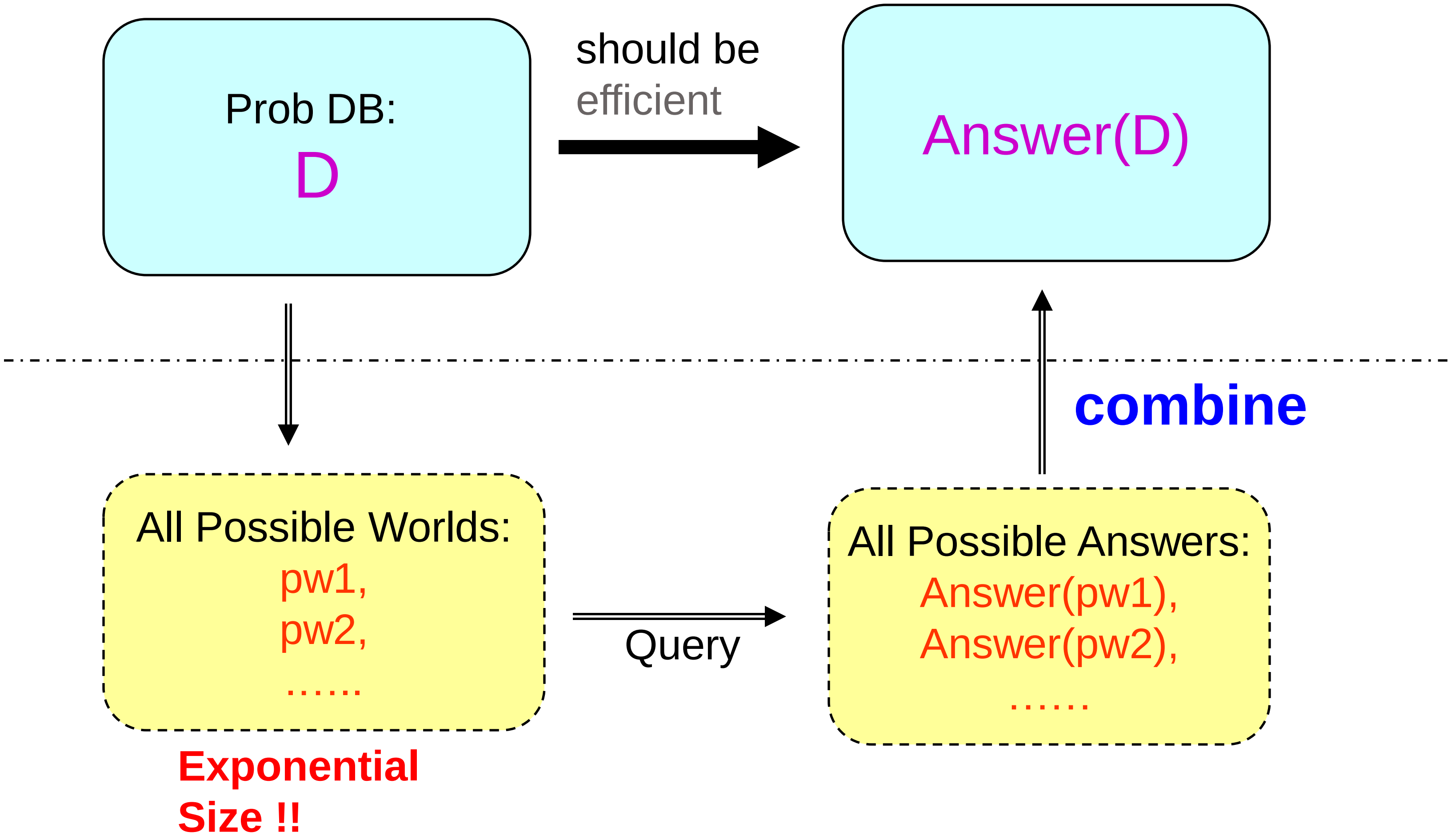
Consensus Answers for Queries over Probabilistic Databases

Jian Li and Amol Deshpande
University of Maryland, College Park, USA

Probabilistic Databases

- Motivation: Increasing amounts of uncertain data
  - Sensor Networks; Information Networks
    - Noisy input data; measurement errors; incomplete data
    - Prevalent use of probabilistic modeling techniques
  - Data Integration and Information Extraction
    - Need to model reputation, trust, and data quality
    - Increasing use of automated tools for schema mapping etc.
  - …
- Probabilistic databases
  - Annotate *tuples* with existence probabilities, and *attribute values* with probability distributions
  - Propagate probabilities through query execution
  - Interpretation according to the "possible worlds semantics"

Semantics of Query Processing



Semantics of Query Processing

How to **Combine**?

