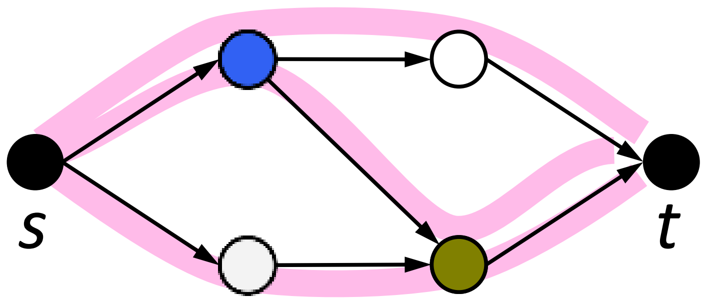
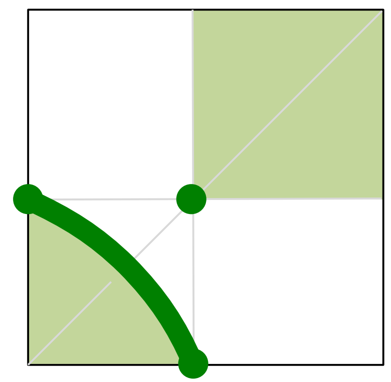
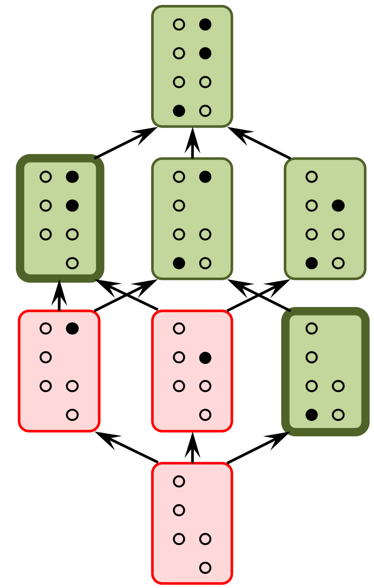


Approximate Lifted Inference with Probabilistic Databases

Wolfgang Gatterbauer

Based on joint work with Dan Suciu

(Oct 5, 2016)


Why Approximate Lifted Inference?

- First-Order Logic and Probabilities ☺

  e.g., Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y), weight=3

  e.g., Q(z) :− Smoker(x,'2009'), Friend(x,z)

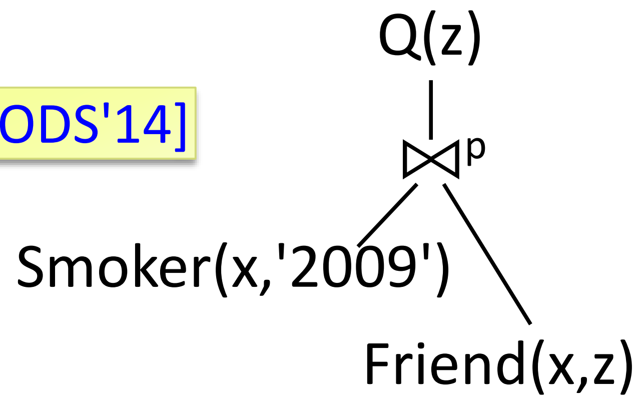
  Russell [CACM'15] Richardson, Domingos [ML'06] Kautz, Singla [CACM'16]

  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 → 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"

Poole [IJCAI'03]

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

V.d.Broeck, Darwiche [NIPS'13]

$x\,f + x\,g = x\,(f + g)$ symmetric in f and 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

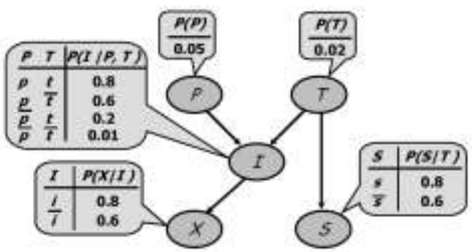
2. **Practice**: Approximate lifted inference for Self-Join-free conjunctive queries

3. Experiments

4. Outlook

Boolean Functions and Applications

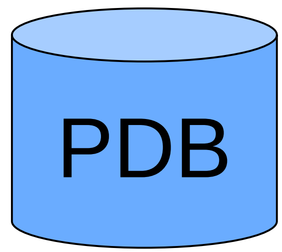
Graphical Models



Weighted
Model Counting

Chavira, Darwiche [AI'00]

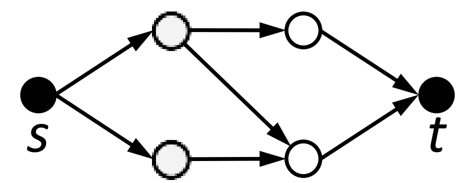
Probabilistic Databases



Possible
Worlds Model

...

Network Reliability

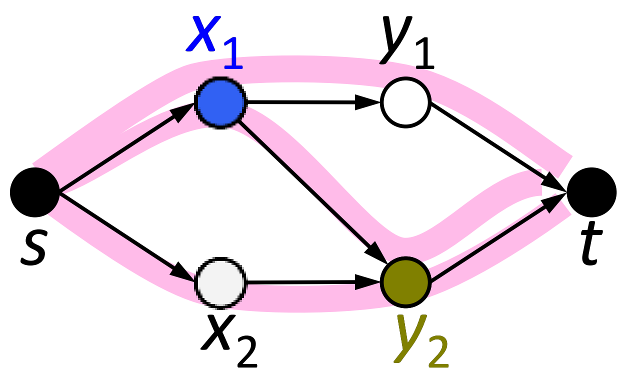


Possible
Worlds Model

Boolean Functions $f = x_1y_1 \vee x_1y_2 \vee x_2y_2$ $P[f]$?

Network reliability

f=true iff s & t connected



$P[x_i] = p_i$, $P[y_j] = q_j$

Boolean functions

f = path 1 V path 2 V path 3

paths are not independent!

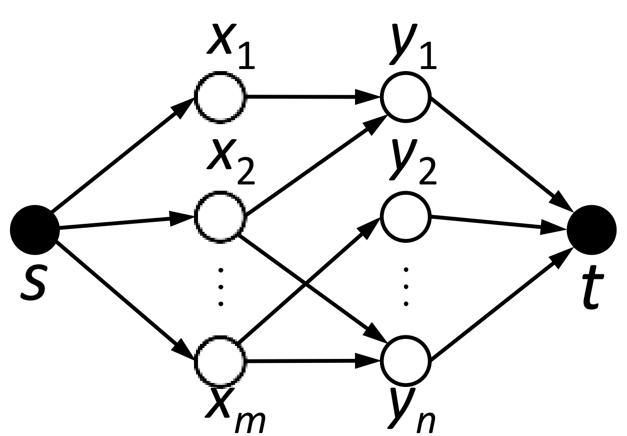
$f = x_1 y_1 \vee x_1 y_2 \vee x_2 y_2$ "not"

$P[f] = P[x_1]P[y_1 \vee y_2] + P[\overline{x_1}]P[x_2 y_2]$

$= p_1 \ (q_1 \otimes q_2) + \overline{p}_1 \ p_2 q_2$

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

More general:

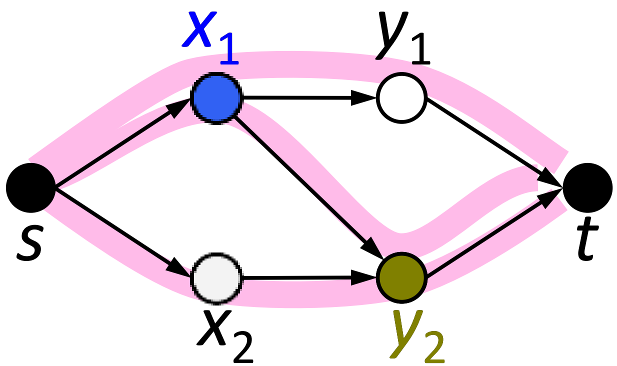


$$f = \bigvee_{(i,j) \in E} x_i y_j \qquad E \subseteq m \times n$$

Calculating P[f] for monotone 2DNF is #P-hard ☹ Provan, Ball [SICOMP'83]

