Ranking Continuous Probabilistic Datasets

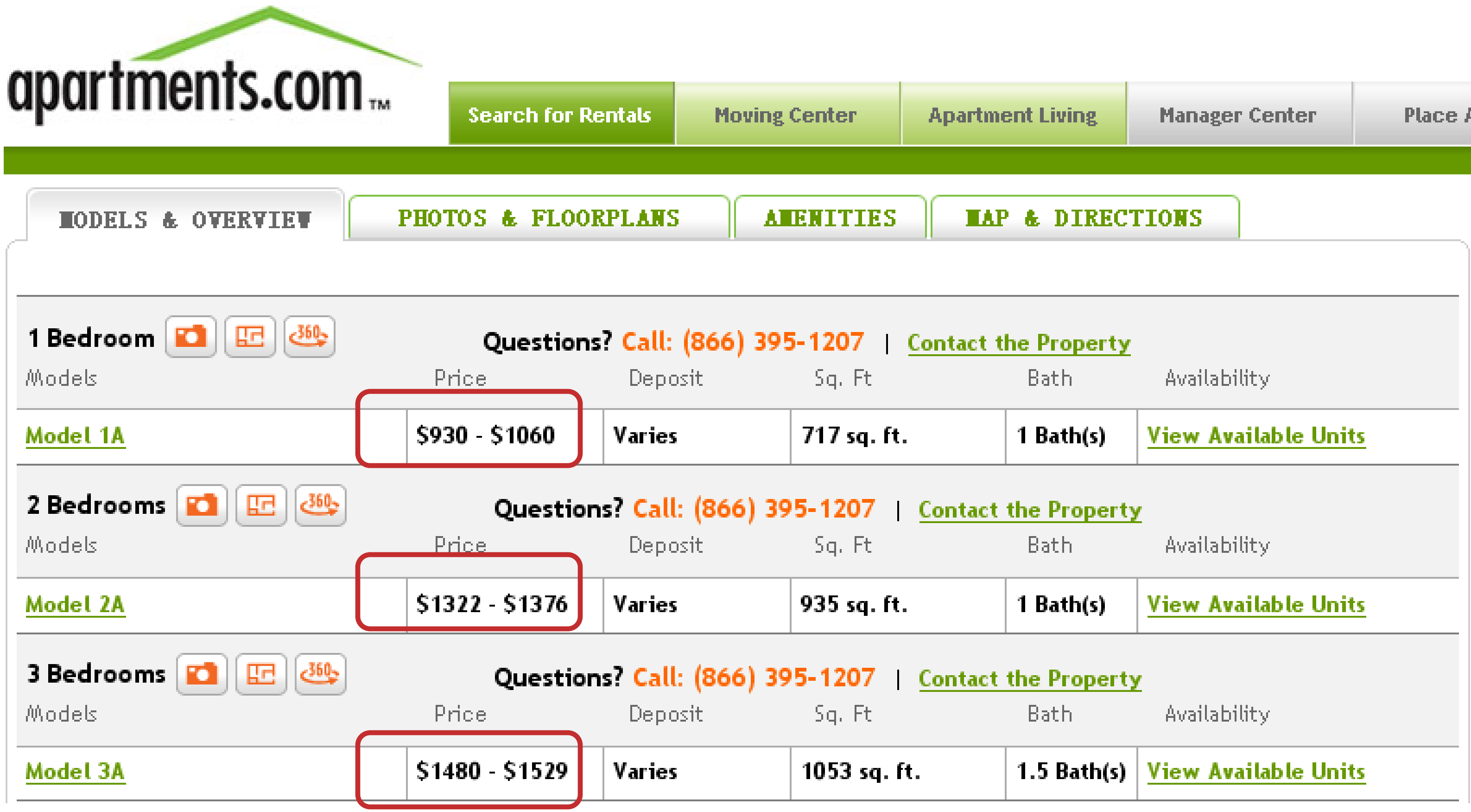
Jian Li, University of Maryland, College Park

Joint work with Amol Deshpande (UMD)

