A Unified Approach to Ranking in Probabilistic Databases

Jian Li, Barna Saha, Amol Deshpande

University of Maryland, College Park, USA

Probabilistic Databases

- Motivation: Increasing amounts of uncertain data
  - Sensor Networks; Information Networks
  - Data Integration and Information Extraction
  - X
- Probabilistic databases
  - Annotate *tuples* with existence probabilities, and/or *attribute values* with probability distributions
  - Interpretation according to the "possible worlds semantics"

Possible World Semantics

ID	Score	Prob
t_1	200	0.2
t_2	150	0.8
t_3	100	0.4

A probabilistic table
(assume tuple-independence)

pw1

ID	Score
t_1	200
t_2	150
t_3	100

w.p. 0.064

pw2

ID	Score
t_1	200
t_2	150

w.p. 0.096

pw3

ID	Score
t_2	150
t_3	100

w.p. 0.256

We focus on tuple uncertainty in the talk

Top-k Query Processing

Score values are used to rank the tuples in every *pw*.

ID	Score	Prob
t_1	200	0.2
t_2	150	0.8
t_3	100	0.4

A probabilistic table
(assume tuple-independence)

pw1

ID	Score
t_1	200
t_2	150
t_3	100

w.p. 0.064

pw2

ID	Score
t_1	200
t_2	150

w.p. 0.096

pw3

ID	Score
t_2	150
t_3	100

w.p. 0.256

The top-1 answer for each possible world

Top-*k* Queries: Many Prior Proposals

- Return *k* tuples *t* with the highest *score(t)Pr(t)* [exp. score]
- Returns the most probable top-*k* answer [U-top-k]

  [Soliman Ŝ ű ⌊ ƭ Φ Ω л т ϐ
- At rank *i*, return tuple with max. prob. of being at rank *i* [U-rank-k]

  [Soliman Ŝ ű ⌊ ƭ Φ Ω л т ϐ
- Return *k* tuples *t* with the largest *Pr (r(t) · k)* values [PT-k/GT-k]

  [Hua Ŝ ű ⌊ ƭ Φ Ω л у ϐ ώ ½ Ƙ ⌊ ƴ Ǝ Ŝ ű ⌊ ƭ Φ Ω л у ϐ
- Return *k* tuples *t* with smallest expected rank: $\ddot{a}_{pw} Pr(pw)\, r_{pw}(t)$

  [Cormode Ŝ ű ⌊ ƭ Φ Ω л ф ϐ

Top-*k* Queries

- Which one should we use???
- Comparing different ranking functions

Normalized Kendall Distance between two top-*k* answers:

Penalizes #reversals and #mismatches

Lies in [0,1], 0: Same answers, 1: Disjoint answers

	E-Score	PT/GT	U-Rank	E-Rank	U-Top
E-Score	----	0.124	0.302	0.799	0.276
PT/GT	0.124	----	0.332	0.929	0.367
U-Rank	0.302	0.332	-----	0.929	0.204
E-Rank	0.799	0.929	0.929	----	0.945
U-Top	0.276	0.367	0.204	0.945	----

Real Data Set: 100,000 tuples, Top-100

Outline

- Parameterized Ranking Functions
- Computing PRF
  - Independent Tuples
  - Computing PRF^e(®)
  - X-Tuples
  - Summery of Other Results
- Approximating Ranking Functions
- Experiments

Parameterized Ranking Function

- Weight Function: ω : (tuple, rank) $\rightarrow \mathbb{C}$
- Parameterized Ranking Function (PRF)

$$\Upsilon_\omega(t) = \sum_{i>0} \omega(t,i) \cdot \Pr(r(t) = i).$$

Probability that t is ranked at position i across possible worlds

Return k tuples with the highest $|\Upsilon_\omega|$ values.

