Topics in Probabilistic and Statistical Databases

Lecture 6: Aspects of Query Processing

Dan Suciu
University of Washington

Outline

- General query evaluation
  - Beyond safe queries.
- Ranking query answers

References for Ranking

- Cormode et al., ICDE 2009
- Zhang and Chomicki, DBRank 2008
- Soliman et al. ICDE'2007
- Yi et al. ICDE'2008
- Hua et al. SIGMOD'2008

General Query Evaluation

- Query q + database DB
  ➔ boolean expression Φ_q^{DB}

- Run any probabilistic inference algorithm on Φ_q^{DB}

This approach is taken in Trio

Background: Probability of Boolean Expressions

Given:

$\Phi = X_1X_2 \vee X_1X_3 \vee X_2X_3$ $\quad P(X_1)= p_1 , P(X_2)= p_2, P(X_3)= p_3$

Compute $P(\Phi)$

$\Omega =$.

X_1	X_2	X_3	P	Φ
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1	$(1-p_1)p_2p_3$	1
1	0	0		0
1	0	1	$p_1(1-p_2)p_3$	1
1	1	0	$p_1p_2(1-p_3)$	1
1	1	1	$p_1p_2p_3$	1

$$Pr(\Phi)=(1-p_1)p_2p_3 + p_1(1-p_2)p_3 + p_1p_2(1-p_3) + p_1p_2p_3$$

#P-complete [Valiant:1979]

Query q + Database PDB ➔ Φ

q= R(**x**, **y**), S(**x**, **z**)

PDB=

R^p

A	**B**	P	
a_1	b_1	p_1	X_1
a_2	b_2	p_2	X_2

S^p

A	**C**	P	
a_1	c_1	q_1	Y_1
a_1	c_2	q_2	Y_2
a_2	c_3	q_3	Y_3
a_2	c_4	q_4	Y_4
a_2	c_5	q_5	Y_5

$\Phi = X_1Y_1 \vee X_1Y_2 \vee X_2Y_3 \vee X_2Y_4 \vee X_2Y_5$

Probabilistic Networks

- Nodes = random variables
- Edges = dependence relationships

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

$\Phi = X_1Y_1 \vee X_1Y_2 \vee X_2Y_3 \vee X_2Y_4 \vee X_2Y_5$

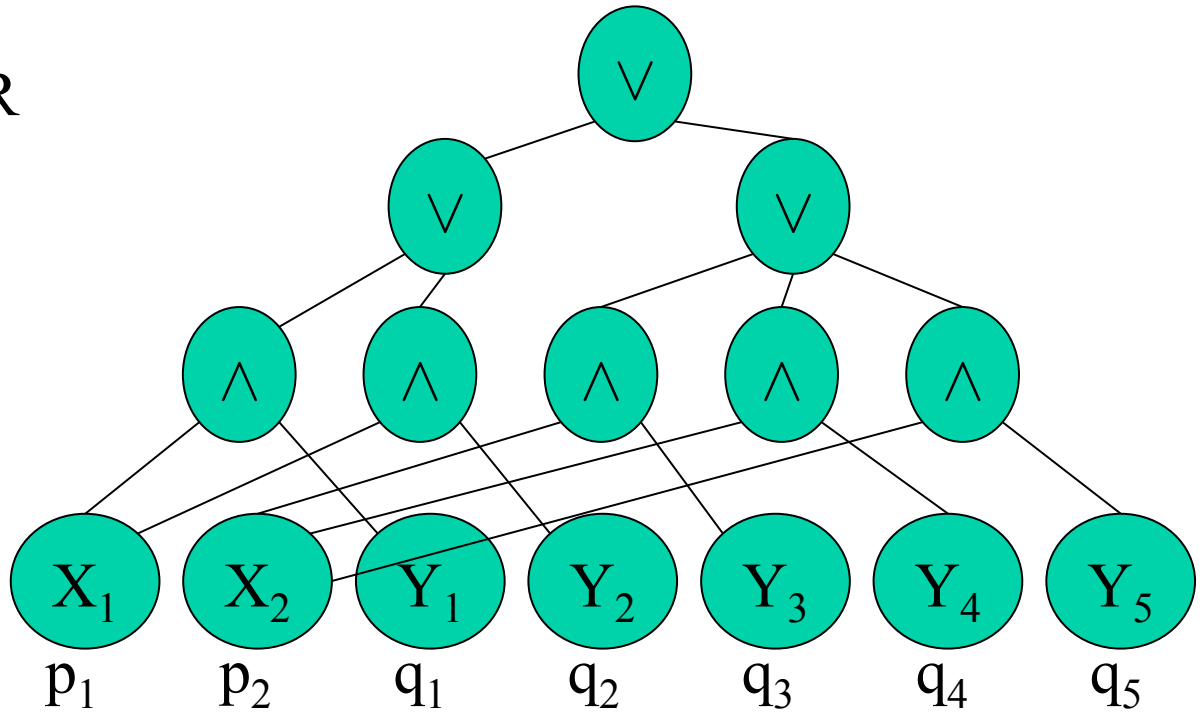
Studied intensively in KR

Typical networks:

Bayesian networks

Markov networks

Boolean expressions



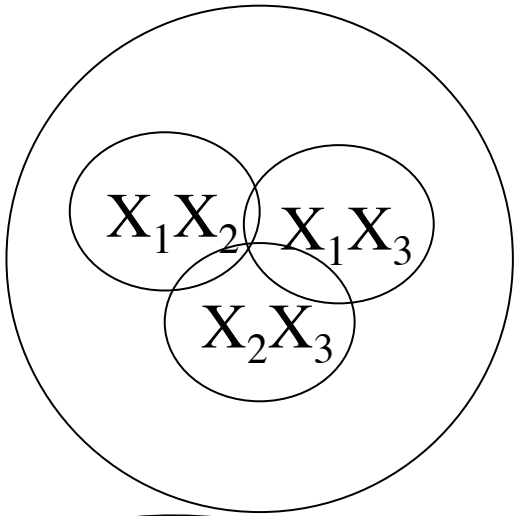
Inference Algorithms for Boolean Expressions