VLDB 2010, Singapore

Motivation

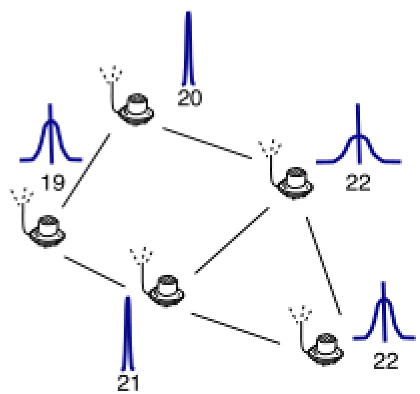
- Uncertain data with continuous distributions is ubiquitous



Uncertain scores

Motivation

- Uncertain data with continuous distributions is ubiquitous



Sensor ID	Temp.
1	Gauss(40,4)
2	Gauss(50,2)
3	Gauss(20,9)
…	…

- Many probabilistic database prototypes support continuous distributions.
  - Orion [Singh et al. SIGMOD'08], Trio [Agrawal et al. MUD'09], MCDB [Jampani et al. SIGMOD'08],], PODS [Tran et al. SIGMOD'10], etc.

Motivation

- Uncertain data with continuous distributions is ubiquitous.
- Many probabilistic database prototypes support continuous distributions.
  - Orion [Singh et al. SIGMOD'08], Trio [Agrawal et al. MUD'09], MCDB [Jampani et al. SIGMOD'08],], PODS [Tran et al. SIGMOD'10], etc.
- Often need to "rank" tuples or choose "top k"
  - Deciding which apartments to inquire about
  - Selecting a set of sensors to "probe"
  - Choosing a set of stocks to invest in
  - …

Ranking in Probabilistic Databases

- Possible worlds semantics

ID	Score
t_1	Uni(100,200)
t_2	150
t_3	Gauss(100,3)

A probabilistic table
(assume tuple-independence)

pw1

ID	Score
t_1	125
t_2	150
t_3	97

ranking

t_2, t_1, t_3

pw2

ID	Score
t_1	200
t_2	150
t_3	102

ranking

t_1, t_2, t_3

⋮

Uncountable number of possible worlds
A probability density function (pdf) over worlds

Motivation

- Much work on ranking queries in probabilistic databases.
  - U-top-k, U-rank-k [Soliman et al. ICDE’07]
  - Probabilistic Threshold (PT-k) [Hua et al. SIGMOD’08]
  - Global-top-k [Zhang et al. DBRank’08]
  - Expected Rank [Cormode et al. ICDE’09]
  - Typical Top-k [Ge et al. SIGMOD’09]
  - Parameterized Ranking Function [Li et al. VLDB’09]
  - .....
- Most of them focus on discrete distributions.
  - Some simplistic methods, such as discretizing the continuous distributions, have been proposed, e.g., [Cormode et al. ICDE’09].
  - One exception: Uniform distributions [Soliman et al. ICDE’09]

Parameterized Ranking Functions

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

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

Positional Probability: *Probability that* t *is ranked at position* i *across possible worlds*

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

Parameterized Ranking Functions

- PRF generalizes many previous ranking functions.
  - PT-k/GT-k: return top-k tuples such that $Pr(r(t) \leq k)$ is maximized.
    - $\omega(t,i) = 1$ if $i \leq k$ and $\omega(t,i)=0$ if $i>k$
  - Exp-rank: Rank tuple by an increasing order of $E[r(t)]$.
    - $\omega(t,i) = n-i$
  - Can approximate many others using linear combinations of PRFe functions.
- Weights can be learned using user feedbacks.

Outline

- A closed-form *generating function* for the positional probabilities.
- Polynomial time *exact* algorithms for uniform and piecewise polynomial distributions.
- Efficient approximations for arbitrary distributions based on *spline* approximation.
- Theoretical comparisons with *Monte-Carlo* and *Discretization*.
- Experimental comparisons.

A Straightforward Method

- Suppose we have three r.v. s_1, s_2, s_3 with pdf μ_1, μ_2, μ_3, respectively.

$$\Pr(s_1 < s_2) = \int_{-\infty}^{+\infty} \mu_1(x_1) \underbrace{\int_{x_1}^{+\infty} \mu_2(x_2) dx_2}_{\Pr(s_1 < s_2 \mid s_1 = x_1)} dx_1$$

- Similarly,

$$\Pr(s_1 < s_2 < s_3) = \int_{-\infty}^{+\infty} \mu_1(x_1) \int_{x_1}^{+\infty} \mu_2(x_2) \int_{x_2}^{+\infty} \mu_3(x_3) dx_3 dx_2 dx_1$$