- Allow probabilistic answers.
  - Return all possible tuples along with prob. [Dalvi, Suciu '04]
  - Return tuples with annotations [Green et al. '06]

- What if we want a single deterministic answer?
  - Probabilistic thresholding [Dalvi, Suciu '04]
    - Return all tuples s.t. t appears in the answer w.p. >=Threshold
  - Sampling
  - Top-k queries ?

Semantics of Top-k Queries

pw 1:	pw 2:	pw 3:	pw 4:	
t_1	t_1	t_2	t_2	
t_2	t_3	t_3	t_4	…
t_3	t_4	t_5	t_5	
⋮	⋮	⋮	⋮	

- Many prior proposals for combining them
  - U-top-k, U-rank-k [Soliman et al. '07]
  - Probabilistic Threshold (PT-k) [Hua et al. '08]
  - Global-top-k [Zhang et al. '08]
  - Expected Rank [Cormode et al. '09]
  - Parameterized Ranking Function (PRF) [Li et al. '09]

But, formal semantics are lacking.

Consensus Answers

- Think of each possible answer as a point in the space.

  Suppose d() is a distance metric between answers.
- **Consensus Answers:**

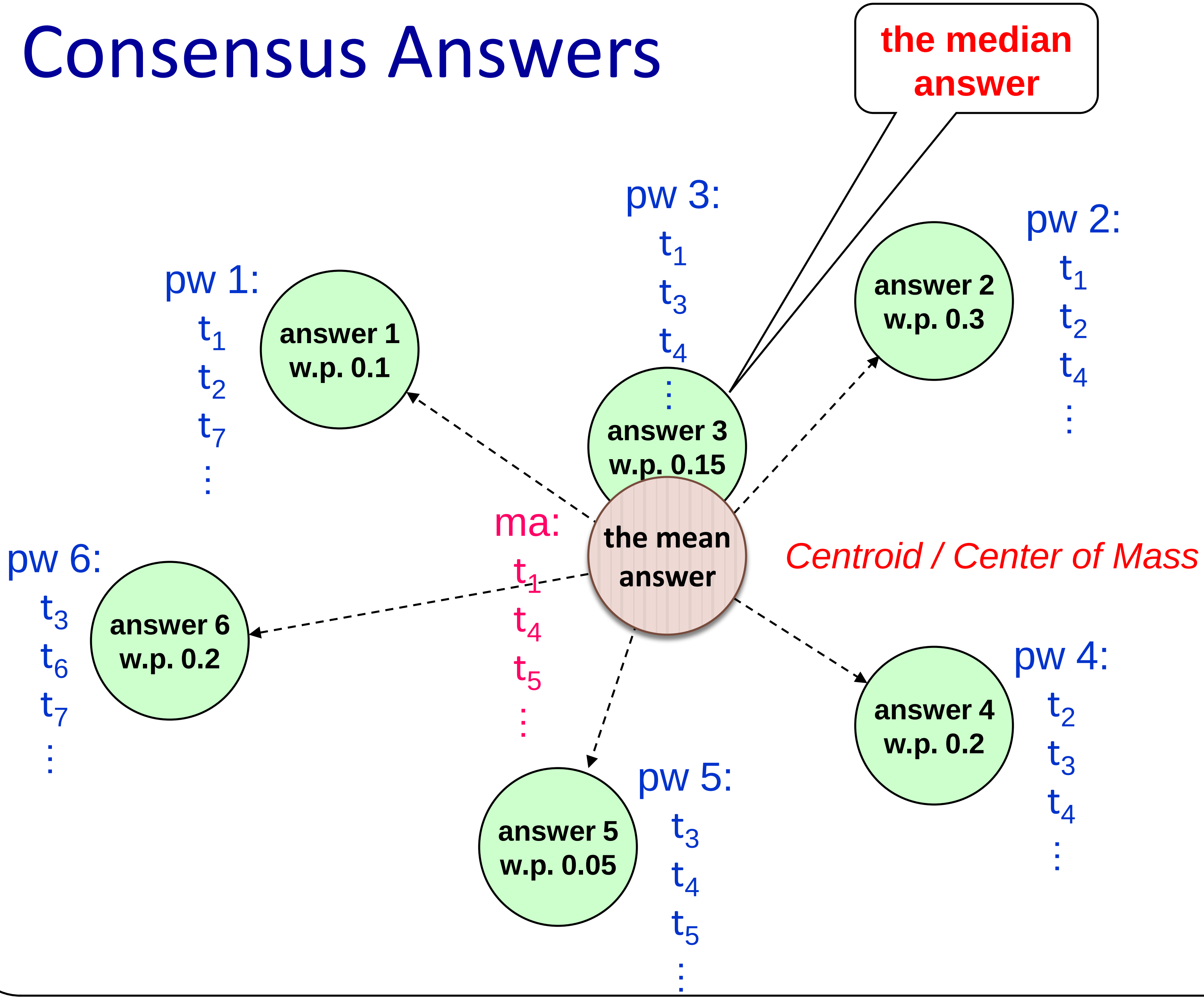
  A single deterministic answer

$$\tau = \arg\min_{\tau' \in \Omega} \{ \mathsf{E}[\mathsf{d}(\tau', \tau_{pw})] \}.$$

where τ_{pw} is the answer for the possible world pw

- Mean Answers: Ω is the set of feasible answers
- Median Answers: Ω is the set of possible answers

