Management of Probabilistic Data: Foundations and Challenges

Nilesh Dalvi and Dan Suciu
Univerisity of Washington
Databases Are Deterministic

- Applications since 1970’s required precise semantics
  - Accounting, inventory
- Database tools are deterministic
  - A tuple is an answer or is not
- Underlying theory assumes determinism
  - FO (First Order Logic)
Future of Data Management

We need to cope with uncertainties!

• Represent uncertainties as probabilities

• Extend data management tools to handle probabilistic data

*Major* paradigm shift affecting both foundations and systems
Uncertainties Everywhere

• In the schema mappings:
  – Data spaces
  – *Pay as you go* data integration

• In the data mapping
  – Life science data integration
  – Object reconciliation, fuzzy joins

• In the data itself
  – Data “by the masses”
  – Information Extraction
  – RFID data, sensor data

[Halevy’2007]
[Philippi&Kohler’2006]
[Arasu’06]
[Gupta&Sarawagi’2006]
[Welbourne’2007]
Example 1
Data Integration in Life Sciences

• U2 integrates several biological databases

Example: find functional annotations of ABCD1

User types: “Gene ABCD1”
U2 finds 80 “related” proteins
Ranks them by *uncertainty score*
Correct 9 functions are among top 11

Need to represent uncertainties explicitly
Example 2
Information Extraction

...52 A Goregaon West Mumbai ...

<table>
<thead>
<tr>
<th>ID</th>
<th>House-No</th>
<th>Street</th>
<th>City</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>52</td>
<td>Goregaon West</td>
<td>Mumbai</td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>52-A</td>
<td>Goregaon West</td>
<td>Mumbai</td>
<td>0.4</td>
</tr>
<tr>
<td>1</td>
<td>52</td>
<td>Goregaon</td>
<td>West Mumbai</td>
<td>0.2</td>
</tr>
<tr>
<td>1</td>
<td>52-A</td>
<td>Goregaon</td>
<td>West Mumbai</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>....</td>
<td>....</td>
<td>....</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>....</td>
<td>....</td>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>

Here probabilities are meaningful

≈20% of such extractions are correct

[Gupta&Sarawagi’2006]
Example 3
RFID Ecosystem at UW

[Welbourne'2007]
• RFID data = noisy
  – SIGHTING(tagID, antennaID, time)

• Derived data = Probabilistic
  – “John entered Room 524 at 9:15”  prob=0.6
  – “John carried laptop x77 at 11:03”  prob=0.8
  – . . .

• Queries
  – “Which people were in Room 478 yesterday?”

Massive amounts of probabilistic data from RFIDs, sensors
A Model for Uncertainties

- Data is probabilistic
- Queries formulated in a standard language
- Answers are annotated with probabilities

This talk: Probabilistic Databases
Probabilistic databases: Long History

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

Focus today: the Query Evaluation Problem
Has this been solved by AI?

<table>
<thead>
<tr>
<th>AI</th>
<th>Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Theorem prover</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>Query processing</td>
</tr>
<tr>
<td>Input: KB</td>
<td>Fix q</td>
</tr>
<tr>
<td>Input: DB</td>
<td>[this talk]</td>
</tr>
<tr>
<td>Probabilistic inference</td>
<td>[this talk]</td>
</tr>
</tbody>
</table>
Outline

• Data model

• Query evaluation

• Challenges
What is a Probabilistic Database (PDB)?

HasObject\(p\)

<table>
<thead>
<tr>
<th>Object</th>
<th>Time</th>
<th>Person</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop77</td>
<td>9:07</td>
<td>John</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jim</td>
<td>0.34</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>Mary</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>John</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fred</td>
<td>0.11</td>
</tr>
</tbody>
</table>

[Barbara et al. 1992]
What is a Probabilistic Database (PDB)?

What does it mean?

