarrow$	1WP	2WP	DWT	PT	Connected
1WP					
2WP					
DWT		PTIME			
PT					#P-hard
Connected					

$\geqslant 2$ labels

Results

$\downarrow Q$ $I \rightarrow$	1WP	2WP	DWT	PT	Connected
1WP					
2WP					
DWT		PTIME			
PT					#P-hard
Connected					

$\geqslant 2$ labels

$\downarrow Q$ $I \rightarrow$	1WP	2WP	DWT	PT	Connected
1WP					
2WP					
DWT		PTIME			
PT					#P-hard
Connected					

No labels

Led to a publication in PODS'2017

Contributions:

- Detailed study of the **combined** complexity of PQE

Led to a publication in PODS'2017

Contributions:

- Detailed study of the **combined** complexity of PQE
- Focus on CQs on arity-two signatures

Led to a publication in PODS'2017

Contributions:

- Detailed study of the **combined** complexity of PQE
- Focus on CQs on arity-two signatures
- Showed the importance of various features on the problem: **labels, global orientation, branching, connectedness**

Led to a publication in PODS'2017

Contributions:

- Detailed study of the **combined** complexity of PQE
- Focus on CQs on arity-two signatures
- Showed the importance of various features on the problem: **labels, global orientation, branching, connectedness**
- Established the complexity for all combinations of the graph classes we considered

Led to a publication in PODS'2017

Contributions:

- Detailed study of the **combined** complexity of PQE
- Focus on CQs on arity-two signatures
- Showed the importance of various features on the problem: **labels, global orientation, branching, connectedness**
- Established the complexity for all combinations of the graph classes we considered

Drawbacks and future work:

- Our graph classes may seem "arbitrary"

Led to a publication in PODS'2017

Contributions:

- Detailed study of the **combined** complexity of PQE
- Focus on CQs on arity-two signatures
- Showed the importance of various features on the problem: **labels, global orientation, branching, connectedness**
- Established the complexity for all combinations of the graph classes we considered

Drawbacks and future work:

- Our graph classes may seem "arbitrary"
- Not yet a dichotomy, just starting to understand the problem
- Practical applications?

Lowering our expectations

What if we want the complexity to be:

- Tractable in the data
- Not *too* horrible in the query

Can we then support a more expressive query language (e.g., disjunctions, negations, recursion)?

Starting point

- Existing data dichotomy result on instances

Starting point

- Existing data dichotomy result on instances
  - $\rightarrow$ PQE for MSO on bounded-treewidth instances has linear data complexity [Amarilli, Bourhis, & Senellart, 2015]

Starting point

- Existing data dichotomy result on instances
  - → PQE for MSO on bounded-treewidth instances has linear data complexity [Amarilli, Bourhis, & Senellart, 2015]
  - Problem: nonelementary in the query $2^{2^{\cdot^{\cdot^{\cdot^{|Q|}}}}}$

Starting point

- Existing data dichotomy result on instances
  → PQE for MSO on bounded-treewidth instances has linear data complexity [Amarilli, Bourhis, & Senellart, 2015]
  - Problem: nonelementary in the query $2^{2^{\cdot^{\cdot^{\cdot^{|Q|}}}}}$

The instance class is parameterized

Starting point

- Existing data dichotomy result on instances
  - → PQE for MSO on bounded-treewidth instances has linear data complexity [Amarilli, Bourhis, & Senellart, 2015]
    - Problem: nonelementary in the query $2^{2^{\cdot^{\cdot^{\cdot^{|Q|}}}}}$

The instance class is parameterized

Idea: one parameter for the instances and one parameter for the queries

Parameterized Complexity

Idea: one parameter k_I for the instance (treewidth) AND one parameter k_Q for the query

Parameterized Complexity

Idea: one parameter k_I for the instance (treewidth) AND one parameter k_Q for the query

