An Overview of Probabilistic Databases

Gerome Miklau

645 Spring 2008

Based on tutorial by Dan Suciu
Deterministic v. Probabilistic Databases

- Conventional databases are deterministic:
  - An item is either in the database, or it is not.
  - A tuple is either in the query answer, or it is not.

- Probabilistic databases:
  - “An item belongs to the database” is a probabilistic event
  - “A tuple is an answer to the query” is a probabilistic event
Two Types of Probabilistic Data

- Database is deterministic, Query answers are probabilistic
- Database is probabilistic, Query answers are probabilistic
Long History

• Probabilistic relational databases have been studied from the late 80’s until today:

  • Cavallo & Pitarelli: 1987
  • Barbara, Garcia-Molina, Porter: 1992
  • Lakshmanan, Leone, Ross & Subrahmanian: 1997
  • Fuhr & Roellke: 1997
  • Dalvi & Suciu: 2004
  • Widom: 2005, 2006
Outline

1. **Motivating Applications**

2. Semantics of Probabilistic Data

3. Representation Systems

4. Complexity
Motivating Applications

• Text extraction & record linkage

• Quality in data integration

• Inconsistent data

• Ranking query answers
**Text extraction**

**address string**  “52-A Goregaon West Mumbai 400 076”

<table>
<thead>
<tr>
<th>House</th>
<th>Area</th>
<th>City</th>
<th>Pincode</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>Goregaon West</td>
<td>Mumbai</td>
<td>400 062</td>
<td>0.1</td>
</tr>
<tr>
<td>52-A</td>
<td>Goregaon</td>
<td>West Mumbai</td>
<td>400 062</td>
<td>0.2</td>
</tr>
<tr>
<td>52-A</td>
<td>Goregaon West</td>
<td>Mumbai</td>
<td>400 062</td>
<td>0.5</td>
</tr>
<tr>
<td>52</td>
<td>Goregaon</td>
<td>West Mumbai</td>
<td>400 062</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Four segmentations of the address string along with probabilities
Record Linkage

- Determine if two data records describe same object
  
- Scenarios:
  
  - Join/merge two relations
  
  - Remove duplicates from a single relation
  
  - Validate incoming tuples against a reference

<table>
<thead>
<tr>
<th>Authors</th>
<th>Conference</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>B. Croft, J. Allan</td>
<td>CIKM</td>
<td>2003</td>
</tr>
<tr>
<td>Bruce Croft and James Allan</td>
<td>Conf. on Information and Knowledge Management</td>
<td>2003</td>
</tr>
</tbody>
</table>
Inconsistent Data

• Goal: consistent query answers from inconsistent databases

• Applications:
  • Integration of autonomous data sources
  • Un-enforced integrity constraints
  • Temporary inconsistencies

[Bertosi&Chomicki:2003]
The Repair Semantics

Consider all “repairs”

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>State</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miklau</td>
<td>UW</td>
<td>WA</td>
<td>Data security</td>
</tr>
<tr>
<td>Dalvi</td>
<td>UW</td>
<td>WA</td>
<td>Prob. Data</td>
</tr>
<tr>
<td>Balazinska</td>
<td>UW</td>
<td>WA</td>
<td>Data streams</td>
</tr>
<tr>
<td>Balazinska</td>
<td>MIT</td>
<td>MA</td>
<td>Data streams</td>
</tr>
<tr>
<td>Miklau</td>
<td>Umass</td>
<td>MA</td>
<td>Data security</td>
</tr>
</tbody>
</table>
The Repair Semantics

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>State</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miklau</td>
<td>UW</td>
<td>WA</td>
<td>Data security</td>
</tr>
<tr>
<td>Dalvi</td>
<td>UW</td>
<td>WA</td>
<td>Prob. Data</td>
</tr>
<tr>
<td>Balazinska</td>
<td>UW</td>
<td>WA</td>
<td>Data streams</td>
</tr>
<tr>
<td>Balazinska</td>
<td>MIT</td>
<td>MA</td>
<td>Data streams</td>
</tr>
<tr>
<td>Miklau</td>
<td>Umass</td>
<td>MA</td>
<td>Data security</td>
</tr>
</tbody>
</table>

Find people in State=WA  ⇒ Dalvi

Key (?!?)

Consider all “repairs”
The Repair Semantics

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>State</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miklau</td>
<td>UW</td>
<td>WA</td>
<td>Data security</td>
</tr>
<tr>
<td>Dalvi</td>
<td>UW</td>
<td>WA</td>
<td>Prob. Data</td>
</tr>
<tr>
<td>Balazinska</td>
<td>UW</td>
<td>WA</td>
<td>Data streams</td>
</tr>
<tr>
<td>Balazinska</td>
<td>MIT</td>
<td>MA</td>
<td>Data streams</td>
</tr>
<tr>
<td>Miklau</td>
<td>Umass</td>
<td>MA</td>
<td>Data security</td>
</tr>
</tbody>
</table>

Find people in State=WA  ⇒  Dalvi
Find people in State=MA  ⇒  ∅
The Repair Semantics

Consider all "repairs"

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>State</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miklau</td>
<td>UW</td>
<td>WA</td>
<td>Data security</td>
</tr>
<tr>
<td>Dalvi</td>
<td>UW</td>
<td>WA</td>
<td>Prob. Data</td>
</tr>
<tr>
<td>Balazinska</td>
<td>UW</td>
<td>WA</td>
<td>Data streams</td>
</tr>
<tr>
<td>Balazinska</td>
<td>MIT</td>
<td>MA</td>
<td>Data streams</td>
</tr>
<tr>
<td>Miklau</td>
<td>Umass</td>
<td>MA</td>
<td>Data security</td>
</tr>
</tbody>
</table>

