Lecture 11: Provenance and Data privacy

December 8, 2010
Outline

• Database provenance
  – Slides based on Val Tannen’s Keynote talk at EDBT 2010

• Data privacy
  – Slides from my UW colloquium talk in 2005
Data Provenance

provenance, n.

*The fact of coming from some particular source or quarter* origin, derivation [Oxford English Dictionary]

• **Data provenance** [BunemanKhannaTan 01]: aims to explain how a particular result (in an experiment, simulation, query, workflow, etc.) was derived.

• Most science today is **data-intensive**. Scientists, eg., biologists, astronomers, worry about data provenance all the time.
Provenance? Lineage? Pedigree?

• Cf. Peter Buneman:
  – Pedigree is for dogs
  – Lineage is for kings
  – Provenance is for art

• For data, let’s be artistic (artsy?)
Database transformations?

- Queries
- Views
- ETL tools
- Schema mappings (as used in data exchange)
Outline

• **What’s with the semirings? Annotation propagation**
  [GK&T PODS 07, GKI&T VLDB 07]

• **Housekeeping in the zoo of provenance models**
Propagating annotations through database operations

The annotation $p \cdot r$ means joint use of data annotated by $p$ and data annotated by $r$.
Another way to propagate annotations

The annotation $p + r$ means alternative use of data
Another use of $+$

$+$ means alternative use of data
An example in positive relational algebra (SPJU)

$$Q = \sigma_{C=e} \Pi_{AC}(\Pi_{AC}R \bowtie \Pi_{BC}R \cup \Pi_{AB}R \bowtie \Pi_{BC}R)$$

For selection we multiply with two special annotations, 0 and 1
Summary so far
Summary so far

A space of annotations, $K$
Summary so far

A space of annotations, \( K \)

\( K \)-relations: every tuple annotated with some element from \( K \).
Summary so far

A space of annotations, $K$

*K-relations*: every tuple annotated with some element from $K$.

Binary operations on $K$: $\cdot$ corresponds to joint use (join), and $+$ corresponds to alternative use (union and projection).
Summary so far

A space of annotations, $K$

$K$-relations: every tuple annotated with some element from $K$.

Binary operations on $K$: $\cdot$ corresponds to joint use (join), and $+$ corresponds to alternative use (union and projection).

We assume $K$ contains special annotations 0 and 1.
Summary so far

A space of annotations, \( K \)

\( K \)-relations: every tuple annotated with some element from \( K \).

Binary operations on \( K \): \( \cdot \) corresponds to joint use (join), and \( + \) corresponds to alternative use (union and projection).

We assume \( K \) contains special annotations 0 and 1.

“Absent” tuples are annotated with 0!
Summary so far

A space of annotations, $K$

*K-relations*: every tuple annotated with some element from $K$.

Binary operations on $K$: $\cdot$ corresponds to joint use (join), and $+$ corresponds to alternative use (union and projection).

We assume $K$ contains special annotations 0 and 1.

“Absent” tuples are annotated with 0!

1 is a “neutral” annotation (no restrictions).
Summary so far

A space of annotations, $K$

*K-relations*: every tuple annotated with some element from $K$.

Binary operations on $K$: $\cdot$ corresponds to joint use (join), and $+$ corresponds to alternative use (union and projection).

We assume $K$ contains special annotations 0 and 1.

“Absent” tuples are annotated with 0!

1 is a “neutral” annotation (no restrictions).

**Algebra of annotations**? What are the laws of $(K, +, \cdot, 0, 1)$?
Annotated relational algebra

• DBMS query optimizers assume certain equivalences:
  – union is associative, commutative
  – join is associative, commutative, distributes over union
  – projections and selections commute with each other and with union and join (when applicable)
  – Etc., but no \( R \bowtie R = R \cup R = R \) (i.e., no idempotence, to allow for bag semantics)

• Equivalent queries should produce same annotations!
**Annotated relational algebra**

- DBMS query optimizers assume certain equivalences:
  - union is associative, commutative
  - join is associative, commutative, distributes over union
  - projections and selections commute with each other and with union and join (when applicable)
  - Etc., but no \( R \bowtie R = R \cup R = R \) (i.e., no idempotence, to allow for bag semantics)

- Equivalent queries should produce same annotations!

**Proposition.** Above identities hold for queries on \( K \)-relations iff \((K, +, \cdot, 0, 1)\) is a **commutative semiring**
Annotated relational algebra

• DBMS query optimizers assume certain equivalences:
  – union is associative, commutative
  – join is associative, commutative, distributes over union
  – projections and selections commute with each other and with union and join (when applicable)
  – Etc., but no $R \bowtie R = R \cup R = R$ (i.e., no idempotence, to allow for bag semantics)

• Equivalent queries should produce same annotations!

