Approximate Lifted Inference with Probabilistic Databases

Wolfgang Gatterbauer
Based on joint work with Dan Suciu
(Oct 5, 2016)
Why Approximate Lifted Inference?

- First-Order Logic and Probabilities 😊
  
  e.g., \(\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y), \text{weight}=3\)
  
  e.g., \(\text{Q}(z) \leftarrow \text{Smoker}(x,'2009'), \text{Friend}(x,z)\)


Requires grounding and sampling 😞

- Dichotomy results in databases, e.g.:
  
  Dalvi, Suciu [VLDB'04, JACM'12] Fink, Olteanu [PODS'14]

PTIME cases ("liftable") require no grounding \(\rightarrow\) super fast

- How to perform approximate lifted inference for hard cases? 🎯

G., Suciu [VLDB'15, VLDBJ'16]
Lifted Inference (LI) and Approximate LI (ALI)

"reason about multiple individuals... treat (them) as a group"

"exploiting symmetries ... in the relational structure of the model"

\[ xf + xg = x(f + g) \]

Discovering or introducing symmetries is algebraically equivalent to finding efficient factorizations

Approximate Lifted Inference:
Finding approximate factorizations from the relational structure of the model that allow evaluation polynomial in the data size
Roadmap

1. **Theory**: Bounds on the probability of monotone Boolean functions

2. **Practice**: Approximate lifted inference for Self-Join-free conjunctive queries

3. Experiments

4. Outlook
Boolean Functions and Applications

Graphical Models

Probabilistic Databases

Network Reliability

Weighted Model Counting

Possible Worlds Model

Possible Worlds Model

Chavira, Darwiche [AI’00]

$f = x_1 y_1 \lor x_1 y_2 \lor x_2 y_2$

$P[f]$?
Network reliability

\[ f = \text{true iff } s \& t \text{ connected} \]

More general:

Boolean functions

\[ f = \text{path 1 } \lor \text{ path 2 } \lor \text{ path 3} \]

paths are not independent!

\[
\begin{align*}
 f &= x_1 y_1 \lor x_1 y_2 \lor x_2 y_2 \\
 P[f] &= P[x_1]P[y_1 \lor y_2] + P[x_1]P[x_2 y_2] \\
 &= p_1(q_1 \land q_2) + p_1 p_2 q_2 \\
\end{align*}
\]

"independent-or": \(1-(1-q1)(1-q2)\)

Calculating \(P[f]\) for monotone 2DNF is \#P-hard 😞

Provan, Ball [SICOMP’83]
Intuition for Dissociation

Original network

\[ f = x_1 y_1 \lor x_1 y_2 \lor x_2 y_2 \]
\[ P[f] = P[x_1]P[y_1 \lor y_2] + P[\overline{x_1}]P[x_2 y_2] \]
\[ = p_1 (q_1 \otimes q_2) + \overline{p_1} p_2 q_2 \]

"Dissociated" network

\[ f' = x_1 y_1 \lor (x_1' \lor x_2)y_2 \]
\[ = x_1 y_1 \lor (x_1' \lor x_2)y_2 \]
\[ P[f'] = (p_1 q_1) \otimes ((p_1' \otimes p_2) q_2) \]

Calculating \( P[f] \) for read-once formula is in PTIME 😊 Gurvich [1977]

How to choose \( P[x_1], P[x_1'] \) to get upper or lower bounds?
Bounding monotone Boolean formulas by models

\[ f = x_1 y_1 \lor x_1 y_2 \lor x_2 y_2 \]

\[ f = x_1 y_1 \lor x_1 y_2 \lor x_2 y_2 \]
Bounding monotone Boolean formulas by models

\[ f = x_1 y_1 \lor x_1 y_2 \lor x_2 y_2 \]

\[ f' = x_1 y_1 \lor x_1' y_2 \lor x_2 y_2 \]

\[ \mathbb{P}[f'] \geq \mathbb{P}[f]: \text{Oblivious upper bound: } x_1' = 1 \]

\[ f' = x_1 y_1 \lor y_2 \lor x_2 y_2 \]