### HasObject

<table>
<thead>
<tr>
<th>Object</th>
<th>Time</th>
<th>Person</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop77</td>
<td>9:07</td>
<td>John</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jim</td>
<td>0.34</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>Mary</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>John</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fred</td>
<td>0.11</td>
</tr>
</tbody>
</table>

[Barbara et al.1992]
Background

Finite probability space = \((\Omega, P)\)

\[ \Omega = \{\omega_1, \ldots, \omega_n\} = \text{set of outcomes} \]
\[ P : \Omega \to [0,1] \]
\[ P(\omega_1) + \ldots + P(\omega_n) = 1 \]

Event: \(E \subseteq \Omega, \quad P(E) = \sum_{\omega \in E} P(\omega)\)

“Independent”: \(P(E_1 E_2) = P(E_1) P(E_2)\)

“Mutual exclusive” or “disjoint”: \(P(E_1 E_2) = 0\)
Possible Worlds Semantics

<table>
<thead>
<tr>
<th>Object</th>
<th>Time</th>
<th>Person</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop77</td>
<td>9:07</td>
<td>John</td>
<td>p₁</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jim</td>
<td>p₂</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>Mary</td>
<td>p₃</td>
</tr>
<tr>
<td></td>
<td></td>
<td>John</td>
<td>p₄</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fred</td>
<td>p₅</td>
</tr>
</tbody>
</table>

\[ \Omega = \{ \} \]
Possible Worlds Semantics

\[ \Omega = \{ p_1 p_3 \} \]

\[
\begin{array}{|c|c|c|}
\hline
\text{Object} & \text{Time} & \text{Person} \\
\hline
\text{Laptop77} & 9:07 & \text{John} \\
\hline
\text{Book302} & 9:18 & \text{Mary} \\
\hline
\end{array}
\]

Possible worlds

PDB

\[
\Omega = \{ \}
\]

\[
\begin{array}{|c|c|c|}
\hline
\text{Object} & \text{Time} & \text{Person} \\
\hline
\text{Laptop77} & 9:07 & \text{John} \\
\hline
\text{Book302} & 9:18 & \text{Mary} \\
\hline
\end{array}
\]
Possible Worlds Semantics

\[ \Omega = \{ \text{Object} \mid \text{Time} \mid \text{Person} \mid P \} \]

<table>
<thead>
<tr>
<th>Object</th>
<th>Time</th>
<th>Person</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop77</td>
<td>9:07</td>
<td>John</td>
<td>p_1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jim</td>
<td>p_2</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>Mary</td>
<td>p_3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>John</td>
<td>p_4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fred</td>
<td>p_5</td>
</tr>
</tbody>
</table>

PDB

Possible worlds

\[ p_1p_4 \]
### Possible Worlds Semantics

$$\Omega = \{ p_1, p_1p_4, p_1(1-p_3-p_4-p_5) \}$$

<table>
<thead>
<tr>
<th>Object</th>
<th>Time</th>
<th>Person</th>
<th>( P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop77</td>
<td>9:07</td>
<td>John</td>
<td>( p_1 )</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>John</td>
<td>( p_4 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fred</td>
<td>( p_5 )</td>
</tr>
</tbody>
</table>

Possible worlds

PDB
# Possible Worlds Semantics

\[ \Omega = \{ p_1, p_1 p_4, p_1(1-p_3-p_4-p_5) \} \]

<table>
<thead>
<tr>
<th>Object</th>
<th>Time</th>
<th>Person</th>
<th>( P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop77</td>
<td>9:07</td>
<td>John</td>
<td>( p_1 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jim</td>
<td>( p_2 )</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>Mary</td>
<td>( p_3 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>John</td>
<td>( p_4 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fred</td>
<td>( p_5 )</td>
</tr>
</tbody>
</table>

Possible worlds:

- \( p_1 \)
- \( p_1 p_4 \)
- \( p_1(1-p_3-p_4-p_5) \)

PDB
Definitions

**Definition:** A tuple-disjoint/independent table is:

\[ R(A_1, A_2, \ldots, A_m, B_1, \ldots, B_n, P) \]

**Definition:** A tuple-independent table is:

\[ R(A_1, A_2, \ldots, A_m, P) \]

**Definition:** Semantics is given by possible worlds
HasObject(\textbf{Object}, \textbf{Time}, Person, P)

<table>
<thead>
<tr>
<th>Object</th>
<th>Time</th>
<th>Person</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop77</td>
<td>9:07</td>
<td>John</td>
<td>(p_1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jim</td>
<td>(p_2)</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>Mary</td>
<td>(p_3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>John</td>
<td>(p_4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fred</td>
<td>(p_5)</td>
</tr>
</tbody>
</table>

