Introduction to Probabilistic Data Management

Evgeny Kharlamov     Pierre Senellart

Bases de données avancées, October 26, 2011
Part I: Uncertainty in the Real World
Uncertain data

Numerous sources of uncertain data:

- Measurement errors
- Data integration from contradicting sources
- Imprecise mappings between heterogeneous schemata
- Imprecise automatic process (information extraction, natural language processing, etc.)
- Imperfect human judgment
Uncertain data

Numerous sources of uncertain data

- Measurement errors
- Data integration from contradicting sources
- Imprecise mappings between heterogeneous schemata
- Imprecise automatic process (information extraction, natural language processing, etc.)
- Imperfect human judgment
Use case: Web information extraction

<table>
<thead>
<tr>
<th>instance</th>
<th>iteration</th>
<th>date learned</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>arabic, egypt</td>
<td>406</td>
<td>08-sep-2011</td>
<td>(Seed) 100.0</td>
</tr>
<tr>
<td>chinese, republic_of_china</td>
<td>439</td>
<td>24-oct-2011</td>
<td>100.0</td>
</tr>
<tr>
<td>chinese, singapore</td>
<td>421</td>
<td>21-sep-2011</td>
<td>(Seed) 100.0</td>
</tr>
<tr>
<td>english, britain</td>
<td>439</td>
<td>24-oct-2011</td>
<td>100.0</td>
</tr>
<tr>
<td>english, canada</td>
<td>439</td>
<td>24-oct-2011</td>
<td>(Seed) 100.0</td>
</tr>
<tr>
<td>english, england001</td>
<td>439</td>
<td>24-oct-2011</td>
<td>100.0</td>
</tr>
<tr>
<td>arabic, morocco</td>
<td>422</td>
<td>23-sep-2011</td>
<td>100.0</td>
</tr>
<tr>
<td>cantonese, hong_kong</td>
<td>406</td>
<td>08-sep-2011</td>
<td>100.0</td>
</tr>
<tr>
<td>english, uk</td>
<td>436</td>
<td>19-oct-2011</td>
<td>100.0</td>
</tr>
<tr>
<td>english, south_vietnam</td>
<td>427</td>
<td>27-sep-2011</td>
<td>99.9</td>
</tr>
<tr>
<td>french, morocco</td>
<td>422</td>
<td>23-sep-2011</td>
<td>99.9</td>
</tr>
<tr>
<td>greek, turkey</td>
<td>430</td>
<td>07-oct-2011</td>
<td>99.9</td>
</tr>
</tbody>
</table>

Never-ending Language Learning (NELL, CMU),
http://rtw.ml.cmu.edu/rtw/kbbrowser/
**Use case: Web information extraction**

**Google Squared (terminated), screenshot from**

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Language</th>
<th>Director</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Mask</td>
<td>English</td>
<td>Chuck Russell</td>
<td>29 July 1994</td>
</tr>
<tr>
<td>Scary Movie</td>
<td>English</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Superbad</td>
<td>English</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Music</td>
<td>English, French</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knocked Up</td>
<td>Italian Language</td>
<td>Low confidence</td>
<td></td>
</tr>
</tbody>
</table>

Other possible values:

- English Language
- Low confidence
- English language for the mask
  - www.infibeam.com
- All 9 sources »

- Chuck Russell
- Directed by for The Mask
  - www.infibeam.com
- All 9 sources »

- John R. Dilworth
- Low confidence
- Director for The Mask
  - www.freebase.com

- Fiorella Infascelli
- Low confidence
- Directed by for The Mask
  - www.freebase.com
  - All 2 sources »

- Charles Russell
- Low confidence
- Directed by for The Mask
  - www.freebase.com
  - All 2 sources »

[Search for more values »]
### Use case: Web information extraction

<table>
<thead>
<tr>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elvis Presley</td>
<td>diedOnDate</td>
<td>1977-08-16</td>
<td>97.91%</td>
</tr>
<tr>
<td>Elvis Presley</td>
<td>isMarriedTo</td>
<td>Priscilla Presley</td>
<td>97.29%</td>
</tr>
<tr>
<td>Elvis Presley</td>
<td>influences</td>
<td>Carlo Wolff</td>
<td>96.25%</td>
</tr>
</tbody>
</table>

Uncertainty in Web information extraction

- The information extraction system is **imprecise**.
- The system has some **confidence** in the information extracted, which can be:
  - a **probability** of the information being true (e.g., conditional random fields)
  - an **ad-hoc** numeric confidence score
  - a **discrete** level of confidence (low, medium, high)

- What if this uncertain information is not seen as something final, but is used as a source of, e.g., a query answering system?
Different types of uncertainty

Two dimensions:

- Different types:
  - Unknown value: NULL in an RDBMS
  - Alternative between several possibilities: either A or B or C
  - Imprecision on a numeric value: a sensor gives a value that is an approximation of the actual value
  - Confidence in a fact as a whole: cf. information extraction
  - Structural uncertainty: the schema of the data itself is uncertain

- Qualitative (NULL) or Quantitative (95%, low-confidence, etc.) uncertainty
Managing uncertainty

Objective

Not to pretend this imprecision does not exist, and manage it as rigorously as possible throughout a long, automatic and human, potentially complex, process.
Managing uncertainty

Objective

Not to pretend this imprecision does not exist, and manage it as rigorously as possible throughout a long, automatic and human, potentially complex, process.

