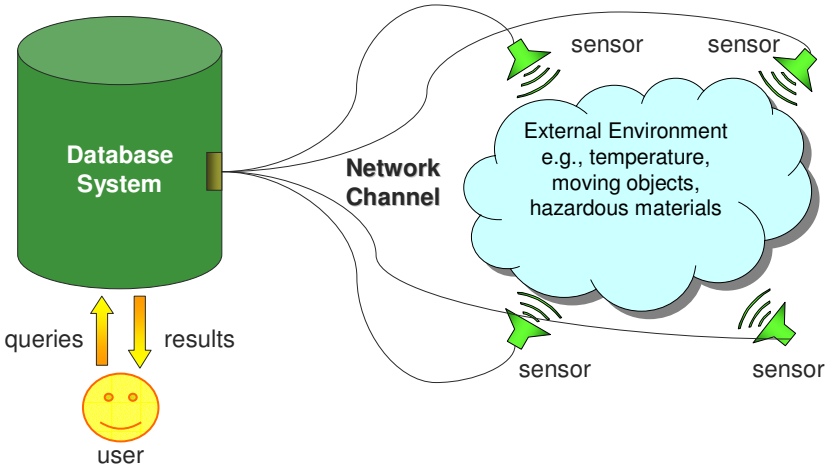
Efficient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data

Reynold Cheng, Yuni Xia, Sunil Prabhakar,
Rahul Shah and Jeffrey Scott Vitter

Department of Computer Science
Purdue University

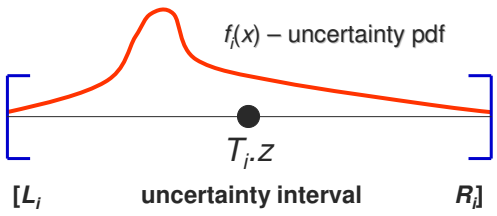
Sensor-based Applications



Data Uncertainty

- Due to limited network bandwidth and battery power, readings are sampled only
- The value of the entity being monitored (e.g., temperature, location) is changing
- Most of the time the database stores old values
- *Query results can be incorrect!*

Uncertainty Model



- T_i ($i = 1,...,n$): database object i
- $T_i.z$: dynamic attribute (e.g., temperature, locations)
- Wolfson et al. (1999) proposed $f_i(x)$ as Gaussian distribution for a moving object on a route
- Deshpande et al. (2004) discussed parametrization of Gaussian distribution in sensor networks

Probabilistic Queries

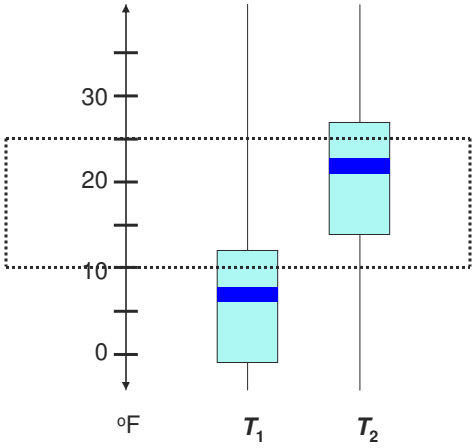
Recorded Temperature

Uncertainty for Current Temperature

- Which room's temp is between 10°F to 25°F?

$$p_1 = \int_{10}^{12} f_1(z)dz$$

- **{(T_1,10%), (T_2,80%)}**



Probabilistic Queries

- **Drawback:** Costly integration operations
- In practice, the user is only concerned with results with sufficiently high probability values
- e.g., return ids of sensors with temperature over 30°F where probability ≥ 0.7

Probability Threshold Queries (PTQ)

- **INPUT:** $[a,b]$, and p,where $a,b,p \in \Re$, $0 < p \leq 1$
- **OUTPUT:** $\{T_i\}$ where probability p_i that $T_i.z$ is inside $[a,b]$ satisfies $p_i \geq p$
- The actual value of p_i is not returned

Interval Indexing

- Interval indexing handles containment, overlap and stabbing queries
- Manolopoulos et al. (2000) proposed an efficient interval tree for range queries
- Arge & Vitter (1996) and Kanellakis et al. (1996) mapped 1D interval queries to 2D queries

