Querying Probabilistic XML Databases

Sept. 21st 2012

Asma Souihli

Network and Computer Science Department
XML

for semi-structured data (*tree-like* structure)

```
<bib> <book year="1994">
  <title> TCP/IP Illustrated </title>
  <author> <last> Stevens </last> <first> W. </first> </author>
  <publisher> Addison-Wesley </publisher>
  <price> 65.95 </price>
</book>

... <book year="2000">
  <title> Data on the Web </title>
  <author> <last> Abiteboul </last> <first> Serge </first> </author>
  <author> <last> Buneman </last> <first> Peter </first> </author>
  <author> <last> Suciu </last> <first> Dan </first> </author>
  <publisher> Morgan Kaufmann Publishers </publisher>
  <price> 39.95 </price>
</book> ... </bib>
```
Probabilistic Data - PrXML


dist([2] ind) = {([3], 0.6)}
dist([5] ind) = {([6], 0.7)}
dist([7] mux) = {([8], 0.2), ([14], 0.4), ([19], 0.3)}
dist([9] mux) = {([10], 0.5), ([12], 0.3)}
dist([27] ind) = {([28], 0.9), ([30], 0.7)}
dist([31] exp) = {([32], [34]), 0.4), ([32], [36]), 0.2), ([32], [34], [36]), 0.3)}
Context

Uncertainty
Context

- In many of these tasks, information is described in a semi-structured manner.

- Especially when the source (e.g., XML or HTML) is already in this form.

- Representation by means of a hierarchy of nodes is natural.
Outline

1. **PrXML Models**
   - Local Dependency
   - Long-distance Dependency

2. **Querying P-documents**
   - Types of Queries
   - Probabilistic Lineage
   - Complexity of Queries

3. **The ProApproX System**
   - Computation Algorithms
   - Lineage Decomposition Techniques
   - Evaluation Plans
   - Experiments

4. **Conclusions**
PrXML Models – Local Dependency

- $\textit{ind}$ children of the node are chosen independently of one another, according to their probabilities.
- $\textit{mux}$ children of the node are chosen in a mutually exclusive way, depending of their probabilities, that must sum up to 1 or less.

Local dependency ($\textit{mux}$ and $\textit{ind}$ nodes)
PrXML Models – Long-distance Dependency

Local dependency
\((mux\ and\ ind\ nodes)\)

\[
\begin{array}{c}
\text{Parent node} \\
mux \\
0.3 \rightarrow 0.7 \\
\text{Child} \\
\text{Child}
\end{array}
\]

\(PrXML^{\{ind,mux\}}\)

Long-distance dependency
\((\text{Conjunction of independent events} \rightarrow cie)\)

\[
\begin{array}{c}
\text{Ancestor node} \\
\text{Parent node} \\
e_2 \\
\text{Child} \\
\text{Child}
\end{array}
\]

\[
\begin{array}{c}
\text{Parent node} \\
e_3 \land e_4 \\
\text{Child} \\
\text{Child}
\end{array}
\]

\(PrXML^{\{cie\}}\)

\(e_2\)

\[
\begin{array}{c}
\text{Parent node} \\
\text{Child}
\end{array}
\]

\(PrXML^{\{cie\}}\)

\[
\begin{array}{c}
\text{Parent node} \\
e_2 \\
\text{Child}
\end{array}
\]

\(PrXML^{\{cie\}}\)

S. Abiteboul, B. Kimelfeld, Y. Sagiv, and P. Senellart. 2009
Example

Repository

Employee

Name
Asma Souihli

Details $e_2$

$\begin{align*}
Pr(e_1) &= .9 \\
Pr(e_2) &= .8 \\
Pr(e_3) &= .4 \\
Pr(e_4) &= .1 \\
Pr(e_5) &= .6 \\
Pr(e_6) &= .3 \\
Pr(e_7) &= .2 \\
Pr(e_8) &= .8
\end{align*}$