Outline

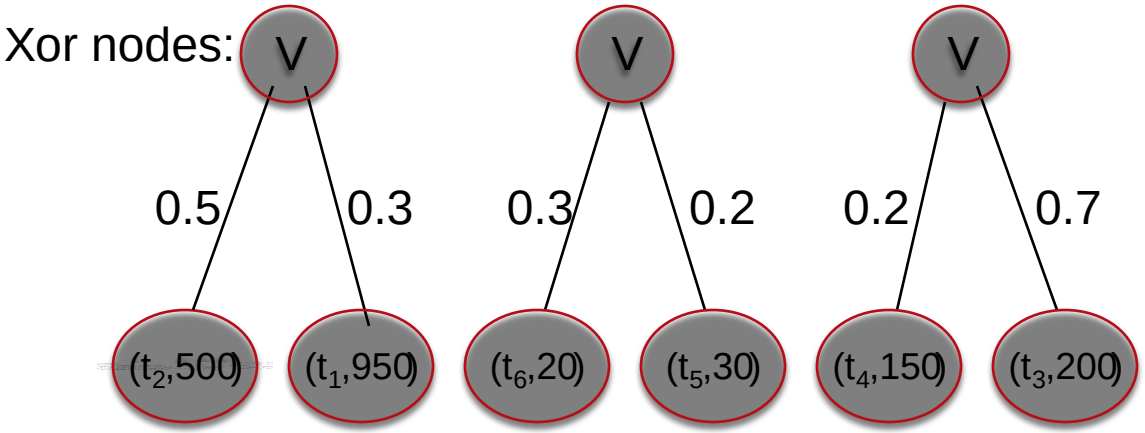
- Parameterized Ranking Functions
-

Outline

- Parameterized Ranking Functions
- Computing PRF
  - Independent Tuples
  - Computing $PRF^e(\circledR)$
  - X-Tuples
  - Summery of Other Results
- Approximating Ranking Functions
- Experiments

Computing PRF: Tuples

- Capture mutual exclusivity correlations



Possible Worlds	Pr
t_4	0.02
t_3	0.08
é é .	
t_1, t_4, t_6	0.03
t_1, t_3, t_6	0.018
é é .	

Summary of Other Results

- PRF computation for probabilistic and/or trees
  - Generalizes x-tuples by allowing both mutual exclusivity and coexistence
  - PRF computation: $O(n^3)$
  - PRF computation : $O(n \log n + nd)$ (d = height of the tree)
- PRF computation on graphical models
  - A polynomial time algorithm when the junction tree has bounded treewidth
    - A nontrivial dynamic program combined with the generating function method.

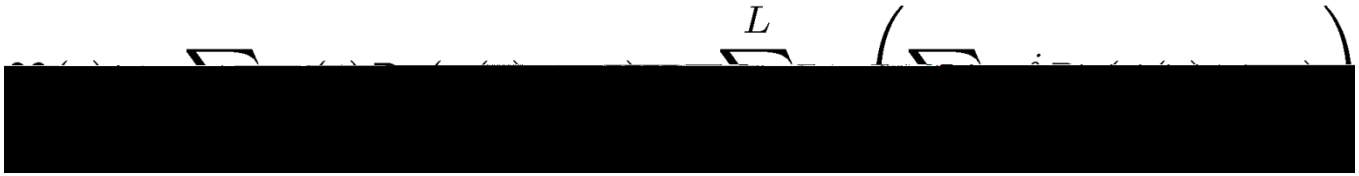
Outline

- Parameterized Ranking Functions
- Computing PRF
  - Independent Tuples
  - Computing PRF^e(®)
  - X-Tuples
  - Summery of Other Results
- Approximating Ranking Functions
- Experiments

Approximating Ranking Functions

Approximating PRF by a linear combination of PRF

- Suppose $\omega(i) \approx \sum_{l=1}^{L} u_l \alpha_l^i$



- Reduce to L individual PRF computations
- Running time $O(n \log n + nL)$ (as opposed to $O(n^2)$)

