Topics in Probabilistic and Statistical Databases

Lecture 4: Dicthotomy Theorems

Dan Suciu
University of Washington

Before we start…

- Kate will give an update on OWA

TABLE II
EXAMPLE OF MISSING PROBABILITIES

EMPLOYEE	DEPARTMENT	QUALITY BONUS	SALES
Jon Smith	Toy	0.3 [Great yes] 0.4 [Good yes] 0.2 [Fair *] 0.1 [* *]	0.3 [$30–34K] 0.5 [$35–39K] 0.2 [*]

TABLE III

RELATION FOR PROJECT EXAMPLES

NAME	DIV PRICE	RATING
P.J.	0.3 [10 200]	0.9 [AAA]
	0.2 [20 250]	0.1 [AA]
	0.2 [10 250]	
	0.1 [0 *]	
	0.1 [* 100]	
	0.1 [* *]	
CONTI	1.0 [0 50]	0.5 [BBB]
		0.5 [CCC]

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

URL	Conf	P
U1	SIGMOD	c1*p1
U1	SIGCOM	c1*p2
U2	VLDB	c2*p3

URL	Chair	P
U1	Kossman	c1*q1
U2	Gehrke	c2*q2
U2	Milo	c2*q3

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→PTIME, non-h→#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

$\texttt{Product}^p$

prod	price	color	shape	p
Gizmo	20	red	oval	$p_1 = 0.25$
		blue	square	$p_2 = 0.75$
Camera	80	green	oval	$p_3 = 0.3$
		red	round	$p_4 = 0.3$
		blue	oval	$p_5 = 0.2$
IPod	300	white	square	$p_6 = 0.8$
		black	square	$p_7 = 0.2$

In each possible world: prod → price, color, shape

In the representation: prod → price

FDs at the Representation Level

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

Suppose x→ y in S(x,y) in the representation

What is the complexity of this query ?

[Dalvi&S' VLDBJ'2007, Olteanu'ICDE2009]

FDs at the Representation Level

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

Suppose x→ y in S(x,y) in the representation

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

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

Now it is hierarchical

[Dalvi&S' VLDBJ'2007, Olteanu'ICDE2009]

FDs at the Representation Level

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

Suppose x→ 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 :- R^p(x), S1(x,u), S2(u,v),S3(v,y), T^p (y,z)

q :- R^p(z,x), S1(x,u), S2(u,z),S3(z,v), S4(v,y), T^p (y)

q :- R^p(z,x), S1(x,u), S2(u,z),S3(z,v), S4(v,y), T^p (z,y)

Case 2: CQ^1+Disjoint/independent

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

Theorem Forall $q \in CQ^1$

q has a safe plan and is in PTIME, OR
q is #P-hard

Finding Safe Plans

<u>Algorithm</u>: find a Safe Plan

1. Root variable u ➔ Π^{i}_{-u}
2. Variable u occurs in a subgoal with constant keys ➔ Π^{D}_{-u}
3. Connected components ➔ Join
4. Single subgoal ➔ Leaf node