Consensus Answers



Related Work

- Rank Aggregation [Dwork et al. ’01], [Ailon ‘07]
  - Original work in voting systems [Condorcet ‘1785]
  - Goal: Combine rankings provided by different experts
- Consensus Clustering [Ailon et al. ’08]
  - Goal: Aggregate a set of clusterings to minimize the disagreements
- Probabilistic Query Processing
  - Dichotomy result: Conjunctive query evaluation is either PTIME or #P-Complete [Dalvi , Suciu ’04]
  - Finding consensus answers a much harder problem (NP-hard even if there is a safe plan)

Outline

- Problem Definition: Consensus Answers
- Models: BID, Probabilistic and/xor tree
- Set Distance Metrics
- Top-k Queries
- Other Types of Queries
- Conclusion

Probabilistic Database Models

- Tuple-independence Model

  The existence of each tuple is independent of other tuples

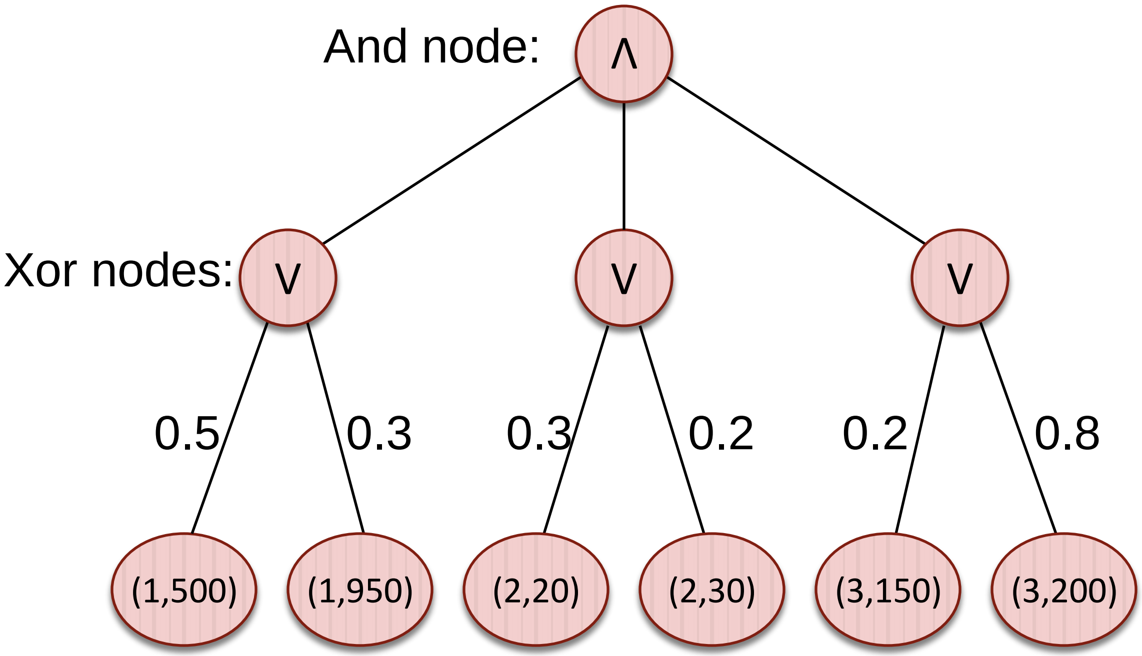
- Block-independent Disjoint (BID) Scheme

Key	Attr 1	Prob
1	500	0.5
1	950	0.3
2	20	0.3
2	30	0.2
3	150	0.2
3	200	0.8

Tuples with the same key are mutually exclusive.

