Ranking Continuous Probabilistic Datasets

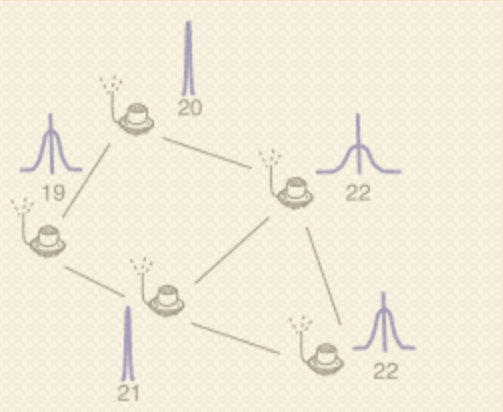
Jian Li, University of Maryland, College Park

Joint work with Amol Deshpande (UMD)

VLDB 2010, Singapore

Motivation

- Uncertain data with continuous distributions is ubiquitous



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

Parameterized Ranking Functions

- PRF generalizes many previous ranking functi
  - PT-k/GT-k: return top-k tuples such that *Pr(r(t)≤k)* is maximized.
    - *ω(*t,i) = 1 if *i≤k* and *ω(*t,i)=0 if i>k
  - Exp-rank: Rank tuple by an increasing order of E[r(t)].
    - *ω(*t,i) = n-i
  - Can approximate many others using linear combinations of PRF 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 thecdf of s_i (the score of t_i) be

$$\text{½}_i(\grave{}) = \Pr(s_i < \grave{}) = \int_{-\infty}^{\ell} \text{1}_i(x)dx; \quad \text{½}_i(\grave{}) = 1 - \text{½}_i(\grave{})$$

Theorem:

Define

$$z_i(x) = x \int_{-\infty}^{\infty} \text{1}_i(\grave{}) \prod_{j \neq i} \left(\text{½}_j(\grave{}) + \text{½}_j(\grave{})x \right) d\grave{}$$

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

$$z_i(x) = \sum_{j \geq 1} \Pr(r(t_i) = j)x^j:$$

Generating Functions

Advantages over the straightforward method:

$$z_i(x) = x \int_{-\infty}^{\infty} 1_i(\ell) \prod_{j \neq i} \left(\frac{1}{2}_j(\ell) + \frac{1}{2}_j(\ell) x \right) d\ell$$