$q(y) \text{ :- } R(\underline{\mathbf{x}},y,z)$

$\Pi_{-x}{}^{i}$

|

$\Pi_{-z}{}^{D}$

|

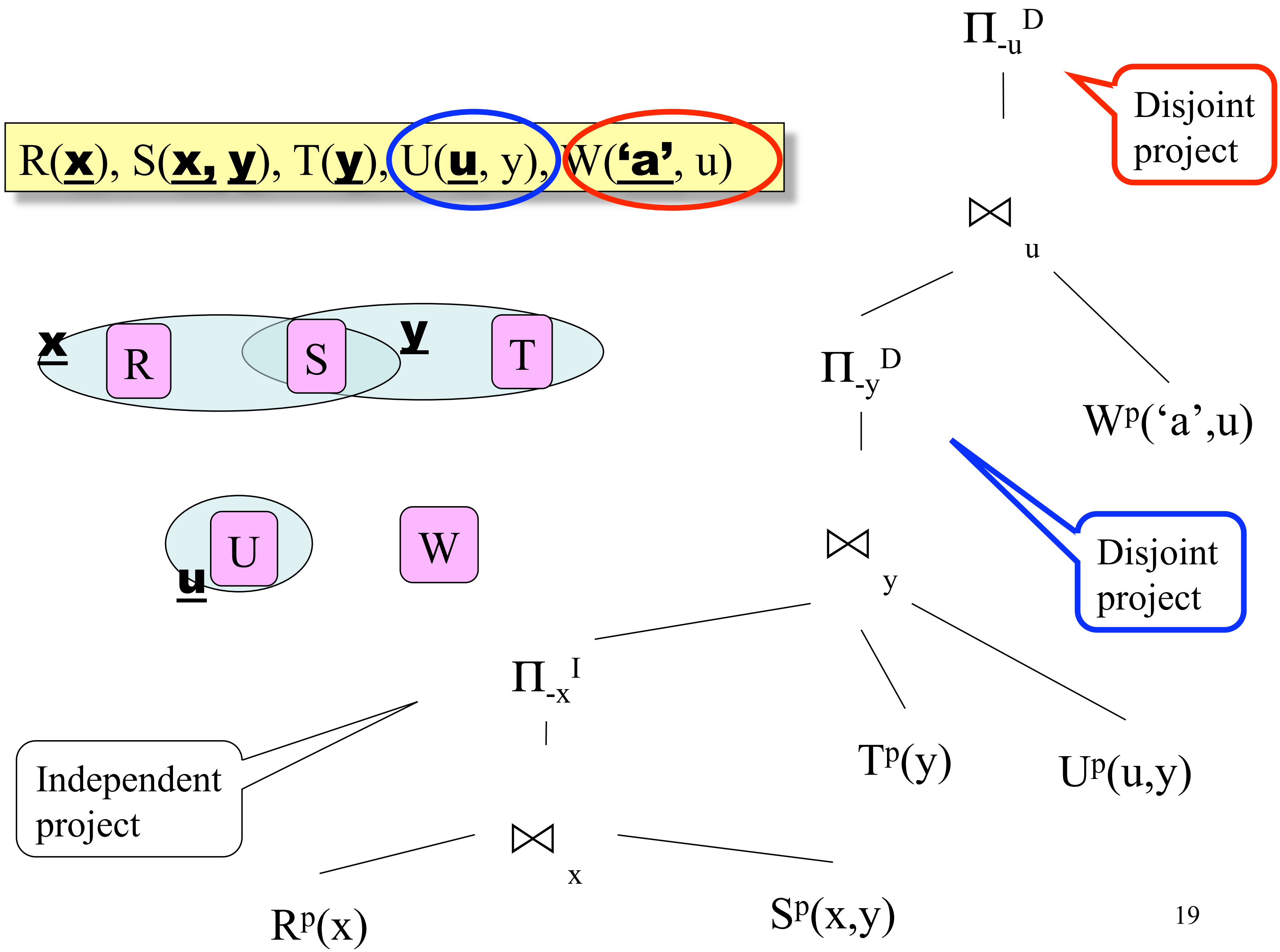
$R(\underline{\mathbf{x}},y,z)$

$q1(x^c,y^c)\text{:-}R(\underline{\mathbf{x}}^c,y^c,z)$

y	P
b	1-(1-p1-p2)(1-p3-p4)

<u>x</u>	y	P
a1	b	p1+p2
a2	b	p3+p4

<u>x</u>	y	z	P
a1	b	c1	p1
	b	c2	p2
a2	b	c1	p3
	b	c2	p4



R(x), S(x, y), T(y), U(u, y), W('a', u)
x
R
S
y
T
u
U
W
Π-u D
Disjoint project
⋈ u
Π-y D
W^p('a',u)
Disjoint project
⋈ y
Π-x I
T^p(y)
U^p(u,y)
Independent project
⋈ x
R^p(x)
S^p(x,y)

Definitions (in class)

- q :- $g_1, \ldots, g_k$
- $Sg(q) = \{g_1, \ldots, g_k\}$
- $Vars(g_i)$ = all variables of gi
- $KVars(g_i)$ = all variables in key positions

Algorithm Safe-Eval

- From [Dalvi&S'2007]
- Show on the whiteboard

- Call a query *safe* is the algorithm succeeds
- What are the *unsafe* queries ?

Some Unsafe Queries

$hd1 = R(\underline{\mathbf{x}}), S(\underline{\mathbf{x}}, \underline{\mathbf{y}}), T(\underline{\mathbf{y}})$

$hd2 = R(\underline{\mathbf{x}}, y), S(\underline{\mathbf{y}})$

$hd3 = R(\underline{\mathbf{x}}, y), S(x, \underline{\mathbf{y}})$

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

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

Step 1

- Show in class the hardness of

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

Then show $hd2^+$

Step 1

- Show in class the hardness of

hd3 = R(<u>**x**</u>,y), S(x,<u>**y**</u>)

Then show $hd3^{+}$

Step 2

- The rewrite rule q ➔ q' (on the whiteboard)
- q is a *final* query if forall q' s.t. q➔ q', q' is safe
- Prove:
  - If q is unsafe, then ∃ 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)

[Dalvi&S’2007b]

Case 3: CQ, independent tables

- Allow selfjoins
- But restrict again to independent tables

Does the query have a safe plan ?

q(x) :- 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) =

$$p(R(a,b))+(1-p(R(a,b))(1-(1-\prod_{y \in Dom, y \neq a}(1-p(R(y,b))))(1-\prod_{x \in Dom, x \neq b}(1-p(R(a,x))))$$

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:

$$q(x) \text{ :- } Movie(x,y),\ Match^p(x,r),\ Review(r,Joe,s),\ s > 4$$
$$Match^p(x,r'),\ Review(r',Jim,s'),\ s'>4$$

