Probabilistic Models for Uncertain Data

MPRI 2.26.2: Web Data Management

Antoine Amarilli, Pierre Senellart

Friday, January 18th

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
- Lies, opinions, rumors

Use case: Web information extraction

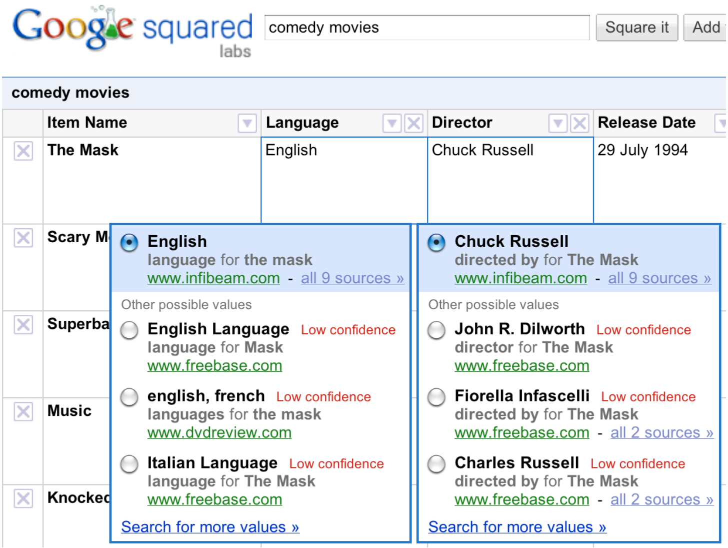
Recently-Learned Facts twitter

Refresh

instance	iteration	date learned	confidence
oliguric_phase is a non-disease physiological condition	1111	06-jul-2018	97.5
alaska_airlines is an organization	1114	25-aug-2018	100.0
heating_insurance_policies is a physical action	1111	06-jul-2018	90.4
n98_12 is a term used by physicists	1111	06-jul-2018	94.2
dragonball_z__super_butoden_2 is software	1111	06-jul-2018	100.0
general_motors_corp_ is a company headquartered in the city detroit	1116	12-sep-2018	100.0
the companies herald and la compete with eachother	1111	06-jul-2018	99.6
stanford hired montgomery	1111	06-jul-2018	98.4
kimn is a radio station in the city denver	1116	12-sep-2018	100.0
radisson_sas_portman_hotel is a park in the city central_london	1116	12-sep-2018	100.0

Never-ending Language Learning (NELL, CMU),
`http://rtw.ml.cmu.edu/rtw/kbbrowser/`

Use case: Web information extraction



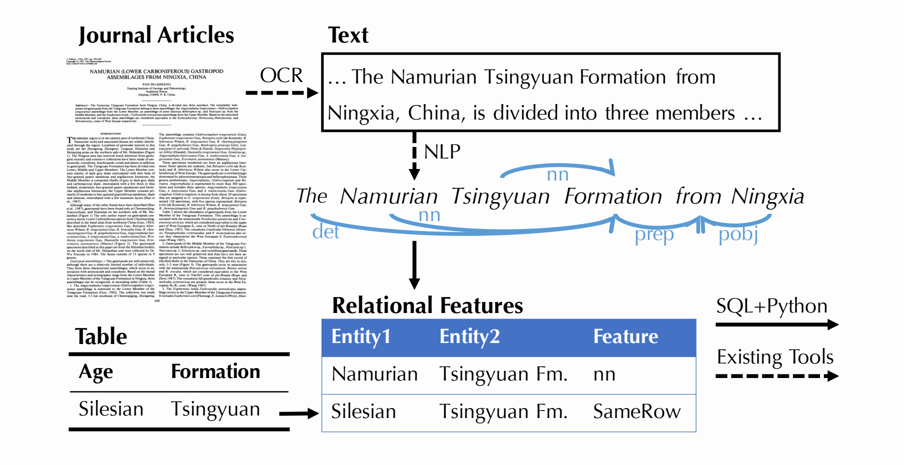
Google Squared (terminated), screenshot from [Fink et al., 2011]