Meets(\textbf{Person1}, \textbf{Person2}, \textbf{Time}, P)

<table>
<thead>
<tr>
<th>Person1</th>
<th>Person2</th>
<th>Time</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Jim</td>
<td>9.12</td>
<td>(p_1)</td>
</tr>
<tr>
<td>Mary</td>
<td>Sue</td>
<td>9:20</td>
<td>(p_2)</td>
</tr>
<tr>
<td>John</td>
<td>Mary</td>
<td>9:20</td>
<td>(p_3)</td>
</tr>
</tbody>
</table>

\(\text{Independent}\) \quad \{\text{Disjoint}\}
Query Semantics

A **boolean query** $q$ is an event: \( \{ \omega \mid \omega \models q \} \)

\[
P(q) = \sum_{\omega \models q} P(\omega)
\]

Did someone take MyBook to the CoffeeRoom?

$q =$

$\text{HasObject('MyBook',x,t), EnterRoom(x,'CoffeeRoom',t)}$

$\Rightarrow$  

$P(q) = 0.96$  (meaning: quite likely !)
Discussion of Data Model

Tuple-disjoint/independent tables:
• Simple model, can store in any DBMS

More advanced models:
• Symbolic boolean expressions
• Trio: add lineage
• Probabilistic Relational Models
• Graphical models

Fuhr and Roellke

[Widom05, Das Sarma’06, Benjelloun 06]

[Getoor’2006]

[Sen&Desphande’07]
Outline

• Data model

• Query evaluation
  – Probability of Boolean expressions
  – From queries to Boolean expressions
  – Data complexity of query evaluation

• Challenges
Probability of Boolean Expressions

\[ \Phi = X_1X_2 \lor X_1X_3 \lor X_2X_3 \]

\[ P(X_1) = p_1, \ P(X_2) = p_2, \ P(X_3) = p_3 \]

Compute \( P(\Phi) \)
**Probability of Boolean Expressions**

\[ \Phi = \overline{X_1} X_2 \lor X_1 \overline{X_3} \lor \overline{X_2} X_3 \]

P\((X_1)\) = \(p_1\), P\((X_2)\) = \(p_2\), P\((X_3)\) = \(p_3\)  
Compute P\((\Phi)\)

<table>
<thead>
<tr>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(X_3)</th>
<th>P</th>
<th>(\Phi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>(1-p_1)p_2p_3)</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(p_1(1-p_2)p_3)</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(p_1p_2(1-p_3))</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(p_1p_2p_3)</td>
<td>1</td>
</tr>
</tbody>
</table>
Probability of Boolean Expressions

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

\[ P(X_1) = p_1, \ P(X_2) = p_2, \ P(X_3) = p_3 \]

Compute \( P(\Phi) \)

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

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

Fix $P(X_1) = P(X_2) = \ldots = P(X_n) = 1/2$

**Theorem**
Exact evaluation of $Pr(\Phi)$ is $\#P$-complete

**Theorem**
For DNF $\Phi$
Approximation of $Pr(\Phi)$ is in PTIME (FPTRAS)

Both theorems extend to rational $P(X_1), \ldots, P(X_n)$

[Valiant:1979]

[Karp&Luby:1983]

Query $q$ + Database $PDB \Rightarrow \Phi$

$q = R(\mathbf{x, y}), S(\mathbf{x, z})$

$PDB = \begin{array}{ccc}
\hline
\text{A} & \text{B} & \text{P} \\
\text{a}_1 & \text{b}_1 & \text{p}_1 \\
\text{a}_2 & \text{b}_2 & \text{p}_2 \\
\hline
\end{array}$

$S^p = \begin{array}{ccc}
\hline
\text{A} & \text{C} & \text{P} \\
\text{a}_1 & \text{c}_1 & \text{q}_1 \\
\text{a}_1 & \text{c}_2 & \text{q}_2 \\
\text{a}_2 & \text{c}_3 & \text{q}_3 \\
\text{a}_2 & \text{c}_4 & \text{q}_4 \\
\text{a}_2 & \text{c}_5 & \text{q}_5 \\
\hline
\end{array}$