- **Instance** classes $\mathcal{I}_1, \mathcal{I}_2, \cdots$

Parameterized Complexity

Idea: one parameter k_I for the instance (treewidth) AND one parameter k_Q for the query

- **Instance** classes $\mathcal{I}_1, \mathcal{I}_2, \cdots$
- **Query** classes $\mathcal{Q}_1, \mathcal{Q}_2, \cdots$

Parameterized Complexity

Idea: one parameter k_I for the instance (treewidth) AND one parameter k_Q for the query

- **Instance** classes $\mathcal{I}_1, \mathcal{I}_2, \cdots$
- **Query** classes $\mathcal{Q}_1, \mathcal{Q}_2, \cdots$

Definition

The problem is *fixed-parameter tractable (FPT) linear* if there exists a computable function f such that it can be solved in time $f(k_I, k_Q) \times |Q| \times |I|$

Publication in ICDT'2017

1) A new language...

- We introduce the language of ***intentional-clique-guarded Datalog*** (ICG-Datalog), parameterized by *body-size* k_P

Publication in ICDT'2017

1) A new language...

- We introduce the language of ***intentional-clique-guarded Datalog*** (ICG-Datalog), parameterized by *body-size* k_P

2) ... with **FPT-linear** (combined) evaluation...

- Given an ICG-Datalog program P with body-size k_P and a relational instance I of treewidth k_I, checking if $I \models P$ can be done in time $f(k_P, k_I) \times |P| \times |I|$

Publication in ICDT'2017

1) A new language...

- We introduce the language of ***intentional-clique-guarded Datalog*** (ICG-Datalog), parameterized by *body-size* k_P

2) ... with **FPT-linear** (combined) evaluation...

- Given an ICG-Datalog program P with body-size k_P and a relational instance I of treewidth k_I, checking if $I \models P$ can be done in time $f(k_P, k_I) \times |P| \times |I|$

3) ... and also **FPT-linear** (combined) computation of provenance

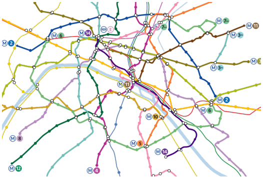
- We design a new concise provenance representation based on cyclic Boolean circuits: ***cycluits***

ICG-Datalog program P
of body-size ≤ k_P

1
C(x) ← Subway("Corvisart",x)
C(x) ← C(y) ∧ Subway(y,x)

2
Goal() ← ¬ C("Châtelet")

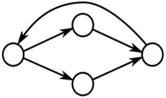
Database I
of treewidth ≤ k_I



(Paris Metro map)

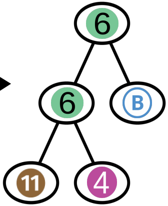
O(g(k_P, k_I) |**P**|)

Two-way Alternating Tree Automaton A



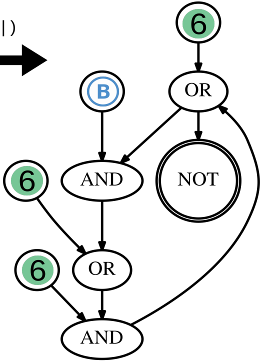
O(g'(k_I) |**I**|)

Tree encoding E



O(|**A**| · |**E**|)

Provenance Cycluit



"***Under which conditions** is it impossible to go from station Corvisart to station Châtelet with the subway?*"

Application to PQE

Theorem

Having fixed k_P and k_I, we can solve PQE in $O(2^{2^{|P|^\alpha}} |I| \, |P|)$.

- 2EXP, but still better than previous nonelementary bounds

Conclusion

Up to now:

- Study of the combined complexity of PQE
- Tractable cases quite restricted
- If we lower our expectations then we can capture more expressive query languages

Ongoing and future work:

- Lots of open technical questions
- Started a collaboration with Dan Olteanu (Univ. of Oxford) on mixed probabilistic models
- Practical applications?

Thanks for your attention!