Especially:

- Represent all different forms of uncertainty
- Use probabilities to represent quantitative information on the confidence in the data
- Query data and retrieve uncertain results
- Allow adding, deleting, modifying data in an uncertain way
- Bonus (if possible): Keep as well lineage/provenance information, so as to ensure traceability
Why probabilities?

- Not the only option: fuzzy set theory [], Dempster-Shafer theory []
- Mathematically rich theory, nice semantics with respect to traditional database operations (e.g., joins)
- Some applications already generate probabilities (e.g., statistical information extraction or natural language probabilities)
- In other cases, we “cheat” and pretend that (normalized) confidence scores are probabilities: see this as a first-order approximation
Objective of this tutorial

- Present **data models** for uncertain data management in general, and probabilistic data management in particular:
  - relational
  - XML
- Show how these models can be **queried**: algorithms, complexity, approximation techniques...
- Discuss the problem of updating a probabilistic database
Part II: Probabilistic Models of Uncertainty

- Probabilistic Relational Models
- Probabilistic XML
Possible worlds semantics

Possible world: A regular (deterministic) relational or XML database

Incomplete database: (Compact) representation of a set of possible worlds

Probabilistic database: (Compact) representation of a probability distribution over possible worlds either:
  - finite: a set of possible worlds, each with their probability
  - continuous: more complicated, requires defining a $\sigma$-algebra, and a measure for the sets of this $\sigma$-algebra
Part II: Probabilistic Models of Uncertainty

- Probabilistic Relational Models
- Probabilistic XML
The relational model

- Data stored into **tables**
- Every table has a precise **schema** (**type of columns**)
- Adapted when the information is very **structured**

<table>
<thead>
<tr>
<th>Patient</th>
<th>Examin. 1</th>
<th>Examin. 2</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>23</td>
<td>12</td>
<td>α</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>23</td>
<td>β</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4</td>
<td>γ</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>15</td>
<td>α</td>
</tr>
<tr>
<td>E</td>
<td>15</td>
<td>17</td>
<td>β</td>
</tr>
</tbody>
</table>
Codd tables, a.k.a. SQL NULLs

<table>
<thead>
<tr>
<th>Patient</th>
<th>Examin. 1</th>
<th>Examin. 2</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>23</td>
<td>12</td>
<td>(\alpha)</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>23</td>
<td>(\perp_1)</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4</td>
<td>(\gamma)</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>15</td>
<td>(\perp_2)</td>
</tr>
<tr>
<td>E</td>
<td>(\perp_3)</td>
<td>17</td>
<td>(\beta)</td>
</tr>
</tbody>
</table>

- Most **simple** form of incomplete database
- **Widely used** in practice, in DBMS since the mid-1970s!
- All NULLs (\(\perp\)) are considered **distinct**
- Possible world semantics: all (infinitely many under the **open world** assumption) possible completions of the table
- In SQL, **three-valued logic**, weird semantics:
  
  SELECT * FROM Tel WHERE tel_nr = ’333’ OR tel_nr <> ’333’
## C-tables

NULLs are labeled, and can be reused inside and across tuples

Arbitrary correlations across tuples

Closed under the relational algebra (Codd tables only closed under projection and union)

Every set of possible worlds can be represented as a database with c-tables

<table>
<thead>
<tr>
<th>Patient</th>
<th>Examin. 1</th>
<th>Examin. 2</th>
<th>Diagnosis</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>23</td>
<td>12</td>
<td>$\alpha$</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>23</td>
<td>$\perp_1$</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4</td>
<td>$\gamma$</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>$\perp_2$</td>
<td>15</td>
<td>$\perp_1$</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>$\perp_3$</td>
<td>17</td>
<td>$\beta$</td>
<td>18 &lt; $\perp_3$ &lt; $\perp_2$</td>
</tr>
</tbody>
</table>
## Tuple-independent databases (TIDs)

