Probabilities and Provenance on Trees and Treelike Instances

Antoine Amarilli\textsuperscript{1}, Pierre Bourhis\textsuperscript{2}, Pierre Senellart\textsuperscript{1,3,4}
September 7th, 2016

\textsuperscript{1}Télécom ParisTech

\textsuperscript{2}CNRS CRIS\textsuperscript{T}AL

\textsuperscript{3}National University of Singapore

\textsuperscript{4}École normale supérieure de Paris
How to travel to Highlights from Paris?
How to travel to Highlights from Paris?
How to travel to Highlights from Paris?
How to travel to Highlights from Paris?

2.2 – Correspondances autorisées

Une fois validé, un ticket t+ permet, sans limites de distance, les correspondances suivantes :

- les correspondances entre les lignes de Métro et les lignes de RER dans Paris, par les cheminement autorisés ;

- les correspondances entre lignes de bus, et entre ces lignes et les lignes de tramway, sur une durée d’une heure trente entre la 1ère et la dernière validation, sous réserve des dispositions suivantes.
2.2 – Correspondances autorisées

Une fois validé, un ticket t+ permet, sans limites de distance, les correspondances suivantes :

- les correspondances entre les lignes de Métro et les lignes de RER dans Paris, par les cheminements autorisés ;

- les correspondances entre lignes de bus, et entre ces lignes et les lignes de tramway, sur une durée d’une heure trente entre la 1ère et la dernière validation, sous réserve des dispositions suivantes.
How to travel to Highlights from Paris?

(Metro|RER)*|(Bus|Tram)*
How to travel to Highlights from Paris?

(Metro|RER)*|(Bus|Tram)*
How to travel to Highlights from Paris?

(Metro|RER)*|(Bus|Tram)*
How to travel to Highlights from Paris?

(Metro|RER)*(Bus|Tram)*
How to travel to Highlights from Paris?

(Metro|RER)*|(Bus|Tram)*
How to travel to Highlights from Paris?

(Metro|RER)*|(Bus|Tram)*
How to travel to Highlights from Paris?

What is the probability that I can attend Highlights 2016?

(Metro|RER)*|(Bus|Tram)*

72%
42%
37%
90%
83%
78%
72%

(Metro|RER)*|(Bus|Tram)*

78%
50%
37%
90%
83%
78%
72%
Input:

Query $Q$ 

$(\text{Metro} | \text{RER})^* (\text{Bus} | \text{Tram})^*$
Input: Query $Q$ and Database $D$ or graph $(\text{Metro}|\text{RER})*|(\text{Bus}|\text{Tram})*$
Problem statement

Input:

- Query $Q$
- Database $D$ or graph
- Probabilities on facts or edges

Output:
The probability that the query is true under the distribution (assuming independence of all probabilistic events)

Complexity: already $\#\text{P}$-hard in the input database! (from $\#\text{MONOTONE-SAT}$)
Problem statement

**Input:**

- Query $Q$
- Database $D$ or graph
- Probabilities on facts or edges

**Output:** the probability that the query is true under the distribution (assuming independence of all probabilistic events)
Problem statement

Input:

- Query $Q$
- Database $D$ or graph
- Probabilities on facts or edges

Output: the probability that the query is true under the distribution (assuming independence of all probabilistic events)

Complexity: already #P-hard in the input database! (from #MONOTONE-SAT)
Using treewidth to make the problem tractable

- Trees have treewidth 1.
- Cycles have treewidth 2.
- $k$-cliques and $(k-1)$-grids have treewidth $k-1$.

→ Treelike: the treewidth is bounded by a constant.
Using treewidth to make the problem tractable

- Trees have treewidth 1
- Cycles have treewidth 2
- $k$-cliques and $(k-1)$-grids have treewidth $k-1$

→ Treelike: the treewidth is bounded by a constant
Using treewidth to make the problem tractable

- Trees have treewidth 1.
- Cycles have treewidth 2.
- $k$-cliques and $(k-1)$-grids have treewidth $k-1$.

Treelike: the treewidth is bounded by a constant.
Using treewidth to make the problem tractable

Treewidth by example:

Trees have treewidth 1.

Cycles have treewidth 2.

$k$-cliques and $(k-1)$-grids have treewidth $k-1$. 

→ Treelike: the treewidth is bounded by a constant.
Using treewidth to make the problem tractable

Treewidth by example:

Trees have treewidth one.

Cycles have treewidth two.

k-cliques and (k−one) grids have treewidth k−one.