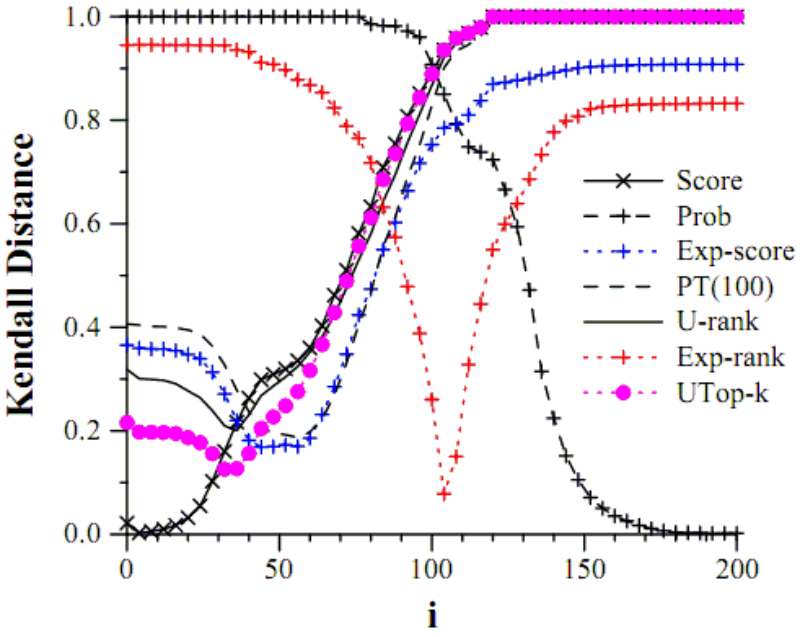
- We developed a scheme based on adapting the *discrete Fourier transformation* of !
- Works very well for monotonically non-increasing !
  - E.g. the step function ($PT/4T$)
- Details in the paper

Outline

- Parameterized Ranking Functions
- Computing PRF
  - Independent Tuples
  - Computing PRF^e(®)
  - X-Tuples
  - Summery of Other Results
- Approximating Ranking Functions
- Experiments