Use case: Web information extraction

Subject	Predicate	Object	Confidence
Elvis Presley	diedOnDate	1977-08-16	97.91%
Elvis Presley	isMarriedTo	Priscilla Presley	97.29%
Elvis Presley	influences	Carlo Wolff	96.25%

YAGO, `https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/`

Other use case: Information extraction from scientific articles



Other use case: Crowdsourcing

All HITs

1-10 of 2751 Results

Sort by: HITs Available (most first) GO! Show all details | Hide all details 1 2 3 4 5 › Next » Last

Transcribe data View a HIT in this group

Requester: p9r **HIT Expiration Date:** Nov 18, 2015 (23 hours 59 minutes) **Reward:** $0.03

Time Allotted: 45 minutes

Description: Please transcribe the data from the following images

Keywords: transcribe, handwriting, data entry

Qualifications Required:

HIT approval rate (%) is greater than 90

Classify Receipt View a HIT in this group

Requester: Jon Brelig **HIT Expiration Date:** Nov 24, 2015 (6 days 23 hours) **Reward:** $0.02

Time Allotted: 20 minutes

Description: Looking at a receipt image, identify the business of the receipt

Keywords: image, receipt, categorize, transcribe, extract, data, entry, transcription, text, easy, qualification, jon, brelig, prod

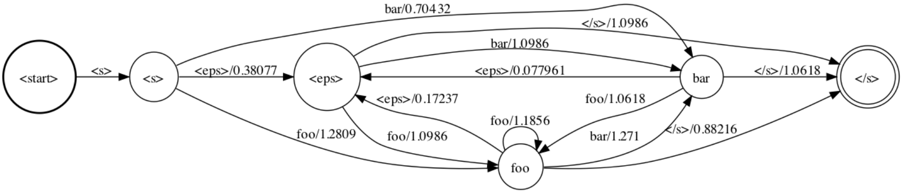
Qualifications Required:

Total approved HITs is not less than 1000

HIT approval rate (%) is not less than 97

Location is US

Other use case: Speech recognition and OCR



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., from a statistical or machine learning model)
  - 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:

- Can be qualitative (NULL) or quantitative (95%, low-confidence, etc.) uncertainty
- Different types of uncertainty:
  - 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
  - Missing data: we know that some data is missing (open-world semantics)

What happens to this uncertainty?

Currently

Forget about uncertainty, or apply a threshold after each computation step

What happens to this uncertainty?

Currently

Forget about uncertainty, or apply a threshold after each computation step

Objective

Instead of neglecting uncertainty, let's manage it rigorously throughout the whole process of answering a query

How to manage uncertainty

- 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
- Ideally: Also keep lineage/provenance information, so as to ensure traceability

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)
- In other cases, we "cheat" and pretend that (normalized) confidence scores are probabilities: see this as a first-order approximation

Objective of this course

- Present data models for uncertain data management in general, and probabilistic data management in particular:
    - relational databases (SQL queries)
    - XML data
- Present provenance management techniques (next set of slides)

Probabilistic Models of Uncertainty

Part II: Probabilistic Models of Uncertainty

- Probabilistic Relational Models
- Probabilistic XML

Possible worlds semantics

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

Uncertain 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

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

Patient	Examin. 1	Examin. 2	Diagnosis
A	23	12	α
B	10	23	β
C	2	4	γ
D	15	15	α
E	15	17	β

Codd tables, a.k.a. SQL NULLs

Patient	Examin. 1	Examin. 2	Diagnosis
A	23	12	α
B	10	23	$\perp_1$
C	2	4	γ
D	15	15	$\perp_2$
E	$\perp_3$	17	β

- 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 possible completions of the table (infinitely many)
- In SQL, three-valued logic, weird semantics:

```
SELECT * FROM Tel WHERE tel_nr = '333' OR tel_nr <> '333'
```

Problem: Codd tables and query evaluation

Appointment

Doctor	Patient
D1	A
D2	A

Illness

Patient	Diagnosis
A	$\perp$

Let's join the two tables...

Problem: Codd tables and query evaluation

Appointment

Doctor	Patient
D1	A
D2	A

Illness

Patient	Diagnosis
A	$\perp$

Let's join the two tables...