Intuition for Dissociation

Original network

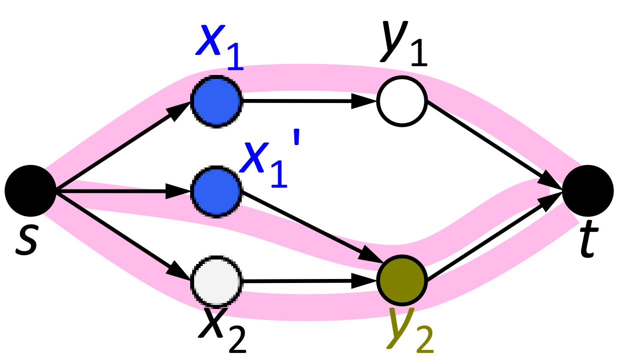


$$f = x_1 y_1 \lor x_1 y_2 \lor x_2 y_2$$

$$P[f] = P[x_1]P[y_1 \lor y_2] + P[\overline{x}_1]P[x_2 y_2]$$

$$= p_1 \ (q_1 \otimes q_2) + \overline{p}_1 \ p_2 q_2$$

"Dissociated" network



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

$$= x_1 y_1 \lor (x_1' \lor x_2) y_2$$

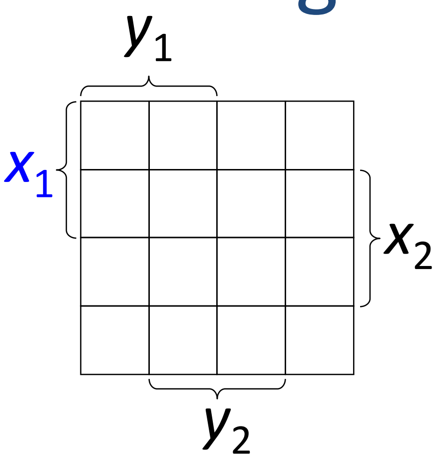
$$P[f'] = (p_1 q_1) \otimes ((p_1' \otimes p_2)\ q_2)$$

Serial-parallel graph

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?*

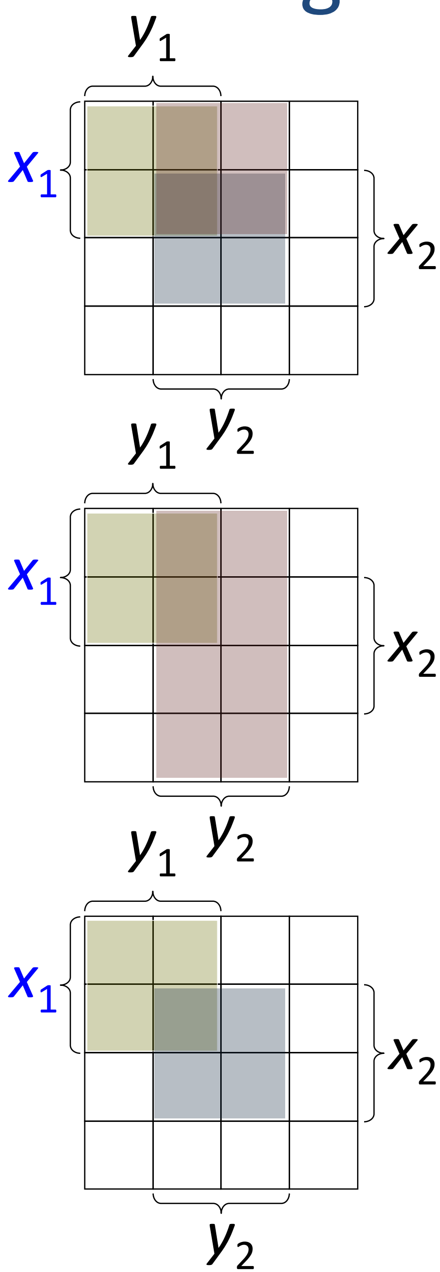
Bounding monotone Boolean formulas by models



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

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

Bounding monotone Boolean formulas by models



$f = x_1y_1 \lor x_1y_2 \lor x_2y_2$

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

P[f'] ≥ **P**[f]: Oblivious upper bound: $x_1' = 1$

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

Read-once, PTIME ☺

P[f'] ≤ **P**[f]: Oblivious lower bound: $x_1' = 0$

$f' = x_1y_1 \lor x_2y_2$

Read-once, PTIME ☺

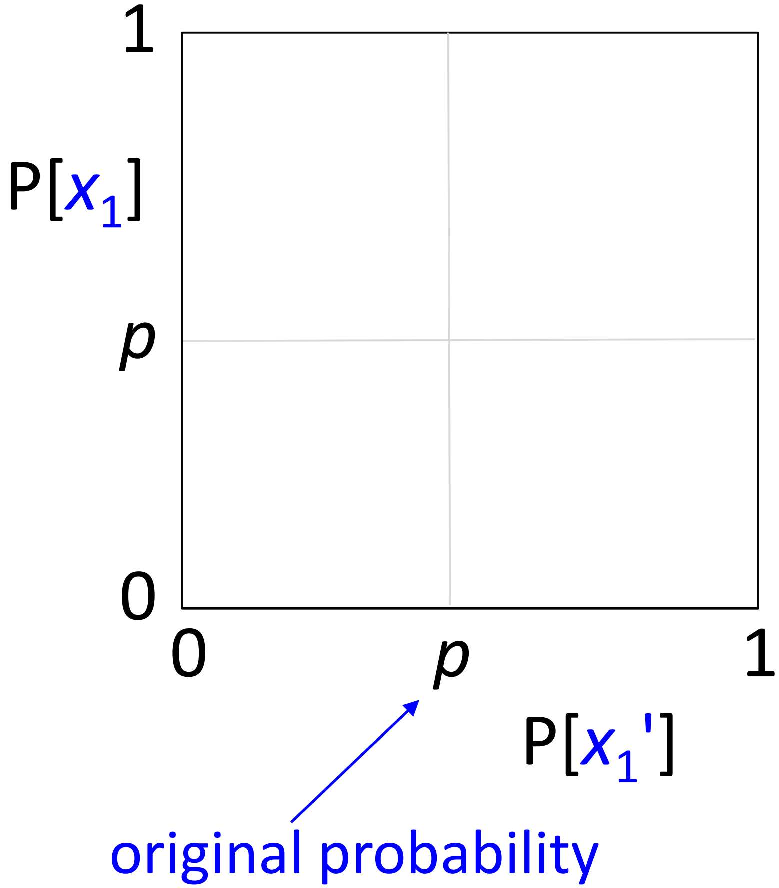
Can we do better (outside standard models)?

"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?*

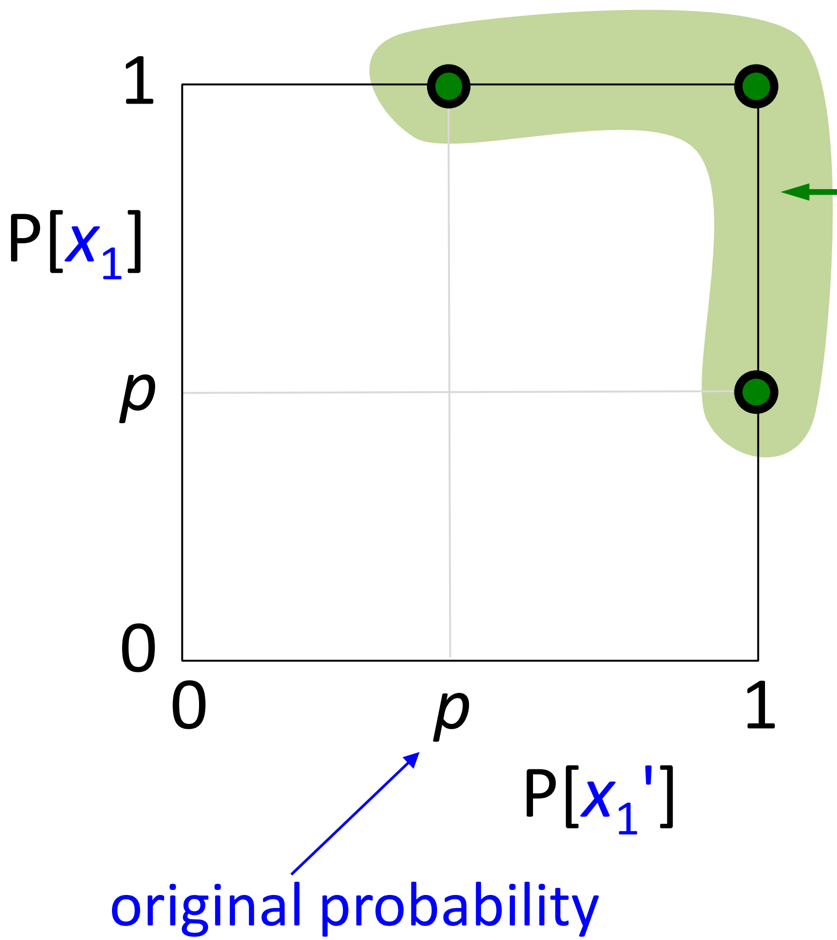
$$f' = x_1 y_1 \lor x_1' y_2 \lor x_2 y_2$$