Probabilistic Database Models

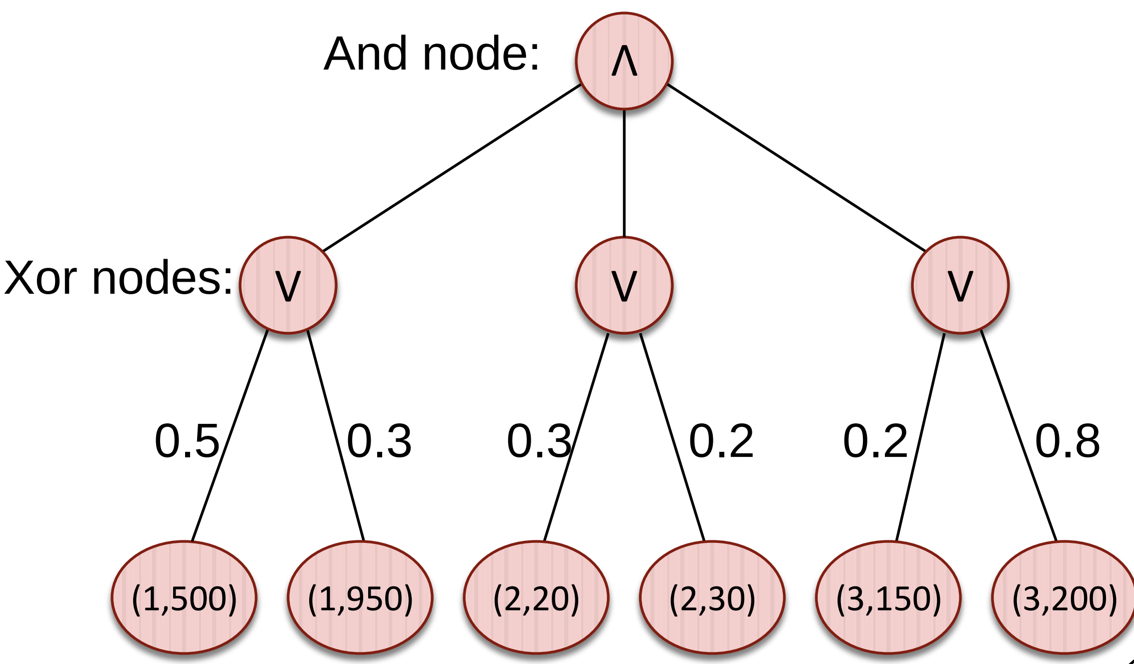
- Probabilistic And/Xor Trees
  - Capture two types of correlations: mutual exclusivity and coexistence.



Possible Worlds	Pr
(3,150)	0.02
(3,200)	0.08
…….	
(1,500);(2,20);(3,150)	0.03
(1,950);(2,20);(3,150)	0.018
…….	

Probabilistic Database Models

- Probabilistic And/Xor Trees
  - Capture two types of correlations: mutual exclusitivity and coexistence.

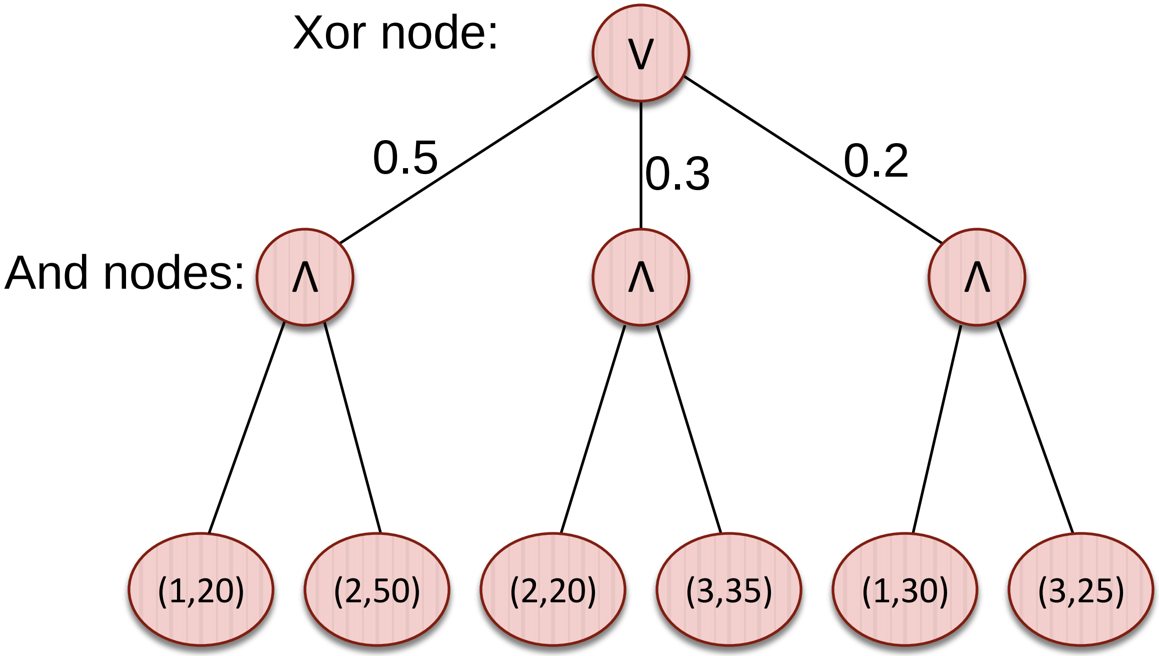


Possible Worlds	Pr
(3,150)	**0.02**
(3,200)	0.08
…….	
(1,500);(2,20);(3,150)	0.03
(1,950);(2,20);(3,150)	0.018
…….	

