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

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

These are the semantics of big graphs!

Dependencies for graph

Dependences \( Q(x) \rightarrow (X \rightarrow Y) \)
- Topological + attribute constraints
- Adding ML predictors to literal
- Matching complexity bounds to relational counterparts
- Axiom system: sound, complete and independent

Examples

Error detection and fixing:

- Input: a graph \( G \), a set of rules \( \Sigma \) and ground truth \( \Gamma \)
- Error detection: find the set \( Vio(\Sigma, G) \) of violations
- Incremental error detection: finding \( \Delta Vio(\Sigma, G, \Delta G) \) when given \( Vio(\Sigma, G) \) and updates \( \Delta G \)
- Fixing: find an instance conforming to \( \Sigma \)
- 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, \( \text{PTIME} \) 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 \( \Delta Vio \) as \( \Delta Vio \) and \( \Delta^+ Vio \)
- Update \( \Delta^+ Vio \) and \( \Delta^- Vio \) 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 \( \Delta G \)
Parallel scalable algorithms for all problems: \( T(p) = O(\frac{\log p}{p}) \)

Reasoning and discovering dependencies

Reasoning and discovering with a set of rules \( \Sigma \):
- Input: a graph \( G \), a set of rules \( \Sigma \), one rule \( \phi \) and a threshold \( p \)
- 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 \( G' \models \phi \)
- Discovery: given support threshold \( p \), find a cover of non-redundant rules \( \Sigma \) that are \( p \)-frequent in \( G \)

Fundamental studies
- coNP-complete and NP-complete for satisfiability and implication resp.
- \( \text{FPT} \) 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: 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 \( \Sigma \), adding missing attributes and edges
- Incremental deduction: update \( \text{ded}(\Sigma, G, \Delta G) \) when given graph \( G \) and \( \Delta G \)

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

Discovering dependencies:
- Input: a graph \( G \), an application \( A \), a support threshold \( \rho \)
- Discovery: find a cover of non-redundant \( A \)-related rules \( \Sigma \) that are \( \rho \)-frequent in \( G \)
- Acceleration discovery by application-driven reducing and sampling \( G \)

Discovering Algorithms:
- Conduct application-driven reduction to deduce \( A \)-relevant graph \( G_A \)
- 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:
- past: 2020 [3, 4, 5], 2019 [6, 7, 8], and 2018 [9, 10, 11]

Put the package of solutions in action!

Publications