Oblivious Bounds from Models

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

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



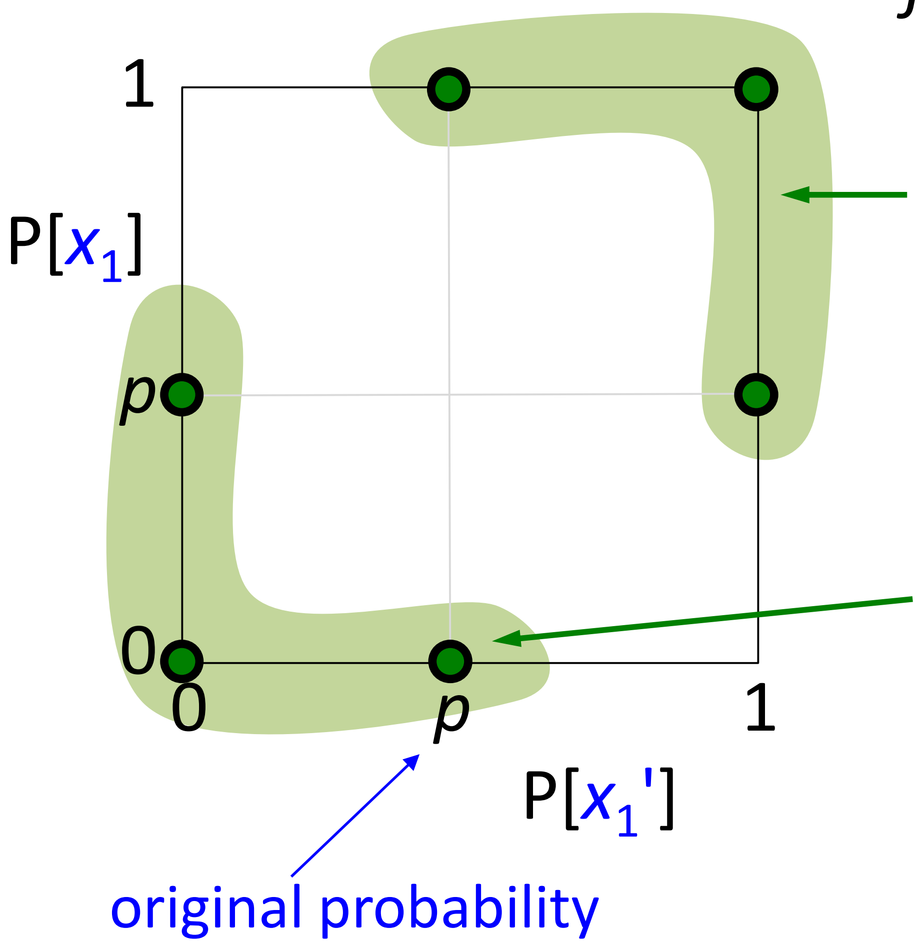
Oblivious upper bound: $x_1' = 1$

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

Oblivious Bounds from Models

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

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



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

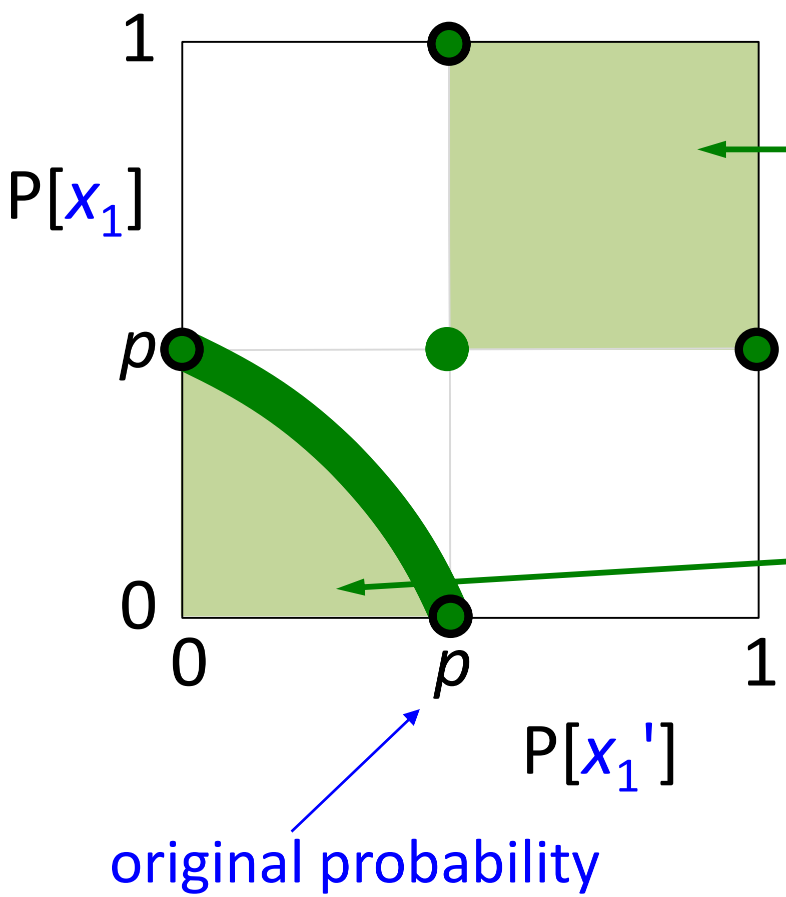
Oblivious lower bound: $x_1' = 0$
P[f'] ≤ P[f] or $x_1 = 0$

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?*

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



Oblivious upper bound: $x_1' = 1$ or $x_1 = 1$

$P[f'] \geq P[f]$

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

Oblivious lower bound: $x_1' = 0$ or $x_1 = 0$

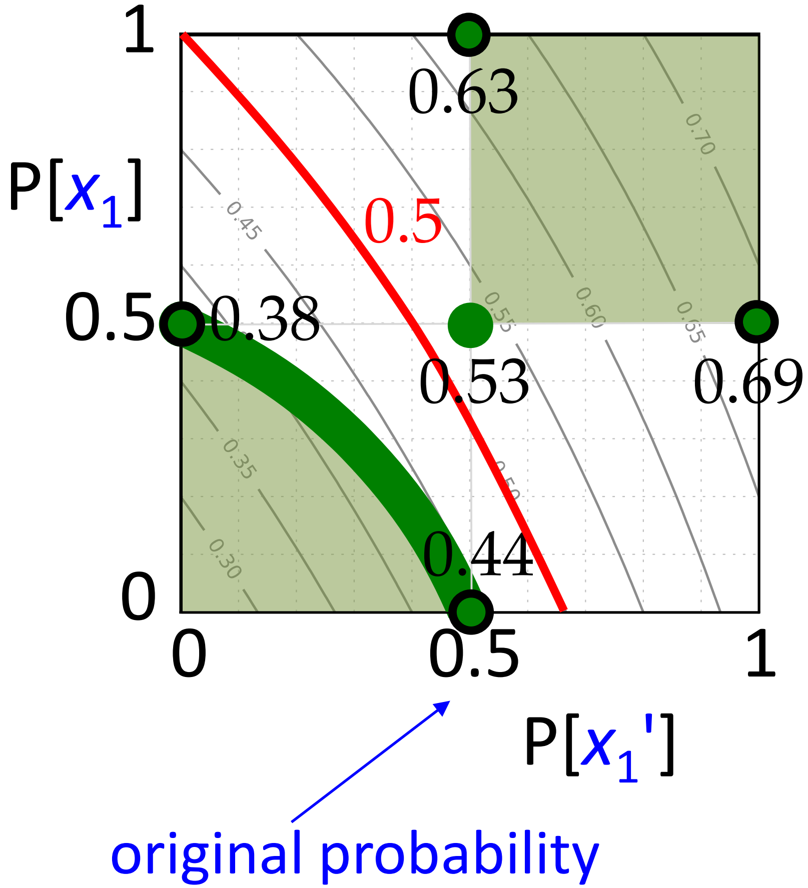
$P[f'] \leq P[f]$

$p_1 \otimes p_1' \leq p$

G., Suciu [TODS'14]