Find people in State=WA  \Rightarrow Dalvi

Find people in State=MA  \Rightarrow \emptyset

Hi precision, but low recall
Alternative probabilistic approach

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>State</th>
<th>Area</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miklau</td>
<td>UW</td>
<td>WA</td>
<td>Data security</td>
<td>0.5</td>
</tr>
<tr>
<td>Dalvi</td>
<td>UW</td>
<td>WA</td>
<td>Prob. Data</td>
<td>1</td>
</tr>
<tr>
<td>Balazinska</td>
<td>UW</td>
<td>WA</td>
<td>Data streams</td>
<td>0.5</td>
</tr>
<tr>
<td>Balazinska</td>
<td>MIT</td>
<td>MA</td>
<td>Data streams</td>
<td>0.5</td>
</tr>
<tr>
<td>Miklau</td>
<td>Umass</td>
<td>MA</td>
<td>Data security</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Alternative probabilistic approach

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>State</th>
<th>Area</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miklau</td>
<td>UW</td>
<td>WA</td>
<td>Data security</td>
<td>0.5</td>
</tr>
<tr>
<td>Dalvi</td>
<td>UW</td>
<td>WA</td>
<td>Prob. Data</td>
<td>1</td>
</tr>
<tr>
<td>Balazinska</td>
<td>UW</td>
<td>WA</td>
<td>Data streams</td>
<td>0.5</td>
</tr>
<tr>
<td>Balazinska</td>
<td>MIT</td>
<td>MA</td>
<td>Data streams</td>
<td>0.5</td>
</tr>
<tr>
<td>Miklau</td>
<td>Umass</td>
<td>MA</td>
<td>Data security</td>
<td>0.5</td>
</tr>
</tbody>
</table>

State=WA $\Rightarrow$ Dalvi, Balazinska(0.5), Miklau(0.5)
Alternative probabilistic approach

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>State</th>
<th>Area</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miklau</td>
<td>UW</td>
<td>WA</td>
<td>Data security</td>
<td>0.5</td>
</tr>
<tr>
<td>Dalvi</td>
<td>UW</td>
<td>WA</td>
<td>Prob. Data</td>
<td>1</td>
</tr>
<tr>
<td>Balazinska</td>
<td>UW</td>
<td>WA</td>
<td>Data streams</td>
<td>0.5</td>
</tr>
<tr>
<td>Balazinska</td>
<td>MIT</td>
<td>MA</td>
<td>Data streams</td>
<td>0.5</td>
</tr>
<tr>
<td>Miklau</td>
<td>Umass</td>
<td>MA</td>
<td>Data security</td>
<td>0.5</td>
</tr>
</tbody>
</table>

State=WA  ⇒  Dalvi, Balazinska(0.5), Miklau(0.5)
State=MA  ⇒  Balazinska(0.5), Miklau(0.5)
Alternative probabilistic approach

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>State</th>
<th>Area</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miklau</td>
<td>UW</td>
<td>WA</td>
<td>Data security</td>
<td>0.5</td>
</tr>
<tr>
<td>Dalvi</td>
<td>UW</td>
<td>WA</td>
<td>Prob. Data</td>
<td>1</td>
</tr>
<tr>
<td>Balazinska</td>
<td>UW</td>
<td>WA</td>
<td>Data streams</td>
<td>0.5</td>
</tr>
<tr>
<td>Balazinska</td>
<td>MIT</td>
<td>MA</td>
<td>Data streams</td>
<td>0.5</td>
</tr>
<tr>
<td>Miklau</td>
<td>Umass</td>
<td>MA</td>
<td>Data security</td>
<td>0.5</td>
</tr>
</tbody>
</table>

State=WA  ⇒  Dalvi, Balazinska(0.5), Miklau(0.5)
State=MA  ⇒  Balazinska(0.5), Miklau(0.5)

Lower precision, but better recall
Ranking Query Answers

- Database is deterministic.

- Query answers are uncertain:
  - Query terms loosened due to user’s lack of understanding of the data or schema.
  - The query returns a ranked list of tuples; user interested in top-k.
The Empty Answers Problem

- Query is over-specified: no answers
- Example: try to buy a house in Seattle...

```
SELECT *
FROM Houses
WHERE bedrooms = 4
    AND style = 'craftsman'
    AND district = 'View Ridge'
    AND price < 400000
```

[Agrawal, Chaudhuri, Das, Gionis 2003]
Ranking answers

- Compute a similarity score between a tuple and the query

Q = SELECT *
   FROM   R
   WHERE A1=v1 AND … AND Am=vm
Computing a similarity score between a tuple and the query:

\[ Q = \text{SELECT} * \]
\[ \text{FROM} \quad R \]
\[ \text{WHERE} \quad A_1 = v_1 \quad \text{AND} \quad \ldots \quad \text{AND} \quad A_m = v_m \]

Query is a vector: \( Q = (v_1, \ldots, v_m) \)

Tuple is a vector: \( T = (u_1, \ldots, u_m) \)
Ranking answers

- Compute a similarity score between a tuple and the query

\[ Q = \text{SELECT} * \]
\[ \quad \text{FROM} \quad R \]
\[ \quad \text{WHERE} \quad A_1 = v_1 \quad \text{AND} \quad \ldots \quad \text{AND} \quad A_m = v_m \]

Query is a vector: \( Q = (v_1, \ldots, v_m) \)

Tuple is a vector: \( T = (u_1, \ldots, u_m) \)

Rank tuples by their TF/IDF similarity to the query Q

Includes partial matches
Keyword Search in Databases

\[ Q = \text{‘Abiteboul’ and ‘Widom’} \]