Solving PTQ with Interval Indexes

1. Use interval indexes to find intervals that overlap [*a*,*b*]
2. For each object retrieved, evaluate its probability of being within [*a*,*b*]
3. Return intervals with probability $\geq p$

The Problem of Interval Indexes

- Current Interval indexes do not consider probabilities during search
- Many irrelevant objects (probability < *p*) may be retrieved from the interval index

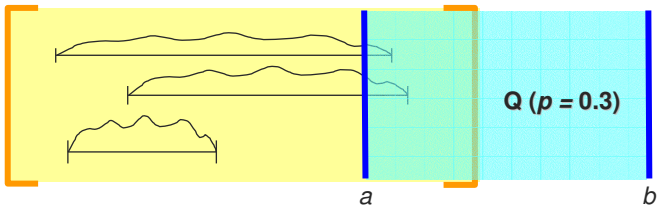
Outline

- **Probability Threshold Indexing (PTI)**
  1D interval R-tree with uncertainty
- **Variance-based Clustering**
  Transform intervals to 2D points and index based on variance
- Experimental Results

Outline

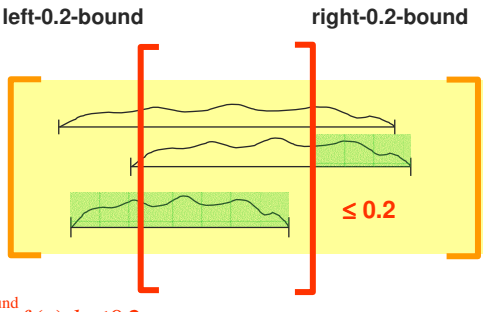
- **Probability Threshold Indexing (PTI)**
  1D interval R-tree with uncertainty
- **Variance-based Clustering**
  Transform intervals to 2D points and index based on variance
- Experimental Results

Pruning in a 1D R-Tree



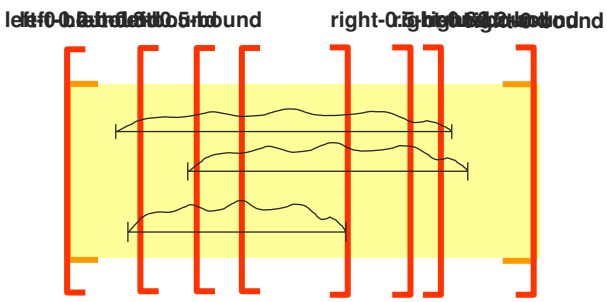
- Some intervals in the MBR may satisfy Q
- Need to retrieve the contents of MBR