work

$e_1$ Place
Telecom
Paristech

$e_7$ Place
Gaumont
Pathé

Contact

$e_8$

Phone

$e_3$

0622330011

$souihli@enst.fr$

$e_1$

$e_6$

$e_4 \wedge \neg e_5$

address

$e_5$

Paris 13

$asma.souihli@gmail.com$

Paris 15
Outline

1. PrXML Models
   Local Dependency
   Long-distance Dependency

2. Querying P-documents
   Types of Queries
   Probabilistic Lineage
   Complexity of Queries

3. The ProApproX System
   Computation Algorithms
   Lineage Decomposition Techniques
   Evaluation Plans
   Experiments

4. Conclusions
Querying P-documents – Types of Queries

- Tree Pattern Queries (TPQ)
  - Tree Pattern Queries with joins (TPQJ)
Example

- **Q1**: Employee [Name= "Asma Souihli"] // e-mail / text()

<table>
<thead>
<tr>
<th>enst.fr:</th>
<th>e₂ ∧ e₈ ∧ e₁</th>
<th>C1</th>
</tr>
</thead>
<tbody>
<tr>
<td>gmail.com:</td>
<td>e₂ ∧ e₈ ∧ e₆</td>
<td>C2</td>
</tr>
<tr>
<td>sap.com:</td>
<td>e₂ ∧ e₉ ∧ e₁₀</td>
<td>C3</td>
</tr>
<tr>
<td>gmail.com:</td>
<td>e₂ ∧ e₉ ∧ e₆</td>
<td>C4</td>
</tr>
</tbody>
</table>

```
Repository

Employee

Name

Details e₂

Asma Souihli

t₁

Contact e₈

e-mail e₁

souihli@enst.fr

asma.souihli@gmail.com

t₂

Contact e₉

e-mail e₆

souihli@sap.com

asma.souihli@gmail.com

e₃ = .9

e₂ = .8

e₉ = .6

e₁₀ = .7

e₆ = .3

e₈ = .8
```
Querying PrXML – Probabilistic Lineage

- Probability to find an e-mail:

\[ \Pr(Q_1) = \Pr(C_1 \lor C_2 \lor C_3 \lor C_4) \rightarrow \text{Probabilistic lineage (DNF shape)} \]

- Possible results:

\[ \Pr(\text{asma.souihli@gmail.com}) = \Pr(C_2 \lor C_4) \]
\[ \Pr(\text{souihli@enst.fr}) = \Pr(C_1) \]
\[ \Pr(\text{souihli@sap.com}) = \Pr(C_3) \]
When is a linear computation possible?

- if $C_1$ and $C_2$ are independent, then:
  
  $$\Pr(C_1 \land C_2) = \Pr(C_1) \times \Pr(C_2)$$

  $$\Pr(C_1 \lor C_2) = 1 - (1 - \Pr(C_1)) \times (1 - \Pr(C_2))$$

- if $C_1$ and $C_2$ are inconsistent (mutually exclusive), then:

  $$\Pr(C_1 \lor C_2) = \Pr(C_1) + \Pr(C_2)$$
Back to the Example

Pr(@enst.fr) = Pr(C1) = Pr(e_2 \wedge e_8 \wedge e_1) = .8 \times .8 \times .9
= 0.576

Pr(@sap.com) = Pr(C3) = 0.336

Pr(@gmail.com) = Pr(C2 \lor C4) = (e_2 \wedge e_8 \wedge e_6) \lor (e_2 \wedge e_9 \wedge e_6)

Factorization:
Pr(@gmail.com) = (e_2 \wedge e_6) \wedge (e_8 \lor e_9) = .8 \times .3 \times (1 - (1-.8)(1-.6))
= 0.2208
Pr(Q1) = Pr( C1 ∨ C2 ∨ C3 ∨ C4 )
= Pr [ (e₂ ∧ e₈ ∧ e₁) ∨ (e₂ ∧ e₈ ∧ e₆) ∨ (e₂ ∧ e₉ ∧ e₁₀) ∨ (e₂ ∧ e₉ ∧ e₆) ]

→ Factorization:
= Pr [e₂ ∧ ((e₈ ∧ (e₁ ∨ e₆)) ∨ (e₉ ∧ (e₁₀ ∨ e₆))) ]

→ Difficult to evaluate!
Solutions..

