Topics in Probabilistic and Statistical Databases

Lecture 4: Dichotomoty Theorems

Dan Suciu
University of Washington
Before we start…

• Kate will give an update on OWA
### TABLE II

**Example of Missing Probabilities**

<table>
<thead>
<tr>
<th>EMPLOYEE</th>
<th>DEPARTMENT</th>
<th>QUALITY BONUS</th>
<th>SALES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jon Smith</td>
<td>Toy</td>
<td>0.3 [Great yes]</td>
<td>0.3 [$30–34K]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.4 [Good yes]</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2 [Fair *]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1 [* *]</td>
<td>0.2 [*]</td>
</tr>
<tr>
<td>NAME</td>
<td>DIV PRICE</td>
<td>RATING</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>P.J.</td>
<td>0.3 [10 200]</td>
<td>0.9 [AAA]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2 [20 250]</td>
<td>0.1 [AA]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2 [10 250]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1 [0 0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1 [0 100]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1 [0 0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONTI</td>
<td>1.0 [0 50]</td>
<td>0.5 [BBB]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5 [CCC]</td>
<td></td>
</tr>
</tbody>
</table>
Brief review…

• What are the three different definitions for the complexity of the query evaluation problem?

• What is #P?
A Probabilistic Database Design Quiz

• You need to store data extracted from conference Websites

• Extractor has two phases:
  – A classifier checks if the Webpage is about a conference, and returns a confidence $c$ in $(0,1]$
  – A conference-name extractor, returns a name with confidence $p$
  – A pc-chair extractor, returns a person name, with confidence $q$
A Probabilistic Database Design Quiz

<table>
<thead>
<tr>
<th>URL</th>
<th>Conf</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>SIGMOD</td>
<td>c1*p1</td>
</tr>
<tr>
<td>U1</td>
<td>SIGCOM</td>
<td>c1*p2</td>
</tr>
<tr>
<td>U2</td>
<td>VLDB</td>
<td>c2*p3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URL</th>
<th>Chair</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>Kossman</td>
<td>c1*q1</td>
</tr>
<tr>
<td>U2</td>
<td>Gehrke</td>
<td>c2*q2</td>
</tr>
<tr>
<td>U2</td>
<td>Milo</td>
<td>c2*q3</td>
</tr>
</tbody>
</table>

There are correlations! Represent them with I/D-tables only.
Problem Statement

• Given:
  – A disjoint/independent probdb  PDB
  – A Boolean conjunctive query  Q

• Compute the probability Q(PDB)
Three Theorems

• Case 1: CQ\(^1\) on independent databases
  – Review: Hierarchical \(\Rightarrow\) PTIME, non-h \(\Rightarrow\) \#P-hard
  – Today: extensions to FDs, deterministic relations

• Case 2: CQ\(^1\) on D/I – databases
  – Today in class

• Case 3: CQ on independent databases
  – Start today, continue next time
Case 1: CQ\(^1\)+independent

- Review hierarchical queries, safe plans in class
- Review the expression-algorithm
FDs: Worlds v.s. Representation

<table>
<thead>
<tr>
<th>Product^p</th>
<th>price</th>
<th>color</th>
<th>shape</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>20</td>
<td>red</td>
<td>oval</td>
<td>p_1 = 0.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
<td>square</td>
<td>p_2 = 0.75</td>
</tr>
<tr>
<td>Camera</td>
<td>80</td>
<td>green</td>
<td>oval</td>
<td>p_3 = 0.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>red</td>
<td>round</td>
<td>p_4 = 0.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
<td>oval</td>
<td>p_5 = 0.2</td>
</tr>
<tr>
<td>iPod</td>
<td>300</td>
<td>white</td>
<td>square</td>
<td>p_6 = 0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>black</td>
<td>square</td>
<td>p_7 = 0.2</td>
</tr>
</tbody>
</table>

In each possible world: prod → price, color, shape
In the representation: prod → price
FDs at the Representation Level

\[ q \leftarrow R(x), S(x,y), T(y) \]

Suppose \( x \rightarrow y \) in \( S(x,y) \) in the representation

What is the complexity of this query?
FDs at the Representation Level

\[ q \leftarrow R(x), S(x,y), T(y) \]

Suppose \( x \rightarrow y \) in \( S(x,y) \) in the representation

"Reduce" \( R(x) \) to \( R(x,y) \)

\[ q \leftarrow R(x,y), S(x,y), T(y) \]

Now it is hierarchical
FDs at the Representation Level

\[ q(x) :\neg \ R(y), \ S(x,y,z), \ T(z) \]

Suppose \( x \rightarrow y \) in \( S(x,y) \) in the representation

What is the complexity now?
FDs at the Representation Level

**Theorem**  Let q be a query over a schema with FDs at the representation level. Let q’ be the “reduced” query (chase ?). Then the evaluation problem of q is reducible in PTIME to the evaluation problem of q’, and vice versa.