• Hence, for each commutative semiring $K$ we have a $K$-annotated relational algebra.
What is a commutative semiring?

An algebraic structure \((K, +, \cdot, 0, 1)\) where:

- \(K\) is the domain
- + is associative, commutative, with 0 identity
- \(\cdot\) is associative, with 1 identity
- \(\cdot\) distributes over +
- \(a \cdot 0 = 0 \cdot a = 0\)
- \(\cdot\) is also commutative

Unlike ring, no requirement for inverses to +
Back to the example

\[
\begin{align*}
\text{R} & \quad \text{Q} \\
\begin{array}{ccc}
A & B & C \\
\hline
a & b & c \\
d & b & e \\
f & g & e \\
\end{array} & \quad \begin{array}{ccc}
A & C \\
\hline
(\text{p} \cdot \text{p} + \text{p} \cdot \text{p}) \cdot 0 \\
\text{p} \cdot \text{r} \cdot 1 \\
\text{r} \cdot \text{p} \cdot 0 \\
(\text{r} \cdot \text{r} + \text{r} \cdot \text{s} + \text{r} \cdot \text{r}) \cdot 1 \\
(\text{s} \cdot \text{s} + \text{s} \cdot \text{r} + \text{s} \cdot \text{s}) \cdot 1
\end{array}
\end{align*}
\]
Using the laws: **polynomials**

Polynomials with coefficients in $\mathbb{N}$ and annotation tokens as indeterminates $p, r, s$ capture a very general form of **provenance**
Provenance reading of the polynomials

- three different ways to derive $d$, $e$
- two of the ways use only $r$
- but they use it twice
- the third way uses $r$ once and $s$ once
Low-hanging fruit: deletion propagation

We used this in Orchestra [VLDB07] for update propagation

<table>
<thead>
<tr>
<th>R</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>d</td>
<td>e</td>
</tr>
<tr>
<td>f</td>
<td>g</td>
</tr>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>pr</td>
</tr>
<tr>
<td>r</td>
<td>2r^2 + rs</td>
</tr>
<tr>
<td>s</td>
<td>rs + 2s^2</td>
</tr>
</tbody>
</table>
Low-hanging fruit: deletion propagation

We used this in *Orchestra* [VLDB07] for update propagation

Delete d b e from R?
Low-hanging fruit: deletion propagation

We used this in **Orchestra** [VLDB07] for update propagation

Delete **d b e** from **R**?

Set $r = 0$!
Low-hanging fruit: deletion propagation

We used this in **Orchestra** [VLDB07] for update propagation

Delete **d b e** from **R** ?

Set  \( r = 0 \)!
Low-hanging fruit: deletion propagation

We used this in **Orchestra** [VLDB07] for update propagation

Delete **d b e** from **R**?

Set \( r = 0 \)!
But are there useful commutative semirings?

<table>
<thead>
<tr>
<th>Semiring</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>((B, \land, \lor, \top, \bot))</td>
<td>Set semantics</td>
</tr>
<tr>
<td>((\mathbb{N}, +, \cdot, 0, 1))</td>
<td>Bag semantics</td>
</tr>
<tr>
<td>((P(\Omega), \cup, \cap, \emptyset, \Omega))</td>
<td>Probabilistic events [FuhrRölleke 97]</td>
</tr>
<tr>
<td>((\text{BoolExp}(X), \land, \lor, \top, \bot))</td>
<td>Conditional tables (c-tables) [ImielinskiLipski 84]</td>
</tr>
<tr>
<td>((R_+^\infty, \min, +, 1, 0))</td>
<td>Tropical semiring (cost/distrust score/confidence need)</td>
</tr>
<tr>
<td>((A, \min, \max, 0, P))  where (A = P &lt; C &lt; S &lt; T &lt; 0)</td>
<td>Access control levels [PODS8]</td>
</tr>
</tbody>
</table>
But are there useful commutative semirings?

<table>
<thead>
<tr>
<th>Semiring</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\mathbb{B}, \land, \lor, \top, \bot)$</td>
<td>Set semantics</td>
</tr>
<tr>
<td>$(\mathbb{N}, +, \cdot, 0, 1)$</td>
<td>Bag semantics</td>
</tr>
<tr>
<td>$(\mathcal{P}(\Omega), \cup, \cap, \emptyset, \Omega)$</td>
<td>Probabilistic events [FuhrRölleke 97]</td>
</tr>
<tr>
<td>$(\text{BoolExp}(X), \land, \lor, \top, \bot)$</td>
<td>Conditional tables (c-tables) [ImielinskiLipski 84]</td>
</tr>
<tr>
<td>$(\mathbb{R}_+^\infty, \min, +, 1, 0)$</td>
<td>Tropical semiring (cost/distrust score/confidence need)</td>
</tr>
<tr>
<td>$(A, \min, \max, 0, P)$</td>
<td>Access control levels [PODS8]</td>
</tr>
<tr>
<td>where $A = P &lt; C &lt; S &lt; T &lt; 0$</td>
<td></td>
</tr>
</tbody>
</table>
But are there useful commutative semirings?

