1. Probabilistic Databases (PDBs) and Conjunctive Queries

<table>
<thead>
<tr>
<th>SQL query</th>
<th>Schema</th>
<th>Instances</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>select R.A</td>
<td>R(A,B)</td>
<td>R.A</td>
<td>B</td>
</tr>
<tr>
<td>from R,S,T</td>
<td>S(B,C)</td>
<td>0.2</td>
<td>b</td>
</tr>
<tr>
<td>where R.B=S.B</td>
<td>T(C)</td>
<td>0.4</td>
<td>b</td>
</tr>
</tbody>
</table>

Datalog

\[ Q(x):=R(x,y),S(y,z),T(z) \]

PDBs keep probabilities for tuples
- allows ranking of query results
- allows eager integration of data (think the Web...)  

<table>
<thead>
<tr>
<th>Query complexity</th>
<th>Data complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-hard</td>
<td>PTIME</td>
</tr>
</tbody>
</table>

Main problem of PDBs: III

2. Propagation: a new semantics for ranking over PDBs

Several desirable properties:
- unique score in \([0,1]\): “pseudo-probabilistic”
- always in PTIME
- even expressible in RA
- conservative to PWS for all types of safe queries
- \( \rho \geq \tau \) (propagation \( \geq \) reliability \( \geq \) PWS)
- allows efficient filtering
- semantics inspired by ranking on graphs
- large success of propagation schemes for ranking of nodes in large graphs (e.g. PageRank)

3. Two ‘relatedness’ scores of nodes in networks

- Network reliability:
  - probability that source \( x \) is connected to target \( t \)
  - \( r(t|x) = 0.5 \)

- Network propagation:
  - score of a node recursively depends on parents;
  - independent-of (or parents “pseudo-probabilistic”)
  - \( \rho(x) = \sum_{p(x)} \rho(p) \rho_{PA}(x) = 1 - \prod_{p(x)}(1 - \rho(p)) \rho_{PA}(x) \)
  - \( \rho(t|x) = 0.75 \)

The connection

\( r(t|x) = 0.75 \)  
\( \rho(t|x) = 0.75 \)

Def. “Network dissocation”:
- creating independent copies of the same node

4. Query reliability & propagation on chain queries

- Directed k+1-partite graph
- k-chain query

Reliability

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

Propagation

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

Note that \( q = q' = \pi(\Delta) \)

Def. “Query dissocation”:
- adding variables to subgoals
- Def. “Table dissocation”:
- cross product with active domain

Def. “Safe dissocation”:
- \( \Delta \) is safe if \( q^\Delta \) is safe

Def. “Propagation score”:
- minimum score of all safe dissociations

5. The incidence matrix of a CQ – a useful abstraction

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

\( x \) and \( y \) subgoals in rows, variables in columns

\( \Delta \) can make unsafe queries safe!

6. Partial dissocation order and propagation

Def. “Partial dissocation order”:

\[ \Delta \preceq \Delta' \iff \forall x: \text{Var}(\Delta') \supseteq \text{Var}(\Delta) \]

Theorem 1:

\[ \Delta \preceq \Delta' \iff \rho(\Delta') \geq \rho(\Delta) \]

Def. “Safe dissocation”:

\( \Delta \) is safe if \( q^\Delta \) is safe

Def. “Propagation score”:

minimum score of all safe dissociations

7. Safe dissociations and query plans

Theorem 2:

- isomorphism between safe dissociations and all probabilistic query plans
- \( q_{0} = R(x),S(x,y),T(x,y) \)
- \( q_{1} = R(x),S(x,y),T(x,y) \)
- \( q_{2} = R(x),S(x,y),T(x,y) \)

8. Overview: Query dissociation & graph propagation

<table>
<thead>
<tr>
<th>Networks / graphs</th>
<th>Conjunctive queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network reliability</td>
<td>Query reliability</td>
</tr>
<tr>
<td>– prob. that two nodes are connected</td>
<td>– prob. that query is true in a random world</td>
</tr>
<tr>
<td>– independent of edge direction</td>
<td>– independent of query plan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Propagation score</th>
<th>Dissociation score</th>
</tr>
</thead>
<tbody>
<tr>
<td>– Score propagates from source to target</td>
<td>– A query plan evaluates from leaves to root</td>
</tr>
<tr>
<td>– Dependent on edge direction</td>
<td>– Dependent on choice of dissociation</td>
</tr>
<tr>
<td>Upper bound to reliability</td>
<td>Upper bound to reliability</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Propagation score</th>
<th>Dissociation score</th>
</tr>
</thead>
<tbody>
<tr>
<td>– Minimum over all safe dissociations</td>
<td>– Unique for given query</td>
</tr>
<tr>
<td>– PTIME</td>
<td>– PTIME</td>
</tr>
</tbody>
</table>

9. Experiments on chain queries

- Exact probabilistic inference (Unary/RP)

1. Performance guarantees

Propagation is always fast independent of instance

Project page: http://db.cs.washington.edu/propagation