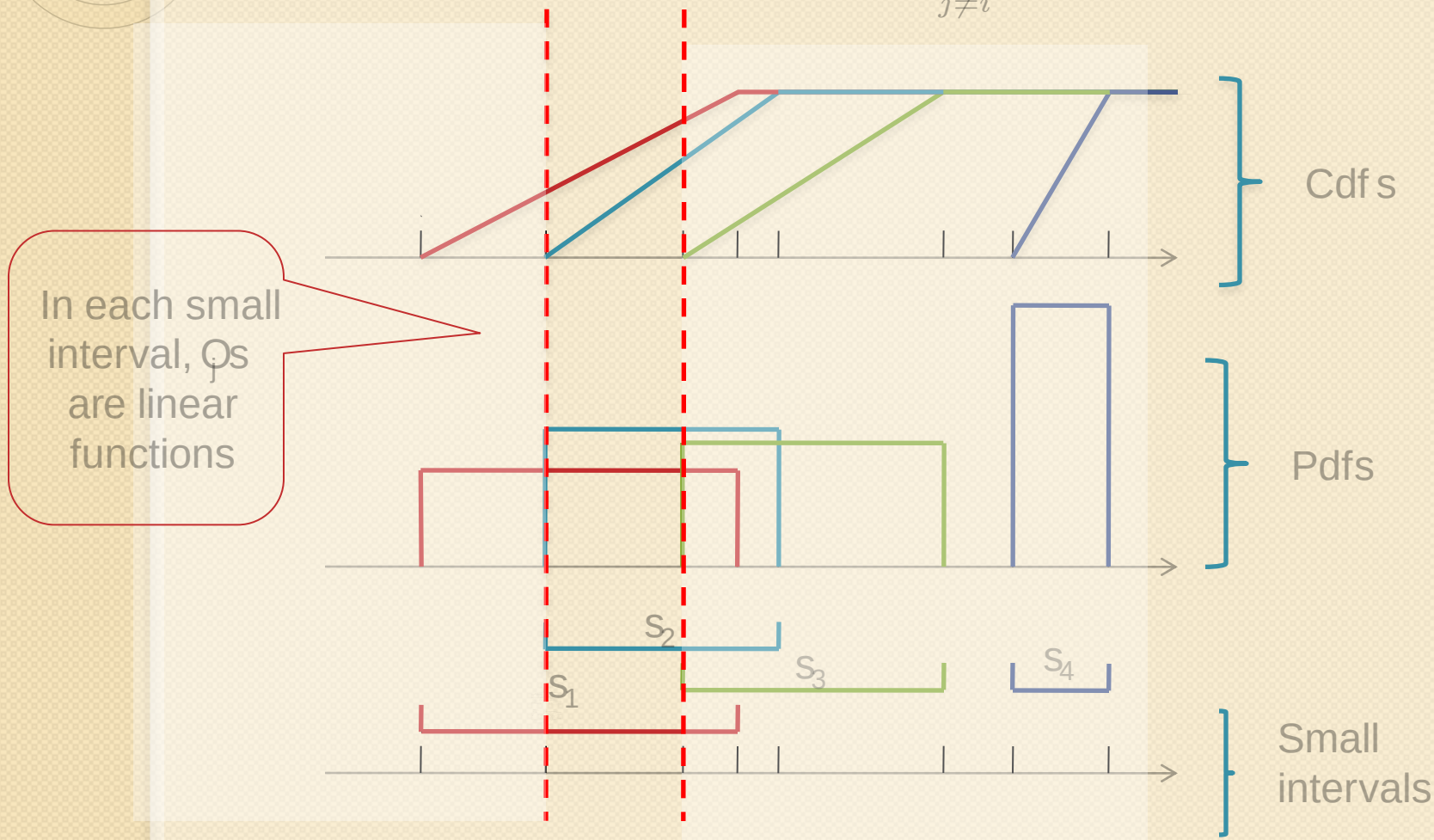
A Polynomial of x

1-dim Integral

No exp. # terms

Uniform Distribution: A Polytime Algorithm

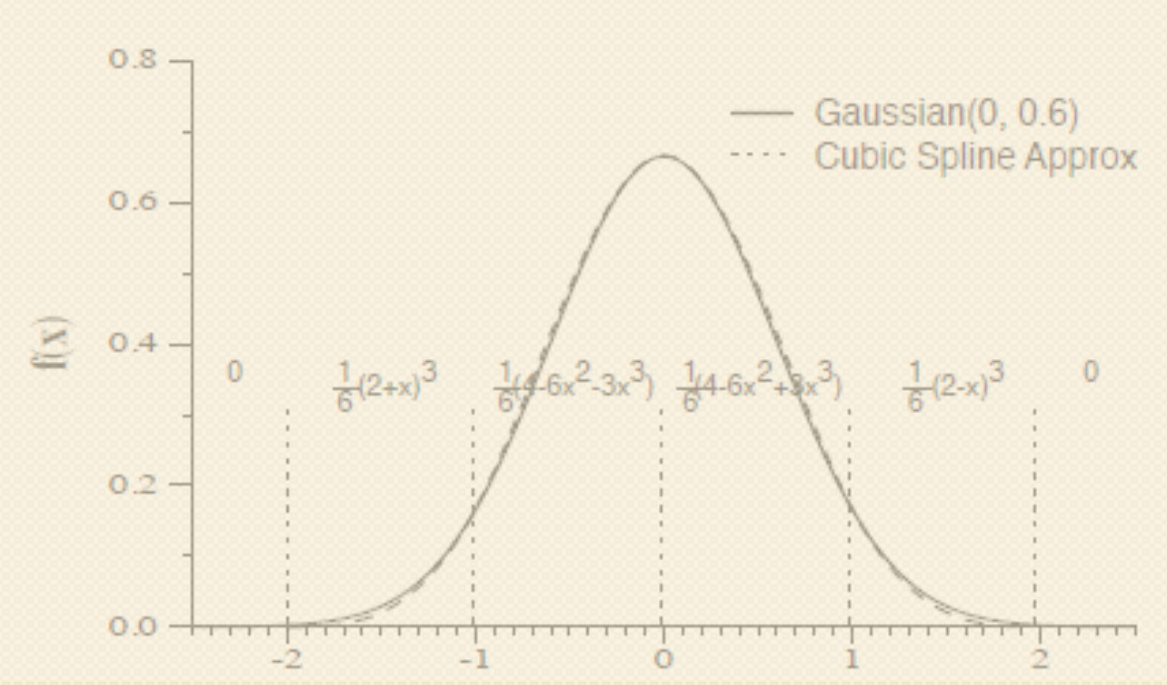
Consider the g.f $z_i(x) = x \int_{-\infty}^{\infty} \mu_i(\ell) \prod_{j \neq i} \left(\frac{1}{2} F_j(\ell) + \frac{1}{2} F_j(\ell) x \right) d\ell$



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.



