Probabilistic XML: A Data Model for the Web

Pierre Senellart

Habilitation à diriger les recherches, 13 juin 2012
An Uncertain World

TRUE...FALSE...
TRUE...TRUE...
FALSE...TRUE...
MA'AM?
WHAT DO WE DO IF WE COME ACROSS A HALF-TRUTH?
Outline

An Uncertain World
Uncertainty is Everywhere
Probabilistic Databases
Probabilistic XML Applications

Probabilistic XML: Models and Complexity

Reconciling Web Data Models with the Actual Web
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, graph mining, etc.)
- Imperfect human judgment
- Lies, opinions, rumors
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, graph mining, etc.)
- Imperfect human judgment
- Lies, opinions, rumors
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 [Fink et al., 2011]
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>

YAGO, http://www.mpi-inf.mpg.de/yago-naga/yago [Suchanek et al., 2007]
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 rough 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?
Outline

An Uncertain World
  Uncertainty is Everywhere
  Probabilistic Databases
  Probabilistic XML Applications

Probabilistic XML: Models and Complexity

Reconciling Web Data Models with the Actual Web
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 different forms of uncertainty
- Probabilities are used to measure uncertainty in the data
- Query data and retrieve uncertain results
- Allow adding, deleting, modifying data in an uncertain way
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 **different forms** of uncertainty
- **Probabilities** are used to measure uncertainty in the data
- Query data and retrieve **uncertain** results
- Allow adding, deleting, modifying data in an **uncertain** way
Why probabilities?

- Not the only option: fuzzy set theory [Galindo et al., 2005], Dempster-Shafer theory [Zadeh, 1986]
- 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)
- Naturally arising in case of conflicting information, based on the trust in the sources
- In other cases, we “cheat” and pretend that (normalized) confidence scores are probabilities: see this as a first-order approximation
Probabilistic relational DBMSs

Numerous studies on the modeling and querying of probabilistic relations:

- block-independent disjoint (BIDs) databases [Dalvi et al., 2009] tuples either mutually exclusive or independent, depending on key values

- probabilistic c-tables [Imieliński and Lipski, 1984, Green and Tannen, 2006, Senellart, 2007], tuples annotated with arbitrary probabilistic formulas

Two main probabilistic relational DBMS:

- Trio [Widom, 2005] Various uncertainty operators: unknown value, uncertain tuple, choice between different possible values, with probabilistic annotations

- MayBMS [Koch, 2009] Uncertain tables can be constructed using a REPAIR-KEY operator, similar to BIDs, and more elaborate correlations can result from querying
Probabilistic relational DBMSs

Numerous studies on the modeling and querying of probabilistic relations:

block-independent disjoint (BIDs) databases [Dalvi et al., 2009] tuples either mutually exclusive or independent, depending on key values

probabilistic c-tables [Imieliński and Lipski, 1984, Green and Tannen, 2006, Senellart, 2007], tuples annotated with arbitrary probabilistic formulas

Two main probabilistic relational DBMS:

Trio [Widom, 2005] Various uncertainty operators: unknown value, uncertain tuple, choice between different possible values, with probabilistic annotations

MayBMS [Koch, 2009] Uncertain tables can be constructed using a REPAIR-KEY operator, similar to BIDs, and more elaborate correlations can result from querying
Probabilistic relational DBMSs

Numerous studies on the modeling and querying of probabilistic relations:

- **block-independent disjoint (BIDs) databases**
  - tuples either mutually exclusive or independent, depending on key values

- Probabilistic c-tables
  - tuples annotated with arbitrary probabilistic formulas

Two main probabilistic relational DBMS:

- **Trio** [Widom, 2005]
  - Various uncertainty operators: unknown value, uncertain tuple, choice between different possible values, with probabilistic annotations

- **MayBMS** [Koch, 2009]
  - Uncertain tables can be constructed using a `REPAIR-KEY` operator, similar to BIDs, and more elaborate correlations can result from querying
Outline

An Uncertain World
Uncertainty is Everywhere
Probabilistic Databases
Probabilistic XML Applications

Probabilistic XML: Models and Complexity

Reconciling Web Data Models with the Actual Web
Why probabilistic XML?