[Hristidis, Papakonstantinou’2002]
Keyword Search in Databases

Join sequences (tuple trees):

Q = ‘Abiteboul’ and ‘Widom’

[Author] \(\xrightarrow{\text{In}}\) [Conference] \(\xrightarrow{}\) [Editor]

[Person]

[Hristidis, Papakonstantinou’2002]
Keyword Search in Databases

Q = ‘Abiteboul’ and ‘Widom’

Join sequences (tuple trees):

[Charidis, Papakonstantinou’2002]
Keyword Search in Databases

Q = ‘Abiteboul’ and ‘Widom’

Join sequences (tuple trees):

[Abiteboul, Widom]

[Hristidis, Papakonstantinou’2002]
Keyword Search in Databases

Q = ‘Abiteboul’ and ‘Widom’

Join sequences (tuple trees):

[Person Abiteboul]

[Person Widom]

[Person Widom]

[Person Abiteboul]

[Person Abiteboul]

[Person Widom]

[Person Abiteboul]

[Person Widom]

[Person Abiteboul]

[Person Widom]

[Hristidis, Papakonstantinou’2002]
Keyword Search in Databases

- Goal: users don’t know schema; query database using keywords

- Techniques:
  - Matching objects may be scattered across physical tables due to normalization; need on-the-fly joins
  - Score of a tuple = number of joins, plus “prestige” based on in-degree
Summary: Motivating Applications

- **Text extraction & record linkage**: imprecise representations of objects. Probabilities can offer uniform treatment of uncertainty, need correlations and disjoint tuples.

- **Inconsistent data**: inconsistent data can be modeled probabilistically; may improve recall; requires tuple correlations.

- **Ranking query answers**: deterministic data, uncertain answers.
Outline

1. Motivating Applications
2. **Semantics of Probabilistic Data**
3. Representation Systems
4. Complexity
Possible Worlds Semantics

- Attribute domains: \( \text{int, char(30), varchar(55), datetime} \)

- Relation schema: \( \text{Employee(name:varchar(55), dob:datetime, salary:int)} \)

- Database schema: \( \text{Employee(\ldots), Projects(\ldots), Groups(\ldots)} \)

- Number of instances: \( N \) (big, but finite)
The Definition

The set of all possible database instances:

\[ \text{INST} = \{I_1, I_2, I_3, \ldots, I_N\} \]

**Definition** A probabilistic database \( I^p \) is a probability distribution on \( \text{INST} \)

\[ \text{Pr} : \text{INST} \rightarrow [0,1] \quad \text{s.t.} \quad \sum_{i=1,N} \text{Pr}(I_i) = 1 \]
The Definition

The set of all possible database instances:

\[ \text{INST} = \{I_1, I_2, I_3, \ldots, I_N\} \]

**Definition** A probabilistic database \( I^p \) is a probability distribution on \( \text{INST} \)

\[ \text{Pr} : \text{INST} \rightarrow [0,1] \quad \text{s.t.} \quad \sum_{i=1,N} \text{Pr}(I_i) = 1 \]

**Definition** A possible world is \( I \) s.t. \( \text{Pr}(I) > 0 \)
**Example**

\[ |p| = \]

\[
\begin{align*}
\text{Pr}(I_1) &= 1/3 \\
\text{Pr}(I_2) &= 1/12 \\
\text{Pr}(I_3) &= 1/2 \\
\text{Pr}(I_4) &= 1/12 \\
\end{align*}
\]

\[
\text{Possible worlds} = \{ I_1, I_2, I_3, I_4 \}
\]
Tuples as Events

One tuple $t$: event $t \in I$

$$\Pr(t) = \sum_{I: t \in I} \Pr(I)$$

Two tuples $t_1, t_2$: event $t_1 \in I \land t_2 \in I$

$$\Pr(t_1 t_2) = \sum_{I: t_1 \in I \land t_2 \in I} \Pr(I)$$
Tuple correlation

Disjoint  \[ \Pr(t_1 \cap t_2) = 0 \]

Negatively correlated  \[ \Pr(t_1 \cap t_2) < \Pr(t_1) \Pr(t_2) \]

**Independent**  \[ \Pr(t_1 \cap t_2) = \Pr(t_1) \Pr(t_2) \]

Positively correlated  \[ \Pr(t_1 \cap t_2) > \Pr(t_1) \Pr(t_2) \]

Identical  \[ \Pr(t_1 \cap t_2) = \Pr(t_1) = \Pr(t_2) \]
Example

\[ p = \]

\[
\begin{align*}
\Pr(I_1) &= \frac{1}{3} \\
\Pr(I_2) &= \frac{1}{12} \\
\Pr(I_3) &= \frac{1}{2} \\
\Pr(I_4) &= \frac{1}{12}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Customer</th>
<th>Address</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Gizmo</td>
</tr>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Gizmo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer</th>
<th>Address</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Boston</td>
<td>Gadget</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Gizmo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer</th>
<th>Address</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Gizmo</td>
</tr>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer</th>
<th>Address</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Boston</td>
<td>Gadget</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
</tbody>
</table>
Query Semantics

Given a query $Q$ and a probabilistic database $I^p$, what is the meaning of $Q(I^p)$?
Query Semantics

**Semantics 1: Possible Answers**
A probability distribution on *sets of tuples*

\[ \Pr(Q = A) = \sum_{I \in \text{INST.}} I \in Q(I) = A \Pr(I) \]

**Semantics 2: Possible Tuples**
A probability function on *tuples*

\[ \Pr(t \in Q) = \sum_{I \in \text{INST.}} t \in Q(I) \Pr(I) \]
Example: Query Semantics