- One possible (naïve) way, is to find the truth value assignments that satisfy the propositional formula (probabilistic lineage) (out of $2^{#literals}$ possible assignments/worlds !)

- And sum the probabilities of these satisfying assignments to get the answer

<table>
<thead>
<tr>
<th>$e_1$</th>
<th>$e_2$</th>
<th>$e_6$</th>
<th>$e_8$</th>
<th>$e_9$</th>
<th>$e_{10}$</th>
<th>Probability</th>
<th>$C1$ V $C2$ V $C3$ V $C4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>0.0845</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>0.3345</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>0.87</td>
<td>false</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Probabilities of the satisfying assignments for the DNF (lineage formula): \#P-Hard problem

- No polynomial time algorithm for the exact solution if \( P \neq NP \)
- \#P problems ask "how many" rather than "are there any"

How many graph coloring using \( k \) colors are there for a particular graph \( G \)?
Querying PrXML – Complexity of Queries

- A union of sets (clauses) problem: #P-Hard problem

\[
\begin{align*}
\Pr(C_1) &= \Pr(e_1) \cdot \Pr(e_2) \\
\Pr(C_2) &= \Pr(e_2) \cdot \Pr(e_3) \\
\Pr(C_1 \cap C_2) &= \Pr(e_1) \cdot \Pr(e_2) \cdot \Pr(e_3) \\
\Pr(C_1 \cup C_2) &= \Pr(C_1) + \Pr(C_2) - \Pr(C_1 \cap C_2)
\end{align*}
\]

For dependent probabilistic clauses \(C_1 \ldots C_n\) the inclusion-exclusion principle becomes:

\[
\Pr(\bigcup_{i=1}^{n} C_i) = \sum_{k=1}^{n} (-1)^{k-1} \sum_{\substack{J \subseteq \{1, \ldots, n\} \atop |J|=k}} \Pr(C_J)
\]

where:

\[C_J = \bigcap_{j \in J} C_j\]
Outline

1. PrXML Models
   Local Dependency
   Long-distance Dependency

2. Querying P-documents
   Types of Queries
   Probabilistic Lineage
   Complexity of Queries

3. The ProApproX System
   Computation Algorithms
   Lineage Decomposition Techniques
   Evaluation Plans
   Experiments

4. Conclusions
The ProApproX System

[CIKM 2012, SIGMOD 2011]

- Translates into a probabilistic database with only \( cie \) nodes
- Translates the user query into a lineage query
Back to the Example

Q1: / Employee [Name= "Asma Souihli"] // e-mail / text()

- To get the lineage for the boolean projection:

```xml
for $x1 in /employee
for $x2 in $x1/name[.="Asma Souihli"]
for $x3 in $x1//email/text()
let $leaves:=(x2,x3)
let $atts:=(for $i in $leaves return $i/ancestor-or-self::*/attribute(event))
return text{distinct-values(for $att in $atts return string($att))}
```

- To get lineages of answers:

```xml
for $val in distinct-values(/employee [name="Asma Souihli"]//email/text())
order by $val
return <match>{$val}{
for $x1 in /employee
for $x2 in $x1/name[.="Asma Souihli "]
for $x3 in $x1//email/text()
let $leaves:=(x2,x3)
let $atts:=(for $i in $leaves return $i/ancestor-or-self::*/attribute(event))
where $x3=$val
return <clause>{distinct-values(for $att in $atts return string($att))}</clause>
}</match>
```
The ProApproX System

[CIKM 2012, SIGMOD 2011]

- Translates into a probabilistic database with only cie nodes
- Translates the user query into a lineage query
- Is built on top of a native XML DBMS
- Processes the lineage formula to get the probability of the query (and of each matching answer)
The ProApproX System – Computation Algorithms