- Different typical querying languages: SQL and conjunctive queries vs XPath and 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!
Web information extraction [Senellart et al., 2008]

- Annotate HTML Web pages with possible labels
- Labels can be learned from a corpus of annotated documents
- Conditional random fields for XML: estimate probabilities of annotations given annotations of neighboring nodes
- Provides probabilistic labeling of Web pages
Use trees with probabilistic annotations to represent the uncertainty in the correctness of a document under open version control (e.g., Wikipedia articles)
Probabilistic summaries of XML corpora

[Abiteboul et al., 2012a,b]

- Transform an XML schema (deterministic top-down tree automaton) into a probabilistic generator (probabilistic tree automaton) of XML documents.
- Probability distribution optimal with respect to a given corpus.
Probabilistic XML: Models and Complexity
Outline

An Uncertain World

Probabilistic XML: Models and Complexity
   Modeling
   Querying
   Updating

Reconciling Web Data Models with the Actual Web
A general discrete Probabilistic XML Model
[Abiteboul and Senellart, 2006, Senellart, 2007, Abiteboul et al., 2009]

Expresses \textbf{arbitrarily complex} dependencies

Analogous to probabilistic c-tables

\begin{tabular}{|c|c|}
\hline
Event & Prob. \\
\hline
$w_1$ & 0.8 \\
$w_2$ & 0.7 \\
\hline
\end{tabular}

$\mathbb{E} = \{w_1, \neg w_2\}$

\textbf{semantics}

$p_1 = 0.06$

$p_2 = 0.70$

$p_3 = 0.24$
A general discrete Probabilistic XML Model

[Abiteboul and Senellart, 2006, Senellart, 2007, Abiteboul et al., 2009]

Expresses \textit{arbitrarily complex} dependencies

Analogous to probabilistic c-tables
Continuous distributions [Abiteboul et al., 2011]

- Two kinds of dependencies in discrete probabilistic XML: global (e) and local (mux)
- Add continuous leaves; non-obvious semantics issues

- e: event “it did not rain” at time 1
- mux: mutually exclusive options
- $N(70, 4)$: normal distribution
Recursive Markov chains [Benedikt et al., 2010]

- Probabilistic model that subsumes local discrete models
- Allows generating documents of unbounded width or depth
Outline

An Uncertain World

Probabilistic XML: Models and Complexity
  Modeling
  Querying
  Updating

Reconciling Web Data Models with the Actual Web
Querying probabilistic XML

Semantics of a (Boolean) query = \textit{probability} the query is true:

1. Generate all possible worlds of a given probabilistic document

2. In each world, evaluate the query

3. Add up the probabilities of the worlds that make the query true

\textit{EXPTIME} algorithm! Can we do better, i.e., can we apply directly the algorithm on the probabilistic document?

We shall talk about data complexity of query answering.
Querying probabilistic XML

Semantics of a (Boolean) query = probability the query is true:

1. Generate all possible worlds of a given probabilistic document (possibly exponentially many)
2. In each world, evaluate the query
3. Add up the probabilities of the worlds that make the query true

**EXPTIME** algorithm! Can we do better, i.e., can we apply directly the algorithm on the probabilistic document?

We shall talk about data complexity of query answering.
Previously known results

[Kimelfeld et al., 2009, Cohen et al., 2009]

- Tree-pattern queries are evaluable in linear time over local dependencies with a bottom-up, dynamic programming algorithm.
- Indeed, any monadic second-order query is tractable over local dependencies.
- Even trivial queries are $\#P$-hard over global dependencies.
- Monte-Carlo sampling works.
- Multiplicative approximation is also tractable (existence of a FPRAS).
Aggregation queries [Abiteboul et al., 2010]

Aggregate Queries: sum, count, avg, countd, min, max, etc.
Distributions? Possible values? Expected value?

- Computing expected values of sum and count tractable with global dependencies; everything else intractable
- Computing expected values of every of these aggregate functions tractable with local dependencies
- Computing distributions and possible values tractable for count, min, max, intractable for the others

Continuous distributions do not add any more complexity!
Queries with joins are hard \cite{Kharlamov2011}

Tree-pattern queries (TP) \( /A[C/D]//B \)

Tree-pattern queries with joins (TPJ) for $x$ in $\text{doc}/A/C/D$

\[
\text{return } \text{doc}/A//B[.=$x$]
\]

Over local dependencies, dichotomy for queries with a single join:
- If equivalent to a join-free query, linear-time
- Otherwise, $\#P$-hard
Queries with joins are hard [Kharlamov et al., 2011]

Tree-pattern queries (TP) \(/A[C/D]/B\)

Tree-pattern queries with joins (TPJ) for $x$ in $doc/A/C/D$

\[
\text{return } doc/A//B[.=$x] \]

Over local dependencies, **dichotomy** for queries with a single join:
- If equivalent to a join-free query, **linear-time**
- Otherwise, **#P-hard**
Rely on approximations... when needed

- Monte-Carlo is very good at approximating high probabilities
- Sometimes the structure of a query makes the probability of a query easy to evaluate
- For small formulas, naïve evaluation techniques good enough
- Refined approximation methods best when everything else fails

⇒ Use a cost model to decide which evaluation algorithm to use!