Read-once, PTIME 😊

\[ \mathbb{P}[f'] \leq \mathbb{P}[f]: \text{Oblivious lower bound: } x_1' = 0 \]

\[ f' = x_1 y_1 \lor x_2 y_2 \]

Read-once, PTIME 😊

*Can we do better (outside standard models)?*

"Oblivous framework": \( p' \) is computed only as function of \( p \), thus uses only "local" information, thus can be fast
Oblivious Bounds for disjunctive Dissociations

How to choose $P[x_1]$, $P[x_1']$ to get upper or lower bounds?

$$f' = x_1y_1 \lor x_1'y_2 \lor x_2y_2$$

original probability
Oblivious Bounds from Models

How to choose \( P[x_1], P[x_1'] \) to get upper or lower bounds?

\[
f' = x_1y_1 \lor x_1'y_2 \lor x_2y_2
\]

Oblivious upper bound: \( x_1' = 1 \)
\[
P[f'] \geq P[f]
\]
or \( x_1 = 1 \)

original probability
Oblivious Bounds from Models

How to choose $P[x_1], P[x_1']$ to get upper or lower bounds?

$f' = x_1 y_1 \lor x_1' y_2 \lor x_2 y_2$

Oblivious upper bound: $x_1' = 1$
$P[f'] \geq P[f]$ or $x_1 = 1$

Oblivious lower bound: $x_1' = 0$
$P[f'] \leq P[f]$ or $x_1 = 0$

What about the rest?
Oblivious Bounds from Dissociations
(an algebraic framework)

How to choose $P[x_1], P[x_1']$ to get upper or lower bounds?

$$f' = x_1 y_1 \lor x_1' y_2 \lor x_2 y_2$$

Oblivious upper bound: $x_1' = 1$

$P[f'] \geq P[f]$ or $x_1 = 1$

$p_1 \geq p, p_1' \geq p$

Oblivious lower bound: $x_1' = 0$

$P[f'] \leq P[f]$ or $x_1 = 0$

$p_1 \otimes p_1' \leq p$

G., Suciu [TODS'14]
Example: Assume all probabilities are 0.5 (Then $P[f]=0.5$)

$$f' = x_1 y_1 \lor x_1' y_2 \lor x_2 y_2$$

$$P[f'] = (p_1 0.5) \otimes ((p_1' \otimes 0.5) 0.5)$$

**Also allows model counting ($\#f = 8$)**

$$\#f' = P[f'] \cdot 2^4$$
Example: Assume all probabilities are 0.5 (Then $P[f]=0.5$)

$$f' = x_1y_1 \lor x_1'y_2 \lor x_2y_2$$

$$P[f'] = (p_1 0.5) \otimes ((p_1' \otimes 0.5) 0.5)$$

Also allows model counting ($\#f = 8$)

$\#f' = P[f'] 2^4$
Oblivious Bounds, Relaxation & Compensation, & Models for Monotone Boolean functions

Variable is split into $d=2$ new variables (similar results hold for any $d$)

<table>
<thead>
<tr>
<th>Oblivious bounds</th>
<th>Upper</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model-based bounds</td>
<td>Upper</td>
<td>Lower</td>
</tr>
<tr>
<td>Relaxation &amp; Comp.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conjunctive D.</th>
<th>Disjunctive D.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f = f_1 \land f_2$</td>
<td>$f = f_1 \lor f_2$</td>
</tr>
<tr>
<td>$f' = f_1[x'/x] \land f_2[x''/x]$</td>
<td>$f' = f_1[x'/x] \lor f_2[x''/x]$</td>
</tr>
</tbody>
</table>

- Oblivious bounds
  - Upper: $p' \cdot p'' \geq p$
  - Lower: $p' \leq p$, $p'' \leq p$

- Model-based bounds
  - Upper: $p' = p$, $p'' = 1$ (optimal)
  - Lower: $p' = p$, $p'' = 0$ (non-optimal)

