The Other Side of Big Graphs
Dependency Theory, Practice and Applications

Wenfei Fan

University of Edinburgh

Backgrounds

The veracity of big graphs

- Inconsistencies, entity resolution
- Knowledge extraction, fraud detection
- Missing values, association rules

These are about the *semantics* of big graphs!

Rule bases approach are practical: 67% tools are rule based.
Challenges for graph rules:

- Capturing topological structures
- Coping with schemaless graphs
- Correctness and non-redundancy
- Catching, detecting and fixing errors
- Making predictions for missing links

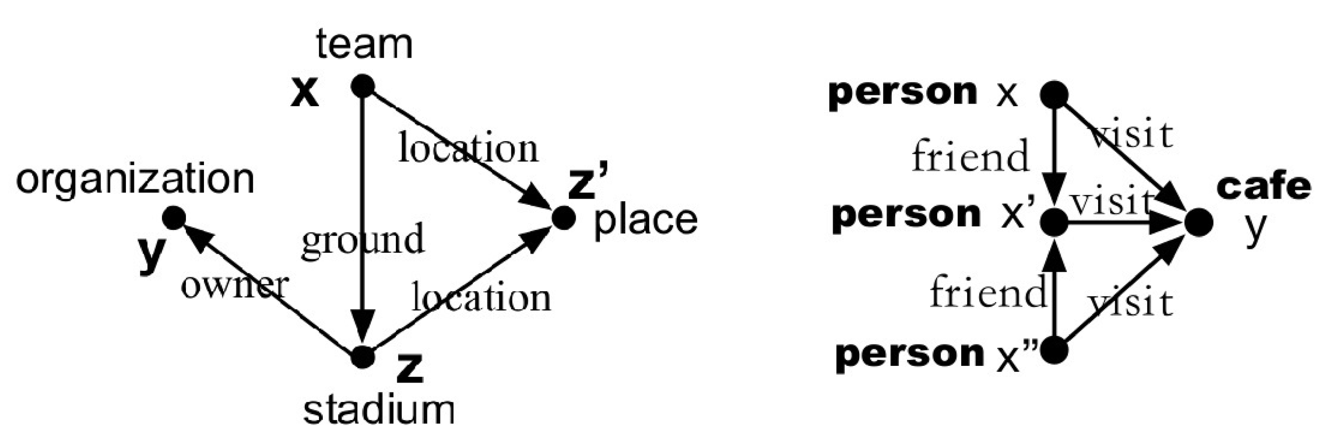
The need for a dependency theory for graphs

Dependencies for graph

Dependencies $Q[\bar{x}](\mathrm{X} \rightarrow \mathrm{Y})$

- Topological + attribute constraints
- Adding ML predictors to literal
- Matching complexity bounds to relational counterparts
- Axiom system: sound, complete and independent

Examples



Q_1 Q_2

Q_1: 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

$$Q_1(\mathcal{M}(x, y, \text{association}) \rightarrow \text{tenant}(z, x))$$

Q_2: If x' 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

$$Q_2(x.\text{interest} = x'.\text{interest} \wedge x'.\text{interest} = x''.\text{interest} \rightarrow \mathcal{M}(x, x'', \text{friend}))$$

Error detection and fixing

Error detection and fixing:

- **Input**: a graph G, a set of rules Σ and ground truth Γ
- Error detection: find the set $\text{Vio}(\Sigma, G)$ of violations
  - Incremental error detection: finding $\Delta\text{Vio}(\Sigma, G, \Delta G)$ when given $\text{Vio}(\Sigma, G)$ and batch updates ΔG
- Fixing: find an instance conforming to Σ
  - 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 ΔVio 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 practical methods
  - Online: fix errors pertaining to a small set of user's interest
  - Offline: deduce fixes to the entire graph

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

Reasoning and discovering dependencies

Reasoning and discovering with a set of rules Σ:

- **Input**: a graph G, a set of rules Σ, one rule φ and a threshold ρ
- 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 $G \models \varphi$
- Discovery: given support threshold ρ, find a cover of non-redundant rules Σ that are ρ-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

Parallel scalable algorithms for all problems: $T(p) = O(\frac{t}{p})$
The more processors used, better the performance is

A Uniform Framework: Deduction and Discovery

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

Dependency Deductions: deducing $\text{ded}(\Sigma, G)$

- Deductions: extensions of G from literals in Σ, adding missing attributes and edges
- Incremental deduction: update $\text{ded}(\Sigma, G, \Delta G)$ when given graph update ΔG

Deduction Algorithm:

- Sequential: extending chase sequences
- Parallel: GRAPE based solution
- Incremental: catching invalidness and refining

Discovering dependencies:

- Input: a graph G, an application $\mathcal{A}$, a support threshold ρ
- Discovery: find a cover of non-redundant $\mathcal{A}$-related rules Σ that are ρ-frequent in G
- Accelerate discovery by application-driven reducing and sampling G

Discovering Algorithms:

- Conduct application-driven reduction to deduce $\mathcal{A}$-relevant graph $G_{\mathcal{A}}$
- Sample graph G with probabilistic bounds on recall and support

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

Publications

11 top-tier publications:

- 2022: VLDB [1], 2021: VLDB [2]
- past: 2020 [3, 4, 5], 2019 [6, 7, 8], and 2018 [9, 10, 11]

Put the package of solutions in action!

Publications

[1] Wenfei Fan, Wenzhi Fu, Ruochun Jin, Ping Lu, and Chao Tian. Discovering association rules from big graphs. *PVLDB*, pages 1479–1492, 2022.

[2] Wenfei Fan, Chao Tian, Yanghao Wang, and Qiang Yin. Parallel discrepancy detection and incremental detection. *PVLDB*, 14(8):1351–1364, 2021.

[3] Wenfei Fan, Chunming Hu, Xueli Liu, and Ping Lu. Discovering graph functional dependencies. *TODS*, 2020.

[4] Wenfei Fan, Ping Lu, Chao Tian, and Jingren Zhou. Catching numeric inconsistencies in graphs. *TODS*, 2020.

[5] Wenfei Fan, Ruochun Jin, Muyang Liu, Ping Lu, Chao Tian, and Jingren Zhou. Capturing associations in graphs. *Proceedings of the VLDB Endowment*, 13(12):1863–1876, 2020.

[6] Wenfei Fan and Ping Lu. Dependencies for graphs. *TODS*, 44(2):5:1–5:40.

[7] Wenfei Fan. Dependencies for graphs: Challenges and opportunities. *JDIQ*, 11(2):1–12, 2019.

[8] Wenfei Fan, Ping Lu, Chao Tian, and Jingren Zhou. Deducing certain fixes to graphs. *PVLDB*, 12(7):752–765, 2019.

[9] Wenfei Fan, Xueli Liu, and Yingjie Cao. Parallel reasoning of graph functional dependencies. In *ICDE*, pages 593–604. IEEE, 2018.

[10] Wenfei Fan, Xueli Liu, Ping Lu, and Chao Tian. Catching numeric inconsistencies in graphs. In *SIGMOD*, pages 381–393, 2018.

[11] Wenfei Fan, Chunming Hu, Xueli Liu, and Ping Lu. Discovering graph functional dependencies. In *SIGMOD*, pages 427–439, 2018.