- **Additive approximation:**
  - For a fixed error $\epsilon$ and a DNF $F$, $A(F)$ is an additive $\epsilon$-approximation of $\Pr(F)$ with a probability of at least $\delta$ (a fixed reliability factor) if:

    \[
    \Pr(F) - \epsilon \leq A(F) \leq \Pr(F) + \epsilon
    \]

- **Multiplicative Approximation**
  - For a fixed error $\epsilon$, a DNF $F$, $A(F)$ is an multiplicative $\epsilon$-approximation of $\Pr(F)$ with a probability of at least $\delta$ if:

    \[
    (1-\epsilon) \Pr(F) \leq A(F) \leq (1+\epsilon) \Pr(F)
    \]
DEMO 1

[SIGMOD 2011]
The ProApproX System – Computation Algorithms

- **Exact Computations:**
  - The naïve algorithm – Possible worlds
    - Finding the satisfying assignments out of \(2^{\#\text{variables}}\) possible truth value assignments
    - \(O(2^n)\)
  - The sieve algorithm – The inclusion-exclusion principle
    - Exponential in the number of clauses \(m\)
    - \(O(2^m)\)
Approximations:

- Naïve Monte Carlo sampling for additive app.:
  - Linear but could take exponentially many samples to converge to a good approximation for low probabilities

- Biased Monte Carlo sampling for multiplicative app.:
  - Running time grows in $O(n^3 \ln n)$ in the number of clauses

- Self-Adjusting Coverage Algorithm for the DNF probability problem:
  - Linear in the length of F times $\ln(1/\delta) / \varepsilon^2$

Kimelfeld, Kosharovsky, and Sagiv. 2009
M. Karp, M. Luby, and N. Madras. 1989
The ProApproX System – Computation Algorithms

- Possibility to derive a multiplicative approximation from an additive approximation *(and vice versa)*

- Cost models and cost constants:

<table>
<thead>
<tr>
<th>Algorithm alg</th>
<th>cost(_{alg})</th>
<th>(C_{alg}) (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>naïve</td>
<td>(C_{\text{ naïve}} \times 2^N \times L)</td>
<td>(4 \times 10^{-5})</td>
</tr>
<tr>
<td>sieve</td>
<td>(C_{\text{ sieve}} \times 2^m \times \frac{L}{m})</td>
<td>(5 \times 10^{-5})</td>
</tr>
<tr>
<td>AddMC</td>
<td>(C_{\text{ AddMC}} \times \ln \frac{2}{\delta} \times \frac{L}{\varepsilon^2})</td>
<td>(4 \times 10^{-5})</td>
</tr>
<tr>
<td>MulMC</td>
<td>(C_{\text{ AddMC}} \times \ln \frac{2}{\delta} \times \frac{L}{\ell^2 \varepsilon^2})</td>
<td>(4 \times 10^{-5})</td>
</tr>
<tr>
<td>coverage</td>
<td>(C_{\text{ coverage}} \times \ln \frac{2}{\delta} \times \frac{(1+\varepsilon)L}{\varepsilon^2})</td>
<td>(10^{-3})</td>
</tr>
</tbody>
</table>
The ProApproX System – Lineage Decomposition Techniques

Factorization

\[ F = (e_1 \land e_2) \lor (e_3 \land e_4) \lor (e_3 \land e_5) \lor (\neg e_3) \lor (e_6 \land e_7) \lor (e_6 \land e_8) \lor e_8 \]

Pr(F)

Exact /naïve Algo. OR Approximation
DEMO 2
[CIKM 2012]
The ProApproX System – Evaluation Plans

- Propagation of \( \varepsilon \) (and \( \delta \)):

**Proposition 1.** Let \( \phi = \psi_1 \lor \psi_2 \), and assume \( \tilde{p}_1 \) and \( \tilde{p}_2 \) are additive approximations of \( \Pr(\psi_1) \) and \( \Pr(\psi_2) \), to a factor of \( \varepsilon_1 \) and \( \varepsilon_2 \), respectively. Then \( 1-(1-\tilde{p}_1)(1-\tilde{p}_2) \) is an additive approximation of \( \Pr(\phi) \) to a factor of \( \varepsilon \) if:

\[
\varepsilon_1 + \varepsilon_2 + \varepsilon_1 \varepsilon_2 \leq \varepsilon
\]

