Interactive Path Query Specification on Graph Databases
Angela Bonifati, Radu Ciucanu, Aurélien Lemay
University of Lille & INRIA, France

1. Motivations
- Specifying a database query is a challenging task for non-expert users i.e., unfamiliar with language formalisms.
- In the context of graph databases, the problem becomes even harder:
  - No distinction between instances and schemata.
  - No proper metadata carried by the instances.
  - Difficulty to visualize possibly large instances.
- Traditional query specification paradigms for non-expert users e.g., query by example [3] become unfeasible.
- To assist users to specify path queries on graphs, we study the problem of simple node labeling as positive or negative examples.

2. Graph databases
**Example:** A geographical database storing information about neighborhoods of a city and different facilities.

3. Path queries on graph databases
We focus on path queries that select nodes having at least one path in the language of a given regular expression.

**Example:** “Select neighborhoods from which one can reach a cinema via public transportation”:

$(\text{tram } + \text{ bus})^* \cdot \text{cinema}$

Selects four nodes on the graph because they are entailed by the following paths:

- $N_1$, $N_4$, cinema, $C_1$
- $N_2$, $N_3$, $N_4$, cinema, $C_1$
- $N_4$, cinema, $C_1$
- $N_6$, cinema, $C_2$

4. Learning from examples
Given a set of positive and negative examples, construct a consistent query i.e., that selects all positive examples and none of the negative ones.

- positive examples: $N_2$, $N_6$
- negative examples: $N_5$

There exists an infinity of consistent queries:
- the goal $(\text{tram } + \text{ bus})^* \cdot \text{cinema} - \text{bus}$
- every query bus $\cdot (\varepsilon + q)$, with $q$ any path query

What is the exact query that the user has in mind?

5. Interactive scenario
We instantiate our general paradigm [2] for the case of specifying path queries on graph databases.

6. Learning algorithm
**Problem 1** – Inconsistent sample: we may enumerate an infinite set of paths and never halt.

**Problem 2** – Very long SCPs e.g., goal query $a^* \cdot b$ and SCPs $a^* \cdot b$, $ab^* \cdot ab$, $a^* \cdot b$.
If we generalize $a$ and $ab$, we obtain $a^* \cdot b$, which covers the third positive node too, without having to explicitly construct its long SCP.

**Theorem (main result of [1]):** If we bound the SCPs to a length $k$, all path queries with canonical DFA of at most $n$ states s.t. $2 \cdot n + 1 \leq k$ are learnable with abstain in polynomial time and data.

7. Bound the length of SCPs

8. GPS
We propose three types of interactions:

1. Static labeling – annotate a graph (possibly inconsistently) and run the learning algorithm.
2. Interactive labeling (without path validation) – ask the user to label informative nodes only and let the learning algorithm choose the relevant paths for each positive node (i.e., the SCPs).
3. Interactive labeling (with path validation) – additionally ask the user to validate the path of interest for each positive node and then use the learning algorithm only for generalizing them.

It would return $(\text{tram } + \text{ bus})^* \cdot \text{cinema}$ if the user validates the paths of interest bus-tram-cinema and cinema for the two positive nodes.

References