$\mathbf{x}_1 \quad \mathbf{x}_2$

$\mathbf{Y}_1 \quad \mathbf{Y}_2 \quad \mathbf{Y}_3 \quad \mathbf{Y}_4 \quad \mathbf{Y}_5$
Query $q + \text{ Database PDB} \rightarrow \Phi$

$q = R(x, y), S(x, z)$

PDB =

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>p1</td>
<td></td>
</tr>
<tr>
<td>a2</td>
<td>b2</td>
<td>p2</td>
<td></td>
</tr>
</tbody>
</table>

$\Phi = X_1Y_1 \lor X_1Y_2 \lor X_2Y_3 \lor X_2Y_4 \lor X_2Y_5$
Application to Query Evaluation

**Corollary** Fix FO query $q$
Exact evaluation of $\Pr(q)$ on input PDB is in $\#P$

**Corollary** Fix a conjunctive query $q$.
Approximation of $\Pr(q)$ on input PDB is in PTIME (FPTRAS)

Background: Probabilistic Networks

Inference: hard in general

KR techniques exploit local properties:

E.g. bounded treewidth $\Rightarrow$ PTIME

Note: for this query the treewidth is unbounded

$\Phi = X_1 Y_1 \lor X_1 Y_2 \lor X_2 Y_3 \lor X_2 Y_4 \lor X_2 Y_5$

$R(x, y), S(x, z)$

[Zabiyaka&Darwiche’06]
\[
q = \text{Join} (\Pi_A \big\otimes \Pi_A) \big\otimes \Pi_\emptyset \\
= R(x, y), S(x, z)
\]
q = \text{Join}\ R({\textbf{x}}, \textbf{y}), S({\textbf{x}}, \textbf{z})

\[\begin{array}{|c|c|c|}
\hline
\textbf{A} & \textbf{B} & \textbf{P} \\
\hline
a_1 & b_1 & p_1 \\
a_2 & b_2 & p_2 \\
\hline
\end{array}\]

\[\begin{array}{|c|c|c|}
\hline
\textbf{A} & \textbf{C} & \textbf{P} \\
\hline
a_1 & c_1 & q_1 \\
a_1 & c_2 & q_2 \\
a_2 & c_3 & q_3 \\
a_2 & c_4 & q_4 \\
a_2 & c_5 & q_5 \\
\hline
\end{array}\]
$q = \text{Join} \left( R(\mathbf{x}, \mathbf{y}), S(\mathbf{x}, \mathbf{z}) \right)$

$\Pi \emptyset$

$\Pi_A$

$\Pi_A$

$R^p(\mathbf{A}, \mathbf{B})$

$S^p(\mathbf{A}, \mathbf{C})$

$\begin{array}{|c|c|c|}
\hline
 \mathbf{A} & \mathbf{B} & \mathbf{P} \\
\hline
 a_1 & b_1 & p_1 \\
 a_2 & b_2 & p_2 \\
\hline
\end{array}$

$\begin{array}{|c|c|c|}
\hline
 \mathbf{A} & \mathbf{C} & \mathbf{P} \\
\hline
 a_1 & c_1 & q_1 \\
 a_1 & c_2 & q_2 \\
 a_2 & c_3 & q_3 \\
 a_2 & c_4 & q_4 \\
 a_2 & c_5 & q_5 \\
\hline
\end{array}$