<table>
<thead>
<tr>
<th>Patient</th>
<th>Examin. 1</th>
<th>Examin. 2</th>
<th>Diagnosis</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>23</td>
<td>12</td>
<td>$\alpha$</td>
<td>0.9</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>23</td>
<td>$\beta$</td>
<td>0.8</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4</td>
<td>$\gamma$</td>
<td>0.2</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>14</td>
<td>$\gamma$</td>
<td>0.4</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>15</td>
<td>$\alpha$</td>
<td>0.6</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>15</td>
<td>$\beta$</td>
<td>0.4</td>
</tr>
<tr>
<td>E</td>
<td>15</td>
<td>17</td>
<td>$\beta$</td>
<td>0.7</td>
</tr>
<tr>
<td>E</td>
<td>15</td>
<td>17</td>
<td>$\alpha$</td>
<td>0.3</td>
</tr>
</tbody>
</table>

- Allow representation of the **confidence** in each row of the table
- Impossible to express **dependencies** across rows
- Very simple model, well understood
Block-independent databases (BIDs)

<table>
<thead>
<tr>
<th>Patient</th>
<th>Examin. 1</th>
<th>Examin. 2</th>
<th>Diagnosis</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>23</td>
<td>12</td>
<td>α</td>
<td>0.9</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>23</td>
<td>β</td>
<td>0.8</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4</td>
<td>γ</td>
<td>0.2</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>14</td>
<td>γ</td>
<td>0.4</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>15</td>
<td>β</td>
<td>0.6</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>15</td>
<td>α</td>
<td>0.4</td>
</tr>
<tr>
<td>E</td>
<td>15</td>
<td>17</td>
<td>β</td>
<td>0.7</td>
</tr>
<tr>
<td>E</td>
<td>15</td>
<td>17</td>
<td>α</td>
<td>0.3</td>
</tr>
</tbody>
</table>

- The table has a **primary key** tuples sharing a primary key are mutually exclusive (probabilities must sum up to \(1\))
- Simple dependencies (exclusion) can be expressed, but not more complex ones
## Probabilistic c-tables

<table>
<thead>
<tr>
<th>Patient</th>
<th>Examin. 1</th>
<th>Examin. 2</th>
<th>Diagnosis</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>23</td>
<td>12</td>
<td>α</td>
<td>$w_1$</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>23</td>
<td>β</td>
<td>$w_2$</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4</td>
<td>γ</td>
<td>$w_3$</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>14</td>
<td>γ</td>
<td>$\neg w_3 \land w_4$</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>15</td>
<td>β</td>
<td>$w_5$</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>15</td>
<td>α</td>
<td>$\neg w_5 \land w_6$</td>
</tr>
<tr>
<td>E</td>
<td>15</td>
<td>17</td>
<td>β</td>
<td>$w_7$</td>
</tr>
<tr>
<td>E</td>
<td>15</td>
<td>17</td>
<td>α</td>
<td>$\neg w_7$</td>
</tr>
</tbody>
</table>

- The $w_i$’s are **Boolean random variables**
- Each $w_i$ has a probability of being true (e.g., $Pr(w_1) = 0.9$)
- The $w_i$’s are independent
- Any **finite** probability distribution of tables can be represented using probabilistic c-tables
Two actual PRDBMS: Trio and MayBMS

Two main probabilistic relational DBMS:

Trio  [?] Various uncertainty operators unknown value, uncertain tuple, choice between different possible values, with probabilistic annotations. See example later on.

MayBMS  [?] Implementation of the probabilistic c-tables model. In addition, uncertain tables can be constructed using a REPAIR-KEY operator, similar to BIDs.
Two actual PRDBMS: Trio and MayBMS

Trio

Various uncertainty operators: unknown value, uncertain tuple, choice between different possible values, with probabilistic annotations. See example later on.

MayBMS

Implementation of the probabilistic c-tables model. In addition, uncertain tables can be constructed using a REPAIR-KEY operator, similar to BIDs.

```sql
test=# select * from R;
dummy | weather | ground | p
-------+---------+---------+---------
dummy | rain    | wet     | 0.35    
dummy | rain    | dry     | 0.05    
dummy | no rain | wet     | 0.1     
dummy | no rain | dry     | 0.5     
(4 rows)
```

```sql
test=# create table S as
    repair key Dummy in R weight by P;
SELECT
```

```sql
test=# select Ground, conf() from S group by Ground;
ground | conf
-------+------
dry    | 0.55 
et     | 0.45 
(2 rows)
```
Part II: Probabilistic Models of Uncertainty

- Probabilistic Relational Models
- Probabilistic XML
The semistructured model and XML

- Tree-like structuring of data
- No (or less) schema constraints
- Allow mixing tags (structured data) and text (unstructured content)
- Particularly adapted to tagged or heterogeneous content
Why Probabilistic XML?