- Many possible values for \( \varepsilon_1 \) and \( \varepsilon_2 \) can be found
- Best assignments are not always obvious
The ProApproX System – Possible Evaluation Plans

Deterministic exploration:

\[
cost_{\phi} = 200
\]

- \(cost_{\psi_1} = 1\)
- \(cost_{\psi_2} = 35\)
- \(cost_{\psi_3} = 8\)
- \(cost_{\psi_4} = 6\)
- \(cost_{\psi_5} = 3\)
- \(cost_{\psi_6} = 2\)
- \(cost_{\psi_7} = 1\)
- \(cost_{\psi_8} = 15\)
- \(cost_{\psi_9} = 8\)
- \(cost_{\psi_{10}} = 12\)
- \(cost_{\psi_{11}} = 10\)
- \(cost_{\psi_{12}} = 9\)
The ProApproX System – Experiments

Running time of the different algorithms on the MondialDB dataset
The ProApproX System – Experiments

<table>
<thead>
<tr>
<th>Query type</th>
<th>Avg DNF size</th>
<th>XQuery</th>
<th>Comp</th>
<th>Exp+Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td>96.82%</td>
<td>2.47%</td>
<td>0.71%</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>98.19%</td>
<td>1.38%</td>
<td>0.42%</td>
</tr>
<tr>
<td>4</td>
<td>43</td>
<td>97.41%</td>
<td>2.01%</td>
<td>0.58%</td>
</tr>
<tr>
<td>5</td>
<td>252</td>
<td>64.47%</td>
<td>33.76%</td>
<td>1.78%</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>99.69%</td>
<td>0.29%</td>
<td>0.03%</td>
</tr>
<tr>
<td>4+join</td>
<td>3656</td>
<td>61.49%</td>
<td>36.12%</td>
<td>2.39%</td>
</tr>
</tbody>
</table>

Proportion of time
(MondialDB - Best Tree)

Relative error on the probabilities computed by the algorithm on the MondialDB over each non-join query with respect to the exact probability values
($\varepsilon = 0.1$, $\delta = 95\%$)
Running time of the different algorithms on a given query of the movie dataset. (times greater than 5s are not shown)
The ProApproX System – Experiments

Running time of the different algorithms on the synthetic dataset
Outline

1. **PrXML Models**
   - Local Dependency
   - Long-distance Dependency

2. **Querying P-documents**
   - Types of Queries
   - Probabilistic Lineage
   - Complexity of Queries

3. **The ProApproX System**
   - Lineage Decomposition Techniques
   - Computation Algorithms
   - Demo
   - Evaluation Plans
   - Experiments

4. **Conclusions**
Contributions

- We have introduced an original optimizer-like approach to evaluating query results over probabilistic XML
- Over a more expressive PrXML model
- Positive tree-pattern queries, possibly with joins

[Submitted ICDE 2013]
Contributions

- Main observation - optimal probability evaluation algorithm to use depends on the characteristics of the formula:
  - Few variables naïve algorithm
  - Few clauses sieve algorithm
  - Monte-Carlo is very good at approximating high probabilities
  - Sometimes the structure of a query makes the probability of a query easy to evaluate (EvalDP)
  - Refined approximation methods best when everything else fails (coverage)
Perspectives

- Exploiting the structure of the query to obtain factorized lineage

- Most evaluation algorithms scale effortlessly (with the exception of the self-adjusting coverage algorithm, which requires synchronization)
  - distribute the probability computation over multi-core or distributed architectures

- Processing DNFs, but the technique could probably be extended to arbitrary formulas

- Define the range of negated TPQ queries having a DNF lineage
Thank you.

Actually, that assumption isn't really necessary. We can see here that the point-cow approximation works equally well.