$A | P$

$\begin{array}{|c|c|}
\hline
 a_1 & 1-(1-q_1)(1-q_2) \\
 a_2 & 1-(1-q_3)(1-q_4)(1-q_5) \\
\hline
\end{array}$
\[ q = \text{Join} \] 

\[ \text{Join} \] 

\[ R^p(\text{A, B}) \] 

\[ S^p(\text{A, C}) \] 

\[ P(q) = 1 - (1-p_1(1-(1-q_1)(1-q_2))) * (1-p_2(1-(1-q_3)(1-q_4)(1-q_5))) \] 

\[ \begin{array}{|c|c|c|c|}
\hline
\text{A} & \text{P} & \\
\hline
a_1 & p_1(1-(1-q_1)(1-q_2)) & \\
\hline
a_2 & p_2(1-(1-q_3)(1-q_4)(1-q_5)) & \\
\hline
\end{array} \] 

\[ \begin{array}{|c|c|c|c|}
\hline
\text{A} & \text{C} & \text{P} & \\
\hline
a_1 & c_1 & q_1 & \\
\hline
a_1 & c_2 & q_2 & \\
\hline
a_2 & c_3 & q_3 & \\
\hline
a_2 & c_4 & q_4 & \\
\hline
a_2 & c_5 & q_5 & \\
\hline
\end{array} \]
The data complexity of this query is PTIME.
Dichotomy Theorem

Let q be a conjunctive query without self-joins

**Theorem** One of the following holds:

1. Either q is in PTIME
2. Or q is #P hard

In Case (1) q can be computed by a “safe plan” and we call it a “safe query”
PTIME Queries \#P-Hard Queries
How do we decide if a query is in PTIME or #P hard?
Hierarchical Queries

$sg(x)$ = set of subgoals containing the variable $x$ in key positions

**Definition**  
A query $q$ is *hierarchical* if for all $x, y$:

$$sg(x) \supseteq sg(y) \text{ or } sg(x) \subseteq sg(y) \text{ or } sg(x) \cap sg(y) = \emptyset$$
Hierarchical Queries

\[ \text{sg}(x) = \text{set of subgoals containing the variable } x \text{ in key positions} \]

**Definition** A query \( q \) is *hierarchical* if for all \( x, y \):

\[ \text{sg}(x) \supseteq \text{sg}(y) \text{ or } \text{sg}(x) \subseteq \text{sg}(y) \text{ or } \text{sg}(x) \cap \text{sg}(y) = \emptyset \]

**Hierarchical**

\[ q = R(x, y), S(x, z) \]

**Non-hierarchical**

\[ h1 = R(x), S(x, y), T(y) \]
Case 1: Independent Tuples Only

PTIME Queries:

**Fact** If \( q \) is hierarchical then \( q \) is in \( \text{PTIME} \)
Case 1: Independent Tuples Only

PTIME Queries:

Fact If q is hierarchical then q is in PTIME

\[ q = R(x, y), S(x, z) \]

The hierarchy gives the safe plan!

1. Root variable u \( \rightarrow \Pi_x \)
2. Connected components \( \rightarrow \text{Join} \)
Case 1: Independent Tuples Only

#P-hard Queries:

Recall: \( h_1 = R(x), S(x, y), T(y) \)

\( h_1 \) is #P-hard (reduction from Partitioned Positive 2DNF)

**Fact** If \( q \) is non-hierarchical then it is #P-hard.

Proof: it “contains” \( h_1 \):

\( q = \ldots R(x, \ldots), S(x, y, \ldots), T(y, \ldots) \ldots \)
Case 1: Independent Tuples Only

#P-hard Queries:

Recall: \[ h_1 = R(x), S(x, y), T(y) \]

\( h_1 \) is #P-hard (reduction from Partitioned Positive 2DNF)

**Fact** If \( q \) is non-hierarchical then it is #P-hard.

Proof: it “contains” \( h_1 \):
\[ q = \ldots R(x, \ldots), S(x, y, \ldots), T(y, \ldots) \ldots \]

**Theorem** Testing if \( q \) is PTIME or #P-hard is in AC^0
Case 2: Independent/disjoint Tuples

PTIME Queries:

1. Root variable $\rightarrow \Pi^I$
2. CC’s $\rightarrow$ Join
3. Constant key attrs $\rightarrow \Pi^D$
Case 2: Independent/disjoint Tuples

PTIME Queries:

\[ R(x), S(x, y), T(y), U(u, y), W('a', u) \]

1. Root variable \( \rightarrow \Pi^I \)
2. CC’s \( \rightarrow \) Join
3. Constant key attrs \( \rightarrow \Pi^D \)

Diagram:

- \( \Pi_{-u}^D \)
- \( \Pi_{-y}^D \)
- \( W^p('a', u) \)
- \( \Pi_{-x}^I \)
- \( T^p(y) \)
- \( U^p(u, y) \)
- \( R^p(x) \)
- \( S^p(x, y) \)
- \( \Pi_u \)
- \( \Pi_y \)
- \( \text{Join}_x \)
- \( \text{Join}_y \)
- \( \text{Join}_u \)

Disjoint project
Case 2: Independent/disjoint Tuples

PTIME Queries:

1. Root variable $\Rightarrow \Pi^I$
2. CC’s $\Rightarrow$ Join
3. Constant key attrs $\Rightarrow \Pi^D$
Case 2: Independent/disjoint Tuples

PTIME Queries:

1. Root variable $\Rightarrow \Pi^l$
2. CC's $\Rightarrow$ Join
3. Constant key attrs $\Rightarrow \Pi^D$

$R(x), S(x, y), T(y), U(u, y), W('a', u)$
Case 2: Independent/disjoint Tuples

#P-hard Queries:

Recall:

\[ h_1 = R(x), S(x, y), T(y) \]
\[ h_2 = R(x, y), S(y) \]
\[ h_3 = R(x, y), S(x, y) \]

#P-hard by reduction from PERMANENT

If the safe-plan algorithm fails on q, then q can be “rewritten” to either \( h_1 \) or \( h_2 \) or \( h_3 \) and hence is #P-hard (see paper for details)

**Theorem** Testing if q is PTIME or #P-hard is PTIME complete
Summary on Query Evaluation

We understand completely only queries w/o self-joins

Lessons learned from our system MystiQ:

• When the query is safe:
  – Evaluate it exactly, in the database engine
  – Performance: close to regular SQL

• When the query is unsafe
  – Approximate it, compute only top-k
  – Performance: one or two orders of magnitude worse

[Re’2007]
Outline

• Data model

• Query evaluation

• Challenges
Query Optimization

Even a \#P-hard query often has subqueries that are in PTIME. Needed:

- Combine safe plans + probabilistic inference
- “Interesting independence/disjointness”
- Model a probabilistic engine as black-box

CHALLENGE: Integrate a black-box probabilistic inference in a query processor.
Probabilistic Inference Algorithms

Open the box! Logical to physical
Examine specific algorithms from KR:
• Variable elimination
• Junction trees
• Bounded treewidth

CHALLENGE: (1) Study the space of optimization alternatives. (2) Estimate the cost of specific probabilistic inference algorithms.
Open Theory Problems

• Self-joins are much harder to study
  – Solved only for independent tuples

• Extend to richer query language
  – Unions, predicates (<, ≤, ≠), aggregates

• Do hardness results still hold for Pr = 1/2?

CHALLENGE: Complete the analysis of the query complexity over probabilistic databases
Complex Probabilistic Model

• Independent and disjoint tuples are insufficient for real applications
• Capturing complex correlations:
  – Lineage
  – Graphical models

CHALLENGE: Explore the connection between complex models and views

[Das Sarma’06, Benjelloum’06]
[Getoor’06, Sen&Deshpande’07]
[Verma&Pearl’1990]
Constraints

Needed to clean uncertainties in the data

• Hard constraints:
  – Semantics = conditional probability

• Soft constraints:
  – What is the semantics?

Lots of prior work, but still little understood

CHALLENGE: Study the impact of hard/soft constraints on query evaluation
Information Leakage

A view V should not leak information about a secret S

\[ P(S) \approx P(S \mid V) \]

- Issues: Which prior P? What is \( \approx \)?

**Probability Logic:**

- \( U \Rightarrow V \) means \( P(V \mid U) \approx 1 \)

**CHALLENGE:** Define a probability logic for reasoning about information leakage

[Evfimievski’03, Miklau&S’04, DMS’05]

[Pearl’88, Adams’98]
Conclusions

• Prohibitive cost of cleaning data

• Represent uncertainties explicitly

• Need to re-examine many assumptions
Conclusions

• Prohibitive cost of cleaning data

• Represent uncertainties explicitly

• Need to re-examine many assumptions

A call to arms:

*The management of probabilistic data*