- Extensive literature about probabilistic relational databases
- Different typical querying languages: conjunctive queries vs tree-pattern queries (possibly with joins)
- Cases where a tree-like model might be appropriate:
  - No schema or few constraints on the schema
  - Independent modules annotating freely a content warehouse
  - Inherently tree-like data (e.g., mailing lists, parse trees) with naturally occurring queries involving the descendant axis

Remark

Some results can be transferred from one model to the other. In other cases, connection much trickier!
Models all XML documents where these patterns exist (i.e., this subtree can be matched)

Can be used for query answering, etc.
Simple probabilistic annotations

- Probabilities associated to tree nodes
- Express parent/child dependencies
- Impossible to express more complex dependencies
- \( \Rightarrow \) some sets of possible worlds are not expressible this way!
Annotations with event variables

\[ w_1, \neg w_2 \]

\[ w_1 \quad \begin{array}{c} A \\ B \quad C \\ D \end{array} \]

\[ w_2 \]

<table>
<thead>
<tr>
<th>Event</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_1 )</td>
<td>0.8</td>
</tr>
<tr>
<td>( w_2 )</td>
<td>0.7</td>
</tr>
</tbody>
</table>
Annotations with event variables

<table>
<thead>
<tr>
<th>Event</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>0.8</td>
</tr>
<tr>
<td>$w_2$</td>
<td>0.7</td>
</tr>
</tbody>
</table>

$p_1 = 0.06$  
$p_2 = 0.70$  
$p_3 = 0.24$

- Expresses **arbitrarily complex** dependencies
- Obviously, analogous to probabilistic c-tables
A general probabilistic XML model

- e: event “it did not rain” at time 1
- mux: mutually exclusive options
- $N(70, 4)$: normal distribution

- Compact representation of a set of possible worlds
- Two kinds of dependencies: global (e) and local (mux)
- Generalizes all previously proposed models of the literature
Recursive Markov chains

<!ELEMENT directory (person*)>
<!ELEMENT person (name,phone*)>

- Probabilistic model that extends PXML with local dependencies
- Allows generating documents of unbounded width or depth
Part III: Querying Probabilistic Databases
Algorithm for TP over local dependencies

Bottom-up dynamic programming algorithm. Query: /A//B

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$D_2$</th>
<th>$\text{mux}_3$</th>
<th>$B_4$</th>
<th>$C_5$</th>
<th>$B_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>/B</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>//B</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/A//B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

mux  convex sum

ordinary inclusion-exclusion
Algorithm for TP over local dependencies

Bottom-up dynamic programming algorithm. Query: /A//B

<table>
<thead>
<tr>
<th></th>
<th>A₁</th>
<th>D₂</th>
<th>mux₃</th>
<th>B₄</th>
<th>C₅</th>
<th>B₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>/B</td>
<td>0.3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>//B</td>
<td>0.3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>/A//B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

mux  convex sum
ordinary inclusion-exclusion
Algorithm for TP over local dependencies

Bottom-up dynamic programming algorithm. Query: /A//B

Pr(D₂ | /B) = 1 - (1 - 0.8 × Pr(mux₃ | /B)) × (1 - 0.6 × Pr(B₆ | /B))
= 1 - (1 - 0.8 × 0.3) × (1 - 0.6) = 0.696
Algorithm for TP over local dependencies

Bottom-up dynamic programming algorithm. Query: /A//B

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$D_2$</th>
<th>$\text{mux}_3$</th>
<th>$B_4$</th>
<th>$C_5$</th>
<th>$B_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>/B</td>
<td>0</td>
<td>0.696</td>
<td>0.3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>//B</td>
<td>0.696</td>
<td>0.696</td>
<td>0.3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>/A//B</td>
<td>0.696</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

mux convex sum
ordinary inclusion-exclusion
Part V: To go further
Systems

Trio  http://infolab.stanford.edu/trio/, useful to see lineage computation

MayBMS  http://maybms.sourceforge.net/, full-fledged probabilistic relational DBMS, on top of PostgreSQL, usable for actual applications.

ProApproX  http://www.infres.enst.fr/~souihli/Publications.html to play with various approximation and exact query evaluation methods for probabilistic XML.
Reading material

- An influential paper on incomplete databases
- A book on probabilistic relational databases, focused around TIDs/BIDs and MayBMS
- An in-depth presentation of MayBMS
- A gentle presentation of relational and XML probabilistic models
- A survey of probabilistic XML
Research directions

- Demonstrating the usefulness of probabilistic databases over ad-hoc approach on concrete applications Web information extraction, data warehousing, scientific data management, etc.
- Understanding better the connection between probabilistic relational databases and probabilistic XML why does the picture look so different?
- Understanding under which restrictions on the data (e.g., (hyper)tree-width characteristics) query answering can be tractable.
- Connecting probabilistic databases with probabilistic models in general, e.g., as used in machine learning: Bayesian networks, Makov logic networks, factor graphs, etc.
- Other operations on probabilistic data: mining, deduplication, learning, matching, etc.