x-bounds in a PTI Node

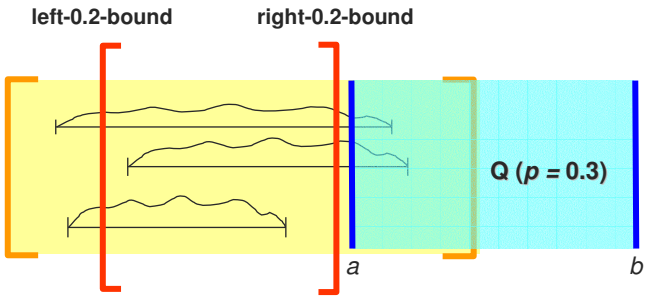


$\int_{L_i}^{\text{left-0.2-bound}} f_i(y)dy \leq 0.2$ ≥ 0.8

x-bounds in a PTI Node

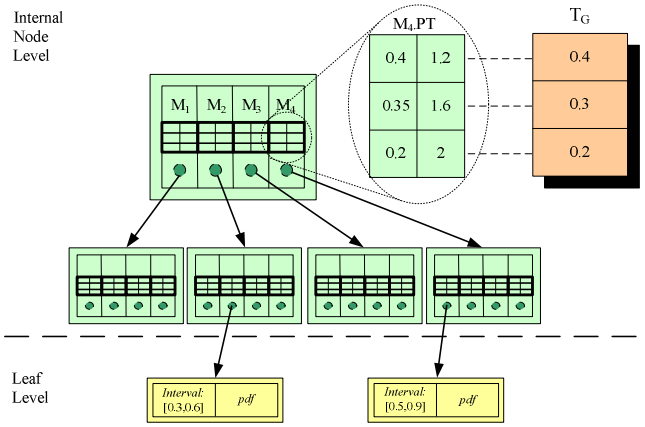


Pruning with *x*-bounds



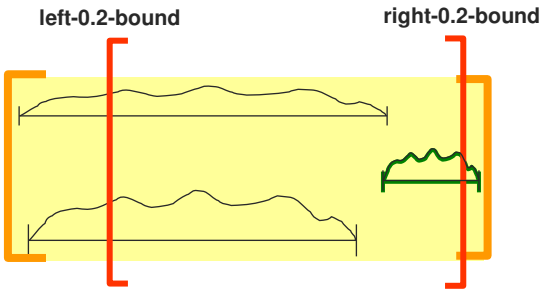
- An MBR is not further retrieved if:
  1. Q does not cut left and right *x*-bounds
  2. $p > x$

Implementation of PTI



Drawback of PTI

- Extra overhead in storing *x*-bounds
- Doesn't distinguish small and large intervals



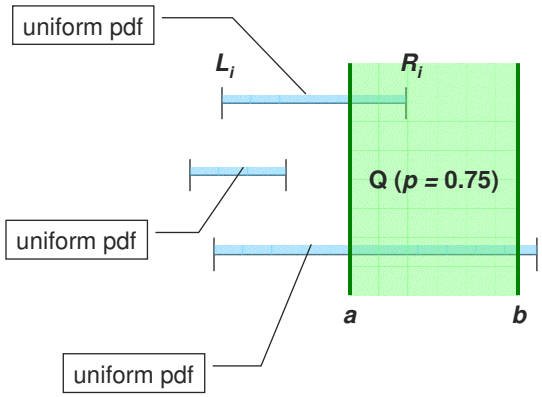
Outline

- **Probability Threshold Indexing (PTI)**
  1D interval R-tree with uncertainty
- **Variance-based Clustering**
  Transform intervals to 2D points and index based on variance
- Experiment results

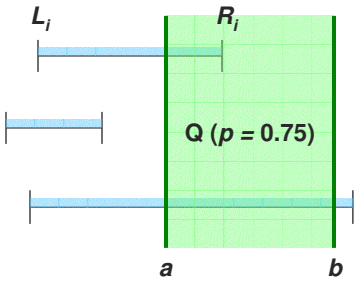
Mapping intervals to 2D-space

- Each 1D interval $[L_i, R_i]$ can be mapped to a point (x,y) in 2D space
  - $L_i \rightarrow x$
  - $R_i \rightarrow y$
- $y \geq x$: mapped points lie above $x=y$ line