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Gizmo</td>
</tr>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Gizmo</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Camera</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Boston</td>
<td>Gizmo</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Gizmo</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
<td>Gadget</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Gizmo</td>
</tr>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Boston</td>
<td>Camera</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
</tbody>
</table>

Purchase\(^p\)

\[ \Pr(I_1) = \frac{1}{3} \]

\[ \Pr(I_2) = \frac{1}{12} \]

\[ \Pr(I_3) = \frac{1}{2} \]

\[ \Pr(I_4) = \frac{1}{12} \]
Example: Query Semantics

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Gizmo</td>
</tr>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Gizmo</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Camera</td>
</tr>
<tr>
<td>John</td>
<td>Boston</td>
<td>Gizmo</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Gadget</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
</tbody>
</table>

**Purchase**

\[
\Pr(I_1) = 1/3
\]

\[
\Pr(I_2) = 1/12
\]

\[
\Pr(I_3) = 1/2
\]

\[
\Pr(I_4) = 1/12
\]

SELECT DISTINCT x.product
FROM Purchase p x, Purchase p y
WHERE x.name = 'John'
   and x.product = y.product 
   and y.name = 'Sue'
Example: Query Semantics

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Gizmo</td>
</tr>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Gizmo</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Camera</td>
</tr>
</tbody>
</table>

### Purchase\textsuperscript{p}

| Pr(I\textsubscript{1}) = 1/3 |

SELECT DISTINCT x.product
FROM Purchase\textsuperscript{p} x, Purchase\textsuperscript{p} y
WHERE x.name = 'John'
and x.product = y.product
and y.name = 'Sue'

### Possible answers

Semantics:

<table>
<thead>
<tr>
<th>Answer set</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo, Camera</td>
<td>1/3</td>
</tr>
<tr>
<td>Gizmo</td>
<td>1/12</td>
</tr>
<tr>
<td>Camera</td>
<td>7/12</td>
</tr>
</tbody>
</table>

Pr(I\textsubscript{2}) = 1/12
Pr(I\textsubscript{3}) = 1/2
Pr(I\textsubscript{4}) = 1/12
Example: Query Semantics

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Gizmo</td>
</tr>
<tr>
<td>John</td>
<td>Seattle</td>
<td>Camera</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Gizmo</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>Camera</td>
</tr>
</tbody>
</table>

\[ \Pr(I_1) = \frac{1}{3} \]

\[ \Pr(I_2) = \frac{1}{12} \]

\[ \Pr(I_3) = \frac{1}{2} \]

\[ \Pr(I_4) = \frac{1}{12} \]

SELECT DISTINCT x.product
FROM Purchase^p x, Purchase^p y
WHERE x.name = 'John'
and x.product = y.product
and y.name = 'Sue'

Possible answers semantics:

<table>
<thead>
<tr>
<th>Answer set</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo, Camera</td>
<td>( \frac{1}{3} )</td>
</tr>
<tr>
<td>Gizmo</td>
<td>( \frac{1}{12} )</td>
</tr>
<tr>
<td>Camera</td>
<td>( \frac{7}{12} )</td>
</tr>
<tr>
<td>( P(I_3) + P(I_4) )</td>
<td>( \Pr(I_3) + P(I_4) )</td>
</tr>
</tbody>
</table>

Possible tuples semantics:

<table>
<thead>
<tr>
<th>Tuple</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>( \frac{11}{12} )</td>
</tr>
<tr>
<td>Gizmo</td>
<td>( \frac{5}{12} )</td>
</tr>
<tr>
<td>( \Pr(I_1)+P(I_3) + P(I_4) )</td>
<td>( \Pr(I_1)+P(I_3) + P(I_4) )</td>
</tr>
</tbody>
</table>
Query semantics

- Query semantics

  - Very powerful: every SQL query has well-defined answer

  - **Possible answers** semantics

    - Precise; can be used to compose queries; difficult user interface?

  - **Possible tuples** semantics

    - Less precise, but simple; sufficient for most apps; cannot be used to compose queries. Simple user interface.
Outline

1. Motivating Applications
2. Semantics of Probabilistic Data
3. Representation Systems
4. Complexity
Representation Systems

• Need a good representation formalism for describing probabilistic databases, i.e. sets of possible worlds along with probabilities.

• **Completeness**: a representation system is complete if it can describe any probability distribution over instances.

• **Closure**

• Several representation systems exist, but no clear winner. *Probably the main open problem in this area.*
Representation systems

- Tuple independent databases -- very basic, intuitive
- Intensional databases - a complete formalism related to c-tables
- Incomplete formalisms:
  - Explicit tuple probabilities
  - Implicit tuple probabilities
Tuple independent probabilistic database
Tuple independent probabilistic database

\[ \text{TUP} = \{t_1, t_2, \ldots, t_M\} = \text{all tuples} \]

\[ \text{INST} = P(\text{TUP}) \]

\[ N = 2^M \]
Tuple independent probabilistic database

\[ \text{INST} = P(TUP) \]
\[ N = 2^M \]

\[ \text{TUP} = \{ t_1, t_2, \ldots, t_M \} = \text{all tuples} \]

\[ \text{Pr} : \text{TUP} \rightarrow [0,1] \]

No restrictions
Tuple independent probabilistic database

\[ \text{TUP} = \{ t_1, t_2, \ldots, t_M \} \quad = \text{all tuples} \]

\[ \text{Pr} : \text{TUP} \rightarrow [0,1] \quad \text{No restrictions} \]

\[ \text{Pr}(l) = \prod_{t \in l} \text{pr}(t) \times \prod_{t \notin l} (1-\text{pr}(t)) \]

\[ \text{INST} = \mathcal{P}(\text{TUP}) \quad N = 2^M \]
Tuple Prob. → Possible Worlds

\[ J = \]