Difficulty 1: Multi-dimensional integral

$$\Pr(r(s_1) = 3) = \Pr(s_1 < s_2 < s_3) + \Pr(s_1 < s_3 < s_2)$$

Difficulty 2: #terms is possibly exponential

Generating Functions

Let the **cdf** of s_i (the score of *t_i*) be

$$\rho_i(\ell) = \mathsf{Pr}(s_i < \ell) = \int_{-\infty}^{\ell} \mu_i(x)\mathrm{d}x, \quad \bar{\rho}_i(\ell) = 1 - \rho_i(\ell)$$

Theorem:

Define

$$F_i(x) = x\int_{-\infty}^{\infty} \mu_i(\ell) \prod_{j\neq i} \Big(\rho_j(\ell) + \bar{\rho}_j(\ell)x\Big)\mathrm{d}\ell$$

Then, $F_i(x)$ is the **generating function** of the positional probabilities.

$$F_i(x) = \sum_{j\geq 1} \mathsf{Pr}(r(t_i) = j)x^j.$$

Generating Functions

Advantages over the straightforward method:

$$F_i(x) = x \int_{-\infty}^{\infty} \mu_i(\ell) \prod_{j \neq i} \Big(\rho_j(\ell) + \bar{\rho}_j(\ell) x \Big) \mathrm{d}\ell$$

