An Overview of Probabilistic Databases

Gerome Miklau

645 Spring 2010

Based on a tutorial by Dan Suciu
A good overview paper:

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 .... and much recent work.
Outline

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

• Text extraction & record linkage
• 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 the 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>

Find people in State=WA  ⇒  Dalvi

Find people in State=MA  ⇒  ∅

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>

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

\[ Q = \text{SELECT } * \]

FROM \( R \)

WHERE \( A1=v1 \) AND \ldots \) AND \( Am=vm \)

Query is a vector: \( Q = (v1, \ldots, vm) \)

Tuple is a vector: \( T = (u1, \ldots, um) \)

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

Includes partial matches
Keyword Search in Databases

Q = ‘Abiteboul’ and ‘Widom’

Join sequences (tuple trees):

[Abiteboul, 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: int, char(30), varchar(55), datetime
- Relation schema: Employee(name:varchar(55), dob:datetime, salary:int)
- Database schema: Employee(...), Projects(...), Groups(...)

- 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 \( \text{INST} \)

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

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

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

<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>

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

<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>

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

<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>

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

<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>

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

Negatively correlated

**Independent**

Positively correlated

Identical

\[
\begin{align*}
\text{Disjoint} & : \Pr(t_1, t_2) = 0 \\
\text{Negatively correlated} & : \Pr(t_1, t_2) < \Pr(t_1) \Pr(t_2) \\
\text{Independent} & : \Pr(t_1, t_2) = \Pr(t_1) \Pr(t_2) \\
\text{Positively correlated} & : \Pr(t_1, t_2) > \Pr(t_1) \Pr(t_2) \\
\text{Identical} & : \Pr(t_1, t_2) = \Pr(t_1) = \Pr(t_2)
\end{align*}
\]
Example

\[ p = \]

Pr(I_1) = 1/3

<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>

Pr(I_2) = 1/12

<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>

Pr(I_3) = 1/2

<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>

Pr(I_4) = 1/12

<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 \textit{sets of tuples}

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

Semantics 2: Possible Tuples
A probability function on \textit{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>

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

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

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

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

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

**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>

**Possible tuples** semantics:

<table>
<thead>
<tr>
<th>Tuple</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>11/12</td>
</tr>
<tr>
<td>Gizmo</td>
<td>5/12</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. *An important 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

$$\Pr(I) = \prod_{t \in I} \text{pr}(t) \times \prod_{t \notin I} (1 - \text{pr}(t))$$

$$\text{INST} = \mathbb{P}(\text{TUP})$$
$$N = 2^M$$

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

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

No restrictions
## Tuple Prob. → Possible Worlds

$$J = \begin{array}{|c|c|}
\hline
\text{Name} & \text{City} \\
\hline
\text{John} & \text{Seattle} \\
\text{Sue} & \text{Boston} \\
\text{Fred} & \text{Boston} \\
\hline
\end{array}$$

$$\sum = 1$$

$$|p| = (1-p_1)(1-p_2)p_1(1-p_2)(1-p_3)$$

$$p_1 = 0.8$$

$$p_2 = 0.6$$

$$p_3 = 0.9$$
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>p₁</td>
</tr>
<tr>
<td>Sue</td>
<td>Boston</td>
<td>p₂</td>
</tr>
<tr>
<td>Fred</td>
<td>Boston</td>
<td>p₃</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>q₁</td>
</tr>
<tr>
<td>John</td>
<td>Gadget</td>
<td>...</td>
<td>q₂</td>
</tr>
<tr>
<td>John</td>
<td>Gadget</td>
<td>...</td>
<td>q₃</td>
</tr>
<tr>
<td>Sue</td>
<td>Camera</td>
<td>...</td>
<td>q₄</td>
</tr>
<tr>
<td>Sue</td>
<td>Gadget</td>
<td>...</td>
<td>q₅</td>
</tr>
<tr>
<td>Sue</td>
<td>Gadget</td>
<td>...</td>
<td>q₆</td>
</tr>
<tr>
<td>Fred</td>
<td>Gadget</td>
<td>...</td>
<td>q₇</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Tuple</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seattle</td>
<td>( p₁(1-(1-q₂)(1-q₃)) )</td>
</tr>
<tr>
<td>Boston</td>
<td>( 1-(1-p₂(1-(1-q₅)(1-q₆)))(1-p₃q₇) )</td>
</tr>
</tbody>
</table>
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 $\Rightarrow$ Possible Worlds

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

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

\[ |p| = \begin{array}{cccccccc}
\emptyset & \text{John} & \text{Seattle} & \text{Sue} & \text{Denver} & \text{John} & \text{Seattle} & \text{Sue} & \text{Denver} \\
\hline
(1-p_1)(1-p_2)(1-p_3) & p_1(1-p_2)p_3 & (1-p_1)p_2p_3 & & & p_1p_2(1-p_3) + p_1p_2p_3 \\
\hline
\end{array} \]
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>John</td>
<td>Boston</td>
</tr>
</tbody>
</table>

$E_1 = e_1$
$E_2 = \neg e_1 \land e_2$
$E_3 = \neg e_1 \land \neg e_2 \land e_3$
$E_4 = \neg e_1 \land \neg e_2 \land \neg e_3 \land e_4$

$\Pr(e_1) = p_1$
$\Pr(e_2) = p_2/(1-p_1)$
$\Pr(e_3) = p_3/(1-p_1-p_2)$
$\Pr(e_4) = p_4/(1-p_1-p_2-p_3)$

“Prefix code”

$J = \{E_1 \lor E_2, E_1 \lor E_4, E_1 \lor E_2 \lor E_3\}$

Intensional DBs are complete
Closure Under Operators

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 [Imlielinski&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

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

**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:

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

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>( p_1(1-p_2)p_3 )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( p_1p_2(1-p_3) )</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>( p_1(1-p_2)p_3 )</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>

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

Seems inefficient in general…
Theorem [Valiant:1979]
For a boolean expression $E$, computing $\Pr(E)$ is $\#P$-complete.

NP = 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

One still needs to compute probability of event expression
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'

[Jalvi&S:2004]

Wrong!

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

Correct

Jon | Sea | p_1(1-(1-q_1)(1-q_2)(1-q_3))
Query Complexity

Sometimes no correct extensional plan

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

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})

Probabilistic databases have high query complexity