→ Treelike: the treewidth is bounded by a constant
Using treewidth to make the problem tractable

Treewidth by example:
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth \(1\).
- Cycles have treewidth \(2\).
- \(k\)-cliques and \((k-1)\)-grids have treewidth \(k-1\).
Using treewidth to make the problem tractable

Treewidth by example:
Using treewidth to make the problem tractable

Treewidth by example:
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth 1.
- Cycles have treewidth 2.
- $k$-cliques and $(k-1)$-grids have treewidth $k$.

→ Treelike: the treewidth is bounded by a constant.
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth 1.
- Cycles have treewidth 2.
- $k$-cliques and $(k-1)$-grids have treewidth $k-1$. 

→ Treelike: the treewidth is bounded by a constant
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth 1.
- Cycles have treewidth 2.
- $k$-cliques and $(k-1)$-grids have treewidth $k-1$. 

→ Treelike: the treewidth is bounded by a constant.
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth $\text{treewidth} = 1$
- Cycles have treewidth $\text{treewidth} = 2$
- $k$-cliques and $(k-1)$-grids have treewidth $k - 1$
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth 1.
- Cycles have treewidth 2.
- $k$-cliques and $(k-1)$-grids have treewidth $k-1$. 

*Treelike*: the treewidth is bounded by a constant.
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth 1.
- Cycles have treewidth 2.
- $k$-cliques and $(k-1)$-grids have treewidth $k-1$. 

Treelike: the treewidth is bounded by a constant.
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth $1$
- Cycles have treewidth $2$
- $k$-cliques and $(k-1)$-grids have treewidth $k-1$

Treelike: the treewidth is bounded by a constant
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth $O(1)$.
- Cycles have treewidth $O(n)$.
- $k$-cliques and $(k-1)$-grids have treewidth $k - O(1)$. 

→ Treelike: the treewidth is bounded by a constant.
Using treewidth to make the problem tractable

Treewidth by example:

- **Trees** have treewidth 1
- **Cycles** have treewidth 2
- **$k$-cliques** and **$(k - 1)$-grids** have treewidth $k - 1$
Using treewidth to make the problem tractable

Treewidth by example:

- Trees have treewidth 1
- Cycles have treewidth 2
- $k$-cliques and $(k - 1)$-grids have treewidth $k - 1$

→ Treelike: the treewidth is bounded by a constant
Theorem

For any fixed Boolean MSO query \( q \) and \( k \in \mathbb{N} \), given a database \( D \) of treewidth \( \leq k \) with independent probabilities, we can compute in linear time the probability that \( D \) satisfies \( q \).
Tractability on treelike instances

Treelike data

MSO query

Tree automaton

For any fixed Boolean MSO query $q$ and $k \in \mathbb{N}$, given a database $D$ of treewidth $\leq k$ with independent probabilities, we can compute in linear time the probability that $D$ satisfies $q$. 
Theorem

For any fixed Boolean MSO query $q$ and $k \in \mathbb{N}$, given a database $D$ of treewidth $\leq k$ with independent probabilities, we can compute in linear time the probability that $D$ satisfies $q$. 

\begin{align*}
\text{Treelike data} & \rightarrow \text{Tree encoding} \\
\text{MSO query} & \rightarrow \text{Tree automaton}
\end{align*}
Theorem

For any fixed Boolean MSO query $q$ and $k \in \mathbb{N}$, given a database $D$ of treewidth $\leq k$ with independent probabilities, we can compute in linear time the probability that $D$ satisfies $q$. 

$$\frac{\text{five.osf}}{\text{seven.osf}}$$
Theorem

For any fixed Boolean MSO query $q$ and $k \in \mathbb{N}$, given a database $D$ of treewidth $\leq k$ with independent probabilities, we can compute in linear time the probability that $D$ satisfies $q$.
Theorem

For any fixed Boolean MSO query \( q \) and \( k \in \mathbb{N} \), given a database \( D \) of treewidth \( \leq k \) with independent probabilities, we can compute in linear time the probability that \( D \) satisfies \( q \).
Theorem
For any fixed Boolean MSO query $q$ and $k \in \mathbb{N}$, given a database $D$ of treewidth $\leq k$ with independent probabilities, we can compute in linear time the probability that $D$ satisfies $q$. 

