Provenance for Data-Intensive Systems
**ProDIS: Provenance for Data-Intensive Systems**

- Databases, Data Mining, Data Science...
- Highly complex logic, Big Data

**Provenance of output is typically unknown**

- Why, what if, what data was used, can we trust?...
- Without answers to these questions, results may be useless/harmful
  - Medical recommendations, loan request rejections..

Imagine a world where computation results are **accounted for and explained**

Systems would be transparent and controllable, and the results credible and reusable.
Provenance for Real-life Data-Intensive Systems

Data Provenance: theory and algorithms
Provenance for Real-life Data-Intensive Systems

Data Provenance: theory and algorithms
Models

- Interfaces for non-experts
- Data Science Frameworks
- NLIDB
- Exploration
- ML
- Workflows
- Distributed
- SQL
- recursion
- updates
- aggregation
- Basic SPJU queries

Scale

- Big Data (Distributed)
- Organizational Data (Centralized)
- Small Data
- Internal Representation

Applications

- Everyone
- Analysts

Exploration NLIDB (Big Data) (Distributed) Organizational Data (Centralized) Everyone Analysts Interfaces for non-experts Data Science Frameworks NLIDB Exploration ML Workflows Distributed SQL recursion updates aggregation Basic SPJU queries
### Models

<table>
<thead>
<tr>
<th>eventid</th>
<th>sum</th>
<th>type</th>
<th>due</th>
<th>Prov</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50000</td>
<td>overdraft</td>
<td>2012</td>
<td>p1</td>
</tr>
<tr>
<td>2</td>
<td>400000</td>
<td>mortgage</td>
<td>2014</td>
<td>p2</td>
</tr>
<tr>
<td>3</td>
<td>2000000</td>
<td>overdraft</td>
<td>2010</td>
<td>p3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>custname</th>
<th>eventid</th>
<th>prov</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>1</td>
<td>c1</td>
</tr>
<tr>
<td>Smith</td>
<td>3</td>
<td>c2</td>
</tr>
<tr>
<td>Roth</td>
<td>2</td>
<td>c3</td>
</tr>
</tbody>
</table>

"Return customers with overdraft events after 2006"

"Essence of Computation"

[Green et. al, PODS 2007]
Models

Approach I: Algebraic Provenance

- Workflows
  - Pairing semirings
    - [Deutch, Moskovitch, Tannen, VLDB ’14]

- recursion
  - Absorptive Semirings
    - [Deutch et. al, ICDT’14]

- updates
  - Circuits
    - [Bouhris, Deutch, Moskovitch, ICDE ’16]

What are the right models?
Models

Interfaces for non-experts

- NLIDB
- Exploration

Data Science Frameworks

- ML
- Workflows
- Distributed

SQL

- recursion
- updates
- aggregation
- Basic SPJU queries

Scale

Big Data (Distributed)

- Organizational Data (Centralized)

Applications

Everyone

Analysts

Internal Representation

Small Data
“Return owners of accounts with overdraft events exceeding a sum of €2000 after the year 2006”

“Why is ‘Smith’ an answer?”

[Deutch, Frost, Gilad, VLDB ’17 best paper]
Models
Approach II: Interaction-Based Provenance

Data Science Frameworks

ML

Workflows

Figure 2: Optimal changes

<table>
<thead>
<tr>
<th>Integrity Constraint</th>
<th>Formula Example</th>
<th>Project Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>$\forall t \in R, t.v_i \in [a, b]$</td>
<td>$v_i \geq a \land v_i \leq b$</td>
</tr>
<tr>
<td>Check</td>
<td>$\forall t \in R, t.v_i \leq t.v_j$</td>
<td>$v_i - v_j \leq 0$</td>
</tr>
<tr>
<td>CFD</td>
<td>$\forall t \in R, t.v_i = V_1 \rightarrow t.v_j = V_2$</td>
<td>$v_i \neq V_1 \lor v_j = V_2$</td>
</tr>
<tr>
<td>Unary Denial</td>
<td>$\forall t \in R, \neg (\bigwedge_{i=1}^{k} t.v_i = V_i)$</td>
<td>$\bigvee_{i=1}^{k} v_i \neq V_i$</td>
</tr>
</tbody>
</table>