- Relaxation & Comp.
  - $p' = p$, $p'' = \mathbb{P}[x | f_1]$  

- G., Suciu [TODS'14]
- Fink, Olteanu [ICDT'11]
- Choi, Darwiche [NIPS'09, JSAI-isAI'10]
Oblivious Bounds, Relaxation & Compensation, & Models for Monotone Boolean functions

<table>
<thead>
<tr>
<th>Conjunctive D.</th>
<th>Disjunctive D.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f = f_1 \land f_2$</td>
<td>$f = f_1 \lor f_2$</td>
</tr>
<tr>
<td>$f' = f_1[x_1'/x] \land f_2[x_1'/x]$</td>
<td>$f' = f_1[x_1'/x] \lor f_2[x_1'/x]$</td>
</tr>
</tbody>
</table>

Optimal oblivious bounds

<table>
<thead>
<tr>
<th>Upper</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1 \cdot p_2 = p$</td>
<td>$p_1 \cdot p_2 = p$</td>
</tr>
</tbody>
</table>

Method that allows to upper and lower bound monotone Boolean functions. Upper bounds work very well for DNF.

- **Fink, Olteanu** [ICDT'11]
- **Choi, Darwiche** [NIPS'09, JSAI-isAI'10]
1. **Theory**: Bounds on the probability of monotone Boolean functions

2. **Practice**: Approximate lifted inference for Self-Join-free conjunctive queries

3. Experiments

4. Outlook
Conjunctive Queries & Probabilistic Databases (PDBs)

(1) Query in SQL

```
SELECT distinct T.C
FROM R,S,T
WHERE R.A=S.A and S.B=T.B
```

(2) Query in Datalog

```
Q(z):- R(x), S(x,y), T(y,z)
```

(3) Incidence matrix for SJ-free CQs

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>o</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>o</td>
<td>o</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

(4) Results

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>S</th>
<th>A</th>
<th>B</th>
<th>T</th>
<th>B</th>
<th>C</th>
<th>Q</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td></td>
<td>a</td>
<td>b</td>
<td></td>
<td>b</td>
<td>e</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7</td>
<td></td>
<td>0.7</td>
<td>0.8</td>
<td></td>
<td>0.8</td>
<td>0.41</td>
<td>0.39</td>
<td></td>
</tr>
</tbody>
</table>

Promise of PDBs: ranking of output, due to uncertainty of input

Query complexity
- DBs: NP-hard
- PDBs: \( \geq \#P \) hard

Data complexity
- DBs: \( \text{PTIME} \)
- PDBs: \( \#P \) hard

Problem of PDBs: ranking is hard

- Vardi [STOC’82]
- Dalvi, Suciu [VLDB’04]
Background: Evaluating Probabilistic Queries

\[ Q_1: - R(x), S(x,y), T(x,y) \]

\[
P[Q] = r_1 s_3 t_6 \lor r_1 s_4 t_7 \lor r_2 s_5 t_8
\]

\[
= r_1 (s_3 t_6 \lor s_4 t_7) \lor r_2 (s_5 t_8)
\]

PTIME 😊
"hierarchical"

Incidence matrix

\[
\begin{array}{ccc}
  & x & y \\
 R & \circ & \circ \\
 S & \circ & \circ \\
 T & \circ & \circ \\
\end{array}
\]

Read-Once formula

\[ Q_2: - R(x), S(x,y), T(y) \]

\[
P[Q] = r_1 s_3 t_6 \lor r_1 s_4 t_7 \lor r_2 s_5 t_7
\]

\[
= r_1 (s_3 t_6 \lor s_4 t_7) \lor r_2 (s_5 t_7)
\]

#P hard 😞
not "hierarchical"

\[
\begin{array}{ccc}
  & x & y \\
 R & \circ & \circ \\
 S & \circ & \circ \\
 T & \circ & \circ \\
\end{array}
\]