- Deterministic
  - OBDDs [Olteanu]
  - Discuss briefly in class…

- Monte Carlo:
  - Naïve Monte Carlo
  - Luby and Karp

Naive Monte Carlo Simulation

$$E = X_1X_2 \vee X_1X_3 \vee X_2X_3$$



Cnt $\leftarrow$ 0

repeat N times

randomly choose $X_1, X_2, X_3 \in \{0,1\}$

if $E(X_1, X_2, X_3) = 1$

then Cnt = Cnt+1

P = Cnt/N

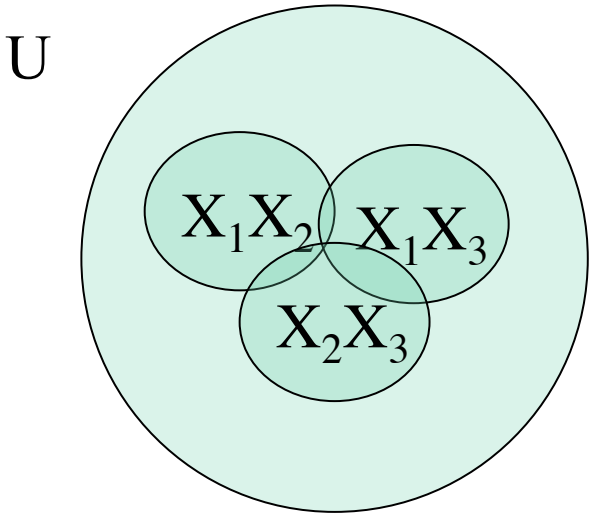
return P /* $\simeq$ Pr(E) */

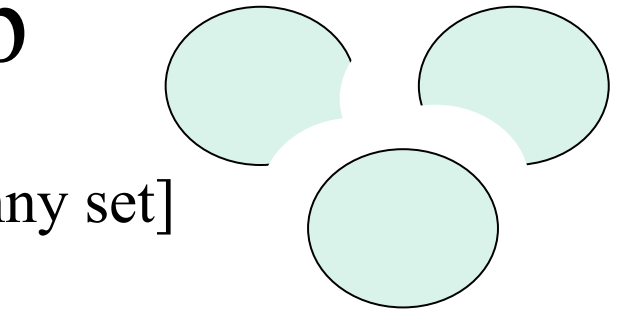
May be big (in theory)

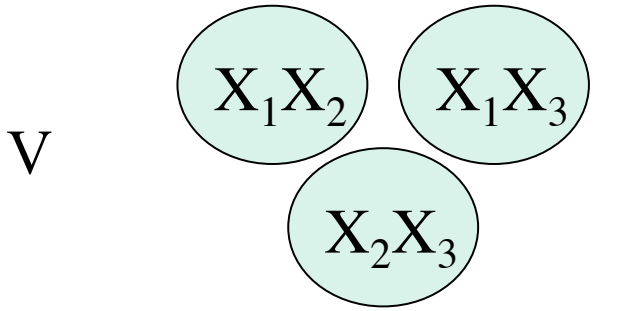
Theorem (0-1 estimator) If $N \geq (1/\Pr(E)) \times (4\ln(2/\delta)/\varepsilon^2)$

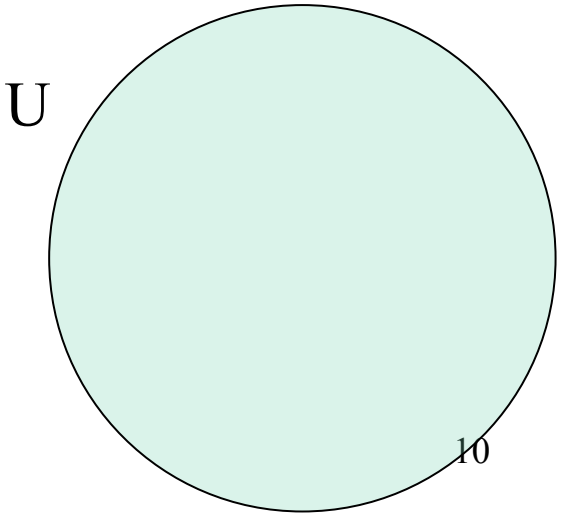
then $\Pr[\,|\,P/\Pr(E) - 1\,|\,> \varepsilon\,] < \delta$

Luby-Karp









$\Pr(E) = \#(X_1X_2 \vee X_1X_3 \vee X_2X_3) / \#U$

$= \#[\text{funny set}] / \#V \quad * \quad \#V / \#U$

= S on the next slide

[Karp&Luby:1983]

[Graedel,Gurevitch,Hirsch:1998]

Luby-Karp

$E = C_1 \vee C_2 \vee \ldots \vee C_m$

Cnt $\leftarrow$ 0; S $\leftarrow$ Pr(C_1) + … + Pr(C_m);

repeat N times

randomly choose i $\in$ {1,2,…, m}, with prob. Pr(C_i) / S

randomly choose $X_1, \ldots, X_n \in \{0,1\}$ s.t. $C_i = 1$

if C_1=0 and C_2=0 and … and C_{i-1} = 0

then Cnt = Cnt+1

P = Cnt/N * S

return P /* $\simeq$ Pr(E) */

Now it's in PTIME

Theorem. If $N \geq (1/m) \times (4\ln(2/\delta)/\varepsilon^2)$ then:

$\Pr[\, |P/\Pr(E) - 1| > \varepsilon \,] < \delta$

[Re,Dalvi&S'2007]

An Example

$q(x,u) :- R^p(\underline{\mathbf{x}},\underline{\mathbf{y}}), S^p(\underline{\mathbf{y}},\underline{\mathbf{z}}), T^p(\underline{\mathbf{z}},u)$

R^p

A	**B**	P
a1	b1	p1
	b2	p2
a2	b1	p3

S^p

B	**C**	P
b1	c1	q1
b2	c1	q2
	c2	q3
	c3	q4

T^p

C	**D**	P
c1	d1	r1
	d2	r2
c2	d1	r3
	d2	r4
	d3	r5

Step 1: evaluate this query *on the representation* to get the data

qTemp(x,y,p,y,z,q,z,u, r) :- R(x,y,p), S(y,z,q), T(z,u,r)

R^p

A	**B**	P
a1	b1	p1
a1	b2	p2
a2	b1	p3

S^p

B	**C**	P
b1	c1	q1
b2	c1	q2
b2	c2	q3
b2	c3	q4

T^p

C	**D**	P
c1	d1	r1
	d2	r2
c2	d1	r3
	d2	r4
	d3	r5

qTemp(x,y,p,y,z,q,z,u, r) :- R(x,y,p), S(y,z,q), T(z,u,r)

Temp

A	B	P	B	C	P	C	D	P
a1	b1	p1	b1	c1	q1	c1	d1	r1
a1	b2	p2	b2	c2	q3	c2	d1	r3
a2	b1		. .					
. .	. .		. .					

Step 2: group Temp by the head variables in q

$q(x,u) :- R^p(\underline{\mathbf{x}},\underline{\mathbf{y}}), S^p(\underline{\mathbf{y}},\underline{\mathbf{z}}), T^p(\underline{\mathbf{z}},u)$

	A	B	P	B	C	P	C	D	P
q(a1,d1)	a1	b1	p1	b1	c1	q1	c1	d1	r1
	a1	b2	p2	b2	c2	q3	c2	d1	r3
	. . .								
q(a1,d2)	a1	b1	p1	b1	c1	q1	c1	d2	r2
	a1	b1		. .				d2	
	. .	. .		. .					
	. . .								

Step 3: each group is a DNF formula; run Monte Carlo

q(a1,d1)

A	B	P	B	C	P	C	D	P
a1	b1	p1	b1	c1	q1	c1	d1	r1
a1	b2	p2	b2	c2	q3	c2	d1	r3
. . .								
a1	. . .						d2	

$\Phi_{a1,d1} = X_{11}Y_{11}Z_{11} \vee X_{12}Y_{22}Z_{21} \vee \ldots$ ➔ $P(\Phi_{a1,d1}) = s1$

$\Phi_{a1,d2} = X_{11}Y_{11}Z_{12} \vee \ldots$ ➔ $P(\Phi_{a1,d2}) = s2$

. . . . . .

Where X_{11} = R(a1,b1) X_{12} = R(a1,b2) Y_{11} = S(b1,c1) etc

Step 4: collect all results

Temp

A	B	P	B	C	P	C	D	P
a1	b1	p1	b1	c1	q1	c1	d1	r1
…								
a1	b1	p1	b1	c1	q1	c1	d2	r2
…	…		…					

➔

Answer to q(x,u)

A	D	P
a1	d1	s1
a1	d2	s2
…		

Remark:

The DBMS executes only the query qTemp:
only selections and joins are done in the engine

The probabilistic inference is done in the middleware

Summary on Monte Carlo

- General method for evaluating P(q), $\forall$ q $\in$ CQ
- Naïve MC: N = O(1/P(q)) steps
- Luby&Karp: N = O(m) steps

- Lessons from MystiQ: no big difference
- Typically: P(q) $\approx$ 0.1 or higher
- Typically: m $\approx$ 5 - 10 or higher

- Typical number of steps: N $\approx$ 100,000: this is for *one single* tuple in the answer !

Optimization 1: Safe Subqueries

Main idea:

1. Find subqueries of q that are
   - Safe
   - "Representable"

2. Evaluate the subqueries using safe plans

3. Rewrite q to q_{opt} by using the subqueries, then evaluate q_{opt} using Monte Carlo

Example

We illustrate with a boolean query (for simplicity):

$$q :- R^p(\underline{\mathbf{x}},y), S^p(\underline{\mathbf{y}},z), T^p(\underline{\mathbf{y}},\underline{\mathbf{z}},\underline{\mathbf{u}})$$

1. Find the following subquery:

$$sq(y) :- S^p(\underline{\mathbf{y}},z), T^p(\underline{\mathbf{y}},\underline{\mathbf{z}},\underline{\mathbf{u}})$$

sq is safe: $sq = \Pi^d_y(S \bowtie T)$

sq(b) is independent from sq(b'), whenever $b \neq b'$

2. Compute sq(y) on the representation using the safe plan:

```
SELECT S.B, sum(S.P*T.P) as P
FROM S,T
WHERE S.C=T.C
GROUP BY S.B
```

→

SQ^p

B	P
b1	t1
b2	t2
. .	