Proof in class.
How does this give us a dichotomy theorem?
Deterministic Relations

• Now add deterministic relations (in class)
  – Notation: \( R^p \) = probabilistic, \( R \) = deterministic

• What is the complexity of the following queries? Give theorem in class

\[
q :\neg R^p(x), S1(x,u), S2(u,v), S3(v,y), T^p(y,z)
\]

\[
q :\neg R^p(z,x), S1(x,u), S2(u,z), S3(z,v), S4(v,y), T^p(y)
\]

\[
q :\neg R^p(z,x), S1(x,u), S2(u,z), S3(z,v), S4(v,y), T^p(z,y)
\]
Case 2: $\text{CQ}^1$+Disjoint/independent

- Dichotomy: in [Dalvi et al.’06, Dalvi&S’07]
- Some safe plans also in [Andritsos’2006]
- $\text{CQ}^1$ (conjunctive queries, no self-joins)
- Independent/independent tables are OK

**Theorem** For all $q \in \text{CQ}^1$

$q$ has a safe plan and is in PTIME, OR
$q$ is #P-hard
Finding Safe Plans

**Algorithm**: find a Safe Plan

1. Root variable $u \rightarrow \Pi_i^i_{-u}$
2. Variable $u$ occurs in a subgoal with constant keys $\rightarrow \Pi^D_{-u}$
3. Connected components $\rightarrow$ Join
4. Single subgoal $\rightarrow$ Leaf node

$q(y) :- R(\textbf{x},y,z)$

$\Pi_i^i_{-x}$

$\Pi^D_{-z}$

$R(\textbf{x},y,z)$

$q_1(x^c,y^c) :- R(x^c,y^c,z)$

$a1$  $b$  $p1$  $p2$

$a2$  $b$  $p3$  $p4$

$b$  $1-(1-p1-p2)(1-p3-p4)$
\(R(x), S(x, y), T(y), U(u, y), W('a', u)\)

\[\Pi_{-u}^{D} \bowtie \pi_{-u}^{D} \vdash u\]

\[\Pi_{-y}^{D} \bowtie \vdash y\]

\[\Pi_{-x}^{I} \vdash x\]

\[T^{P}(y) \vdash U^{P}(u, y)\]

\[R^{P}(x) \vdash S^{P}(x, y)\]

Independent project

Disjoint project

Disjoint project
Definitions (in class)

- \( q \vdash g_1, \ldots, g_k \)
- \( S_g(q) = \{g_1, \ldots, g_k\} \)
- \( \text{Vars}(g_i) = \text{all variables of } g_i \)
- \( \text{KVars}(g_i) = \text{all variables in key positions} \)
Algorithm Safe-Eval

• From [Dalvi&S’2007]
• Show on the whiteboard

• Call a query \textit{safe} is the algorithm succeeds
• What are the \textit{unsafe} queries?
Some Unsafe Queries

hd1 = \( R(x), S(x, y), T(y) \)

hd2 = \( R(x, y), S(y) \)

hd3 = \( R(x, y), S(x, y) \)

Variants: hd2^+, hd3^+ (on the whiteboard)
Plan for Proving Dichotomy

Step 1:
• Show that hd1, hd2, hd3 are $\#P$-hard

Step 2:
• Show that every unsafe query can be “rewritten” to hd1, hd2, or hd3
Step 1

• Show (review) in class the hardness of

\[ \text{hd1} = R(x), S(x, y), T(y) \]
Step 1

• Show in class the hardness of

\[ \text{hd}^2 = R(x,y), S(y) \]

Then show \( \text{hd}^2^+ \)
Step 1

• Show in class the hardness of

\[ hd3 = R(x,y), S(x,y) \]

Then show hd3\(^+\)
Step 2

• The rewrite rule $q \Rightarrow q'$ (on the whiteboard)

• $q$ is a final query if for all $q'$ s.t. $q \Rightarrow q'$, $q'$ is safe

• Prove:
  – If $q$ is unsafe, then $\exists q'$ final s.t. $q \Rightarrow^* q'$
  – The only final queries are $hd1$, $hd2^+$, $hd3^+$
  – This completes the dichotomy (why?)
The Complexity of the Complexity

• Deciding if a query is hierarchical is in \( AC^0 \) (in class)

• Deciding if a query is safe is PTIME complete (in class)
Case 3: CQ, independent tables

• Allow selfjoins

• But restrict again to independent tables
Does the query have a safe plan?

\[ q(x) :\text{-} R(a, x, y), \ R(b, x, z), \ S(y, z, u)\]

(a, b = constants)
Does the query have a safe plan?

q :- R(a,x), R(y,b)
Does the query have a safe plan?