Oblivious Bounds from Dissociations: Example

<u>Example:</u> *Assume all probabilities are 0.5 (Then P[f]=0.5)*



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

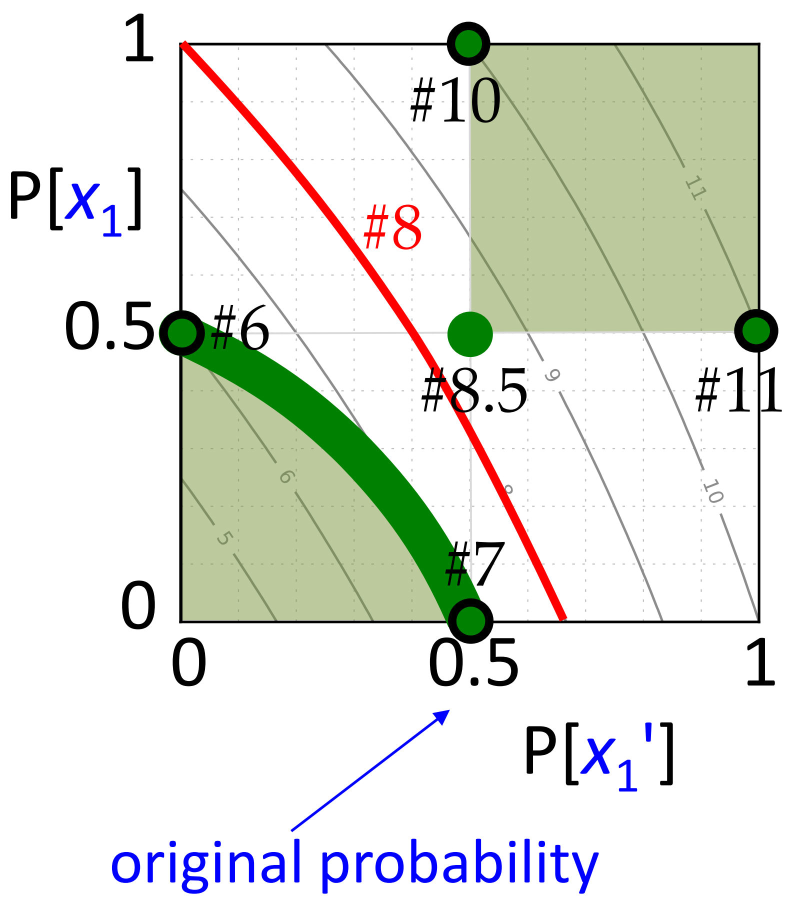
$P[f'] = (p_1 0.5) \otimes ((p_1' \otimes 0.5) 0.5)$

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

$\#f' = P[f']\, 2^4$

Oblivious Bounds from Dissociations: Example

<u>Example:</u> *Assume all probabilities are 0.5 (Then P[f]=0.5)*



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

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

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

$\#f' = P[f']\, 2^4$

Oblivious Bounds, Relaxation & Compensation, & Models for Monotone Boolean functions

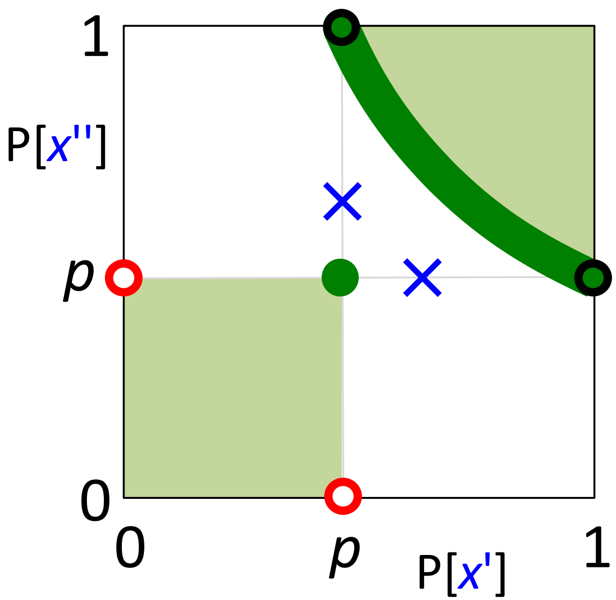
Variable is split into d=2 new variables (similar results hold for any d)

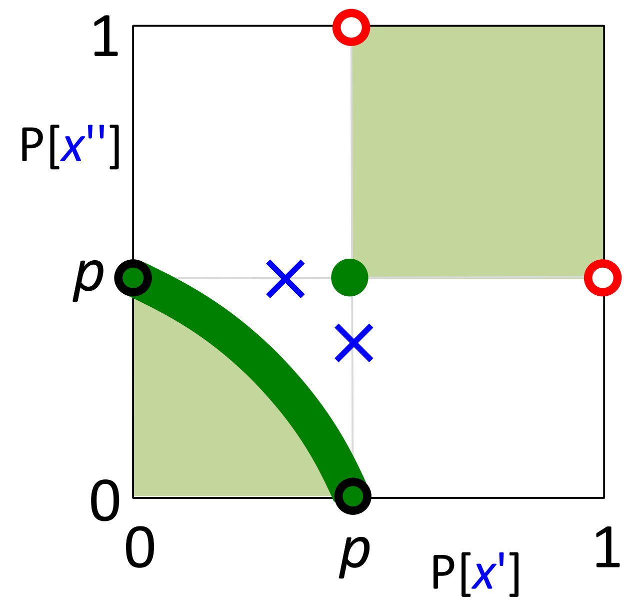
		Conjunctive D.	Disjunctive D.
		$f = f_1 \wedge f_2$ $f' = f_1[x'/x] \wedge f_2[x''/x]$	$f = f_1 \vee f_2$ $f' = f_1[x'/x] \vee f_2[x''/x]$
● **Oblivious bounds**	Upper	$p' \cdot p'' \geq p$	$p' \geq p$, $p'' \geq p$
	Lower	$p' \leq p$, $p'' \leq p$	$p' \otimes p'' \leq p$
○ **Model-based bounds**	Upper	$p'=p$, $p''=1$ (optimal)	$p'=p$, $p''=1$ (non-optimal)
	Lower	$p'=p$, $p''=0$ (non-optimal)	$p'=p$, $p''=0$ (optimal)
× **Relaxation & Comp.**		$p'=p$, $p''=\mathrm{P}[x \mid f_1]$	$p'=p$, $p''=\mathrm{P}[x \mid \overline{f_1}]$

● G., Suciu [TODS'14]

○ Fink, Olteanu [ICDT'11]

× Choi, Darwiche [NIPS'09, JSAI-isAI'10]





Oblivious Bounds, Relaxation & Compensation, & Models for Monotone Boolean functions

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

	Conjunctive D.	Disjunctive D.
	$f = f_1 \wedge f_2$	$f = f_1 \vee f_2$
	$f' = f_1[x_1'/x] \wedge f_2[x_1'/x]$	$f' = f_1[x_1'/x] \vee f_2[x_1'/x]$

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]