3. Rewrite q to q_{opt}: q_{opt} :- $R^p(\underline{\mathbf{x}},y)$, $SQ^p(\underline{\mathbf{y}})$

Continue as before:

Send this to the engine: $qTemp_{opt}(x,p,y,q)$:- R(x,y,p),sq(y,q)

Run Monte Carlo on result

What's improved:

Some of the probabilistic inference pushed in RDBMS

Monte Carlo runs on a smaller DNF

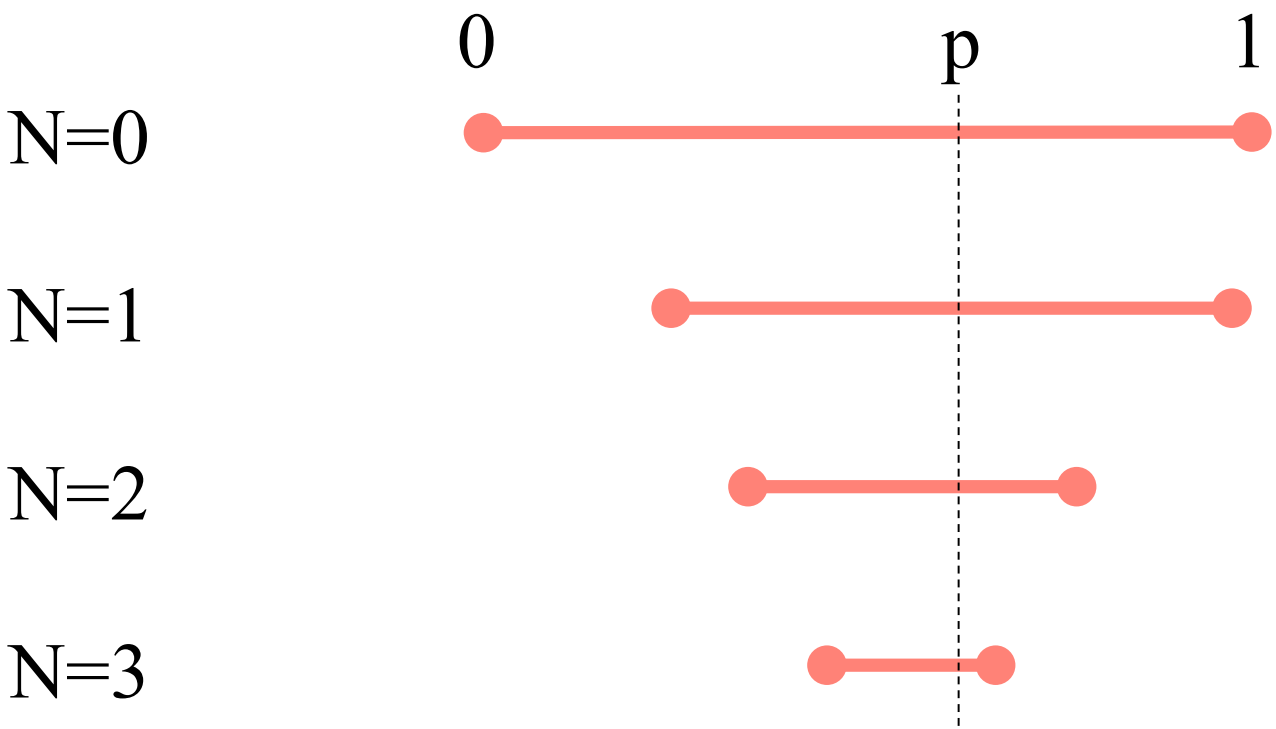
[Re'2007]

Optimization 2: Top-K Ranking

- Main idea:
- Number of potential answers is huge
  - 100s or 1000s
- Users want to see only the top-k
  - Typical: top 10, or top 20
- Catch 22:
- Run the expensive Monte Carlo *only* on top k
- But to discover the top-k we need to run MC !

Interleave Monte Carlo steps with ranking

Modeling Monte Carlo Simulation



$q(x,u) :- R^p(\underline{\mathbf{x}},\underline{\mathbf{y}}), S^p(\underline{\mathbf{y}},\underline{\mathbf{z}}), T^p(\underline{\mathbf{z}},u)$

Current Approximation

A	D	P
a1	d1	0.2 – 0.7
a2	d2	0.6 – 0.8
a3	d3	0 – 1.0
a1000	d1000	0.3 – 0.9

Final, ranked Answer

Top-k

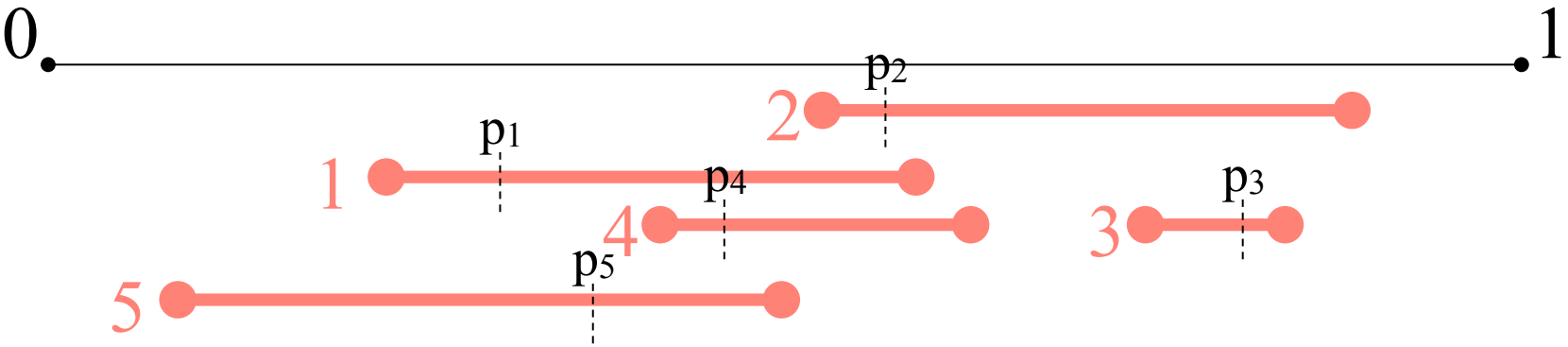
Bottom n-k

A	D	P
a49	d49	0.99
a22	d22	0.90
a87	b87	0.85
a522	b522	0.01

Quiz: which one should we simulate next ?