Note: no “safe plans” are known! PTIME algorithm for an inversion-free query is given in terms of expressions, not plans. Example:

\[ q : - R(a,x), R(y,b) \]

\[
p(q) = p(R(a,b)) + (1-p(R(a,b))(1-(1-\prod_{y \in \text{Dom}, y \neq a}(1-p(R(y,b)))))(1-\prod_{x \in \text{Dom}, x \neq b}(1-p(R(a,x))))
\]

**Open Problem**: what are the natural operators that allow us to compute inversion-free queries in a database engine?
Does the query have a safe plan?

Find movies with high reviews from Joe and Jim:

\[ q(x) \leftarrow \text{Movie}(x,y), \text{Match}^p(x,r), \quad \text{Review}(r,\text{Joe},s), \quad s > 4 \]
\[ \text{Match}^p(x,r'), \quad \text{Review}(r',\text{Jim},s'), \quad s' > 4 \]

\( \text{Match}^p = \text{probabilistic, tuple independent} \)

\( \text{Movie, Review} = \text{deterministic} \)
The \#P-hard Queries

Hierarchical queries with “inversions”:

\[ hi1 = R(x), S(x,y), S(x',y'), T(y') \]

\[ x \supseteq y \text{ unifies with } x' \subseteq y' \]

\[ hi2 = R(x), S(x,y), S(u,v), S'(u,v), S'(x',y'), T(y') \]

\[ x \supseteq y \text{ unifies with } u \equiv v, \text{ which unifies with } x' \subseteq y' \]
The \#P-hard Queries

A query with a long inversion:

\[
\begin{align*}
\text{hi}_k = & \ R(\underline{x}), \ S_0(\underline{x}, \underline{y}), \\
& \ S_0(\underline{u}_1, \underline{v}_1), \ S_1(\underline{u}_1, \underline{v}_1) \\
& \ S_1(\underline{u}_2, \underline{v}_2), \ S_2(\underline{u}_2, \underline{v}_2), \ldots \\
& \ S_k(\underline{x}', \underline{y}'), \ T(\underline{y}')
\end{align*}
\]
The \#P-hard Queries

Sometimes inversions are exposed only after making a copy of the query

\[ q = R(x,y), R(y,z) \]

\[ R(x,y), R(y,z), R(x',y'), R(y',z') \]
Case 3: CQ, independent tables

Let $q$ be hierarchical
$x \subseteq y$ denotes: $x$ is above $y$ in the hierarchy
$x \equiv y$ denotes: $x \subseteq y$ and $x \subseteq y$

**Definition** An inversion is a chain of unifications:
$x \supseteq y$ with $u_1 \equiv v_1$ with … with $u_n \equiv v_n$ with $x' \subset y'$

**Theorem** For all $q \in \text{CQ}$:
If $q$ is non-hierarchical, or has an inversion* then it is $\#P$-hard
Otherwise it is in $\text{PTIME}$

*without “eraser”: see paper.
<table>
<thead>
<tr>
<th>Query</th>
<th>Complexity</th>
<th>Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>R(a,x), R(y,b)</td>
<td>PTIME</td>
<td>Inversion</td>
</tr>
<tr>
<td>R(a,x), R(x,b)</td>
<td>PTIME</td>
<td>Inversion</td>
</tr>
<tr>
<td>R(x,y), R(y,z)</td>
<td>#P</td>
<td>Inversion</td>
</tr>
<tr>
<td>R(x,y), R(y,z), R(z,u)</td>
<td>#P</td>
<td>Non-hierarchical</td>
</tr>
<tr>
<td>R(x,y), R(y,z), R(z,x)</td>
<td>#P</td>
<td>Non-hierarchical</td>
</tr>
<tr>
<td>R(x,y), R(y,z), R(x,z)</td>
<td>#P</td>
<td>Non-hierarchical</td>
</tr>
</tbody>
</table>
History

• [Graedel, Gurevitch, Hirsch’98]
  – \(L(x,y), R(x,z), S(y), S(z)\) is \#P-hard
    This is non-hierarchical, with a self-join

• [Dalvi&S’2004]
  – \(R(x), S(x,y), T(y)\) is \#P-hard
    This is non-hierarchical, w/o self-joins
  – Without self-joins: non-hierarchical = \#P-hard, and
    hierarchical = PTIME

• [Dalvi&S’2007]
  – \textit{All} non-hierarchical queries are \#P-hard
Discussion

• Dichotomy theorems
  – Remaining open problems ?
  – Extensions ?

• What role (if any) do ‘safe plans’ in practice ?
  – Only some queries have safe plans, so why bother ?