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>( p_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>( p_1 = 0.8 )</td>
</tr>
<tr>
<td>Sue</td>
<td>Boston</td>
<td>( p_2 = 0.6 )</td>
</tr>
<tr>
<td>Fred</td>
<td>Boston</td>
<td>( p_3 = 0.9 )</td>
</tr>
</tbody>
</table>
Tuple Prob. → Possible Worlds

\[
J = \begin{array}{|c|c|c|}
\hline
\text{Name} & \text{City} & \text{pr} \\
\hline
\text{John} & \text{Seattle} & p_1 = 0.8 \\
\text{Sue} & \text{Boston} & p_2 = 0.6 \\
\text{Fred} & \text{Boston} & p_3 = 0.9 \\
\hline
\end{array}
\]

\[|p| = (1-p_1)(1-p_2)(1-p_3) \]

\[
\frac{1}{(1-p_2)(1-p_3)} + \frac{(1-p_1)p_3}{p_1(1-p_2)(1-p_3)} + \frac{(1-p_1)(1-p_2)p_3}{(1-p_1)(1-p_2)(1-p_3)} + \frac{p_1p_2p_3}{p_1p_2p_3} = 1
\]
### Tuple Prob. → Query Evaluation

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>$p_1$</td>
</tr>
<tr>
<td>Sue</td>
<td>Boston</td>
<td>$p_2$</td>
</tr>
<tr>
<td>Fred</td>
<td>Boston</td>
<td>$p_3$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer</th>
<th>Product</th>
<th>Date</th>
<th>pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Gizmo</td>
<td>...</td>
<td>$q_1$</td>
</tr>
<tr>
<td>John</td>
<td>Gadget</td>
<td>...</td>
<td>$q_2$</td>
</tr>
<tr>
<td>John</td>
<td>Gadget</td>
<td>...</td>
<td>$q_3$</td>
</tr>
<tr>
<td>Sue</td>
<td>Camera</td>
<td>...</td>
<td>$q_4$</td>
</tr>
<tr>
<td>Sue</td>
<td>Gadget</td>
<td>...</td>
<td>$q_5$</td>
</tr>
<tr>
<td>Sue</td>
<td>Gadget</td>
<td>...</td>
<td>$q_6$</td>
</tr>
<tr>
<td>Fred</td>
<td>Gadget</td>
<td>...</td>
<td>$q_7$</td>
</tr>
</tbody>
</table>
### Tuple Prob. → Query Evaluation

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>pr</th>
<th>Customer</th>
<th>Product</th>
<th>Date</th>
<th>pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>p₁</td>
<td>John</td>
<td>Gizmo</td>
<td></td>
<td>q₁</td>
</tr>
<tr>
<td>Sue</td>
<td>Boston</td>
<td>p₂</td>
<td>John</td>
<td>Gadget</td>
<td></td>
<td>q₂</td>
</tr>
<tr>
<td>Fred</td>
<td>Boston</td>
<td>p₃</td>
<td>John</td>
<td>Gadget</td>
<td></td>
<td>q₃</td>
</tr>
<tr>
<td>Sue</td>
<td></td>
<td></td>
<td>Sue</td>
<td>Camera</td>
<td></td>
<td>q₄</td>
</tr>
<tr>
<td>Sue</td>
<td></td>
<td></td>
<td>Sue</td>
<td>Gadget</td>
<td></td>
<td>q₅</td>
</tr>
<tr>
<td>Fred</td>
<td></td>
<td></td>
<td>Sue</td>
<td>Gadget</td>
<td></td>
<td>q₆</td>
</tr>
<tr>
<td>Sue</td>
<td></td>
<td></td>
<td>Sue</td>
<td>Gadget</td>
<td></td>
<td>q₇</td>
</tr>
</tbody>
</table>

```
SELECT DISTINCT x.city
FROM Person x, Purchase y
WHERE x.Name = y.Customer
       and y.Product = 'Gadget'
```
SELECT DISTINCT x.city 
FROM Person x, Purchase y 
WHERE x.Name = y.Customer 
and y.Product = 'Gadget'
SELECT DISTINCT x.city
FROM Person x, Purchase y
WHERE x.Name = y.Customer 
         and y.Product = 'Gadget'

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>p_1</td>
</tr>
<tr>
<td>Sue</td>
<td>Boston</td>
<td>p_2</td>
</tr>
<tr>
<td>Fred</td>
<td>Boston</td>
<td>p_3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer</th>
<th>Product</th>
<th>Date</th>
<th>pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Gizmo</td>
<td></td>
<td>q_1</td>
</tr>
<tr>
<td>John</td>
<td>Gadget</td>
<td></td>
<td>q_2</td>
</tr>
<tr>
<td>John</td>
<td>Gadget</td>
<td></td>
<td>q_3</td>
</tr>
<tr>
<td>Sue</td>
<td>Camera</td>
<td></td>
<td>q_4</td>
</tr>
<tr>
<td>Sue</td>
<td>Gadget</td>
<td></td>
<td>q_5</td>
</tr>
<tr>
<td>Sue</td>
<td>Gadget</td>
<td></td>
<td>q_6</td>
</tr>
<tr>
<td>Fred</td>
<td>Gadget</td>
<td></td>
<td>q_7</td>
</tr>
</tbody>
</table>