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

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
```

Join

Schema

R(A)
S(A,B)
T(B,C)

(2) Query in Datalog

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

Instance

R

	A
0.5	a
0.7	d

S

	A	B
0.7	a	b
0.8	a	c
0.7	d	d

T

	B	C
0.7	b	e
0.8	c	e
0.8	d	f

Results

Q

	C
0.41	e
0.39	f

Independent tuples

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

(3) Incidence matrix for SJ-free CQs

	x	y	z
R	○		
S	○	○	
T		○	○

	DBs	PDBs
Query complexity	NP-hard	≥#P hard
Data complexity	**PTIME ☺**	**#P hard ☹**

Vardi [STOC'82]

Dalvi, Suciu [VLDB'04]

Problem of PDBs: ranking is hard

Background: Evaluating Probabilistic Queries

Q_1:- R(x), S(x,y), T(x,y)

R	A
r_1	a
r_2	d

S	A	B
s_3	a	b
s_4	a	c
s_5	d	c

T	A	B
t_6	a	b
t_7	a	c
t_8	d	c

$$P[Q] = r_1 s_3 t_6 \vee r_1 s_4 t_7 \vee r_2 s_5 t_8$$
$$= r_1(s_3 t_6 \vee s_4 t_7) \vee r_2(s_5 t_8)$$

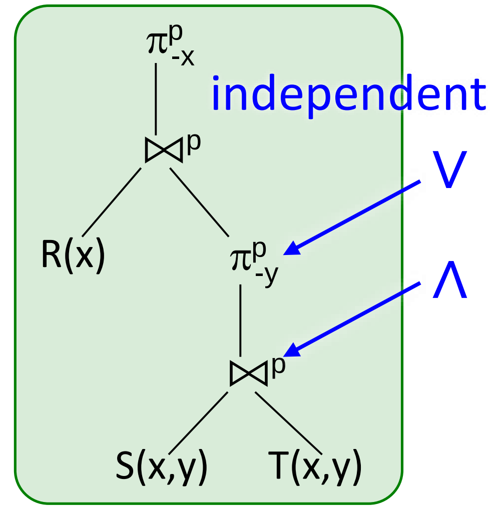
Read-Once formula

PTIME ☺

"hierarchical"

Incidence matrix

	x	y
R	○	
S	○	○
T	○	○



probabilistic query plan

Q_2:- R(x), S(x,y), T(y)

R	A
r_1	a
r_2	d

S	A	B
s_3	a	b
s_4	a	c
s_5	d	c

T	B
t_6	b
t_7	c

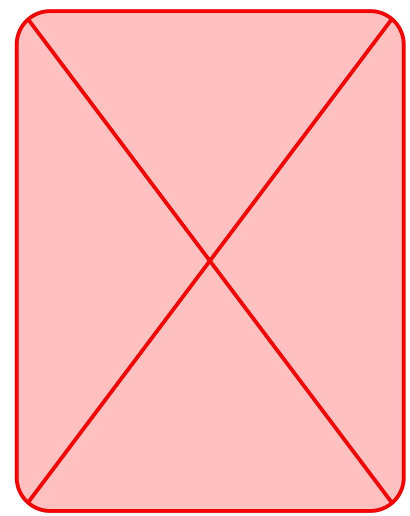
$$P[Q] = r_1 s_3 t_6 \vee r_1 s_4 t_7 \vee r_2 s_5 t_7$$

NO Read-Once formula

#P hard ☹

not "hierarchical"

	x	y
R	○	
S	○	○
T		○



Dalvi, Suciu [VLDB'04]

The idea: Dissociation

Q_1:- R(x), S(x,y), T(x,y)

PTIME ☺

"hierarchical"

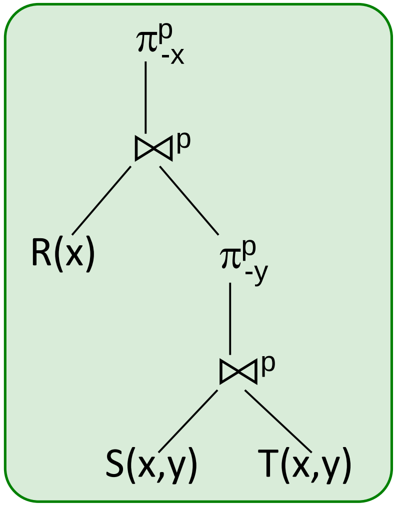
R	A
r_1	a
r_2	d

S	A	B
s_3	a	b
s_4	a	c
s_5	d	c

T	A	B
t_6	a	b
t_7	a	c
t_8	d	c

$$P[Q] = r_1 s_3 t_6 \vee r_1 s_4 t_7 \vee r_2 s_5 t_8$$
$$= r_1(s_3 t_6 \vee s_4 t_7) \vee r_2(s_5 t_8)$$

	x	y
R	○	
S	○	○
T	○	○



Q_2:- R(x), S(x,y), T(y)

R	A
r_1	a
r_2	d

S	A	B
s_3	a	b
s_4	a	c
s_5	d	c

T	B
t_6	b
t_7	c