A Polynomial of *x*

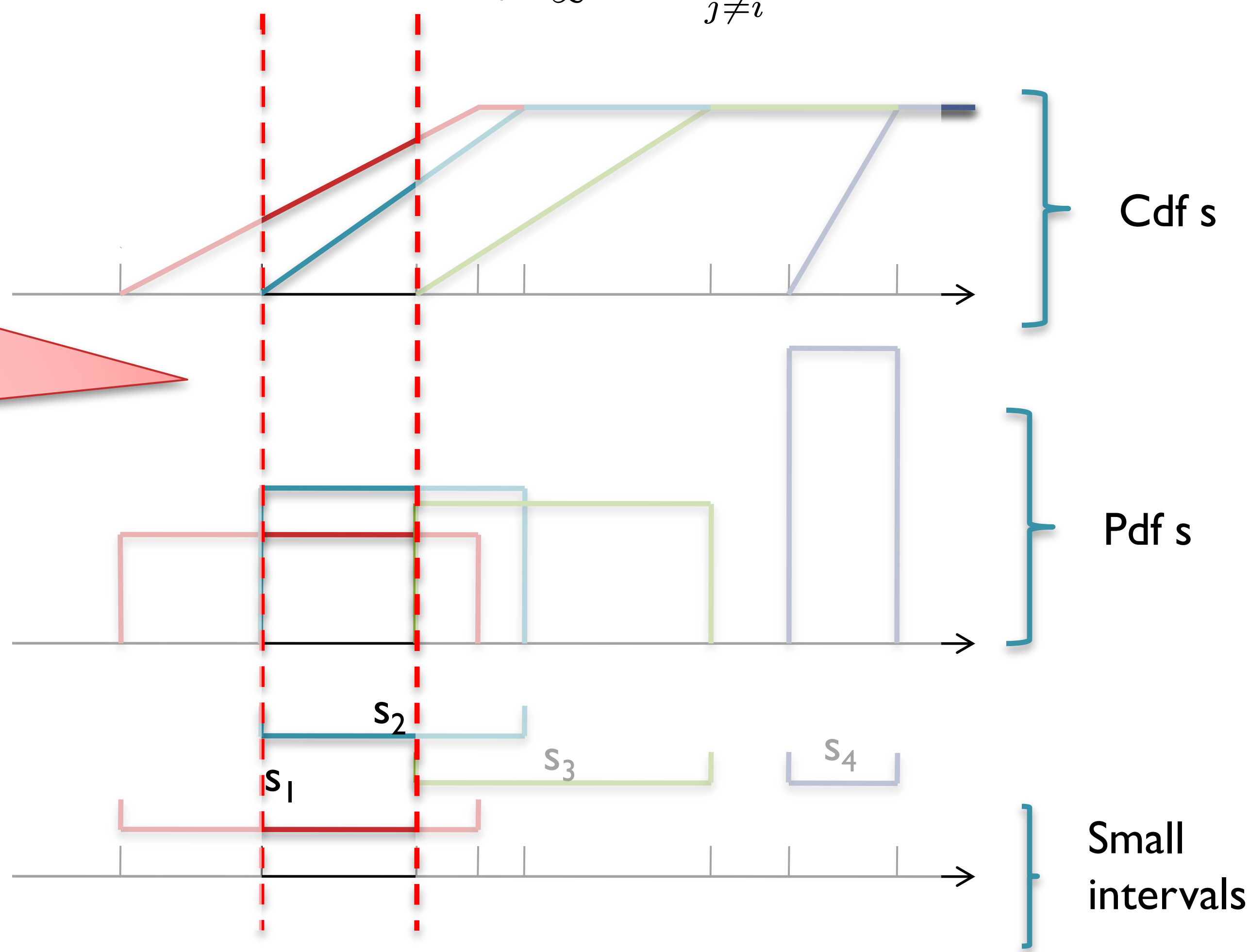
1-dim Integral

No exp. # terms

Uniform Distribution: A Poly-time Algorithm

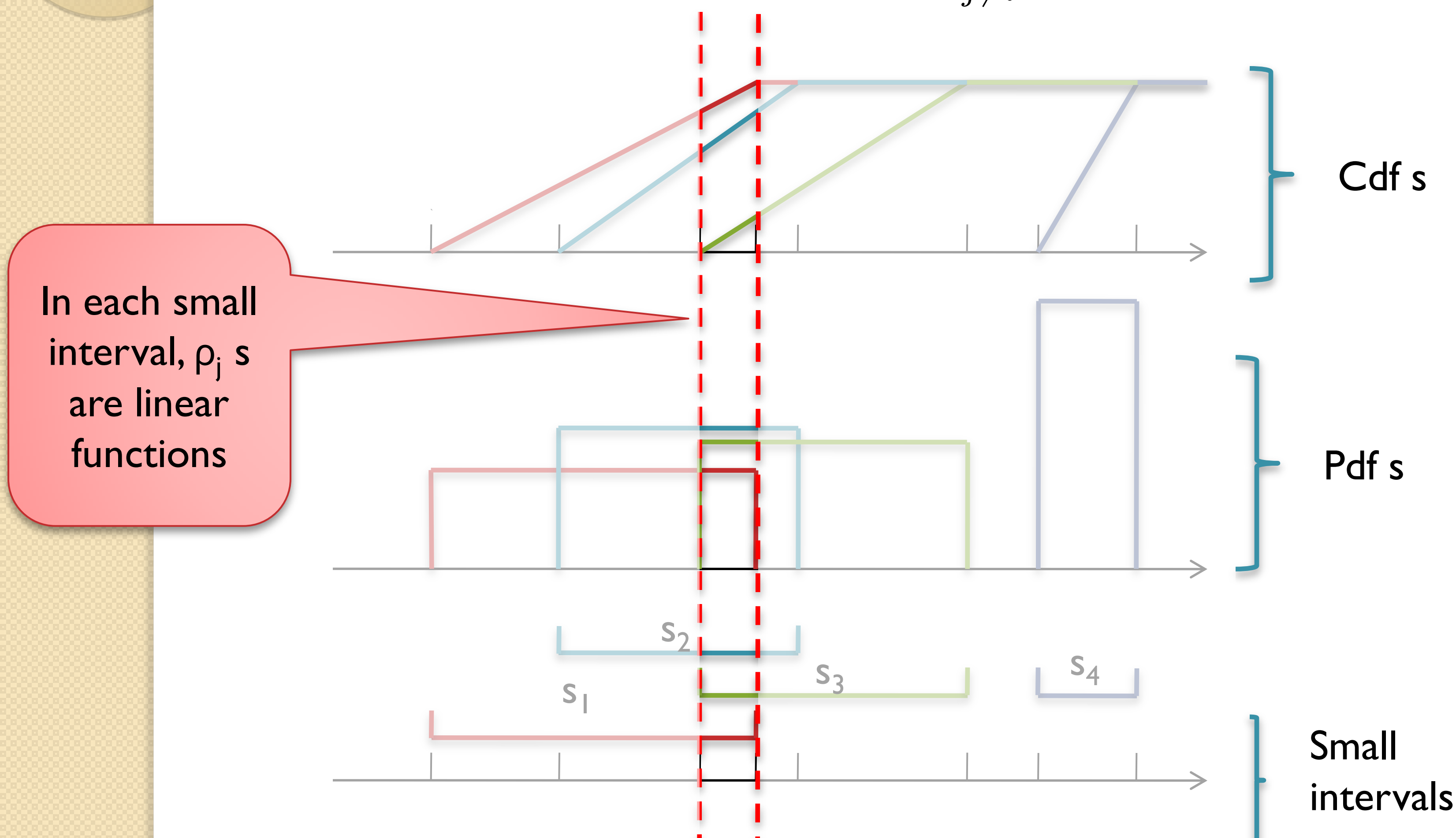
Consider the g.f. $F_i(x) = x \int_{-\infty}^{\infty} \mu_i(\ell) \prod_{j \neq i} \Big(\rho_j(\ell) + \bar{\rho}_j(\ell) x\Big) \mathrm{d}\ell$