Appointment $\bowtie$ Illness

Doctor	Patient	Diagnosis

Problem: Codd tables and query evaluation

Appointment

Doctor	Patient
D1	A
D2	A

Illness

Patient	Diagnosis
A	$\perp$

Let's join the two tables...

Appointment $\bowtie$ Illness

Doctor	Patient	Diagnosis
D1	A	$\perp_1$
D2	A	$\perp_2$

Problem: Codd tables and query evaluation

Appointment

Doctor	Patient
D1	A
D2	A

Illness

Patient	Diagnosis
A	$\bot$

Let's join the two tables...

Appointment $\bowtie$ Illness

Doctor	Patient	Diagnosis
D1	A	$\bot_1$
D2	A	$\bot_2$

- We know that $\bot_1 = \bot_2$, but we cannot represent it
- Simple solution: named nulls aka v-tables
- More expressive solution: c-tables

C-tables [Imielinski and Lipski, 1984]

Patient	Examin. 1	Examin. 2	Diagnosis	Condition
A	23	12	α	
B	10	23	$\perp_1$	
C	2	4	γ	
D	$\perp_2$	15	$\perp_1$	
E	$\perp_3$	17	β	$18 < \perp_3 < \perp_2$

- NULLs are labeled, and can be reused inside and across tuples
- Arbitrary correlations across tuples
- Closed under the relational algebra
- Every set of possible worlds can be represented as a database with c-tables

Tuple-independent databases (TIDs)

[Lakshmanan et al., 1997, Dalvi and Suciu, 2007]

Patient	Examin. 1	Examin. 2	Diagnosis	Probability
A	23	12	α	0.9
B	10	23	β	0.8
C	2	4	γ	0.2
C	2	14	γ	0.4
D	15	15	α	0.6
D	15	15	β	0.4
E	15	17	β	0.7
E	15	17	α	0.3

- 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)

[Barbará et al., 1992, Ré and Suciu, 2007]

Patient	Examin. 1	Examin. 2	Diagnosis	Probability
A	23	12	α	0.9
B	10	23	β	0.8
C	2	4	γ	0.2
C	2	14	γ	0.4
D	15	15	β	0.6
D	15	15	α	0.4
E	15	17	β	0.7
E	15	17	α	0.3

- 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 no more

Probabilistic c-tables [Green and Tannen, 2006]

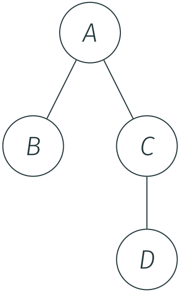
Patient	Examin. 1	Examin. 2	Diagnosis	Condition
A	23	12	α	w_1
B	10	23	β	w_2
C	2	4	γ	w_3
C	2	14	γ	$\neg w_3 \wedge w_4$
D	15	15	β	w_5
D	15	15	α	$\neg w_5 \wedge w_6$
E	15	17	β	w_7
E	15	17	α	$\neg w_7$

- The w_i's are independent Boolean random variables
- Each w_i has a probability of being true (e.g., $\Pr(w_1) = 0.9$)
- Any finite probability distribution of tables can be represented using probabilistic c-tables

Part II: Probabilistic Models of Uncertainty

- Probabilistic Relational Models
- Probabilistic XML

Reminder: The semistructured model and XML




```
<a>
  <b>...</b>
  <c>
    <d>...</d>
  </c>
</a>
```

- 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

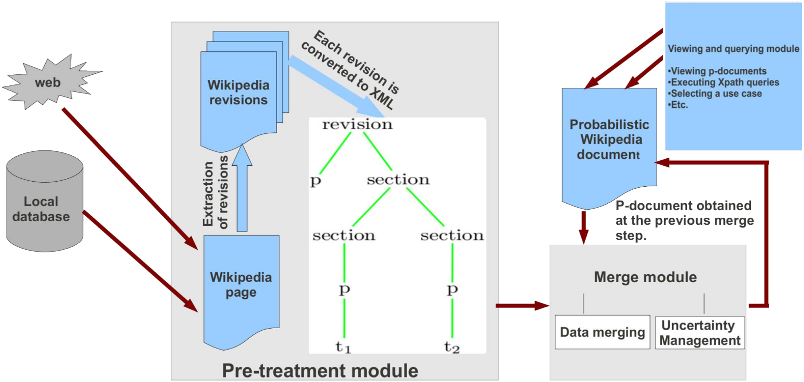
Why Probabilistic XML?