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

	x	y
R	○	
S	○	○
T		○

The idea: Dissociation

Q_1:- R(x), S(x,y), T(x,y)

PTIME ☺

"hierarchical"

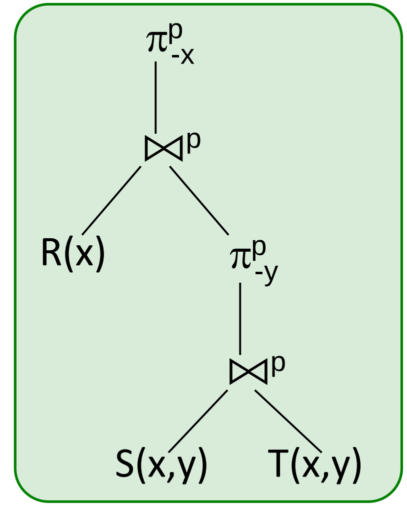
R	A
r_1	a
r_2	d

S	A	B
s_3	a	b
s_4	a	c
s_5	d	c

T	A	B
t_6	a	b
t_7	a	c
t_8	d	c

$$P[Q] = r_1 s_3 t_6 \vee r_1 s_4 t_7 \vee r_2 s_5 t_8$$
$$= r_1(s_3 t_6 \vee s_4 t_7) \vee r_2(s_5 t_8)$$

	x	y
R	○	
S	○	○
T	○	○



Q^{Δ}_2:- R(x), S(x,y), T(x,y)

PTIME ☺

"hierarchical"

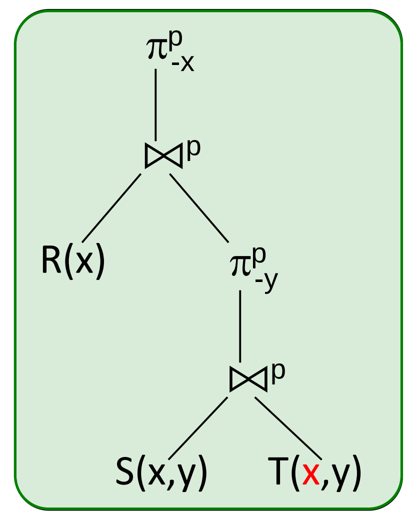
R	A
r_1	a
r_2	d

S	A	B
s_3	a	b
s_4	a	c
s_5	d	c

T	A	B
t_6	a	b
t_7	a	c
t_7'	d	c

Query Dissociation

	x	y
R	○	
S	○	○
T	●	○



$$P[Q^{\Delta}] = r_1 s_3 t_6 \vee r_1 s_4 t_7 \vee r_2 s_5 t_7'$$
$$= r_1(s_3 t_6 \vee s_4 t_7) \vee r_2 s_5 t_7'$$

Read-Once formula ☺

dissociation of tuples

The idea: Dissociation

$Q^{\Delta'}_2$:- R(x,y), S(x,y), T(y)

2

PTIME ☺

"hierarchical"

Query Dissociation

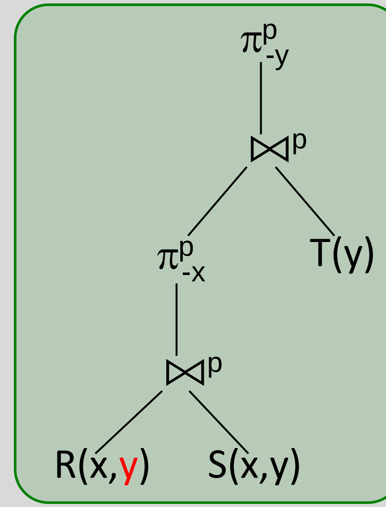
R	A	B
r_1	a	b
r_1'	a	c
r_2	d	c

S	A	B
s_3	a	b
s_4	a	c
s_5	d	c

T	B
r_6	b
r_7	c

	x	y
R	○	●
S	○	○
T		○

$$P[Q^{\Delta'}] = r_1 s_3 t_6 \vee r_1' s_4 t_7 \vee r_2 s_5 t_7$$
$$= r_1 s_3 t_6 \vee (r_1' s_4 \vee r_2 s_5) t_7$$



Q^{Δ}_2:- R(x), S(x,y), T(x,y)

1

PTIME ☺

"hierarchical"

Query Dissociation

R	A
r_1	a
r_2	d

S	A	B
s_3	a	b
s_4	a	c
s_5	d	c

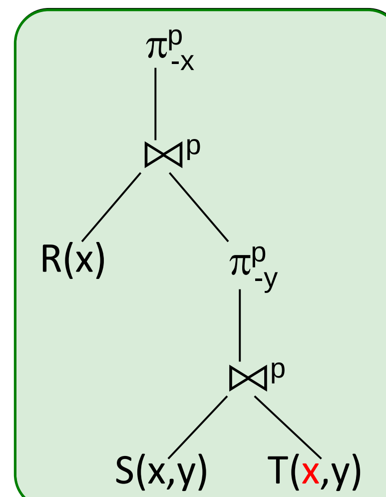
T	A	B
t_6	a	b
t_7	a	c
t_7'	d	c

	x	y
R	○	
S	○	○
T	●	○

$$P[Q^{\Delta}] = r_1 s_3 t_6 \vee r_1 s_4 t_7 \vee r_2 s_5 t_7'$$
$$= r_1 (s_3 t_6 \vee s_4 t_7) \vee r_2 s_5 t_7'$$

Read-Once formula ☺

dissociation of tuples



Can be evaluated with a DMBS

Partial Dissociation Order and Propagation

Q_3:- R(x), S(x), T(x,y), U(y)

Def. "Partial dissociation order" $\leq$:

$$\Delta \leq \Delta' \iff \forall \text{relations } R : Var(R^{\Delta}) \supseteq Var(R^{\Delta'})$$