Tuple Probability

<table>
<thead>
<tr>
<th>Tuple</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seattle</td>
<td>p_1(1-(1-q_2)(1-q_3))</td>
</tr>
<tr>
<td>Boston</td>
<td></td>
</tr>
</tbody>
</table>
SELECT DISTINCT x.city
FROM Person x, Purchase y
WHERE x.Name = y.Customer
and y.Product = 'Gadget'
Tuple-independent distributions

- Expressive power
  - Possible worlds are limited to subsets.
  - Probability distribution cannot accommodate correlations

- Queries:
  - Not closed under query application.
Intensional Database

Atomic event ids

\( e_1, e_2, e_3, \ldots \)

Probabilities:

\( p_1, p_2, p_3, \ldots \in [0,1] \)

Event expressions: \( \land, \lor, : \)

\( e_3 \land (e_5 \lor \neg e_2) \)

Intensional probabilistic database \( J \):
each tuple \( t \) has an event attribute \( t.E \)

[Fuhr&Roellke:1997]
Intensional DB ⇒ Possible Worlds

\[
\begin{array}{|c|c|c|}
\hline
\text{Name} & \text{Address} & \text{E} \\
\hline
\text{John} & \text{Seattle} & e_1 \land (e_2 \lor e_3) \\
\hline
\text{Sue} & \text{Denver} & (e_1 \land e_2) \lor (e_2 \land e_3) \\
\hline
\end{array}
\]
Intensional DB $\Rightarrow$ Possible Worlds

\[ J = \]

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
<td>$e_1 \land (e_2 \lor e_3)$</td>
</tr>
<tr>
<td>Sue</td>
<td>Denver</td>
<td>$(e_1 \land e_2) \lor (e_2 \land e_3)$</td>
</tr>
</tbody>
</table>

\[ e_1 e_2 e_3 = 000 \quad 001 \quad 010 \quad 011 \quad 100 \quad 101 \quad 110 \quad 111 \]

\[ |p| \]

\[ \emptyset \quad \text{John} \quad \text{Seattle} \quad \text{Sue} \quad \text{Denver} \quad \text{John} \quad \text{Seattle} \quad \text{Sue} \quad \text{Denver} \]
Intensional DB ⇒ Possible Worlds

\[ J = \begin{array}{|c|c|c|}
\hline
\text{Name} & \text{Address} & \text{E} \\
\hline
\text{John} & \text{Seattle} & e_1 \land (e_2 \lor e_3) \\
\hline
\text{Sue} & \text{Denver} & (e_1 \land e_2) \lor (e_2 \land e_3) \\
\hline
\end{array} \]

\[ e_1 e_2 e_3 = \begin{array}{cccccccc}
000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\
\hline
\end{array} \]

\[ |p| \quad \emptyset \quad \begin{array}{cc}
\text{John} & \text{Seattle} \\
\hline
\end{array} \quad \begin{array}{cc}
\text{Sue} & \text{Denver} \\
\hline
\end{array} \quad \begin{array}{cc}
\text{John} & \text{Seattle} \\
\hline
\end{array} \quad \begin{array}{cc}
\text{Sue} & \text{Denver} \\
\hline
\end{array} \]

\[ (1-p_1)(1-p_2)(1-p_3) + (1-p_1)(1-p_2)p_3 + (1-p_1)p_2(1-p_3) + p_1(1-p_2)(1-p_3) \]

\[ p_1(1-p_2) p_3 + (1-p_1)p_2 p_3 + p_1 p_2(1-p_3) + p_1 p_2 p_3 \]
Possible Worlds $\Rightarrow$ Intensional DB

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
</tr>
<tr>
<td>John</td>
<td>Boston</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sue</td>
<td>Seattle</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Boston</td>
</tr>
</tbody>
</table>

E₁ = e₁
E₂ = ¬e₁ ∧ e₂
E₃ = ¬e₁ ∧ ¬e₂ ∧ e₃
E₄ = ¬e₁ ∧ ¬e₂ ∧ ¬e₃ ∧ e₄

Pr(e₁) = p₁
Pr(e₂) = p₂/(1-p₁)
Pr(e₃) = p₃/(1-p₁-p₂)
Pr(e₄) = p₄/(1-p₁-p₂-p₃)

“Prefix code”
Possible Worlds ⇒ Intensional DB

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Seattle</td>
</tr>
<tr>
<td>John</td>
<td>Boston</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
E_1 &= e_1 \\
E_2 &= \neg e_1 \land e_2 \\
E_3 &= \neg e_1 \land \neg e_2 \land e_3 \\
E_4 &= \neg e_1 \land \neg e_2 \land \neg e_3 \land e_4
\end{align*}
\]

“Prefix code”

\[
\begin{align*}
\Pr(e_1) &= p_1 \\
\Pr(e_2) &= p_2/(1-p_1) \\
\Pr(e_3) &= p_3/(1-p_1-p_2) \\
\Pr(e_4) &= p_4/(1-p_1-p_2-p_3)
\end{align*}
\]

\[
J = E_1 \lor E_2 \\
E_1 \lor E_4 \\
E_1 \lor E_2 \lor E_3
\]
Possible Worlds ⇒ Intensional DB

Intensional DBs are complete
Closure Under Operators
Closure Under Operators

\[ \sigma \]

\[ v E \]

\[ v_1 E_1 \]

\[ v_2 E_2 \]

\[ v E_1 \]

\[ v E_2 \]

\[ \ldots \ldots \]

\[ v E_1 \]

\[ v E_2 \]