- Extensive literature about probabilistic relational databases [Dalvi et al., 2009, Widom, 2005, Koch, 2009]
- Different typical querying languages: 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
  - Documents with uncertain annotations
  - Inherently tree-like data (e.g., mailing lists, parse trees) with naturally occurring queries involving the descendant axis

Remark

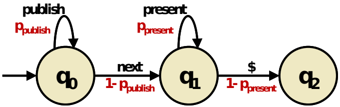
Some results can be transferred between probabilistic relational databases and probabilistic XML [Amarilli and Senellart, 2013]

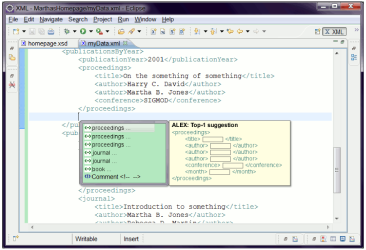
Uncertain version control
[Ba et al., 2013]



Use trees with probabilistic annotations to represent the uncertainty in the correctness of a document under open version control (e.g., Wikipedia articles)

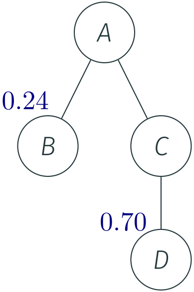
Probabilistic summaries of XML corpora [Abiteboul et al., 2012a,b]





- 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
- Application: Optimal auto-completions in an XML editor

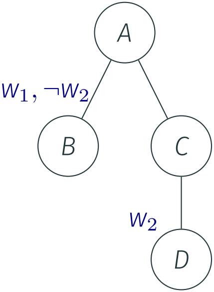
Simple probabilistic annotations



- 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!

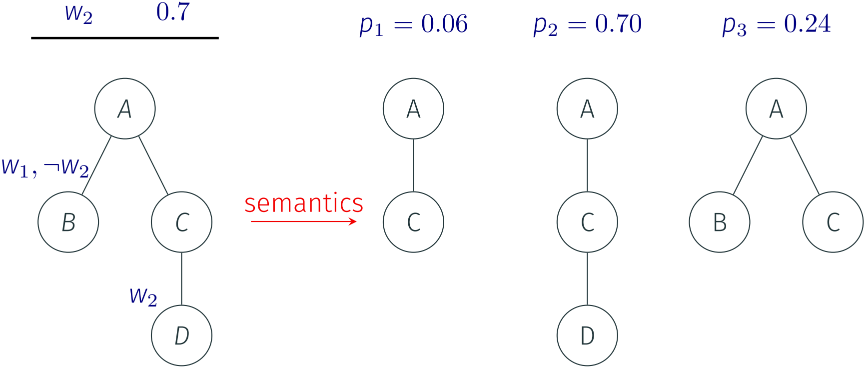
Annotations with event variables

Event	Prob.
w_1	0.8
w_2	0.7



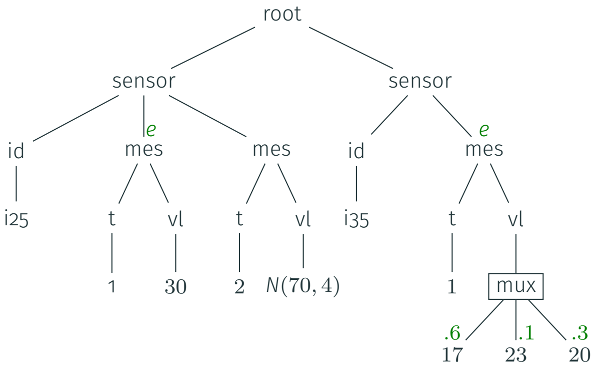
Annotations with event variables

Event	Prob.
w_1	0.8
w_2	0.7



- Expresses arbitrarily complex dependencies
- Obviously, analogous to probabilstic c-tables