Theorem 1:

$$\Delta \leq \Delta' \iff P[Q^{\Delta}] \leq P[Q^{\Delta'}]$$

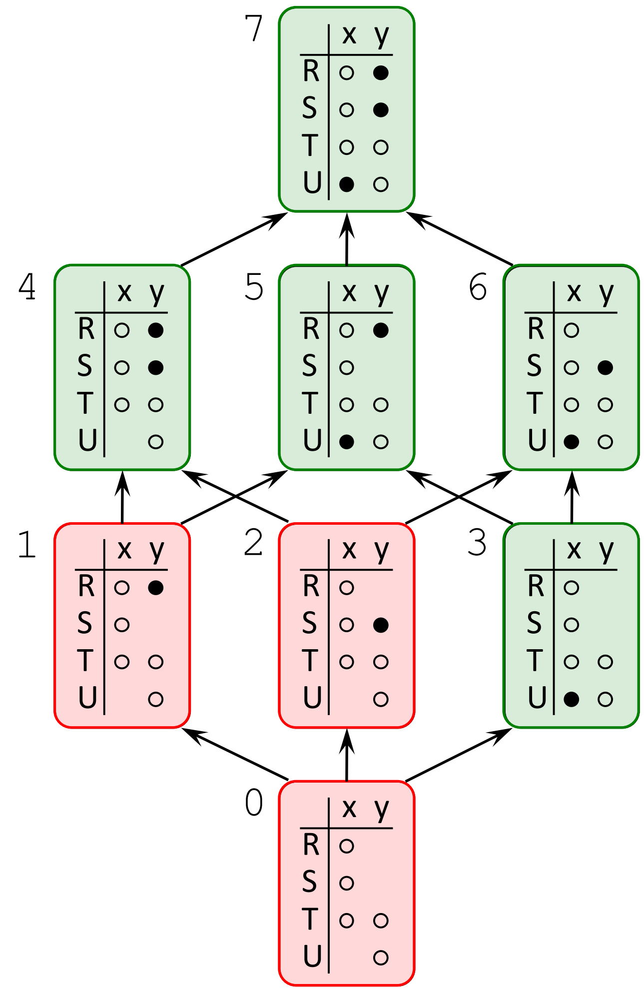
Def. "PTIME dissociation":

$$\Delta \text{ is PTIME} \iff Q^{\Delta} \text{ is PTIME}$$

Def. "Propagation score":

minimum prob. of all PTIME dissociations

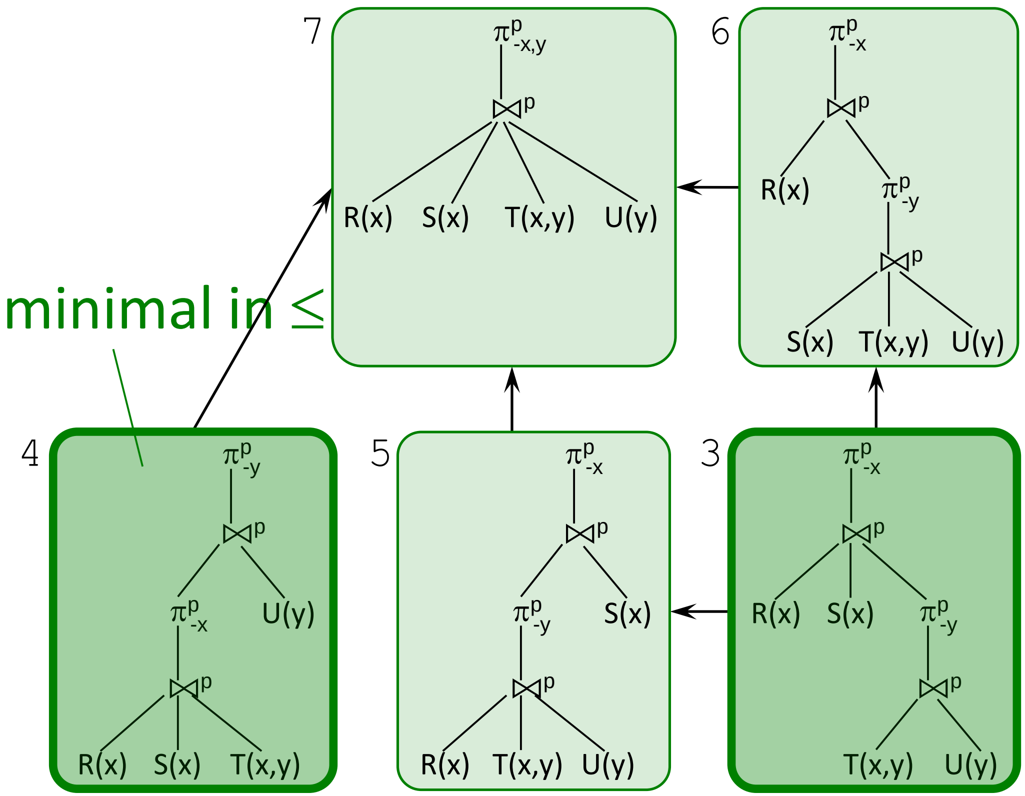
$$\rho[Q] := \text{MIN}_{\Delta \,:\, \text{safe}} \; P[Q^{\Delta}]$$

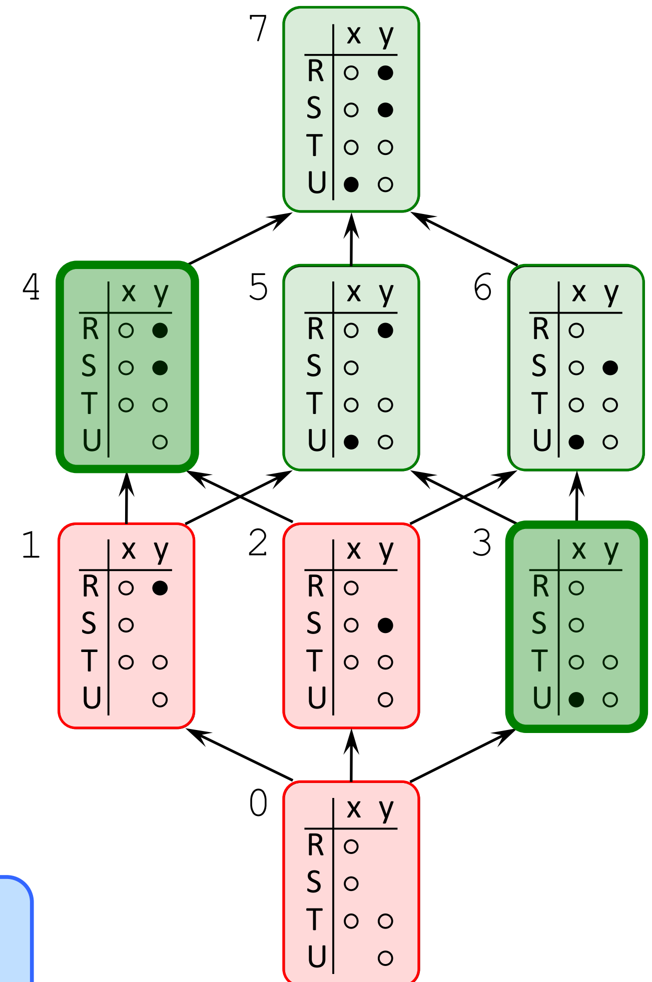


Partial Dissociation Order and Propagation

Q_3:- R(x), S(x), T(x,y), U(y)

Theorem 2: Isomorphism b/w PTIME dissociations and probabilistic query plans: $P[Q^{\Delta}] = P[P^{\Delta}]$



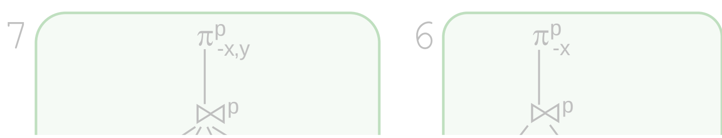


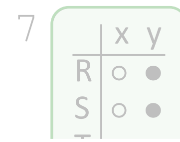
Corollary: Propagation is minimum over all minimal plans: $\rho[Q] = \text{MIN}_{\Delta\text{ : minimal in} \leq} P[P^{\Delta}]$

Partial Dissociation Order and Propagation

Theorem 2: Isomorphism b/w safe dissociations and probabilistic query plans: $P[Q^{\Delta}] = P[P^{\Delta}]$

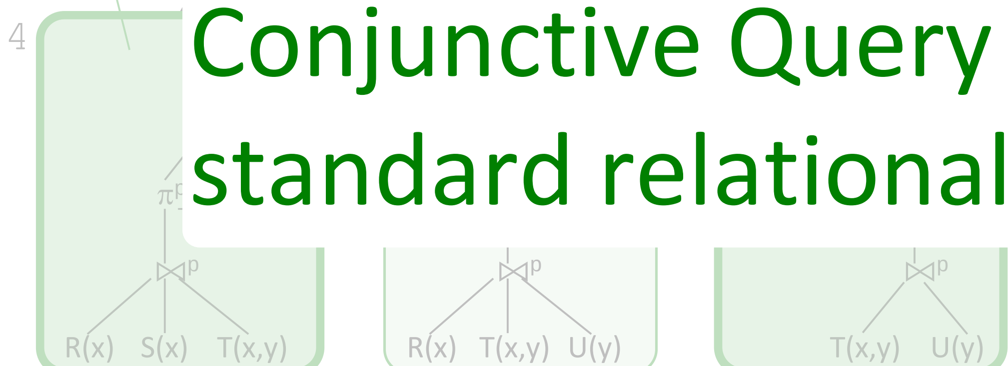
Q_3:- R(x), S(x), T(x,y), U(y)

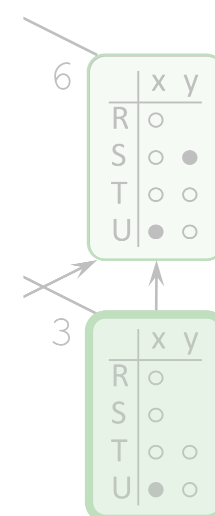


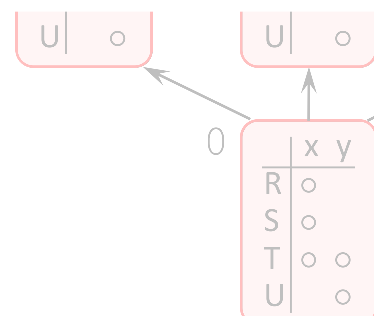


minima

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: $\rho[Q] = \text{MIN}_{\Delta\,:\,\text{minimal in} \leq} P[P^{\Delta}]$

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

Questions for Experiments

Average Precision (ranking) ← Quality (AP@10)

	Quality (AP@10)	**Efficiency** (Time)
1. Dissociation	✓	✓
2. Monte Carlo	MC(10k), MC(1k), ...	✓
3. Exact Probabilistic Inference	serves as ground truth, if possible ...	SampleSearch Gogate, Dechter [AI'11]
4. Ranking by Lineage Size (# of clauses)	✓	✓
5. Deterministic Query Evaluation	random ranking	✓

Experimental Setup

1: TPC-H random database