We have n objects

How to find the top k ?

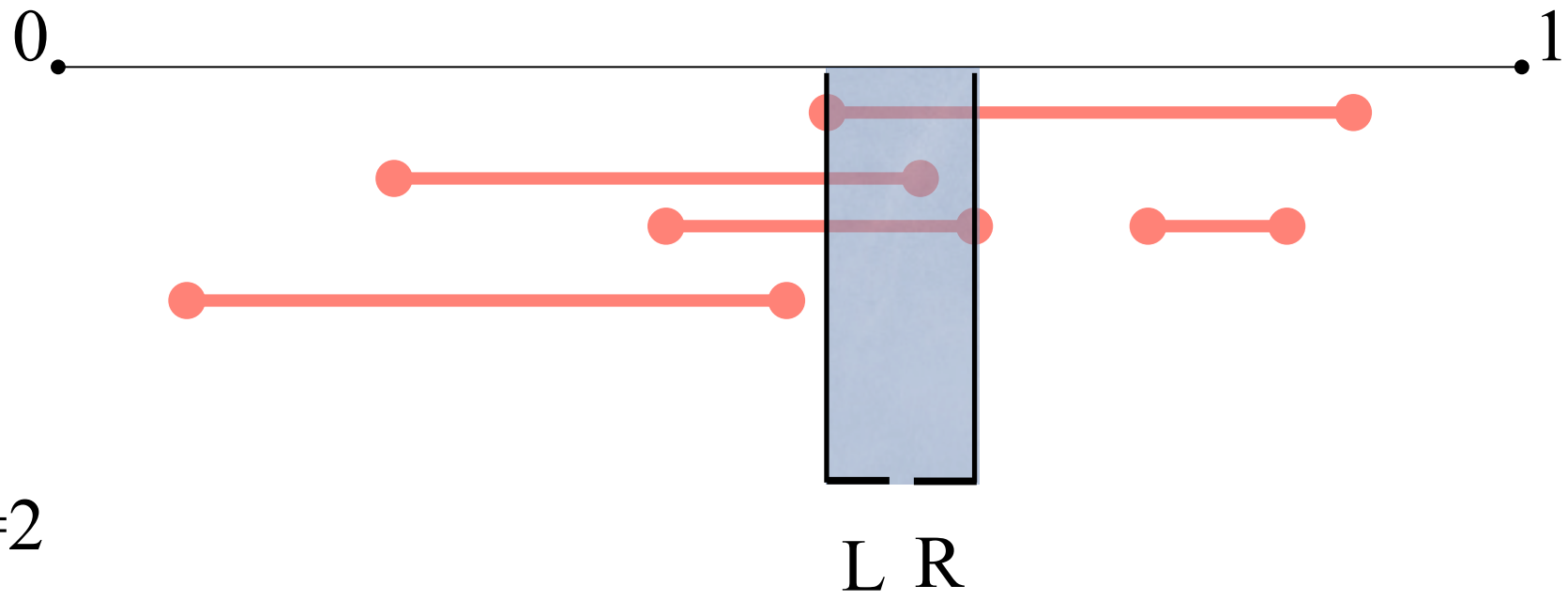


Example: looking for top k=2;

Which one simulate next ?