A general probabilistic XML model
[Abiteboul et al., 2009]

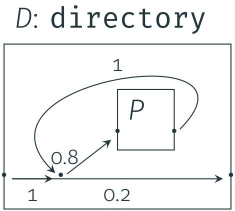


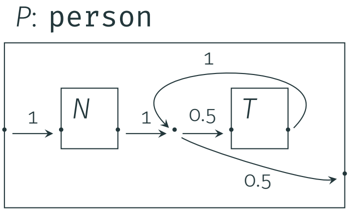
- *e*: event "it did not rain" at time 1
- mux: mutually exclusive options
- $N(70, 4)$: normal distribution

- 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 [Benedikt et al., 2010]

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





- Probabilistic model that extends PXML with local dependencies
- Generate documents of unbounded width or depth

Incomplete formalisms and Open-World Query Answering

Incomplete databases

- Another kind of uncertainty is incomplete data, i.e., missing information
  - $\rightarrow$ For instance, the open-world assumption
- We know some constraints that the true data must satisfy
  - $\rightarrow$ *"Every person has a father..."* (implies the existence of some elements)
  - $\rightarrow$ *"... and only one father"* (... and their uniqueness)
- The possible worlds of the data are all possible completions satisfying the constraints
- Focus on relational data, but there are also models for XML [Barceló et al., 2009].

Open-world query answering (OWQA)

- We have the data, the constraints (in logic), and a query
- Usual evaluation: find all results of the query on the data
- OWQA: find all results of the query that are true on all completions satisfying the constraints!

Open-world query answering (OWQA)

- We have the data, the constraints (in logic), and a query
- Usual evaluation: find all results of the query on the data
- OWQA: find all results of the query that are true on all completions satisfying the constraints!

Person
Name
Luke
Kylo

Filiation	
Son	Father
Kylo	Han

- *"Every person has a father"* and *"Every father is a person"*
  - $\forall x \; \mathrm{Person}(x) \rightarrow \exists y \; \mathrm{Filiation}(x, y)$
  - $\forall xy \; \mathrm{Filiation}(x, y) \rightarrow \mathrm{Person}(y)$
- Query: "find everyone who has a father" $Q(x) : \exists y \; \mathrm{Filiation}(x, y)$

Complexity of OWQA

- OWQA is related to satisfiability in logics:
    - Can we satisfy the data, the constaints, and the negation of the query?
- In OWQA, we want to be tractable in the data
- In general OWQA is undecidable if we allow arbitrary first-order constraints
- If we restrict the constraint language it can become decidable or even tractable ($\leftrightarrow$ description logics)
- Another challenge: handling partial completeness, i.e., where we know that something is complete, see [Razniewski et al., 2015]

The chase

- The chase: complete the data by creating all missing values
- Only defined for dependencies, i.e., constraints of the form "some pattern $\rightarrow$ some other pattern"
- In this case the chase is universal: creates a (possibly infinite) database that only satisfies the certain queries

Chase example

- $\forall x \; \text{Person}(x) \rightarrow \exists y \; \text{Filiation}(x, y)$
- $\forall xy \; \text{Filiation}(x, y) \rightarrow \text{Person}(y)$

Person
Name
Luke
Kylo

Filiation	
Son	Father
Kylo	Han

Chase example

- $\forall x\ \text{Person}(x) \rightarrow \exists y\ \text{Filiation}(x, y)$
- $\forall xy\ \text{Filiation}(x, y) \rightarrow \text{Person}(y)$

Person
Name
Luke
Kylo
Han

Filiation	
Son	Father
Kylo	Han
Luke	x_1

Chase example

- $\forall x\ \text{Person}(x) \to \exists y\ \text{Filiation}(x, y)$
- $\forall xy\ \text{Filiation}(x, y) \to \text{Person}(y)$

Person
Name
Luke
Kylo
Han
x_1

Filiation	
Son	Father
Kylo	Han
Luke	x_1

Chase example

- $\forall x\ \text{Person}(x) \to \exists y\ \text{Filiation}(x, y)$
- $\forall xy\ \text{Filiation}(x, y) \to \text{Person}(y)$

