Backgrounds

The veracity of big graphs
- inconsistencies, entity resolution
- knowledge extraction, fraud detection
- missing values, association rules

These are the semantics of big graphs!

Rule bases approach are practical: 67% tools are rule based.

Challenges for graph rules:
- capturing topological structures
- coping with schematless graphs
- Correctness and non-redundancy
- catching, detecting and fixing errors
- making predictions for missing links

The need for a dependency theory for graphs

Dependencies for graph

Dependencies $Q_1(x)(X \rightarrow Y)$
- topological + attribute constraints
- adding ML predictors to literal
- matching complexity bounds to relational counterparts
- axiom system: sound, complete and independent

Examples

```
Q1: If a team $x$ uses a stadium $z$ as its ground at the same location $z'$, $z$ is owned by a sport organization $y$, and ML predicts the association between $x$ and $y$, then $x$ is a tenant of $z$

Q2: If $z$ is a friend of both $x$ and $x'$, and all the three visit the same cafe $y$ and share comment interest, then $x$ and $x'$ are likely to be friends

Q3: If $x$ is a tenant of $z$, $x$ is a friend of $y$, and $z$ is owned by a sport organization $p$, then $x$ is a tenant of $z''$
```

Error detection and fixing

Error detection and fixing:
- Input: a graph $G$, a set of rules $\Sigma$ and ground truth $\Gamma$
- Error detection: find the set $\text{Vio}(\Sigma, G)$ of violations
- Incremental error detection: finding $\text{Vio}(\Sigma, G, \Delta G)$ when given $\text{Vio}(\Sigma, G)$ and batch updates $\Delta G$
- Fixing: find an instance conforming to $\Sigma$
- Fix: combining object identification and data repairing
- Certain fix: every fix is correct, no errors are newly introduced

Fundamental studies
- validation and incremental validation: coNP-complete
- consistency with ground truth: coNP-complete
- cleaning: NP-complete, PTIME data complexity

Algorithms
- Error detection: parallel algorithm
  - sequential: match and detect
  - work unit: neighbors around pivot nodes
  - balance workload and minimizing communication cost
- Incremental detection
  - divide $\text{Vio}$ as $\Delta^+ \text{Vio}$ and $\Delta^- \text{Vio}$
  - update $\Delta^+ \text{Vio}$ and $\Delta^- \text{Vio}$ from edge insertion and deletion respectively
- certain fix: two modes in practice
  - online: find errors pertaining to a small set of user’s interest
  - offline: deduce fixes to the entire graph

Localizable algorithms: affecting small areas surrounding $\Delta G$
Parallel scalable algorithms for all problems: $T(p) = O(\frac{\ell}{p})$

Reasoning about dependencies and their discovery

Reasoning about a set of rules $\Sigma$:
- Input: a graph $G$, a set of rules $\Sigma$, one rule $\varphi$ and a threshold $\rho$
- Satisfiability: determine if there exists a graph $G$ such that $G \models \Sigma$
- Implication: determine if for any graph $G$ and $G \models \Sigma$, then $G \models \varphi$

Discovering sensible rules from graph
- Input: support threshold $\rho$, graph $G$
- Output: find a cover of non-redundant rules $\Sigma$ that are $\rho$-frequent in $G$

Fundamental studies
- coNP-complete and NP-complete for satisfiability and implication resp.
- FPT for both problems when parameterizing the vertex number in $Q$

Algorithms
- reasoning: sequential algorithms from characterizations
- discovering: vertical and horizontal spawning from generation tree

A Uniform Framework

Uniform framework of logic-based and ML-based methods
- capturing absent links, missing values, and semantic inconsistencies
- all ML methods can be plugged in literals
- interpret ML predictions using rules

Deductions: deducing $\text{ded}(\Sigma, G)$
- deductions: extensions of $G$ from literals in $\Sigma$, adding missing attributes and edges
- algorithms by extending chase sequence
- Incremental deduction: update $\text{ded}(\Sigma, G, \Delta G)$ when given graph update $\Delta G$

Algorithms
- sequential: extending chase sequences
- parallel: GRAPE based solution
- incremental: catching invalidness and refining

A first step of unifying rule-based and ML-based methods

Delivered

10 top-tier publications:
- 2021: VLDB [1]
- past: 2020 [2, 3, 4], 2019 [5, 6, 7], and 2018 [8, 9, 10]

Put the package of solutions in action!

Publications