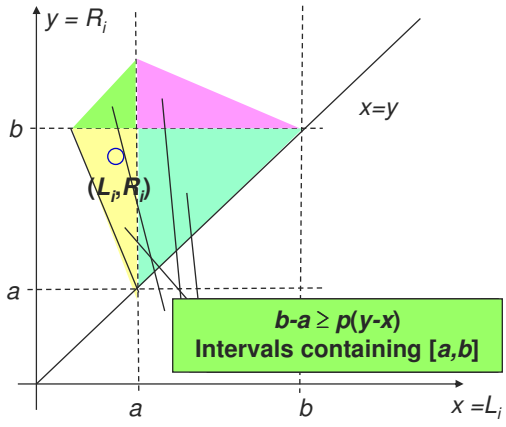
The PTQ-Uniform Problem



2D View of PTQ-Uniform



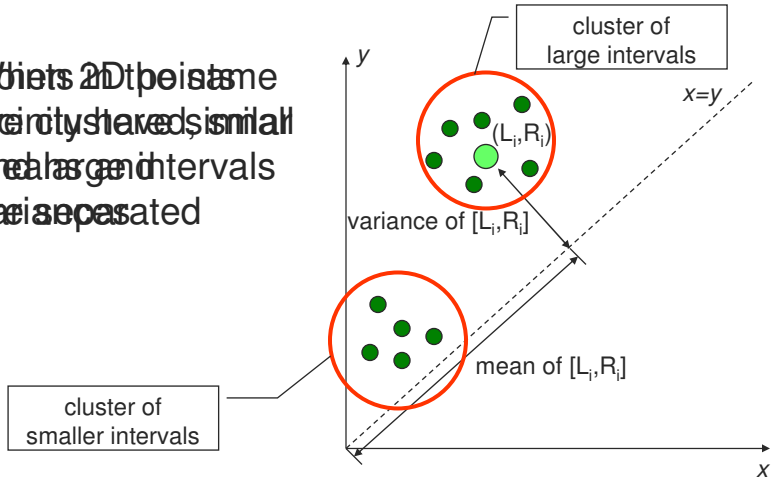
1D View
(Uniform pdf)



2D View

Clustering of 2D points

- Points in the same vicinity have similar means and variances



Answering PTQ-Uniform with 2D R-Tree

- Construct a 2D R-tree over 2D points
- Perform a trapezoidal range query over the 2D R-tree
- Since points *with similar means and variances* are clustered together, it is better than PTI

Variance-based Clustering

- Can be extended to other pdfs
- *Variance-based clustering* is an uncertainty indexing technique based on 2D R-tree
- Each item is indexed based on its *mean* and *variance*

Variance-based Clustering

- For uniform and Guassian distributions, range queries over 2D points can be constructed
- For arbitrary pdfs, a well-defined range query may be infeasible
- In those cases, place *x*-bounds in each 2D R-tree node for pruning

Theoretical Results

- Not possible to create a linear space index that gives logarithmic query times for PTQs in the worst case
- For most cases, any space-partitioning data structure e.g., 2D R-tree suffices
- PTQ with fixed threshold and uniform distribution can be answered in logarithmic time with a linear structure

Outline

- **Probability Threshold Indexing (PTI)**
  1D interval R-tree with uncertainty
- **Variance-based Clustering**
  Transform intervals to 2D points and index based on variance
- Experimental Results

Performance Comparison

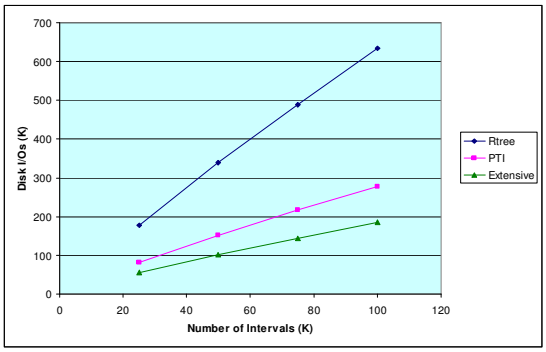
- Compare number of I/Os between
  1. 1D R-tree on intervals only
  2. PTI (1D R-tree with probability thresholds)
  3. 2D variance-based clustering (called *Extensive*)

Simulation Model

- 100K uncertain data, with length uniformly distributed in [0,10000] and uniform uncertainty pdf
- 10K PTQs with length of [a,b] normally distributed and $p \in$ [0.1,1]
- Each PTI node contains five x-bounds, where $x \in$ {0.1,0.3,0.5,0.7,0.9}

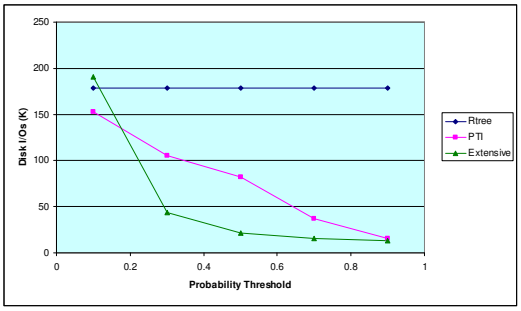
Scalability of Indexes

- Both PTI and *Extensive* outperform R-tree
- Answering PTQ with R-tree requires more computation
- *Extensive* needs about 50% less I/Os than PTI



Effect of Query Probability Threshold

- R-tree does not benefit from the increasing value of *p*
- When *p* is 0.5, *Extensive* is 4 times better than PTI



Conclusions

- Based on the pdf information of uncertain intervals, PTI places tighter bounds in 1D R-tree nodes.
- Variance-based clustering uses a 2D R-tree to avoid placing intervals of extreme sizes together.
- The concept of these indexes can be extended to multiple dimensions.