Person
Name
Luke
Kylo
Han
x_1

Filiation	
Son	Father
Kylo	Han
Luke	x_1
Han	x_2
x_1	x_3

Chase example

- $\forall x \; \mathrm{Person}(x) \rightarrow \exists y \; \mathrm{Filiation}(x, y)$
- $\forall xy \; \mathrm{Filiation}(x, y) \rightarrow \mathrm{Person}(y)$

Person	Filiation	
Name	Son	Father
Luke	Kylo	Han
Kylo	Luke	x_1
Han	Han	x_2
x_1	x_1	x_3
$\vdots$	$\vdots$	$\vdots$

Using the chase

- If the chase is finite we can build it
  - → Can happen when the constaints are acyclic, e.g., $\forall x \text{ Person}(x) \rightarrow \exists y \text{ Filiation}(x, y)$
- If the chase is infinite but has bounded treewidth, we can reason on it using tree automata methods

Extensions for OWQA

- If possible, rewrite the query using the constraints to work on the initial data
  - $\rightarrow$ $Q(x) : \exists y \ \text{Filiation}(x, y)$ rewrites to $Q(x) : \text{Person}(x) \lor \exists y \ \text{Filiation}(x, y) \lor \exists y \ \text{Filiation}(y, x)$
  - Advantage: ensures good complexity in the data
- Application: Ontology-Mediated Query Answering (OMQA)
- Another challenge: handle finiteness of the data, i.e., what if we want to work on all finite completions?

To go further

Querying and Updating

- Numerous works on the complexity of querying probabilistic databases, see [Suciu et al., 2011] (relational case) and [Kimelfeld et al., 2009] (XML case) for surveys
- Hard problem in general ($FP^{\#P}$), some (very few!) tractable cases
- Approximation algorithms [Olteanu et al., 2010, Souihli and Senellart, 2013]: practical solution
- Also important to consider updates [Abiteboul et al., 2009, Kharlamov et al., 2010]

Repairs

- Another kind of uncertainty: we know that the database must satisfy some constraints (e.g., functionality)
- The data that we have does not satisfy it
- Reason about the ways to repair the data, e.g., removing a minimal subset of tuples
- Can we evaluate queries on this representation? E.g., is a query true on every maximal repair? See, e.g., [Koutris and Wijsen, 2015].

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.

ProvSQL `https://github.com/PierreSenellart/provsql` maintains provenance information while evaluating queries (see later), and can use this to perform probabilistic reasoning

Reading material

- An influential paper on incomplete databases [Imielinski and Lipski, 1984]
- A book on probabilistic relational databases, focused around TIDs/BIDs and MayBMS [Suciu et al., 2011]
- An in-depth presentation of MayBMS [Koch, 2009]
- A gentle presentation of relational and XML probabilistic models [Kharlamov and Senellart, 2011]
- A survey of probabilistic XML [Kimelfeld and Senellart, 2013]

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 which restrictions on the data (e.g., (hyper)tree-width) make query answering tractable.
- Finding more expressive logical formalisms for which open-world query answering is decidable (in particular on finite completions)
- 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.; reasoning with probabilistic constraints

References

Serge Abiteboul, Benny Kimelfeld, Yehoshua Sagiv, and Pierre Senellart. On the expressiveness of probabilistic XML models. *VLDB Journal*, 18(5):1041–1064, October 2009.

Serge Abiteboul, Yael Amsterdamer, Daniel Deutch, Tova Milo, and Pierre Senellart. Finding optimal probabilistic generators for XML collections. In *Proc. ICDT*, pages 127–139, Berlin, Germany, March 2012a.

Serge Abiteboul, Yael Amsterdamer, Tova Milo, and Pierre Senellart. Auto-completion learning for XML. In *Proc. SIGMOD*, pages 669–672, Scottsdale, USA, May 2012b. Demonstration.

Antoine Amarilli and Pierre Senellart. On the connections between relational and XML probabilistic data models. In *Proc. BNCOD*, pages 121–134, Oxford, United Kingdom, July 2013.