Dalvi, Suciu [VLDB’04]
The idea: Dissociation

**Q₁:** \(- R(x), S(x,y), T(x,y)\)

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>S</th>
<th>A</th>
<th>B</th>
<th>T</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>r₁</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>b</td>
<td>a</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>r₂</td>
<td>d</td>
<td>a</td>
<td>c</td>
<td>a</td>
<td>c</td>
<td>d</td>
<td>c</td>
</tr>
</tbody>
</table>

\[ P[Q] = r₁s₃t₆ \lor r₁s₄t₇ \lor r₂s₅t₈ \]
\[ = r₁(s₃t₆ \lor s₄t₇) \lor r₂(s₅t₈) \]

**Q₂:** \(- R(x), S(x,y), T(y)\)

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>S</th>
<th>A</th>
<th>B</th>
<th>T</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>r₁</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>r₂</td>
<td>d</td>
<td>a</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td></td>
</tr>
</tbody>
</table>

\[ P[Q] = r₁s₃t₆ \lor r₁s₄t₇ \lor r₂s₅t₈ \]

**PTIME 😊
"hierarchical"**

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>o</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>T</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

[Diagram of hierarchical structure with nodes labeled as follows: R(x), S(x,y), T(x,y)]
The idea: Dissociation

$Q_1 \vdash R(x), S(x,y), T(x,y)$

$$P[Q] = r_1 s_3 t_6 \lor r_1 s_4 t_7 \lor r_2 s_5 t_8$$

$= r_1 (s_3 t_6 \lor s_4 t_7) \lor r_2 (s_5 t_8)$

$PTIME \smile$

"hierarchical"

$P[Q] = \pi_p^{x} \setminus (\pi_p^{y} S(x,y) \otimes \pi_p^{y} T(x,y))$

$R^\Delta \vdash R(x), S(x,y), T(x,y)$

$$P[Q^\Delta] = r_1 s_3 t_6 \lor r_1 s_4 t_7 \lor r_2 s_5 t_8'$$

$= r_1 (s_3 t_6 \lor s_4 t_7) \lor r_2 s_5 t_8'$

$PTIME \smile$

"hierarchical"

$P[Q^\Delta] = \pi_p^{x} R(x) \setminus (\pi_p^{y} S(x,y) \otimes \pi_p^{y} T(x,y))$

$\bigcirc$

$\otimes$

$R^\Delta$ dissociation of tuples

Read-Once formula $\smile$
The idea: Dissociation

**Q** \( \Delta \): \(- R(x, y), S(x, y), T(y) \)

2

PTIME 😊

"hierarchical"

Query

Dissociation

P[Q]\( \Delta \) = \( r_1 s_3 t_6 \lor r_1' s_4 t_7 \lor r_2 s_5 t_7 \)

\( = r_1 s_3 t_6 \lor (r_1' s_4 \lor r_2 s_5) t_7 \)


**Q** \( \Delta \): \(- R(x), S(x, y), T(x, y) \)

1

PTIME 😊

"hierarchical"

Query

Dissociation

P[Q] = \( r_1 s_3 t_6 \lor r_1 s_4 t_7 \lor r_2 s_5 t_7' \)

\( = r_1 (s_3 t_6 \lor s_4 t_7) \lor r_2 s_5 t_7' \)

Read-Once formula 😊

dissociation of tuples

Can be evaluated with a DMBS
Partial Dissociation Order and Propagation

Def. “Partial dissociation order” \( \leq \):
\[
\Delta \leq \Delta' \iff \forall \text{relations } R : \text{Var}(R^\Delta) \supseteq \text{Var}(R^{\Delta'})
\]

Theorem 1:
\[
\Delta \leq \Delta' \iff P[Q^\Delta] \leq P[Q^{\Delta'}]
\]

Def. “PTIME dissociation”:
\[
\Delta \text{ is PTIME} \iff Q^\Delta \text{ is PTIME}
\]

Def. “Propagation score”:
minimum prob. of all PTIME dissociations
\[
\rho[Q] := \text{MIN}_{\Delta : \text{safe}} P[Q^\Delta]
\]
Corollary: Propagation is minimum over all minimal plans: $\rho[Q] = \min_{\Delta : \text{minimal in } \leq} P[P^\Delta]$
Proposition:

\[ \rho[Q] = \min_{\Delta : \text{minimal in } \leq P[P^\Delta]} \]