D. Deutch and N. Frost, Constraints-based Explanations of Classifications, ICDE 2019
Models
- Data Science Frameworks
  - Data Mining
  - Workflows
  - Distributed

- SQL
  - recursion
  - nesting
  - order-by
  - Basic SPJU queries

Applications
- Everyone
  - Analysts
  - Internal Representation

Scale
- Big Data (Distributed)
- Organizational Data (Centralized)
- Small Data

Interfaces for non-experts
- NLIDB
- Exploration

Frameworks for exploration and big data mining
SELECT Customer.cname
FROM Customer, Ownership, Product, Assoc, Event, DebtEvent, Currency
WHERE Customer.cid = Ownership.cid
AND Ownership.pid = Product.pid
AND Product.type LIKE '%account'
AND Product.pid = Assoc.pid
AND Assoc.eid = Event.eid
AND Event.date > '01.01.2007'
AND Event.eid = DebtEvent.eid
AND DebtEvent.sum > 2000

“Return owners of accounts with overdraft events exceeding a sum of €2000 after the year 2006”
<table>
<thead>
<tr>
<th>cname</th>
<th>prov</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>C0123·01325·P85335·A8212·E23874·DE23874·CU2+</td>
</tr>
<tr>
<td></td>
<td>C0123·01325·P85335·A4326·E9873·DE9873·CU2+</td>
</tr>
<tr>
<td>…</td>
<td></td>
</tr>
<tr>
<td>Jones</td>
<td>C8432·012387·P1248·E2384·DE2384·CU2+</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

“Return owners of accounts with overdraft events exceeding a sum of €2000 after the year 2006”

PTIME but practically inefficient

Organizational Data (Centralized)

Basic SPJU queries

Internal Representation
Super-polynomial lower bound for datalog [Deutch et. al, ICDT ’14]

Organizational Data (Centralized)

SQL
- recursion
- updates
- aggregation
- negation
- order-by
- Basic SPJU queries

Internal Representation
Scalable Provenance Solutions

Approach I: Selective Provenance Tracking

[Deutch, Gilad, Moskovitch, VLDB ’15, VLDB Journal ‘18]

[Bouhris, Deutch, Moskovitch, ICDE ‘16]
Scalable Provenance Solutions
Approach II: Summarization

[Deutch et. al, Provenance for Datalog Circuits, ICDT ‘14]
Scalable Provenance Solutions
Approach III: Abstraction

[Deutch, Moskovitch, Rinetzky, Hypothetical Reasoning Via Provenance Abstraction, SIGMOD '19]
Scalable Provenance Solutions
Approach IV: Interaction Based

"Return owners of accounts with overdraft events exceeding a sum of €2000 after the year 2006"

..."Why is ‘Smith’ an answer?"

"Smith is the owner of 3 accounts with 13 overdraft events of a total sum of €30000 in 01.02.2009-01.05.2010"

[Deutch, Frost, Gilad, Provenance for NL Queries, VLDB ’17 best paper]
Expressiveness

- Interfaces for non-experts
  - NLIDB
  - Exploration

- Data Science Frameworks
  - Data Mining
  - Workflows
  - Distributed

- SQL
  - recursion
  - updates
  - aggregation
  - Basic SPJU queries

Scale

- Big Data (Distributed)
- Organizational Data (Centralized)

Applications

- Everyone
- Analysts
- Internal Representation
Provenance Applications

“Why is ‘Smith’ an answer?”

“Smith is the owner of 3 accounts with 13 overdraft events of a total sum of €30000 in 01.02.2009-01.05.2010”

“On what sources is the ‘Smith’ answer based on?”

“Citibank combined with American Express and independently BNP Paribas combined with Visa”

“How could ‘Smith’ Become a non-answer?”

“What if a particular overdraft event of ‘Smith’ is excused?”

“Remove overdraft event of date 01.04.2009 of sum €10000”

“‘Smith’ would still be an answer”
Implementation and Evaluation

Prototyping

Experiments
  Performance Analysis
  User Studies

Benchmarks Development
Vision: a world where computation results are accounted for and explained

Essence of computation
• why • what if • trust • ...

Provenance for Real-life Data-Intensive Systems

Models

Scale

Applications

Data Provenance: theory and algorithms