[Fuhr&Roellke:1997]
Closure Under Operators

[Fuhr&Roellke:1997]
Closure Under Operators

\[ \sigma \]

\[ \times \]

\[ \Pi \]

\[ \mathcal{F} \text{uhr}&\text{Roellke:1997} \]
Closure Under Operators

\[ \sigma \]

\[ v \quad E \]

\[ v_1 \quad v_2 \quad E_1 \land E_2 \]

\[ \vee \quad E_1 \vee E_2 \vee \ldots \]

\[ \Pi \]

\[ v \quad E_1 \land \neg E_2 \]

\[ v \quad E_1 \]

\[ v \quad E_2 \]

\[ \ldots \quad \ldots \]
One still needs to compute probability of event expression
Summary of Intensional Databases

- Event expression for each tuple
- Possible worlds: any subset
- Probability distribution: any
- Complete, and (therefore) closed under relational queries
- but impractical: provably inefficient to compute

- Related to c-tables [Imielinski&Lipski:1984]
Outline

1. Motivating Applications
2. Semantics of Probabilistic Data
3. Representation Systems
4. Complexity
Outline

- Probability of boolean expressions
- Query probability
Probability of Boolean Expressions

\[ E = X_1 X_3 \lor X_1 X_4 \lor X_2 X_5 \lor X_2 X_6 \]

Randomly make each variable true with the following probabilities:

\[ \Pr(X_1) = p_1, \quad \Pr(X_2) = p_2, \ldots \ldots \ldots, \quad \Pr(X_6) = p_6 \]
Probability of Boolean Expressions

Randomly make each variable true with the following probabilities

Pr(X_1) = p_1,  Pr(X_2) = p_2,  \ldots,  Pr(X_6) = p_6

What is Pr(E) ???
Probability of Boolean Expressions

\[ E = X_1X_3 \lor X_1X_4 \lor X_2X_5 \lor X_2X_6 \]

Randomly make each variable true with the following probabilities

\[
\Pr(X_1) = p_1, \quad \Pr(X_2) = p_2, \quad \ldots, \quad \Pr(X_6) = p_6
\]

What is \( \Pr(E) \) ???

Answer: re-group cleverly

\[ E = X_1 (X_3 \lor X_4) \lor X_2 (X_5 \lor X_6) \]

\[
\Pr(E)=1 - (1-p_1(1-(1-p_3)(1-p_4)))
\]
\[
(1-p_2(1-(1-p_5)(1-p_6)))
\]
Now let’s try this:

\[ E = X_1X_2 \lor X_1X_3 \lor X_2X_3 \]
Now let’s try this:

\[ E = X_1 X_2 \lor X_1 X_3 \lor X_2 X_3 \]

No clever grouping seems possible. Brute force:
Now let’s try this:

$$E = X_1X_2 \lor X_1X_3 \lor X_2X_3$$

No clever grouping seems possible. Brute force:

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$E$</th>
<th>$Pr$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$(1-p_1)p_2p_3$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$(1-p_1)p_2p_3$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$p_1(1-p_2)p_3$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$p_1p_2(1-p_3)$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$p_1p_2p_3$</td>
</tr>
</tbody>
</table>

$$Pr(E) = (1-p_1)p_2p_3 + p_1(1-p_2)p_3 + p_1p_2(1-p_3) + p_1p_2p_3$$
Now let’s try this:

\[ E = X_1X_2 \vee X_1X_3 \vee X_2X_3 \]

No clever grouping seems possible. Brute force:

\[
\begin{array}{cccc|c}
X_1 & X_2 & X_3 & E & Pr \\
0 & 0 & 0 & 0 & \\
0 & 0 & 1 & 0 & \\
0 & 1 & 0 & 0 & \\
0 & 1 & 1 & 1 & (1-p_1)p_2p_3 \\
1 & 0 & 0 & 0 & \\
1 & 0 & 1 & 1 & p_1(1-p_2)p_3 \\
1 & 1 & 0 & 1 & p_1p_2(1-p_3) \\
1 & 1 & 1 & 1 & p_1p_2p_3 \\
\end{array}
\]

\[
Pr(E)=(1-p_1)p_2p_3 + p_1(1-p_2)p_3 + p_1p_2(1-p_3) + p_1p_2p_3
\]

Seems inefficient in general…
Complexity of Boolean Expression Probability

**Theorem** [Valiant:1979]
For a boolean expression $E$, computing $\Pr(E)$ is $\#P$-complete.
Complexity of Boolean Expression Probability

**Theorem** [Valiant:1979]
For a boolean expression $E$, computing $\Pr(E)$ is \#P-complete

$NP = \text{class of problems of the form "is there a witness?" } SAT$

$\#P = \text{class of problems of the form "how many witnesses?" } \#SAT$
Complexity of Boolean Expression Probability

**Theorem** [Valiant:1979]
For a boolean expression $E$, computing $Pr(E)$ is $\#P$-complete

$NP = \text{class of problems of the form “is there a witness?” SAT}$

$\#P = \text{class of problems of the form “how many witnesses?” } \#\text{SAT}$

The decision problem for 2CNF is in PTIME
The counting problem for 2CNF is $\#P$-complete
Query Complexity

Data complexity of a query Q:

• Compute $Q(I^p)$, for probabilistic database $I^p$

Simplest scenario only:

• Possible tuples semantics for Q
• Independent tuples for $I^p$
Intensional query evaluation
Intensional query evaluation