M. Lamine Ba, Talel Abdessalem, and Pierre Senellart. Uncertain version control in open collaborative editing of tree-structured documents. In *Proc. DocEng*, Florence, Italy, September 2013.

Daniel Barbará, Hector Garcia-Molina, and Daryl Porter. The management of probabilistic data. *IEEE Transactions on Knowledge and Data Engineering*, 4(5):487–502, 1992.

Pablo Barceló, Leonid Libkin, Antonella Poggi, and Cristina Sirangelo. XML with incomplete information: models, properties, and query answering. In *Proc. PODS*, pages 237–246, New York, NY, 2009. ACM.

Michael Benedikt, Evgeny Kharlamov, Dan Olteanu, and Pierre Senellart. Probabilistic XML via Markov chains. *Proceedings of the VLDB Endowment*, 3(1):770–781, September 2010. Presented at the VLDB 2010 conference, Singapore.

Nilesh Dalvi, Chrisopher Ré, and Dan Suciu. Probabilistic databases: Diamonds in the dirt. *Communications of the ACM*, 52(7), 2009.

Nilesh N. Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, pages 864–875, 2004.

Nilesh N. Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. *VLDB Journal*, 16(4), 2007.

Robert Fink, Andrew Hogue, Dan Olteanu, and Swaroop Rath. SPROUT^2: a squared query engine for uncertain web data. In *SIGMOD*, 2011.

José Galindo, Angelica Urrutia, and Mario Piattini. *Fuzzy Databases: Modeling, Design And Implementation*. IGI Global, 2005.

Todd J. Green and Val Tannen. Models for incomplete and probabilistic information. In *Proc. EDBT Workshops, IIDB*, Munich, Germany, March 2006.

Tomasz Imielinski and Witold Lipski. Incomplete information in relational databases. *Journal of the ACM*, 31(4):761–791, 1984.

Evgeny Kharlamov and Pierre Senellart. Modeling, querying, and mining uncertain XML data. In Andrea Tagarelli, editor, *XML Data Mining: Models, Methods, and Applications*. IGI Global, 2011.

Evgeny Kharlamov, Werner Nutt, and Pierre Senellart. Updating probabilistic XML. In *Proc. Updates in XML*, Lausanne, Switzerland, March 2010.

Benny Kimelfeld and Pierre Senellart. Probabilistic XML: Models and complexity. In Zongmin Ma and Li Yan, editors, *Advances in Probabilistic Databases for Uncertain Information Management*, pages 39–66. Springer-Verlag, May 2013.

Benny Kimelfeld, Yuri Kosharovsky, and Yehoshua Sagiv. Query evaluation over probabilistic XML. *VLDB J.*, 2009.

Christoph Koch. MayBMS: A system for managing large uncertain and probabilistic databases. In Charu Aggarwal, editor, *Managing and Mining Uncertain Data*. Springer-Verlag, 2009.

Paraschos Koutris and Jef Wijsen. The data complexity of consistent query answering for self-join-free conjunctive queries under primary key constraints. In *Proc. SIGMOD*, pages 17–29. ACM, 2015.

Laks V. S. Lakshmanan, Nicola Leone, Robert B. Ross, and V. S. Subrahmanian. ProbView: A flexible probabilistic database system. *ACM Transactions on Database Systems*, 22(3), 1997.

Dan Olteanu, Jiewen Huang, and Christoph Koch. Approximate confidence computation in probabilistic databases. In *Proc. ICDE*, 2010.

Simon Razniewski, Flip Korn, Werner Nutt, and Divesh Srivastava. Identifying the extent of completeness of query answers over partially complete databases. In *Proc. SIGMOD*, 2015.

Christopher Ré and Dan Suciu. Materialized views in probabilistic databases: for information exchange and query optimization. In *Proc. VLDB*, 2007.

Asma Souihli and Pierre Senellart. Optimizing approximations of DNF query lineage in probabilistic XML. In *Proc. ICDE*, pages 721–732, Brisbane, Australia, April 2013.

Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases*. Morgan & Claypool, 2011.

Jennifer Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *Proc. CIDR*, Asilomar, CA, USA, January 2005.

Lotfi A. Zadeh. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, 7(2), 1986.