Contact Reynold Cheng (ckcheng@cs.purdue.edu) for details

Future Work

- Study probabilistic threshold constraints for other queries, such as nearest neighbors and joins
- Study the indexing of other uncertain data types e.g., fuzzy data and sets
- Study other kinds of constraints on probabilistic queries e.g., answers with top-k probability values

Related Work – Probabilistic Queries

- [CKP03] proposes an uncertainty model for constantly-evolving data. It also presents classification, evaluation and quality issues of different types of probabilistic queries.
- For moving object uncertainty,
  - [WSCY99] study probabilistic range queries.
  - [CKP04] study probabilistic nearest neighbor queries.
- [CP03] proposes computation strategies for evaluating PTQ, but does not discuss the indexing of uncertain data.

Related Work – Uncertainty Indexing

- Few works have addressed the issues of indexing uncertain data that involves probability computation.
- [CKP04] proposes an indexing scheme for constantly-growing uncertainty of moving objects.
- [LMPS03] discusses an extension of the TPR-tree to index trajectories of moving objects, where each point in the trajectory has a rectangular uncertain bound.

Related Work – Interval Indexing

- [AV96, KRVV96] discuss the idea of mapping intervals as points in 2D space. The transformation of 1D stabbing queries and range queries to two-sided orthogonal queries in 2D space are also presented.
- [MTT00] proposes an efficient interval tree to facilitate the execution of intersection queries over intervals.
- [CKP04] proposes an indexing scheme for constantly-growing uncertainty of moving objects.

References

1. **[AEM92]** Pankaj K. Agarwal, David Eppstein, and Jir Matousek. Dynamic half-space reporting, geometric optimization, and minimum spanning trees. In FOCS, pages 80-89, 1992.
2. **[AV96]** L. Arge and J. S. Vitter. On dynamic interval management in external memory (extended abstract). In FOCS, p. 560-569, 1996.
3. **[CP03]** R. Cheng and S. Prabhakar. Managing uncertainty in sensor databases. In SIGMOD Record, Dec 2003.
4. **[CKP03]** R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In Proc. of the ACM SIGMOD, 2003.
5. **[KRVV96]** P. C. Kanellakis, S. Ramaswamy, D. Vengroff, and J. S. Vitter. Indexing for data models with constraints and classes. In J. Comp. Syst. Sci, 52(3):589-612, 1996.
6. **[MTT00]** Y. Manolopoulos, Y. Theodoridis, and V. J. Tsotras. Chapter 4: Access methods for intervals. In Advanced Database Indexing, Kluwer, 2000.
7. **[WSCY99]** O. Wolfson, P. Sistla, S. Chamberlain, and Y. Yesha. Updating and querying databases that track mobile units. Distributed and Parallel Databases, 7(3), 1999.

References

- A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein and W. Hong. Model-Driven Data Acquisition in Sensor Networks. In VLDB, 2004.

Complexity of PTQU

- Half-space queries: report a set of points that satisfy $ax + by \geq c$
- PTQU is at least as hard as half-space queries which require $\Omega(n^{1/3})$ operations [F81] using a linear-space index
- Simplex queries: report a set of points that satisfy a list of constraints $a_i x + b_i y \geq c_i$
- PTQU is a special case of simplex queries, where query time is $O(n^{\varepsilon})$ using linear structure [AEM92].

Details of Variance-based Clustering

- The exact indexing technique depends on the form of pdf
- For *regular sets*, e.g., Gaussian and uniform pdf, can prune a node *without the* extra overhead of PTI
- For arbitrary pdfs, need a PTI table in each node to facilitate pruning