[Fuhr&Roellke:1997]
Intensional query evaluation

\[ \sigma \]

\[ \Pi \]

\[ v \ E \]

\[ v_1 \ E_1 \]

\[ v_2 \ E_2 \]

\[ v_1 \ v_2 \ E_1 \land E_2 \]

\[ v \ E_1 \]

\[ v \ E_2 \]

\[ \ldots \ldots \]

\[ v \ E_1 \]

\[ v \ E_2 \]
Intensional query evaluation

[Fuhr&Roellke:1997]
Intensional query evaluation

\[ \sigma \]

\[
\begin{array}{ccc}
\lor & E \\
\lor & v_1 & v_2 \\
\land & E_1 & E_2
\end{array}
\]

\[
\begin{array}{ccc}
\lor & E_1 & \lor & E_2 & \lor & \ldots
\end{array}
\]

\[
\begin{array}{ccc}
\lor & E_1 \\
\land & \neg E_2
\end{array}
\]

\[
\begin{array}{ccc}
\lor & E_1 \\
\lor & E_2
\end{array}
\]

\[
\begin{array}{ccc}
\lor & E_2
\end{array}
\]

\[
\begin{array}{ccc}
\lor & E_1 \\
\land & \ldots
\end{array}
\]

\[
\begin{array}{ccc}
\lor & E_2
\end{array}
\]

[Fuhr&Roellke:1997]
Intensional query evaluation

One still needs to compute probability of event expression
Extensional Query Evaluation

Relational ops compute probabilities

\[ \sigma \]

\[ v \quad p \]

\[ v_1 \quad p_1 \]

\[ v_2 \quad p_2 \]

\[ \times \]

\[ \Pi \]

\[ v \quad p_1 \]

\[ v \quad p_2 \]

\[ - \]

\[ v \quad p_2 \]

Extensional Query Evaluation

Relational ops compute probabilities

Extensional Query Evaluation

Relational ops compute probabilities

Extensional Query Evaluation

Relational ops compute probabilities

\[
\sigma_{v} \times \Pi_{v} \left( 1 - (1 - p_1)(1 - p_2) \ldots \right)
\]

Extensional Query Evaluation

Relational ops compute probabilities

\[ \sigma \]

\[ \times \]

\[ \Pi \]

\[ \ominus \]
Extensional Query Evaluation

Relational ops compute probabilities

Data complexity: PTIME
SELECT DISTINCT x.City
FROM Person^p x, Purchase^p y
WHERE x.Name = y.Cust 
and y.Product = 'Gadget'
SELECT DISTINCT x.City
FROM Person^p x, Purchase^p y
WHERE x.Name = y.Cust 
         and y.Product = 'Gadget'

Sea | 1-(1-p_1q_1)(1- p_1q_2)(1- p_1q_3)

Π
Jon | Sea | p_1q_1
Jon | Sea | p_1q_2
Jon | Sea | p_1q_3

Jon | Sea | p_1

Jon | q_1
Jon | q_2
Jon | q_3

[Dalvi&S:2004]
SELECT DISTINCT x.City
FROM Person x, Purchase y
WHERE x.Name = y.Cust
and y.Product = 'Gadget'

Wrong!

Sea | 1-(1-p₁q₁)(1- p₁q₂)(1- p₁q₃)

Jon | Sea | p₁q₁
Jon | Sea | p₁q₂
Jon | Sea | p₁q₃

Jon | Sea | p₁

Jon | q₁
Jon | q₂
Jon | q₃

[Dalvi&S:2004]
SELECT DISTINCT x.City
FROM Person^p x, Purchase^p y
WHERE x.Name = y.Cust 
         and y.Product = 'Gadget'

Wrong!

\[ \Pi \]

\[ \begin{array}{ccc}
  \text{Jon} & \text{Sea} & p_1 q_1 \\
  \text{Jon} & \text{Sea} & p_1 q_2 \\
  \text{Jon} & \text{Sea} & p_1 q_3 \\
\end{array} \]

Correct

\[ \Pi \]

\[ \begin{array}{ccc}
  \text{Jon} & \text{Sea} & p_1 (1-(1-q_1)(1-q_2)(1-q_3)) \\
  \text{Jon} & \text{q}_1 \\
  \text{Jon} & \text{q}_2 \\
  \text{Jon} & \text{q}_3 \\
\end{array} \]

Reference:
Dalvi&S:2004
SELECT DISTINCT x.City
FROM Person x, Purchase y
WHERE x.Name = y.Cust
  and y.Product = 'Gadget'

[Dalvi&S:2004]

Wrong!

Sea  \(1-(1-p_1 q_1)(1- p_1 q_2)(1- p_1 q_3)\)

jon Sea  \(p_1(1-(1-q_1)(1-q_2)(1-q_3))\)

Correct

Jon  \(1-(1-q_1)(1-q_2)(1-q_3)\)

Depends on plan !!!
Query Complexity

Sometimes no correct extensional plan

$$Q_{\text{bad}} : - R(x), S(x,y), T(y)$$
Query Complexity

Sometimes no correct extensional plan

Q_{bad} :- R(x), S(x,y), T(y)

Data complexity is #P complete
Query Complexity

Sometimes no correct extensional plan

\[ Q_{\text{bad}} :- R(x), S(x,y), T(y) \]

Data complexity is \#P complete

Theorem The following are equivalent

- \( Q \) has PTIME data complexity
- \( Q \) admits an extensional plan (and one finds it in PTIME)
- \( Q \) does not have \( Q_{\text{bad}} \) as a subquery

[Dalvi&S:2004]
Summary on Query Complexity

Extensional query evaluation:

• Very popular
  • generalized to “strategies” [Lakshmanan et al. 1997]

• However, result depends on query plan!

General query complexity

• #P complete (not surprising, given #SAT)

• Already #P hard for very simple query ($Q_{bad}$)
Summary on Query Complexity

Extensional query evaluation:

• Very popular
  - generalized to “strategies” [Lakshmanan et al.1997]

• However, result depends on query plan!

General query complexity

• \#P complete (not surprising, given \#SAT)

• Already \#P hard for very simple query \(Q_{bad}\)

Probabilistic databases have high query complexity