In each small interval, ρ_j s are linear functions



Uniform Distribution: A Poly-time Algorithm

Consider the g.f. $F_i(x) = x\int_{-\infty}^{\infty} \mu_i(\ell) \prod_{j\neq i}\Big(\rho_j(\ell) + \bar{\rho}_j(\ell)x\Big)\mathrm{d}\ell$



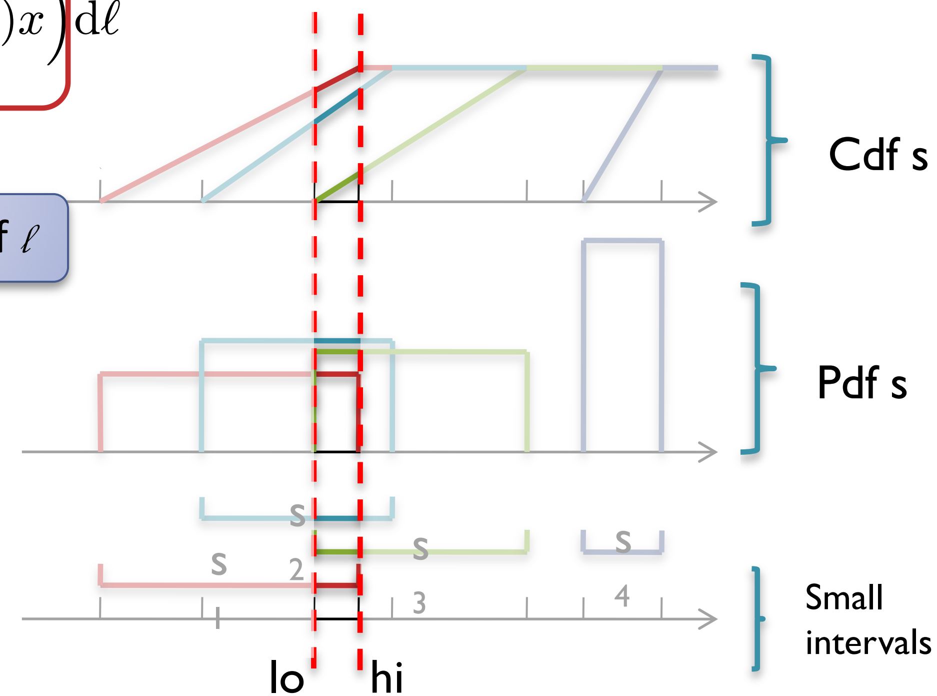
Uniform Distribution: A Poly-time Algorithm

$$F_i(x) = x \int_{lo}^{hi} \boxed{\mu_i(\ell) \prod_{j \neq i} \left(\rho_j(\ell) + \bar{\rho}_j(\ell) x\right)} \mathrm{d}\ell$$