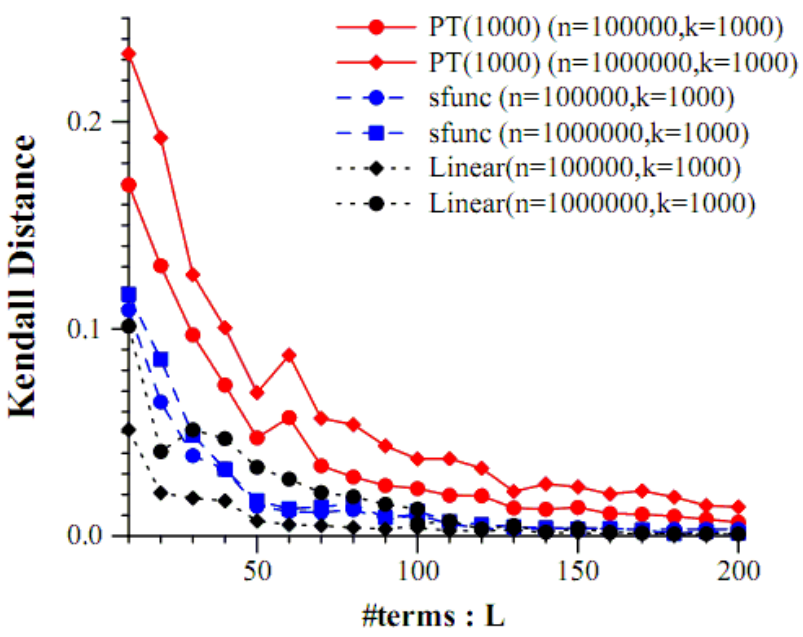
Experiments: Approximation Ability

Using a single PRF



Approximating other functions using PRF(®)
®= 1-0.9^i

A linear combination of PRFs

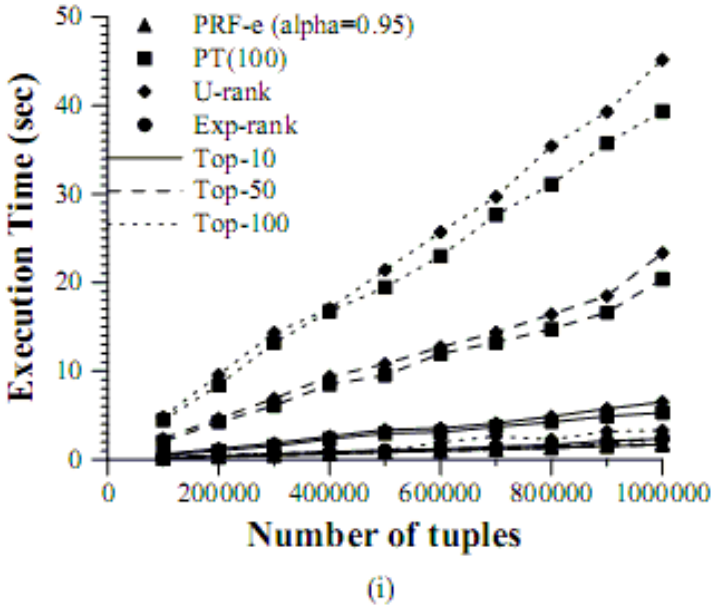


No. of Terms vs Approximation Quality

Real Iceberg sighting dataset: 100,000 tuples

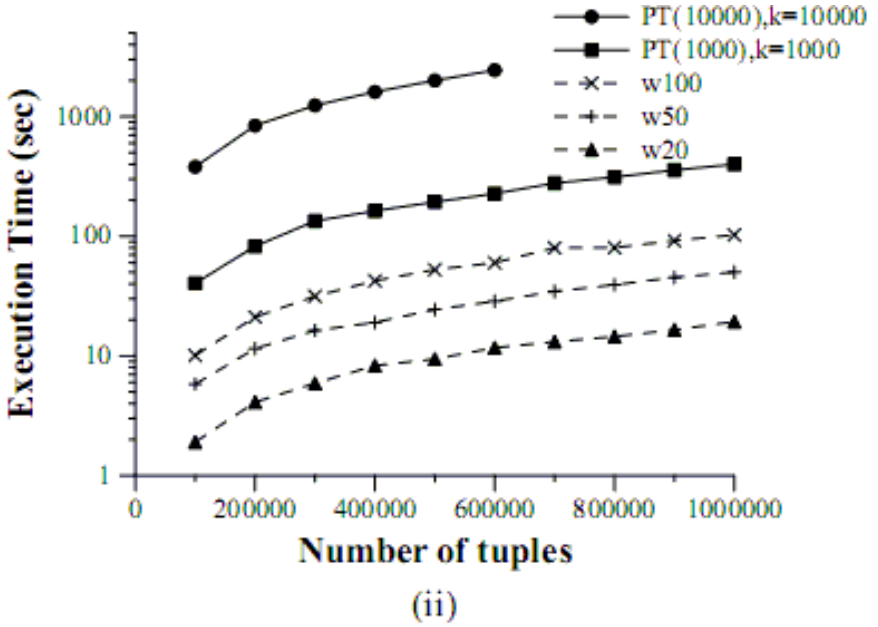
Experiments: Execution Time

PRF^e vs Others



(i)

Approx vs Exact



(ii)

Reallcbergsighting dataset

Conclusions

- Proposed a unifying framework for ranking over probabilist databases through:
  - *Parameterized* ranking functions
  - Incorporation of user feedback
- Designed highly efficient algorithms for computing PRF an PRF
- Developed novel approximation techniques for approximat PRF with PRF