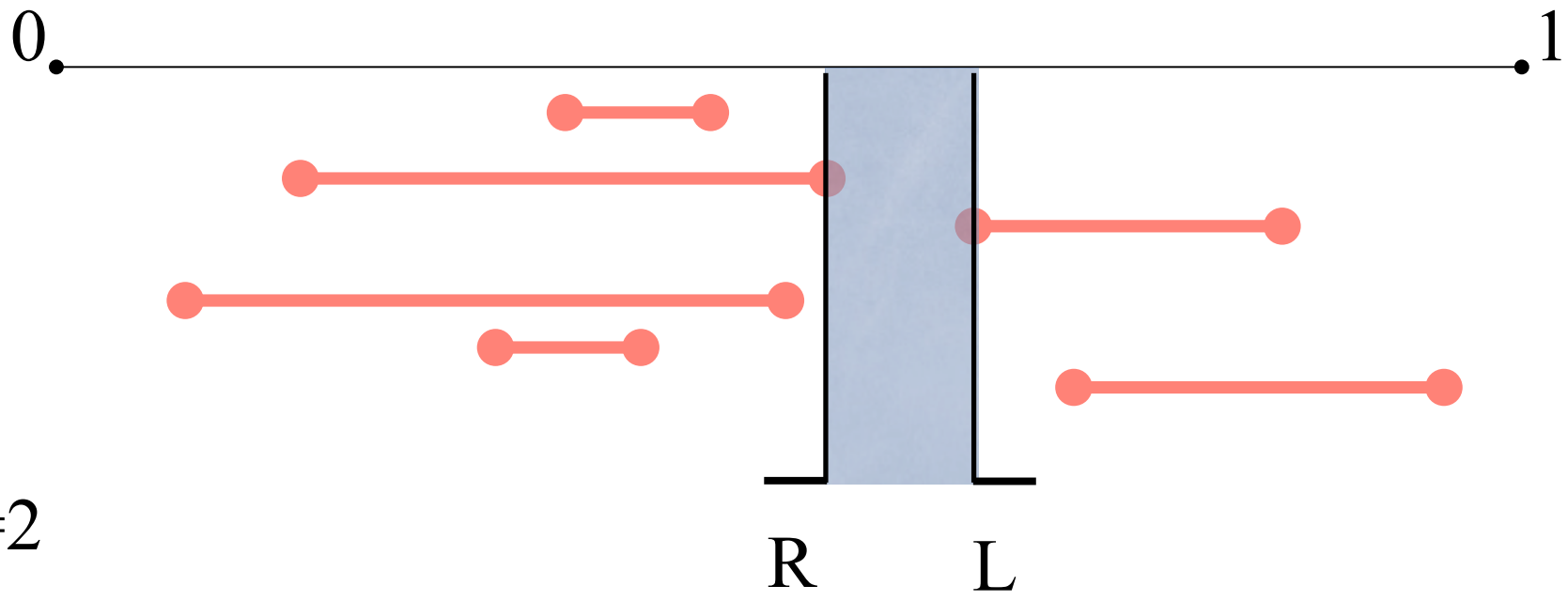
Multisimulation

Critical region:
(k'th left, k+1'th right)



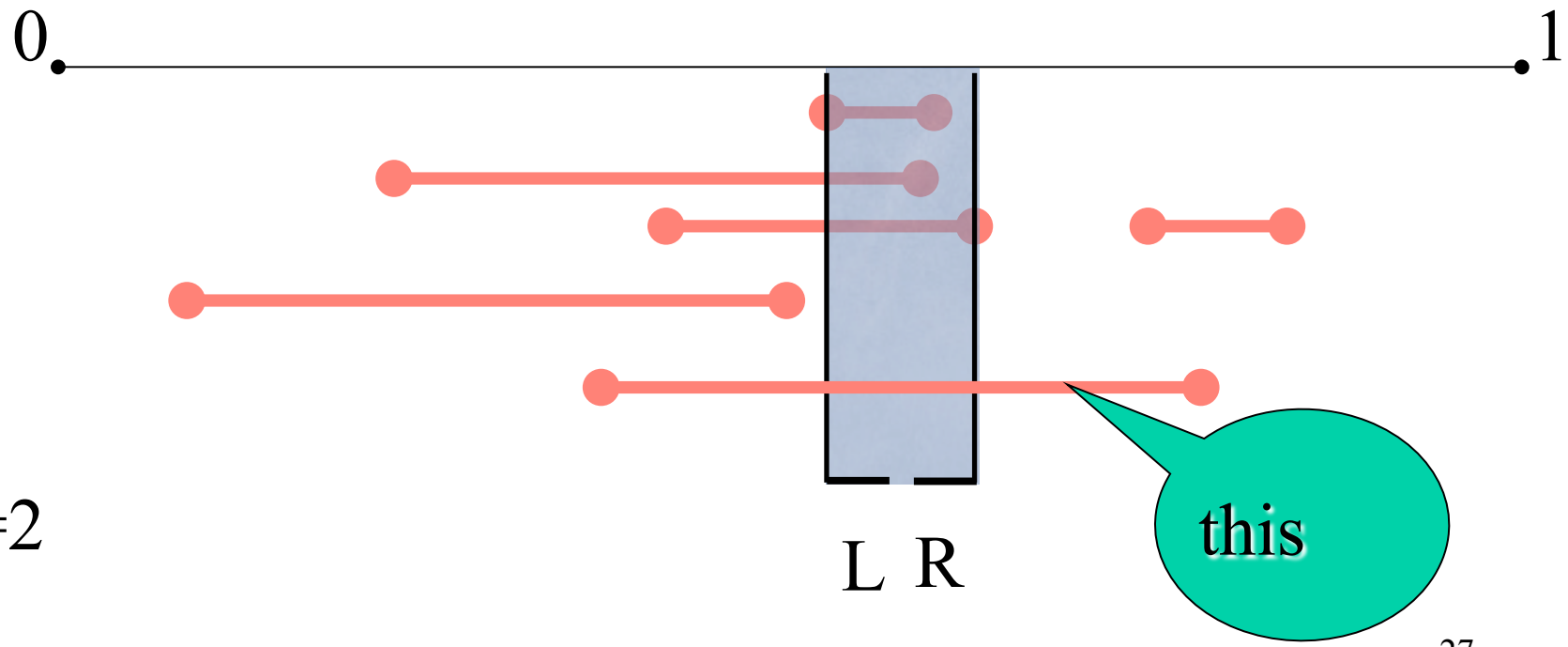
Multisimulation Algorithm

End: when critical region is “empty”