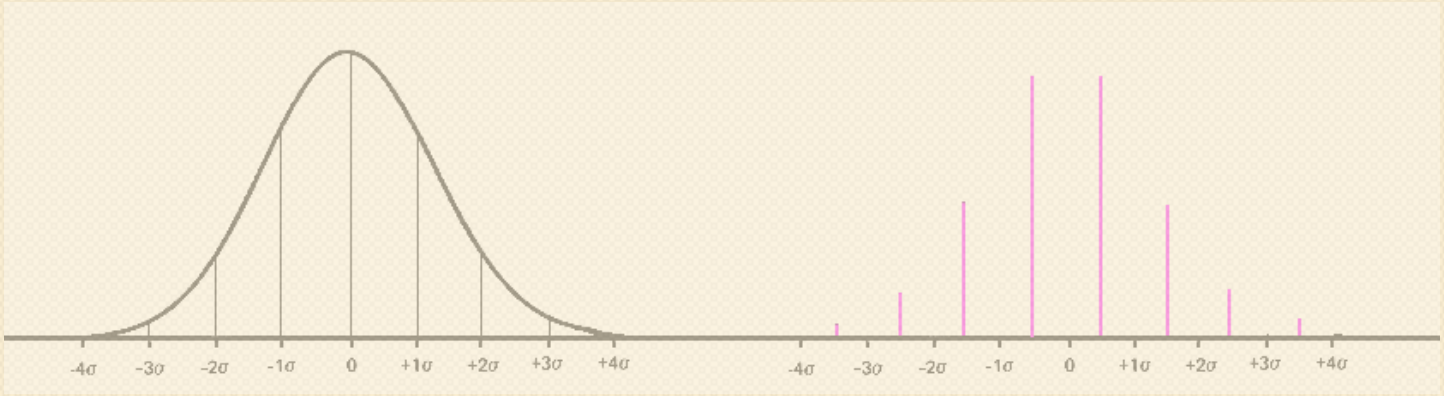
Spline(x) = f(x), Spline'(x) = *f'(x)* for all break points x.

Theoretical Convergence Resul

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 Resul

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

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

  - Under certain continuity assumptions.

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

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

  - Under certain continuity assumptions.

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

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

β=4

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

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

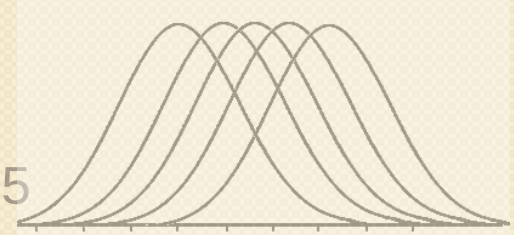
$O(n^{-2\beta})$

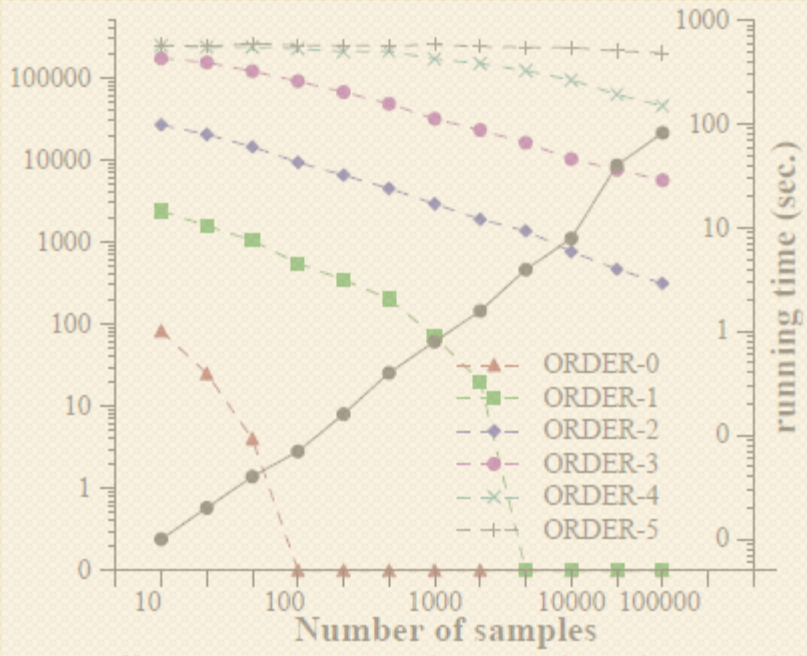
Other Results

- Efficient algorithm for PRF^1 (linear weight func.)
  - If no tuple uncertainty, PRF^1 = 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: 5 dataset ORDER(d=1,2,3,4,5)
Gaussian distributions. 10000 tuples.
Mean: $mean_i() = i * 10^{-d}$ 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 k 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

- Texpoint3.2.1