Treelike data $\xrightarrow{\text{Tree encoding}}$ Tree automaton $\xrightarrow{\text{Provenance circuit}}$ linear $\xrightarrow{95\% \text{ Probability}}$
Tractability on treelike instances

Theorem
For any fixed Boolean MSO query $q$ and $k \in \mathbb{N}$, given a database $D$ of treewidth $\leq k$ with independent probabilities, we can compute in linear time the probability that $D$ satisfies $q$. 

Treelike data

Tree encoding

MSO query

(TreeRER|metro)*

|(bus|tram)*

Tree automaton

Provenance circuit

95% Probability

linear
What can we do for unbounded-treewidth instances?

Theorem

For any graph signature $\sigma$, there is a first-order query $q$ such that for any constructible unbounded-treewidth class $I$, probability evaluation of $q$ on $I$ is $\osfP$-hard under $\RP$ reductions.

Proof idea:

extract instances of a hard problem as topological minors using recent polynomial bounds [Chekuri and Chuzhoy, two.osf/zero.osf/one.osf/four.osf/six.osf/seven.osf]
What can we do for unbounded-treewidth instances? ... not much.
For any graph signature $\sigma$, there is a first-order query $q$ such that for any constructible unbounded-treewidth class $\mathcal{I}$, probability evaluation of $q$ on $\mathcal{I}$ is \#P-hard under RP reductions.
Theorem
For any graph signature $\sigma$, there is a first-order query $q$ such that for any constructible unbounded-treewidth class $\mathcal{I}$, probability evaluation of $q$ on $\mathcal{I}$ is $#P$-hard under RP reductions.

Proof idea: extract instances of a hard problem as topological minors using recent polynomial bounds [Chekuri and Chuzhoy, 2014]
Lower bound

Theorem
For any graph signature $\sigma$, there is a first-order query $q$ such that for any constructible unbounded-treewidth class $I$, probability evaluation of $q$ on $I$ is $\#P$-hard under RP reductions.

Proof idea: extract instances of a hard problem as topological minors using recent polynomial bounds [Chekuri and Chuzhoy, 2014]
Future and ongoing work

- Improving the **lower bound:**
  - From **graphs** to **arbitrary arity** databases
  - From **FO** down to **unions of conjunctive queries with ≠**

---

Thanks for your attention!
Future and ongoing work

- Improving the **lower bound**:
  - From graphs to arbitrary arity databases
  - From FO down to unions of conjunctive queries with \( \neq \)

- Complexity in **query** and **database** — currently \( \Omega \left( 2^{2^{|Q|}} \times |D| \right) \)
  \( \rightarrow \) Which queries can **efficiently** be compiled to automata?

Thanks for your attention!
Future and ongoing work

- Improving the lower bound:
  - From graphs to arbitrary arity databases
  - From FO down to unions of conjunctive queries with $\neq$

- Complexity in query and database — currently $\Omega \left( 2^{2^{|Q|}} \times |D| \right)$
  \[ \rightarrow \text{ Which queries can efficiently be compiled to automata?} \]

- Non-Boolean queries: efficient enumeration of query results?
Future and ongoing work

• Improving the lower bound:
  • From graphs to arbitrary arity databases
  • From FO down to unions of conjunctive queries with $\neq$

• Complexity in query and database — currently $\Omega \left( 2^{\frac{Q}{2^{|Q|}}} \times |D| \right)$
  → Which queries can efficiently be compiled to automata?

• Non-Boolean queries: efficient enumeration of query results?

• Other tasks: probabilistic conditioning
  “Knowing that I’m here, what’s the probability that RER B is up?”
Future and ongoing work

• Improving the lower bound:
  • From graphs to arbitrary arity databases
  • From FO down to unions of conjunctive queries with $\neq$

• Complexity in query and database — currently $\Omega\left(2^{2\cdot Q} \times |D|\right)$
  → Which queries can efficiently be compiled to automata?

• Non-Boolean queries: efficient enumeration of query results?

• Other tasks: probabilistic conditioning
  “Knowing that I’m here, what’s the probability that RER B is up?”

Thanks for your attention!
**Polynomial bounds for the grid-minor theorem.**
In *STOC*.

**The monadic second-order logic of graphs. I. Recognizable sets of finite graphs.**
*Inf. Comput.*, 85(1).
Image credits

- Slide 2:
  - https://commons.wikimedia.org/wiki/File:Paris_Metro_map.svg (cropped), user Umx on Wikimedia Commons, public domain

- Slides 4 and 5: https://commons.wikimedia.org/wiki/File:Carte_Transilien_RER_sch%C3%A9matique.svg (modified), user Benjamin Smith on Wikimedia Commons, license CC BY-SA 4.0 international.