constant

Linear func. of ℓ

Polynomial of *x* and ℓ
Expand in form
$\sum_{j,k} c_{j,k} x^j \ell^k$



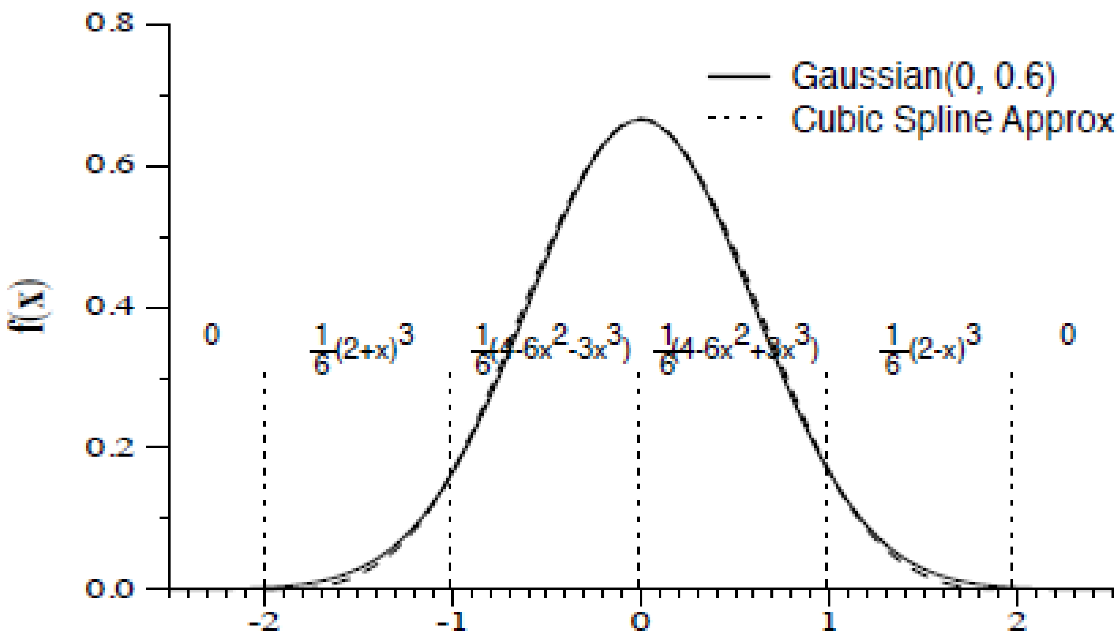
Then, we get $F_i(x) = \mu_i(\ell) \sum_{j,k} c_{j,k} \int_{lo}^{hi} \ell^k \mathrm{d}\ell \cdot x^{j+1}$

Other Poly-time Solvable Cases

- Piecewise polynomial distributions.
  - The cdf ρ_i is piecewise polynomial.
- Combine with discrete distributions.
  - $$S_i = \begin{cases} 100 & \text{w.p. } 0.5, \\ \text{Uni}[150,200] & \text{w.p. } 0.5 \end{cases}$$

General Distribution: Spline Approximations

Spline (Piecewise polynomial): a powerful class of functions to approximate other functions.
Cubic spline: Each piece is a deg-3 polynomial.



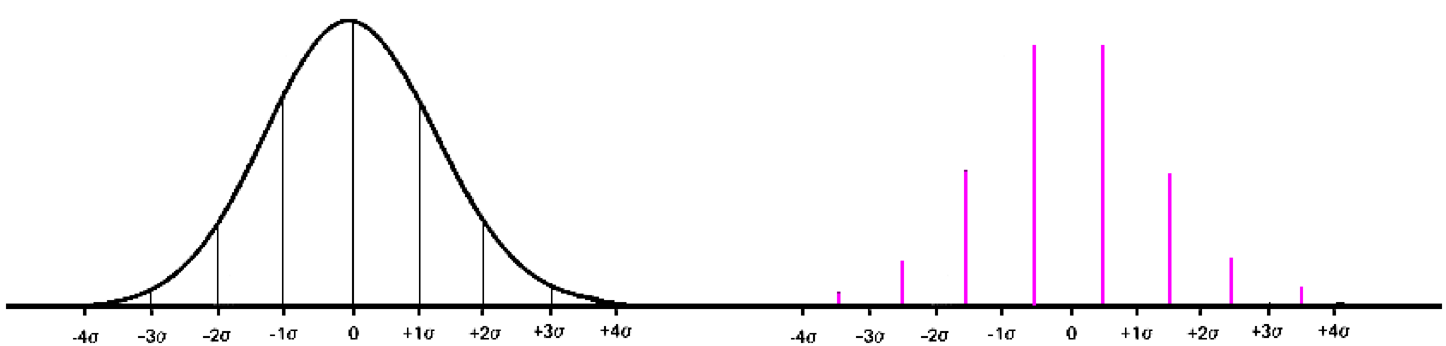
Spline(x) = f(x), Spline'(x) = f'(x) for all break points *x*.