Multisimulation Algorithm

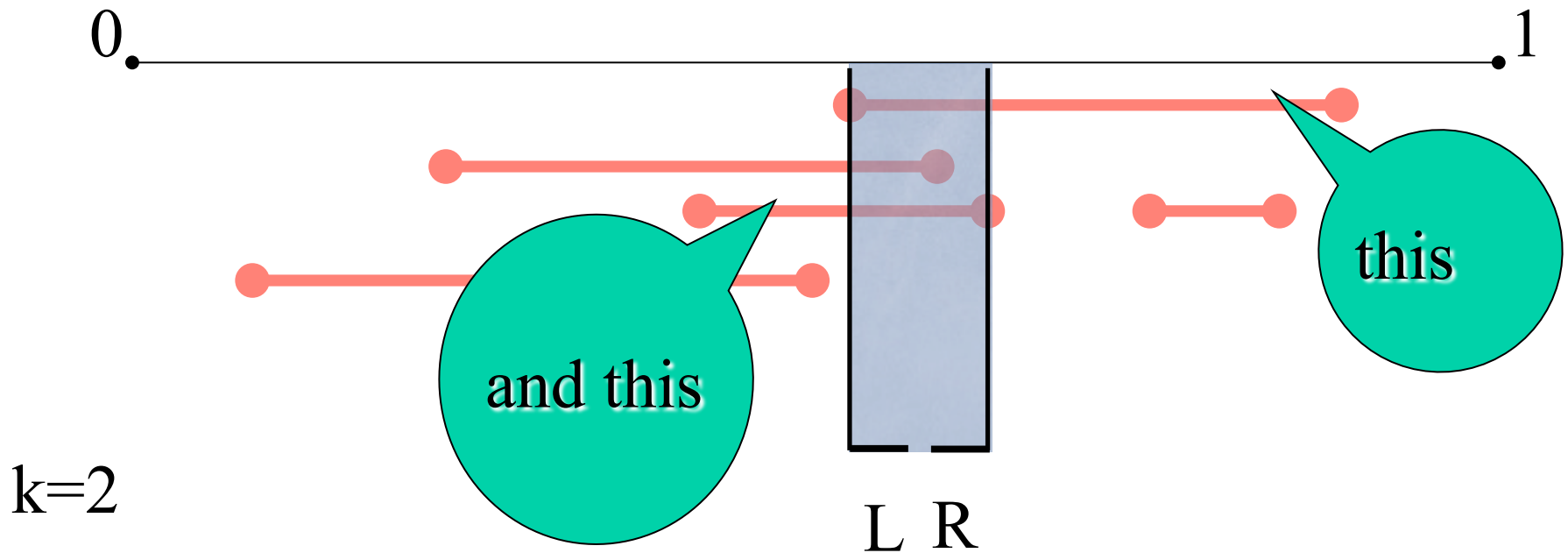
Case 1: pick a “double crosser”
and simulate it



k=2

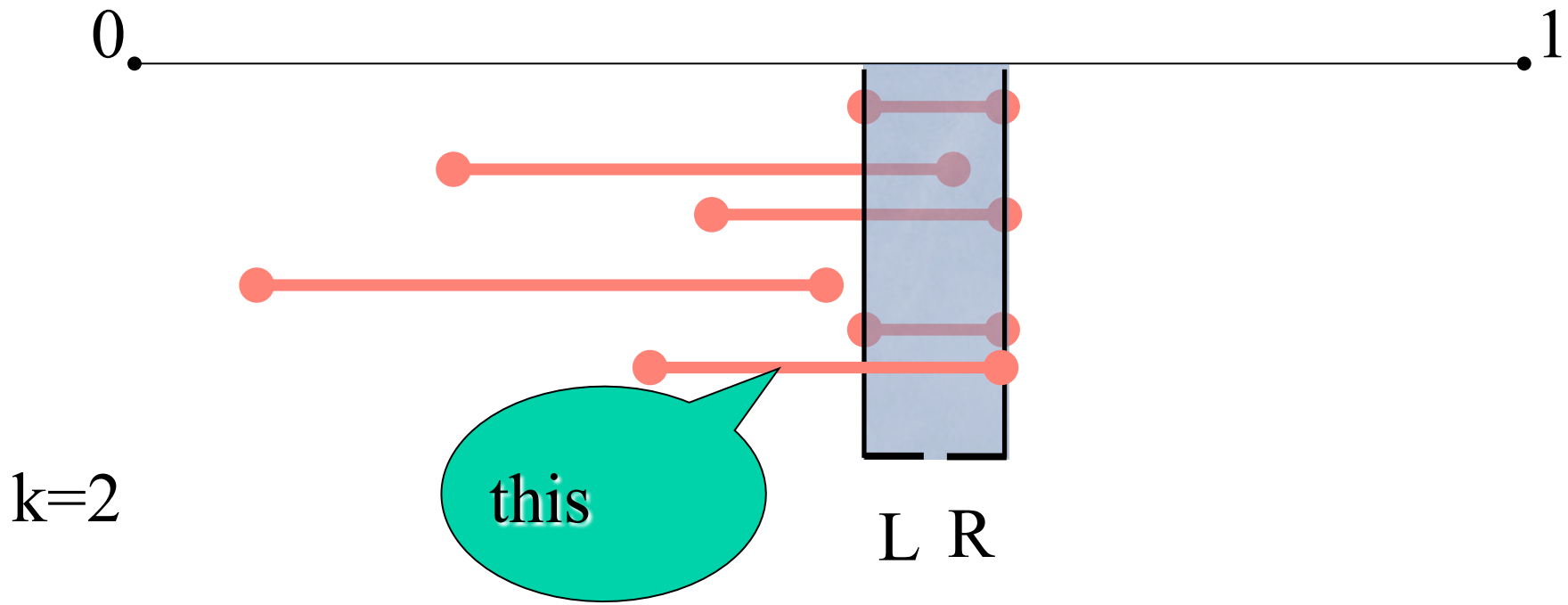
Multisimulation Algorithm

Case 2: pick both a “left” AND a “right” crosser



Multisimulation Algorithm

Case 3: pick a “max crosser” and simulate it



[Re'2007]

Multisimulation Algorithm

Theorem (1) It runs in < 2 Optimal # steps

(2) no other deterministic algorithm does better

Query Evaluation Summary

- Safe queries are OK
  - But how many queries are safe ?
- General purpose query evaluation
  - Inefficient
  - Active research area

- What are your thoughts ?

Ranking Deterministic Data

- Scoring function S(t)

- Rank tuples s.t. $S(t_1) > S(t_2) > \ldots$
- Return top k tuples $t_1, t_2, \ldots, t_k$

- Ranking is deterministic

Ranking Probabilistic Data

- Scoring function S(t) plus probability p(t)
- In each possible world the ranking is deterministic
- But global ranking is probabilistic

Note this differs from the earlier top-k query optimization. WHY ?