<table>
<thead>
<tr>
<th>Semiring</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\mathbb{B}, \land, \lor, \top, \bot)$</td>
<td>Set semantics</td>
</tr>
<tr>
<td>$(\mathbb{N}, +, \cdot, 0, 1)$</td>
<td>Bag semantics</td>
</tr>
<tr>
<td>$(\mathcal{P}(\Omega), \cup, \cap, \emptyset, \Omega)$</td>
<td>Probabilistic events [FuhrRölleke 97]</td>
</tr>
<tr>
<td>$(\text{BoolExp}(X), \land, \lor, \top, \bot)$</td>
<td>Conditional tables (c-tables) [ImielinskiLipski 84]</td>
</tr>
<tr>
<td>$(\mathbb{R}_+^\infty, \min, +, 1, 0)$</td>
<td>Tropical semiring (cost/distrust score/confidence need)</td>
</tr>
<tr>
<td>$(A, \min, \max, 0, P)$ where $A = P &lt; C &lt; S &lt; T &lt; 0$</td>
<td>Access control levels [PODS8]</td>
</tr>
</tbody>
</table>
Outline

• What’s with the semirings? Annotation propagation

• Housekeeping in the zoo of provenance models
  [GK&T PODS 07, FG&T PODS 08, Green ICDT 09]
Semirings for various models of provenance (1)

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>b</td>
<td>e</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>g</td>
<td>e</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>r</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ (\text{Lin}(X), \cup, \cup^*, \emptyset, \emptyset^*) \]

**Lineage** [CuiWidomWiener 00 etc.]

Sets of contributing tuples

**Semiring:** (Lin(X), \cup, \cup^*, \emptyset, \emptyset^*)
Semirings for various models of provenance (2)

(Witness, Proof) **why-provenance**
[BunemanKhannaTan 01] & [Buneman+ PODS08]

Sets of witnesses (w. = set of contributing tuples)

**Semiring:** \( (\text{Why}(X), \cup, \cup, \emptyset, \{\emptyset\}) \)
Semirings for various models of provenance (3)

Minimal witness **why-provenance**
[BunemanKhannaTan 01]

Sets of minimal witnesses

**Semiring:** \((\text{PosBool}(X), \land, \lor, \top, \bot)\)
Semirings for various models of provenance (4)

**Notation:**
- `{ }` set
- `[]` bag

**Trio lineage** [Das Sarma+ 08]

Bags of sets of contributing tuples (of witnesses)

**Semiring:** \((\text{Trio}(X), +, \cdot, 0, 1)\) (defined in [Green, ICDT 09])
Semirings for various models of provenance (5)

Polynomials with boolean coefficients  [Green, ICDT 09]  
( B[X]-provenance )

Sets of bags of contributing tuples

Semiring:  (B[X], +, ⋅, 0, 1)
Semirings for various models of provenance (6)

Provenance polynomials  [GKT, PODS 07]
( N[X]-provenance )
Bags of bags of contributing tuples

Semiring:  (N[X], +, ·, 0, 1)
A provenance hierarchy

most informative

least informative

Why\(^{(X)}\)

N\(^{[X]}\)

B\(^{[X]}\)  Trio\(^{(X)}\)

Lin\(^{(X)}\)  PosBool\(^{(X)}\)
One semiring to rule them all... (apologies!)

Example: $2x^2y + xy + 5y^2 + z$

A path downward from $K_1$ to $K_2$ indicates that there exists an onto (surjective) semiring homomorphism $h : K_1 \rightarrow K_2$
Using homomorphisms to relate models

Example: $2x^2y + xy + 5y^2 + z$

$N[X]$

- drop coefficients
  - $x^2y + xy + y^2 + z$

$B[X]$

- drop exponents
  - $3xy + 5y + z$

$Trio(X)$

- drop both exponents and coefficients
  - $xy + y + z$

$Why(X)$

- collapse terms
  - $xyz$

$Lin(X)$

- apply absorption
  - $(ab + b = b)$

$PosBool(X)$

- $y + z$

Homomorphism?