(1-0.5-0.3)*(1-0.3-0.2)*0.2=0.02

Probabilistic Database Models

- Probabilistic And/Xor Trees



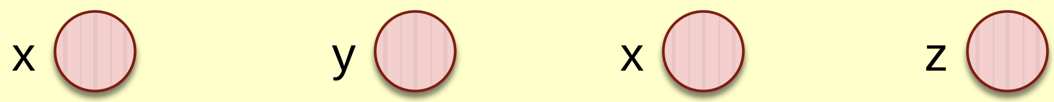
Possible Worlds	Pr
(1,20);(2,50)	0.5
(2,20);(3,35)	0.3
(1,30);(3,25)	0.2

- And/Xor trees can represent any finite set of possible worlds (not necessarily compact).

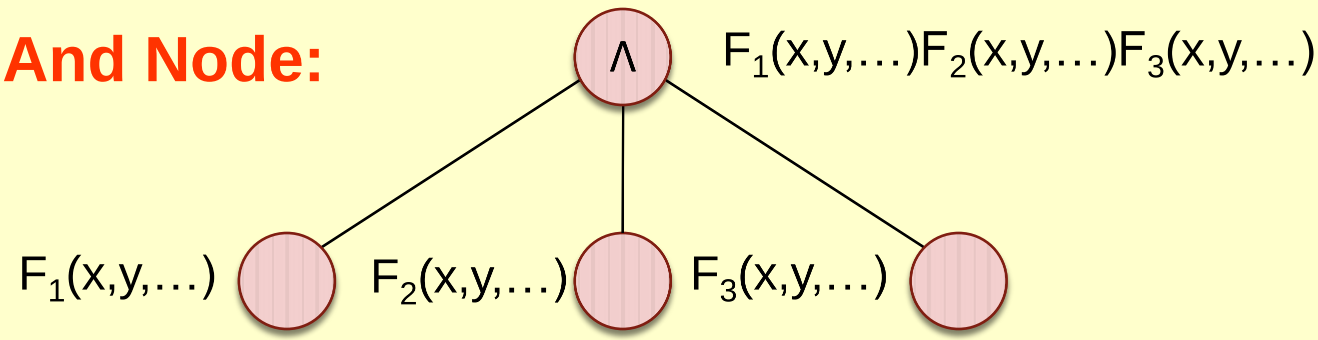
Computing Probabilities on And/Xor Trees

Generating Function Method:

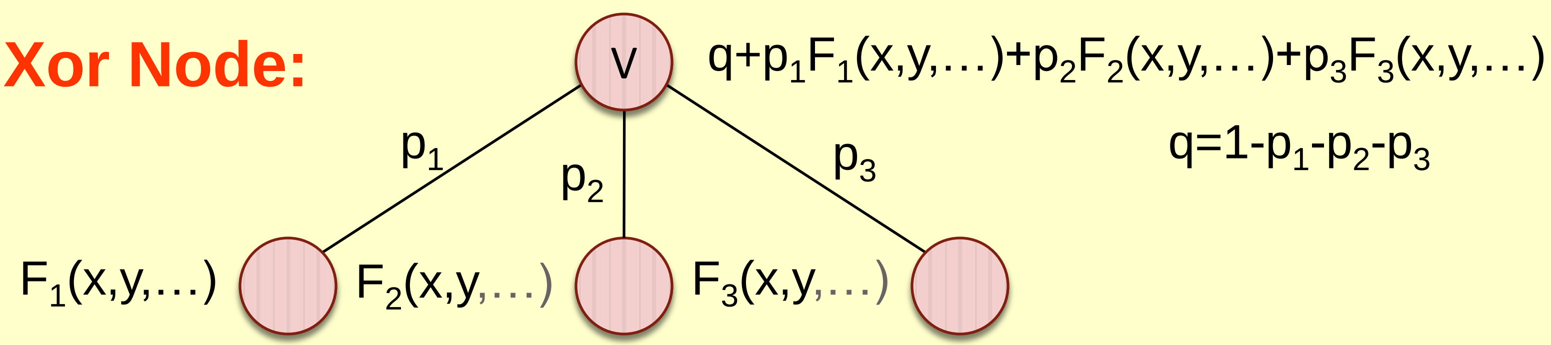
Leaves:



And Node:

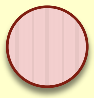


Xor Node:



Computing Probabilities on And/Xor Trees

Generating Function Method:

Root:  $F(x,y,\ldots) = \sum_{ij\ldots} c_{ij\ldots} x^i y^j \ldots$

THM: The coefficient $c_{ij\ldots}$ of the term $x^i y^j \ldots$

= total prob of the possible worlds which contain

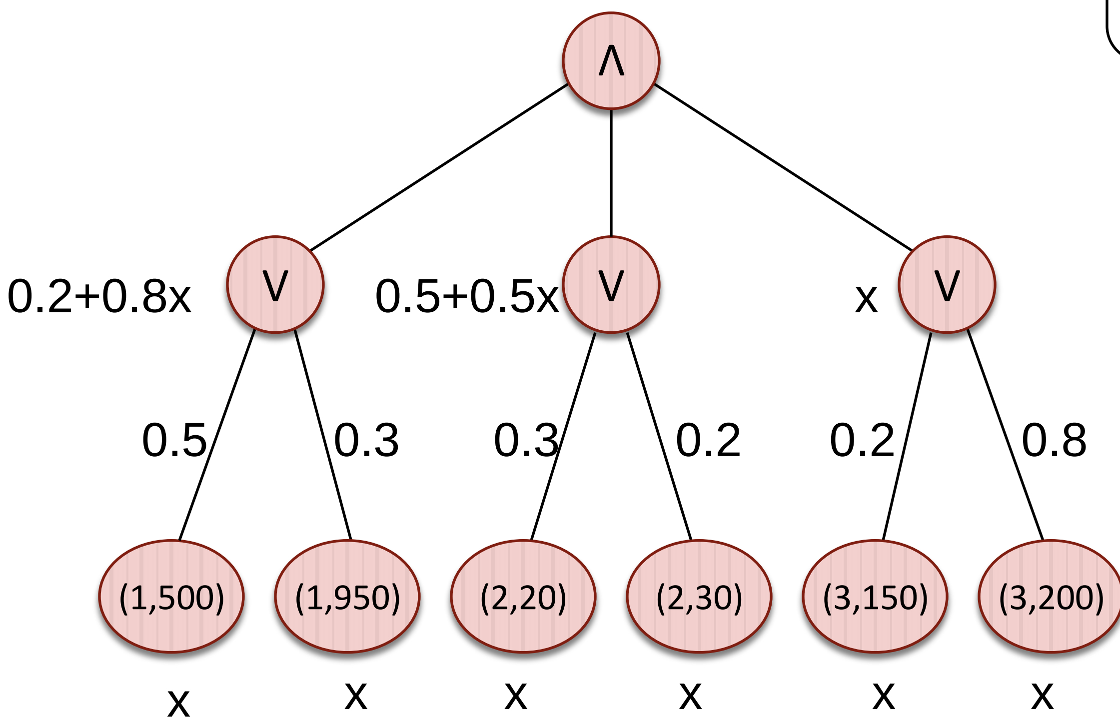
- i tuples annotated with x,
- j tuples annotated with y,……

Computing Probabilities on And/Xor Trees

Example: Computing the prob. dist. of the size of the pw

$(0.2+0.8x)(0.5+0.5x)x = 0.4\,x^3+0.5\,x^2+0.1\,x \Longrightarrow$

Pr(|pw|=3)=0.4
Pr(|pw|=2)=0.5
Pr(|pw|=1)=0.1



Computing Probabilities on And/Xor Trees

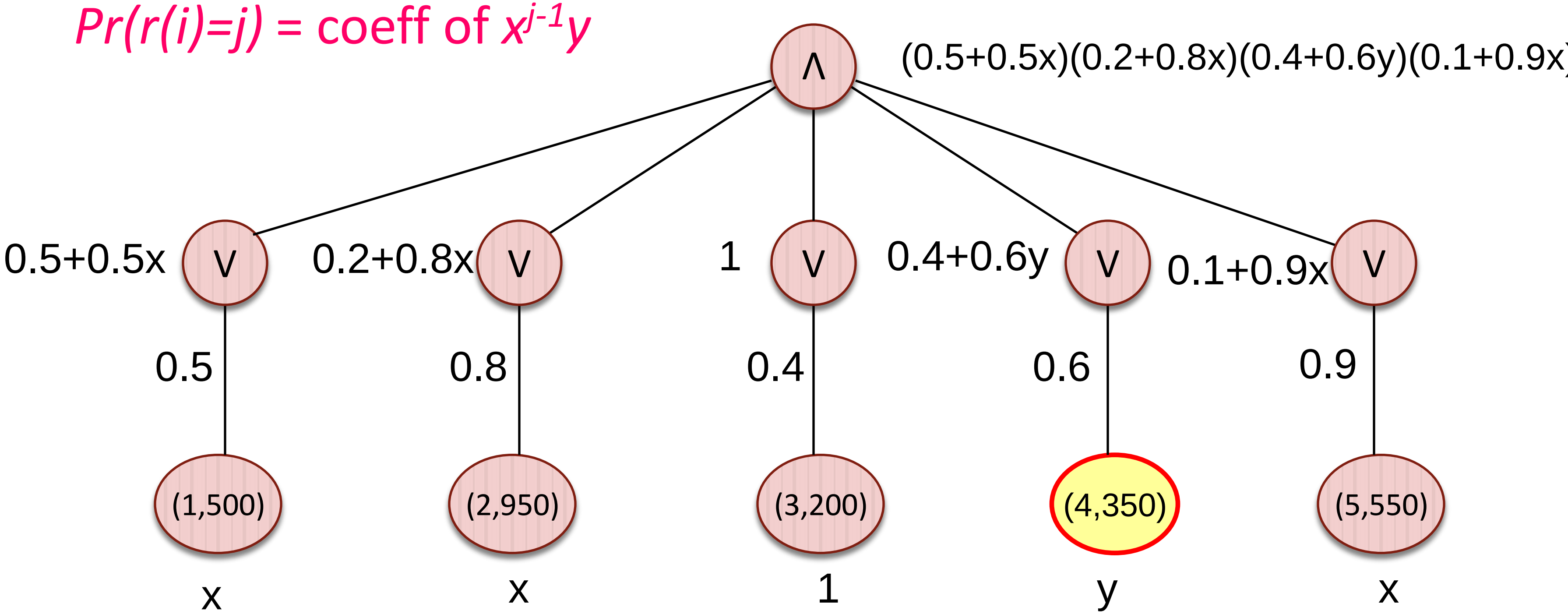
Example: Computing the rank distribution

