Range-Max Queries on Uncertain Data

Pankaj K. Agarwal† Nirman Kumar‡ Stavros Sintos§ Subhash Suri¶

Abstract

Let \( P \) be a set of \( n \) uncertain points in \( \mathbb{R}^d \), where each point \( p_i \in P \) is associated with a real value \( v_i \) and a probability \( \alpha_i \in (0, 1] \) of existence, i.e., each \( p_i \) exists with an independent probability \( \alpha_i \). We present algorithms for building an index on \( P \) so that for a \( d \)-dimensional query rectangle \( R \), the expected or the most-likely maximum value in \( R \) can be computed quickly. The specific contributions of our paper include the following: (i) The first index of sub-quadratic size to achieve a sub-linear query time in any dimension \( d \geq 1 \). In particular, for \( d = 1 \) the index answers a query in \( O(\sqrt{n}) \) time after \( O(n^{3/2} \log n) \) time and \( O(n^{3/2}) \) space preprocessing. Our algorithm also provides a continuum of trade-offs between query time and size of the index. (ii) A conditional lower bound for the most-likely range-max queries, based on the conjectured hardness of the set-intersection problem, which suggests that in the worst case the product (query time)\(^2 \times \) (index size) is \( \Omega(n^2) \), ignoring polylog factors. (iii) A linear-size index for estimating the expected range-max value within approximation factor 1/2 in \( O(\log n) \) time, for some constant \( c > 0 \); that is, if the expected maximum value is \( \mu \) then the query procedure returns a value \( \mu' \) with \( \mu / 2 \leq \mu' \leq \mu \). (iv) Extensions of our algorithm to more general uncertainty models in which the values and the locations of the points are described as probability distributions. Our index for the most likely range-max can also be extended to compute the top-\( k \) range-max values.

1 Introduction

Query-driven data management is an important function in most database systems, and range query is a common form of user query where a user is interested in aggregating information on the objects lying in the query range. A wide array of applications such as sensors networks, data cleaning, data integration, pervasive computing, and scientific data management require answering range and other forms of queries on uncertain data. In these settings, stochastic models are typically used to represent the data uncertainty, and a user is often interested in estimating the aggregate statistics describing the expected behavior of the underlying signal with high confidence. For instance, we may want to know the expected total value of all the records in a range or wish to compute the probability that the largest or the smallest item in the range is within some bound.

In this paper, we investigate range-max queries of the form “find the maximum value in a range” over multi-dimensional uncertain data. More specifically, we consider database records of the form \((p, v, \alpha)\), where \( p \) is a point in \( \mathbb{R}^d \) (attribute values of the record) along with a scalar value \( v \in \mathbb{R} \) and a probability \( \alpha \in (0, 1] \). The value \( v \) is the metric of interest in our range queries and the probability \( \alpha \) reflects our confidence in this record. Given a collection of \( n \) such records, our goal is to construct an index that can answer queries of the following form efficiently:
Given a $d$-dimensional orthogonal range $R$, report the expected maximum value in $R$ (the EM problem). Instead of the expected, we can also ask for the most likely maximum value (MLM problem), the value that occurs as the maximum with the highest probability, or other variations of this kind.

The general framework is abstract, and broadly applicable but as motivation consider the following hypothetical setting. The points might represent a set of geographical locations (cities) $\{p_1, p_2, \ldots, p_n\}$, each associated with a probability $\alpha_i$ of being struck by a natural disaster (flood, earthquake, fire) during next year, and $v_i$ a measure of the cost (damage) incurred at that location for that disaster.\footnote{More generally, each point can be associated with not just a single (value, probability) pair but an entire distribution. For the ease of exposition, we initially focus on the single value case, but remark on how to generalize our results to distributional settings later.} In this case, a range query asks for the expected value of the largest damage suffered within the range. Similarly, an insurance company may associate probabilities of financial claims with various entities, and need to analyze the profile of its maximum loss portfolio. Indeed, in natural disasters such as earthquakes or flooding, the impact is highly non-linear—even hundreds of small quakes hardly cause serious financial or social harm, but a single large one can be catastrophic. Thus, it is far more important to be able to carry out analysis on the profile of the maximum values, and not on simpler aggregates such as sum or average.