[Zhang'2008]

Tuple-level Uncertainty

Name	Score of the biometric match	Probability of staying Saturday night in the lab
Aidan	65	0.3
Bob	55	0.9
Chris	45	0.4

Two people worked in the lab on Saturday night

Find the top-2 tuples

[Zhang'2008, Cormode'2009]

Attribute-level Uncertainty

Location	Temp (F)	Prob
DB Lab	22	0.6
	10	0.4
AI Lab	25	0.1
	15	0.6

What is the temperature in the warmest spot ?

Find the top-1 answers

Ranking Probabilistic Data

- Two orders:
  - Order by the score
  - Order by the probability (confidence)

- How to combine them ?
  - Many possible ways
  - Discuss in class…

Semantics 1: Combination

- score + probability
- score*probability
- 0.3*score/100 + 0.7*probability…

Semantics 2: Expected value

- The expected value depends on the model
- Tuple-level uncertainty:
  - $E[S] = S(t) * p(t)$
- Attribute-level uncertainty:
  - $E[S] = \text{sum}_{t \text{ in block}} S(t) * p(t)$

Notations

- Let W be a deterministic world:
  - $Q^k(W)$ = the set of top k ranked tuples
- Let PDB be a probabilistic database
  - $Pr^k_Q(t) = Pr[t \text{ in } Q^k(W)]$

Semantics 3: U-TopK

Introduced by [Soliman, ICDE'2007]

- Consider all possible worlds: W_1, W_2, …
- Compute top-k answers: $T_1=Q^k(W_1)$, T_2, …
- Return $\text{argmax}_{Ti}\ P(T_i)$

Semantics 4: U-kRanks

Introduced by [Soleiman, ICDE'2007]

- Let $t_1, \ldots, t_n$ be all tuples in the database
- Denote X_{ij} the event that tuple t_j has rank i
- For each i=1,…,k return $\text{argmax}_{tj} \Pr[X_{ij}]$

Semantics 5: Probabilistic Threshold Top-k

Introduced by [Hua, SIGMOD'2008]

- Fix a threshold $p > 0$
- Answer$(Q, p) = \{t \mid Pr^k_Q(t) > p\}$

Semantics 6: Global Top-k

Introduced by [Zhang and Chomicki, 2008]

- Rank all tuples t_1, t_2, … s.t.:
  $Pr^k_Q(t_1) > Pr^k_Q(t_2) > \ldots$

- Return the top-k

Semantics 7: Expected Rank

Introduced by [Cormode, ICDE'2009]

- For each possible world W, the rank of a tuple t_i is:
  - $\text{rank}_W(t_i) = |\{t_j \in W \mid S(t_j) > S(t_i)\}|$
- The expected rank $r(t_i)$ is the expected value of $\text{rank}_W(t_i)$
- Sort tuples $r(t_1) > r(t_2) > \ldots$ and return top k

Semantics 23

- Introduced by [???] in SIGMOD'2017…

You get the point: unclear when to stop…

What is a good semantics ?

- Suppose we have a semantics that, given k, returns a set of tuples R_k
- How do we evaluate the quality of this semantics ?
- In class…

[Zhang'2008, Cormode'2009]

Six Criteria

1. Exact k:
   - If there are k possible tuples, then $|R_k| = k$
2. Containment:
   - $R_k \subset R_{k+1}$
   - Weak containment: $R_k \subseteq R_{k+1}$

[Zhang'2008, Cormode'2009]

Six Criteria

3. Unique ranking:
   - If $r_k(i)$ is the identity of the tuple at rank i, then $r_k(i) \neq r_k(j)$, forall $i \neq j$

4. Value invariance
   - Change the scores from S to S' without affecting the order: $S(t_1) > S(t_2) > \ldots$ becomes $S'(t_1) > S'(t_2) > \ldots$
   - Then $R_k = R_k$'

[Zhang'2008, Cormode'2009]

Six Criteria

5. Stability:
   - Change the probability of one tuple t_i s.t. $P(t_i) < P'(t_i)$
   - Then $t_i \in R_k$ implies $t_i \in R_k$'

6. Faithfulness
   - Change both score and probability of one tuple s.t. $P(t_i) < P'(t_i)$ and $S(t_i) < S'(t_i)$

Example

Find top-2 people

Name	Score of the biometric match	Probability of staying Saturday night in the lab
Aidan	65	0.3
Bob	55	0.9
Chris	45	0.4

Possible World	Prob
$W_1 = \emptyset$	0.042
$W_2 = \{Aidan\}$	0.018
$W_3 = \{Bob\}$	0.378
$W_4 = \{Chris\}$	0.028
$W_5 = \{Aidan, Bob\}$	0.162
$W_6 = \{Aidan, Chris\}$	0.012
$W_7 = \{Bob, Chris\}$	0.252
$W_8 = \{Aidan, Bob, Chris\}$	0.108

U-topk: {Bob} (Pr = 0.378)

U-kRanks: {most likely rank1: Bob; most likely rank2: Bob }

PT-k: Pr^2(Aidan) = 0.3; Pr^2(Bob) = 0.9; Pr^2(Chris) = 0.292;

Global-topK: top 2 above (Bob, Aidan)

Expected rank: r(Aidan)=0.3, r(Bob)=0.72, r(Chris)=…

Expected score: S(Aidan) = 19.5, S(Bob) = 49.5, S(Chris) = 18

[Zhang'2008, Cormode'2009]

Properties

	Exact-k	Containment	Unique-rank	Value-Invariant	Stability	Faithfulness
U-topk	✗	✗	✓	✓	✓	✗
U-kRanks	✓	✓	✗	✓	✓	✗
PT-k	✗	weak	✓	✓	✓	✗
Global-topk	✓	✗	✓	✓	✓	✗
Expected rank	✓	✓	✓	✓	✓	✗
Expected score	✓	✓	✓	✓	✗	✓