r(i) : the rank of tuple i.

r(i)=j if and only if (1) *j-1* tuples with higher scores appear

(2) tuple *i* appears

$Pr(r(i)=j)$ = coeff of $x^{j-1}y$



Outline

- Problem Definition: Consensus Answers
- Models: BID, Probabilistic and/xor tree
- Set Distance Metrics
- Top-k Queries
- Other Types of Queries
- Conclusion

Set Distance Metrics

- Think of the relations (either existing or results of conjunctive queries) as sets.

- Symmetric Difference:

$$d_\Delta(\tau_1, \tau_2) = |(\tau_1 \setminus \tau_2) \cup (\tau_2 \setminus \tau_1)| = |(\tau_1 \cup \tau_2) \setminus (\tau_1 \cap \tau_2)|$$

THM: The mean answer under the symmetric difference distance is the set of all tuples with probability >0.5.

THM: For conjunctive queries over tuple independent databases, finding the median answer under the symmetric difference distance is NP-Hard (even if the query has a safe plan).

Reduction from MAX-2-SAT

Set Distance Metrics

- Jaccard Distance

$$d_J(S_1, S_2) = \frac{|S_1 \Delta S_2|}{|S_1 \cup S_2|}.$$

- LM: For tuple independent databases, if the mean world contains tuple t_1 but not tuple t_2, then Pr (t_1) > Pr (t_2).
- Hence, suffices to sort by probabilities, and consider prefixes

- LM: For any fixed world W, $E[d_J(W,pw)]$ can be computed in polynomial time (using generating functions)
- Gives us a polynomial time algorithm

Outline

- Problem Definition: Consensus Answers
- Models: BID, Probabilistic and/xor tree
- Set Distance Metrics
- Top-k Queries
- Other Types of Queries
- Conclusion

Top-k Queries

Symmetric Difference and Probabilistic Threshold Top-k (PT-k)

Mean answer under $d_\Delta(\tau_1, \tau_2) = \frac{1}{2k}|\tau_1 \Delta \tau_2|$

- Find a k-tuple set τ minimizing $\mathsf{E}[d_\Delta(\tau, \tau_{pw})]$

PT-k: Find k tuples with largest $\Pr(r(t) \leq k)$

THM: The two definitions are equivalent.

Top-k Queries

- Intersection Metric: [Fagin et al '03]

$$d_I(\tau_1, \tau_2) = \frac{1}{k} \sum_{i=1}^{k} d_\Delta(\tau_1^i, \tau_2^i)$$

τ^i : top-i tuples of τ

e.g.

τ_1:	5	4	6	3	1
τ_2:	5	6	2	7	3

$d_I(\tau_1,\tau_2)$=
1/5(0 + 1/4*2 +1/6*2 + 1/8* 4 +1/10*4)

Top-k Queries

- Intersection Metric: [Fagin et al '03]

$$d_I(\tau_1, \tau_2) = \frac{1}{k}\sum_{i=1}^{k} d_\Delta(\tau_1^i, \tau_2^i)$$

For any fixed top-k answer τ, we have

$$\mathsf{E}[\mathsf{d}_I(\tau, \tau_{pw})] = \frac{1}{\mathrm{k}}\sum_{i=1}^{\mathrm{k}} \mathsf{E}[\mathsf{d}_\Delta(\tau^i, \tau_{pw}^i)]$$

$$= \frac{1}{\mathrm{k}}\sum_{i=1}^{\mathrm{k}} \frac{1}{i}\left(\mathrm{k} + \sum_{t \in T} \mathsf{Pr}(r(t) \le \mathrm{k}) - 2\sum_{t \in \tau^i} \mathsf{Pr}(r(t) \le i)\right)$$