Theoretical Convergence Results

Monte-Carlo: $r_i(t)$ is the rank of t in the ith sample
N is the number of samples

Estimation: $$\tilde{\Upsilon}_\omega(t) = \frac{1}{N}\sum_{i=1}^{N}\omega(t, r_i(t)).$$

Discretization: Approximate a continuous distribution by a set of discrete points. N is the number of break points.



Theoretical Convergence Results

- Spline Approximation: We replace each distribution by a spline with $N=O(n^\beta)$ pieces.

$$|\widehat{\Upsilon}_\omega(t) - \Upsilon_\omega(t)| \le O(n^{3/2-4\beta}).$$

  - Under certain continuity assumptions.

- Discretization: We replace each distribution by $N=O(n^\beta)$ discrete pts.

$$|\widehat{\Upsilon}_\omega(t) - \Upsilon_\omega(t)| \le O(n^{3/2-\beta}).$$

  - Under certain continuity assumptions.

- Monte-Carlo: With $N = \quad (n^\beta \log \frac{1}{\delta})$ samples,

$$\Pr\left(|\tilde{\Upsilon}_\omega(t) - \Upsilon_\omega(t)| \le O(n^{-\beta/2})\right) \ge 1-\delta$$

$\beta=4$

$O(n^{-14.5\beta})$

$O(n^{-2.5\beta})$

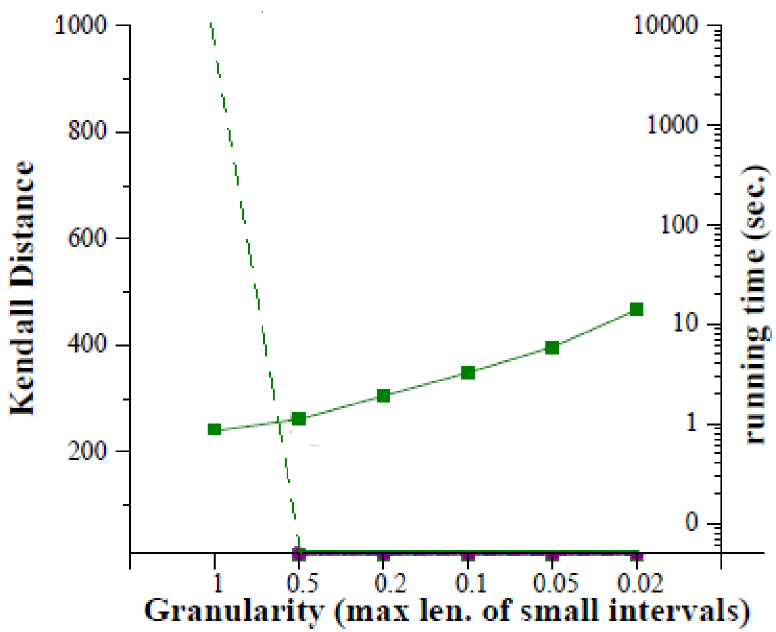
$O(n^{-2\beta})$

Other Results

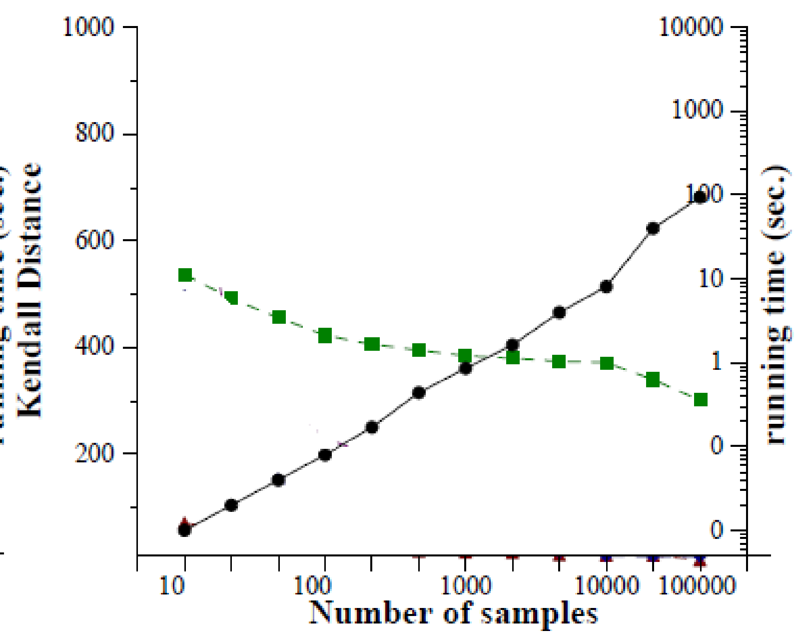
- Efficient algorithm for *PRF-l* (linear weight func.)
  - If no tuple uncertainty, *PRF-l = Expected Rank* [Cormode et al. ICDE09] .
- Efficient algorithm for *PRF-e* (exp. weight func.)
  - Using *Legendre-Gauss quadrature* for numerical integration.
- *K-nearest neighbor* over uncertain points.
  - Semantics: retrieve *k* pts. that have highest prob. being the kNN of the query point *q.*
  - This generalizes the semantics proposed in [Kriegel et al. DASFAA07] and [Cheng et al. ICDE08].
  - score(point *p*) = dist(point *p*, query point *q*).