<table>
<thead>
<tr>
<th>Average time (ms)</th>
<th>Naïve</th>
<th>Sieve</th>
<th>Monte-Carlo</th>
<th>Coverage</th>
<th>EvalDP</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4+join</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

joint work with Pierre Senellart
Rely on approximations... when needed


- Monte-Carlo is very good at approximating high probabilities
- Sometimes the structure of a query makes the probability of a query easy to evaluate
- For small formulas, naïve evaluation techniques good enough
- Refined approximation methods best when everything else fails

⇒ Use a cost model to decide which evaluation algorithm to use!

![Diagram showing average time (ms) for different algorithms and formula complexity](image)

- Naïve
- Sieve
- Monte-Carlo
- Coverage
- EvalDP
- Best

joint work with Pierre Senellart
Outline

An Uncertain World

Probabilistic XML: Models and Complexity
  Modeling
  Querying
  Updating

Reconciling Web Data Models with the Actual Web
Updates defined by a query (cf. XUpdate, XQuery Update).
Semantics: for all matches of a query, insert or delete a node in the tree at a place located by the query.

**Results**

- Most updates are **intractable** with local dependencies: the result of an update can require an exponentially larger representation size
- Insertions with a for-each-match semantics are **tractable** with arbitrary dependencies; deletions are **intractable**
- Some insert-if-there-is-a-match operations are **tractable** for local dependencies but not for arbitrary dependencies
Reconciling Web Data Models with the Actual Web

WE LOST AGAIN!

YOU SAID THERE WAS A BILLION-TO-ONE CHANCE THAT WE MIGHT WIN...

BUT WE DIDN'T! WE LOST!

BOY, YOU JUST CAN'T BELIEVE ANYONE ANY MORE!
Outline

An Uncertain World

Probabilistic XML: Models and Complexity

Reconciling Web Data Models with the Actual Web

Applying Probabilistic XML to Web data

Formal Models for Web Content Acquisition
Applying Probabilistic XML to Web data

- Current mismatch (in my research, but also in general) between:
  - Applications that produce and have to deal with uncertainty:
    - Information extraction
    - Graph mining
    - Natural language parsing
    - Data integration
    - etc.
  - Probabilistic database management systems

- Analogy to data management in the 1960’s:
  - Each data management application (accounting, cataloging, etc.) was developing its own way of managing data
  - Each application dealing with imprecise data is developing its own way of managing imprecision

- Probabilistic DBMSs are becoming more and more mature – time to put them to use for real applications?
Applying Probabilistic XML to Web data

- Current **mismatch** (in my research, but also in general) between:
  - Applications that produce and have to deal with uncertainty:
    - Information extraction
    - Graph mining
    - Natural language parsing
    - Data integration
    - etc.
  - Probabilistic database management systems

- Analogy to **data management in the 1960’s:**
  - Each data management application (accounting, cataloging, etc.) was developing its own way of managing data
  - Each application dealing with imprecise data is developing its own way of managing imprecision

- Probabilistic DBMSs are becoming more and more mature – time to put them to use for **real applications**?
Outline

An Uncertain World

Probabilistic XML: Models and Complexity

Reconciling Web Data Models with the Actual Web

Applying Probabilistic XML to Web data

Formal Models for Web Content Acquisition
Formal Models for Web Content Acquisition

- **Current mismatch** between:
  - Actual Web mining and Web data management applications:
    - Web graph mining
    - Wrapper induction
    - Web crawling
  - Research on the foundations of Web data management

- **Plan:** bridge the two
  - Use formal models for data exchange [Senellart and Gottlob, 2008, Gottlob and Senellart, 2010] to perform wrapper induction or ontology matching
  - Use static analysis techniques for JavaScript to perform deep Web data extraction [Benedikt et al., 2012b]
  - Use results of formal studies of the containment of recursive query languages [Benedikt et al., 2011, 2012a] to optimize query answering over the deep Web
Current **mismatch** between:

- Actual Web mining and Web data management applications:
  - Web graph mining
  - wrapper induction
  - Web crawling
- Research on the foundations of Web data management

**Plan: bridge the two**

- Use formal models for data exchange [Senellart and Gottlob, 2008, Gottlob and Senellart, 2010] to perform wrapper induction or ontology matching
- Use static analysis techniques for JavaScript to perform deep Web data extraction [Benedikt et al., 2012b]
- Use results of formal studies of the containment of recursive query languages [Benedikt et al., 2011, 2012a] to optimize query answering over the deep Web
Questions?