Thus we need to find τ which maximizes

$$A(\tau) = \sum_{i=1}^{\mathrm{k}} \left(\frac{1}{i}\sum_{t \in \tau^i} \mathsf{Pr}(r(t) \le i)\right).$$

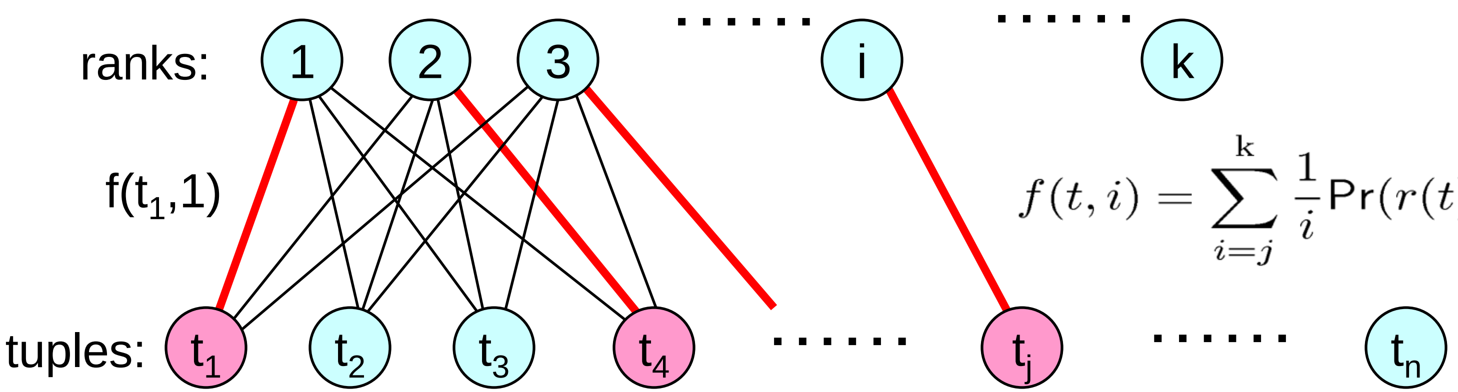
Top-k Queries

- Intersection Metric: [Fagin et al '03]

$$A(\tau) = \sum_{t \in T} \sum_{j=1}^{k} \left(\delta(t = \tau(j)) \sum_{i=j}^{k} \frac{1}{i} \Pr(r(t) \le i) \right)$$

Where $\delta(true) = 1 \text{ and } \delta(false) = 0$

Reduce to the **Max-weight Matching** Problem:



$$f(t,i) = \sum_{i=j}^{k} \frac{1}{i} \Pr(r(t) \le i)$$

Top-k Queries

- Spearman's Footrule [Fagin et al. '03]
  - Extension of traditional footrule distance to partial rankings

$$\mathrm{d}_F(\tau_1, \tau_2) = (k+1)|\tau_1 \Delta \tau_2| + \sum_{t \in \tau_1 \cap \tau_2} |\tau_1(t) - \tau_2(t)| - \sum_{t \in \tau_1 \setminus \tau_2} \tau_1(t) - \sum_{t \in \tau_2 \setminus \tau_1} \tau_2(t).$$

  - Polynomial time algorithm (by reduction to min-cost matching)

- Kendall's tau Distance [Fagin et al. '03]
  - Measures the number of inversions
  - NP-hard [Dwork et al '01]
    - Even for only four possible worlds
  - 3/2-approximation
    - By adapting the algorithm by [Ailon '07]
  - Open question: The complexity for a tuple independent DBs

Outline

- Problem Definition: Consensus Answers
- Models: BID, Probabilistic and/xor tree
- Set Distance Metrics
- Top-k Queries
- Other Types of Queries
- Conclusion

Other Types of Queries

- Aggregate Queries
  - **SELECT** groupname, count(*) **FROM** R **GROUP BY** groupname
  - Distance: squared vector distance
  - Mean answer is trivial: take average count for each group
  - Median answer: 4-approximation
- Clustering
  - A somewhat simplified model
  - Distance: consensus clustering distance
  - 4/3-approximation for finding the mean clustering

Conclusion

- Proposed the notion of Consensus Answers for probabilistic databases
  - Lends precise and formal semantics to query answers
- Algorithms for finding consensus answers for many queries
  - For the rich probabilistic and/xor tree model
- Future work:
  - Examining utility of consensus answers in practice
  - Handling other types of queries: range queries, frequent items, clustering
  - Finding connections to existing query processing semantics

Thanks.