Theorem 2: Isomorphism between safe dissociations and probabilistic query plans:

\[ P[Q^\Delta] = P[P^\Delta] \]

Method that allows to upper bound any hard Self-Join-free Conjunctive Query with any standard relational database 😊

Corollary: Propagation is minimum over all minimal plans:

\[ \rho[Q] = \min_{\Delta : \text{minimal in } \leq P[P^\Delta]} \]
1. Theory: Bounds on the probability of monotone Boolean functions

2. Practice: Approximate lifted inference for Self-Join-free conjunctive queries

3. Experiments

4. Outlook
## Questions for Experiments

<table>
<thead>
<tr>
<th>Quality (AP@10)</th>
<th>Efficiency (Time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Dissociation</td>
<td>✓</td>
</tr>
<tr>
<td>2. Monte Carlo</td>
<td>✓</td>
</tr>
<tr>
<td>3. Exact Probabilistic Inference</td>
<td>✓</td>
</tr>
<tr>
<td>4. Ranking by Lineage Size (# of clauses)</td>
<td>✓</td>
</tr>
<tr>
<td>5. Deterministic Query Evaluation</td>
<td>✓</td>
</tr>
</tbody>
</table>

- **MC(10k), MC(1k), ...**
- **serves as ground truth, if possible**
- **SampleSearch**
  - **Gogate, Dechter [AI'11]**

- **Average Precision (ranking)**
Experimental Setup

1: TPC-H random database

Supplier(s_suppkey, s_nationkey)
PartSupp(ps_suppkey, ps_partkey)
Part(p_partkey, p_name)

(10k tuples)
(800k tuples)
(200k tuples)

We add a random probability to each tuple with avg[p_i] as parameter

2. Parameterized test query

SELECT distinct s_nationkey
FROM Supplier, Partsupp, Part
WHERE s_suppkey = ps_suppkey
and ps_partkey = p_partkey
and s_suppkey <= $1
and p_name like $2

25 nations
500 – 10k

"Which nations (as determined by the attribute nationkey) are most likely to have suppliers with suppkey ≤ $1 that supply parts with a name like $2?"

Q(a) :- S(s, a), PS(s, u), P(u, n), s ≤ $1, n like $2

3. PostgreSQL, Translation happens in Java
Experiments: results on synthetic TPC-H data

**Time (sec)**

- **Exact inference**
- **Approximate Lifted inference**
- **Lineage query Standard SQL**
- **MC(1k)**

**Ranking quality (AP@10)**

- **A.L.I.**
- **MC(1k)**

- **Ranking by lineage size**

<table>
<thead>
<tr>
<th>max[Lineage size]</th>
<th>10</th>
<th>100</th>
<th>1k</th>
<th>10k</th>
<th>50k</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1k</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10k</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50k</td>
<td>0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># of MC samples</th>
<th>10</th>
<th>30</th>
<th>100</th>
<th>300</th>
<th>1k</th>
<th>3k</th>
<th>10k</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.73</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.89</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.94</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.96</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. **Theory**: Bounds on the probability of monotone Boolean functions

2. **Practice**: Approximate lifted inference for Self-Join-free conjunctive queries

3. Experiments

4. Outlook
Approximation-aware learning & inference

Closely related to approximate message passing methods, convex relaxations. See e.g., Wainwright [JMLR'06] Gomely+[TACL'15]
Important Open Problems

1. Self-joins

"Find students who take class1 and class2."

Q(name) :- Student(sid, name), Enrolled(sid, 'class1'), Enrolled(sid, 'class2')

2. Disjoint-independent databases

"A student can take either class 201 or class 202."

3. Learning the probabilities from predictions
Take-aways

1. Probability of Boolean Functions
   • Upper and Lower bounds for monotone Boolean functions by dissociation
   • Improve on model-based bounds

2. Approximate Lifted Inference
   • for Self-Join-free Conjunctive Queries
   • Apply dissociation at query level in multiple ways, then pick "best"
   • Generalizes all PTIME cases
   • Fast and good for ranking

Thanks 😊