Experimental Results

Setup: Gaussian distributions. 1000 tuples.
30% uncertain tuples.
Mean: uniformly chosen in [0,1000].
Avg stdvar: 5. Truncation done at 7*stdvar.
Kendall distance: #reversals between two rankings.



(a) Spline

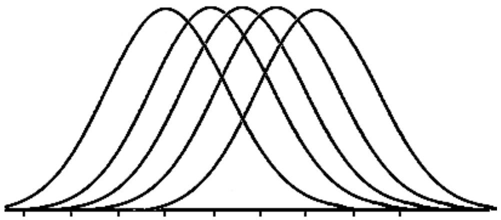


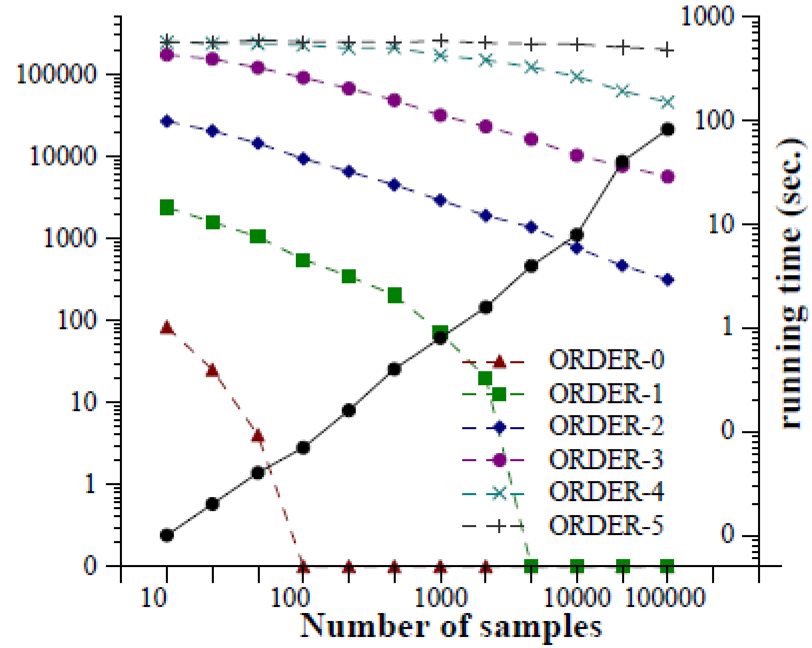
(c) Monte Carlo

Convergence rates of different methods

Experimental Results

Setup: 5 dataset ORDER-d (d=1,2,3,4,5)
Gaussian distributions. 1000 tuples.
Mean: mean(t_i) = i * 10^{-d} where d=1,2,3,4,5
Stdvar: 1.
Kendall distance: #reversals between two rankings.





Take-away: Spline converges faster, but has a higher overhead.
Discretization is somewhere between Spline and Monte-Carlo.

Conclusion

- Efficient algorithms to rank tuples with continuous distributions.
- Compare our algorithms with Monte-Carlo and Discretization.
- Future work:
  - Progressive approximation.
  - Handling correlations.
  - Exploring spatial properties in answering kNN queries.

Thanks

Note

- Texpoint 3.2.1