$Match^p$ = probabilistic, tuple independent

Movie, Review = deterministic

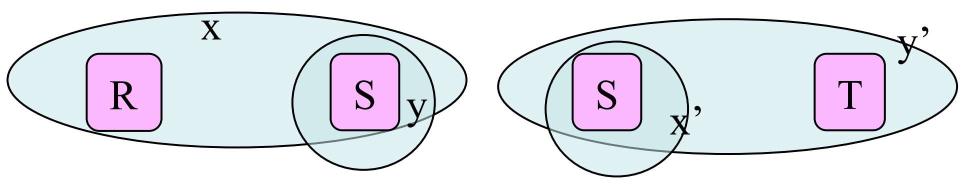
[Dalvi&S’2007b]

The #P-hard Queries

Hierarchical queries with “inversions”:

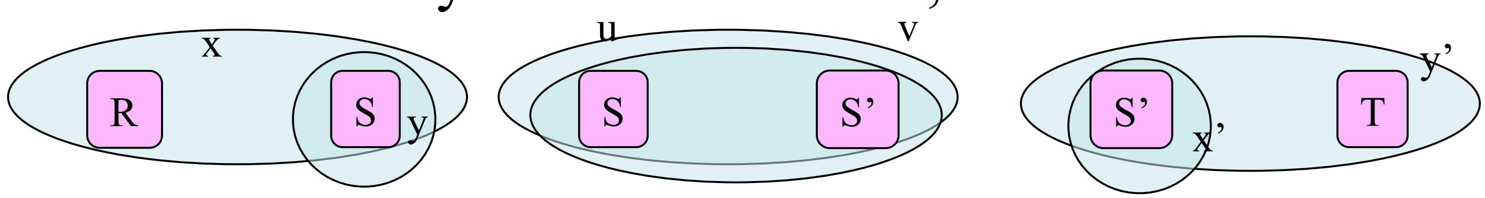
hi1 = R(x), S(x,y), S(x’,y’), T(y’)

$x \supset y$ unifies with $x' \subset y'$



hi2 = R(x), S(x,y), S(u,v), S’(u,v),S’(x’,y’), T(y’)

$x \supset y$ unifies with $u \equiv v$, which unifies with $x' \subset y'$



The #P-hard Queries

A query with a long inversion:

$hi_k = R(\underline{\mathbf{x}}), S_0(\underline{\mathbf{x}},\underline{\mathbf{y}}),$
$S_0(\underline{\mathbf{u}}_1,\underline{\mathbf{v}}_1), \; S_1(\underline{\mathbf{u}}_1,\underline{\mathbf{v}}_1)$
$S_1(\underline{\mathbf{u}}_2,\underline{\mathbf{v}}_2), S_2(\underline{\mathbf{u}}_2,\underline{\mathbf{v}}_2), \ldots$
$S_k(\underline{\mathbf{x}}',\underline{\mathbf{y}}'), T(\underline{\mathbf{y}'})$

The #P-hard Queries

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

q = R(<u>**x**</u>,<u>**y**</u>), R(<u>**y**</u>,<u>**z**</u>)

R(x,y), R(y,z)
R(x',y'), R(y',z')

[Dalvi&S'2007b]

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:

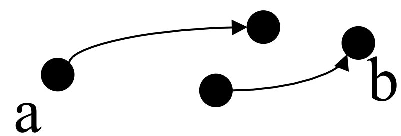
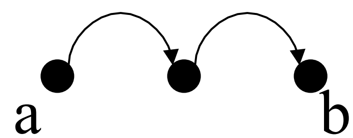
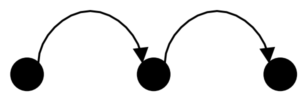
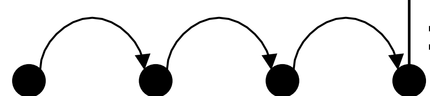
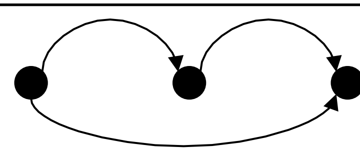
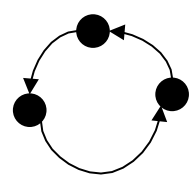
$x \supset y$ with $u_1 \equiv v_1$ with … with $u_n \equiv v_n$ with $x' \subset y'$

Theorem Forall $q \in CQ$:

If q is non-hierarchical, or has an inversion* then it is #P-hard

Otherwise it is in PTIME

*without "eraser": see paper.

Query		Com-plexity	Why
R(a,x), R(y,b)		PTIME	
R(a,x), R(x,b)		PTIME	
R(x,y), R(y,z)		#P	Inversion
R(x,y),R(y,z),R(z,u)		#P	Non-hierarchical
R(x,y),R(y,z),R(z,x)		#P	Non-hierarchical
R(x,y),R(y,z),R(x,z)		#P	Non-hierarchical

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]
  - *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 ?