$h(x+y) = h(x) + h(y)$
$h(xy) = h(x)h(y)$
$h(0) = 0$
$h(1) = 1$

Moreover, for these homomorphisms $h(x) = x$
Containment and Equivalence [Green ICDT 09]

Arrow from $K_1$ to $K_2$ indicates $K_1$ containment (equivalence) implies $K_2$ cont. (equiv.)

All implications not marked $\leftrightarrow$ are strict
Data Security

• Based on my colloquium talk from 2005
Data Security

Dorothy Denning, 1982:

• Data Security is the science and study of methods of protecting data (…) from unauthorized disclosure and modification

• Data Security = Confidentiality + Integrity
Data Security

• Distinct from *systems* and *network* security
  – Assumes these are already secure

• Tools:
  – Cryptography, information theory, statistics, ...

• Applications:
  – An *enabling* technology
Outline

• An attack

• Data security research today
Latanya Sweeney’s Finding

• In Massachusetts, the Group Insurance Commission (GIC) is responsible for purchasing health insurance for state employees
• GIC has to publish the data:

\[ \text{GIC}(\text{zip, dob, sex, diagnosis, procedure, ...}) \]

This is private! Right?
Latanya Sweeney’s Finding

• Sweeney paid $20 and bought the voter registration list for Cambridge Massachusetts:

\[
\text{VOTER}(\text{name, party, ...}, \text{zip, dob, sex})
\]

\[
\text{GIC}(\text{zip, dob, sex, diagnosis, procedure, ...})
\]

This is private! Right?
Latanya Sweeney’s Finding

zip, dob, sex

• William Weld (former governor) lives in Cambridge, hence is in VOTER
• 6 people in VOTER share his dob
• only 3 of them were man (same sex)
• Weld was the only one in that zip
• Sweeney learned Weld’s medical records!
Latanya Sweeney’s Finding

• All systems worked as specified, yet an important data has leaked

• How do we protect against that?

Some of today’s research in data security address breaches that happen even if all systems work correctly
Today’s Approaches

• K-anonymity
  – Useful, but not really private

• Differential privacy
  – Private, but not really useful
k-Anonymity

Definition: each tuple is equal to at least k-1 others

Anonymizing: through suppression and generalization

<table>
<thead>
<tr>
<th>First</th>
<th>Last</th>
<th>Age</th>
<th>Race</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harry</td>
<td>Stone</td>
<td>34</td>
<td>Afr-Am</td>
</tr>
<tr>
<td>John</td>
<td>Reyser</td>
<td>36</td>
<td>Cauc</td>
</tr>
<tr>
<td>Beatrice</td>
<td>Stone</td>
<td>47</td>
<td>Afr-am</td>
</tr>
<tr>
<td>John</td>
<td>Ramos</td>
<td>22</td>
<td>Hisp</td>
</tr>
</tbody>
</table>

Hard: NP-complete for supression only
Approximations exists
**k-Anonymity**

**Definition:** each tuple is equal to at least k-1 others

Anonymizing: through suppression and generalization

<table>
<thead>
<tr>
<th>First</th>
<th>Last</th>
<th>Age</th>
<th>Race</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Stone</td>
<td>30-50</td>
<td>Afr-Am</td>
</tr>
<tr>
<td>John</td>
<td>R*</td>
<td>20-40</td>
<td>*</td>
</tr>
<tr>
<td>*</td>
<td>Stone</td>
<td>30-50</td>
<td>Afr-am</td>
</tr>
<tr>
<td>John</td>
<td>R*</td>
<td>20-40</td>
<td>*</td>
</tr>
</tbody>
</table>

Hard: NP-complete for supression only
Approximations exists

[Samarati&Sweeney’98, Meyerson&Williams’04]
Differential Privacy

- A randomized algorithm $A$ is differentially private if by removing/inserting one tuple in the database, the output of $A$ is “almost the same”, i.e. every possible outcome for $A$ has almost the same probability.

[Dwork’05]
Differential Privacy

- How can we achieve that? Add some random noise to the result of A
- For example:
  - Query: select count(*) from R where blah
  - Add some random noise (Laplacian distribution: $e^{-x/x_0}$)

- Problem: can only ask a limited number of queries
  - Must keep track of the queries answered, then deny
  - Cannot release “the entire data”
Privacy

• All these techniques address confidentiality, but they are often claim privacy

• Privacy is more complex:
  – “Is the right of individuals to determine for themselves when, how and to what extent information about them is communicated to others”
Take Home Lessons

• Data management does not stop at normal forms and query optimization
• Our field (Computer Science) is becoming data-centric. Dominated by massive amounts of data.
• This affects businesses, science, society
• Watch the data management & data mining fields for excitement future innovations