```
Supplier(s_suppkey, s_nationkey)      ← (10k tuples)
PartSupp(ps_suppkey, ps_partkey)      ← (800k tuples)
Part(p_partkey, p_name)               ← (200k tuples)
```

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

2. Parameterized test query

```
SELECT  distinct s_nationkey          ← 25 nations
FROM    Supplier, Partsupp, Part
WHERE   s_suppkey = ps_suppkey
  and   ps_partkey = p_partkey
  and   s_suppkey <= $1               ← 500 – 10k
  and   p_name like $2                ← '%red%green%' '%red%', '%', etc.
```

	a	s	p	n
S	○	○		
PS		○	○	
P			○	○

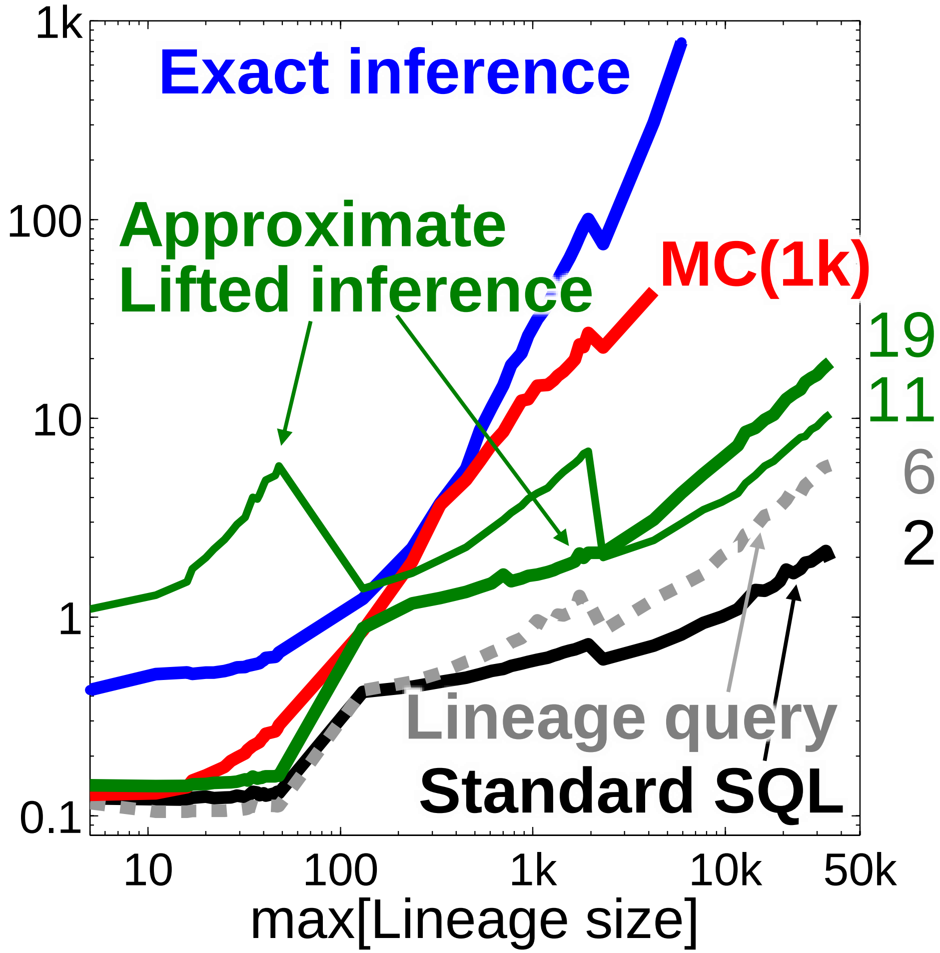
"Which nations (as determined by the attribute nationkey) are most likely to have suppliers with suppkey ≤ $1 that supply parts with a name like $2?

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

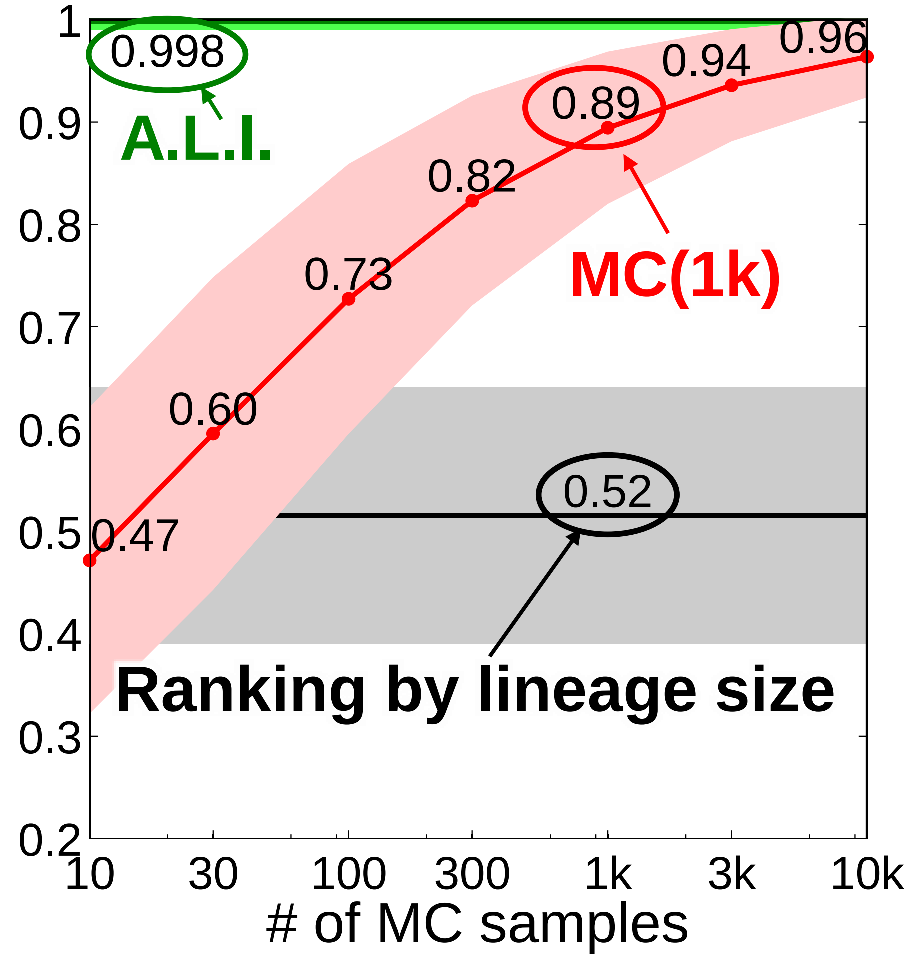
3. PostgreSQL, Translation happens in Java

Experiments: results on synthetic TPC-H data

Time (sec)



Ranking quality (AP@10)



Roadmap

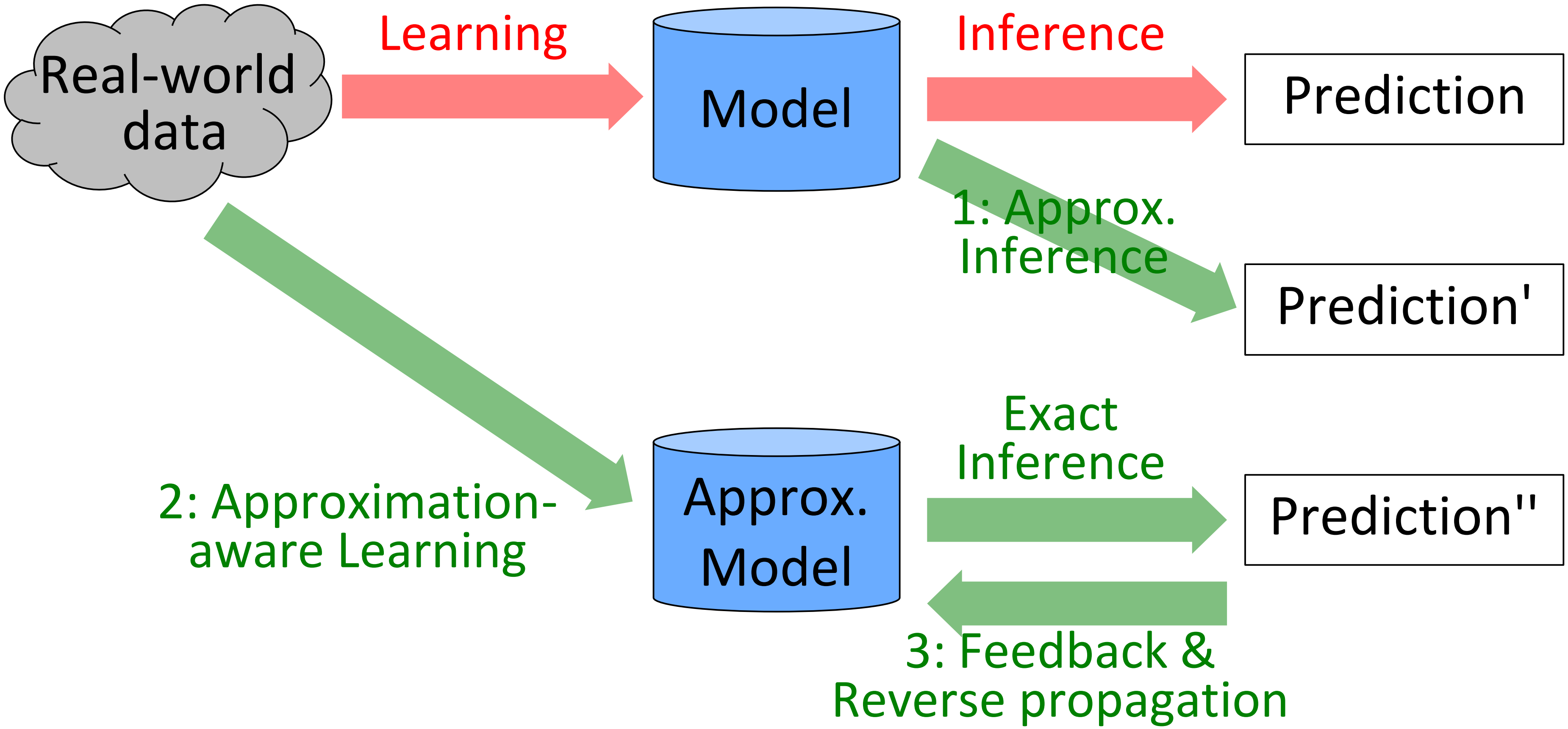
1. **Theory**: Bounds on the probability of monotone Boolean functions

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

3. Experiments

4. Outlook

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

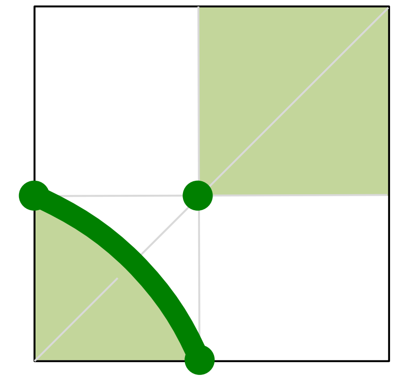
"A student can take either class 201 or class 202."

3. Learning the probabilities from predictions

Take-aways

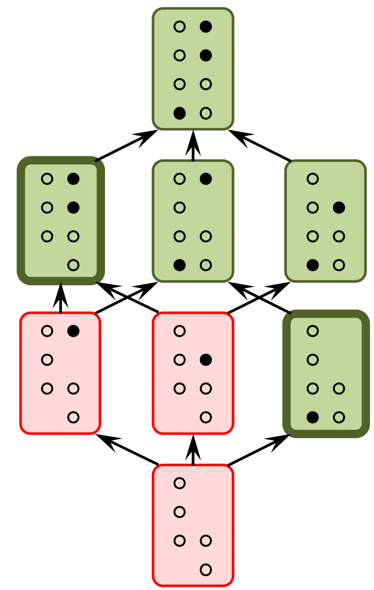
1. Probability of Boolean Functions

- 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 ☺