The stochastic range-max queries are also relevant when dealing with spatially distributed noisy data sources (e.g. sensors) where unusually high measurements might be cause for concern, but only if they deviate from the norm. A probabilistic profile of the expected max in a range can serve as the benchmark for deciding when a sensor measurement is abnormal. Finally, in terms of computational complexity as well, range-max queries over uncertain data appear to be more difficult than those for the average or sum, as we show through a conditional lower-bound argument in the paper. Therefore, the problem seems to require new approaches and ideas.

**Related work.** Motivated by a wide range of applications, there is extensive work on managing, querying, and analyzing uncertain data. See the book \([6]\) and the survey papers \([7][11]\) for an overview of known results on this topic. In the last few years there is also work in computational geometry, and in algorithms more broadly, on dealing with uncertain data \([5][10][16][22][23][29][32]\).

In the context of query processing, early work focused on top-$k$ queries over uncertain data. Numerous algorithms under different semantics of top-$k$ queries have been proposed; see \([9][15][19][21][25][28][30][35]\). Jayram et. al. \([20]\) describe algorithms for computing aggregation statistics on uncertain data in the streaming model. This line of work computes the desired statistics over the entire data, and it does not consider query ranges. By now, many algorithms have been proposed for answering nearest-neighbor and range queries on uncertain data \([2][4][14][31][33][34][36]\). The query time of many of these algorithms is linear in the worst case; only a few of them have better worst case running times. For example, Agarwal et. al. \([3]\) described an index for range reporting in 1D. They assumed that the location of each point is given as a piecewise-constant pdf, and they described an algorithm for reporting all points that lie inside a query interval with probability at least $\tau$. If $\tau$ is fixed the query time to report all $k$ such points is $O(n \log n + k)$ and the size of the index is $O(n)$. If $\tau$ is part of the query, then the query time is $O(\log^2 n + k)$ and the size is $O(n \log^2 n)$. Recently Li and Wang \([24]\) extended this approach to return the $k$ points that lie in the query interval with the highest probability for some special cases, (e.g. unbounded intervals) in $O(k)$ query time if $k = \Omega(\log n \log \log n)$ or $O(\log n + k \log k)$ otherwise, using an index of size $O(n)$.

Abdullah et. al. \([11]\) have considered the problem of range counting on uncertain data. They show the existence of a small-size coreset for this problem, which can be used to answer the query approximately. We note that computing the expected number of points in a query range is the same as Range-SUM problem, with weighted points and without the data uncertainty, and the same is true if we wanted to compute the expected value of $\sum_{p \in P \cap R} v_i$ in our setting. As mentioned above, computing the expected value of the SUM is a special case. We are not aware of any non-trivial algorithm for computing the expected or most likely maximum value inside a query range even for $d = 1$. Note that for $d = 1$, a quadratic-size and $O(\log n)$ query time index is trivial since there are $O(n^2)$ combinatorially distinct intervals. An interesting open
question is whether EM or MLM values can be computed in sublinear time using subquadratic storage. We note that if each point in \( P \) has existence probability \( \alpha \), then the EM value of points inside a query range can be estimated within \((1 + \epsilon)\)-factor in \( O\left( \frac{1}{\alpha \epsilon} \log n \right) \) time using an index of linear space because it suffices to report only \( \sim \frac{1}{\alpha \epsilon} \log n \) points of \( P \) that lie inside a query interval.

Our results. The main contributions of our paper can be summarized as follows.

(A) We design the first sub-quadratic size index to achieve a sub-linear query time in any dimension \( d \geq 1 \) for both EM and MLM problems. The index answers a query in \( O(n^{1-t} + \log n) \) time, has size \( O(n^{(2d-1)t+1}) \), and takes \( O(n^{(2d-1)t+1} \log n) \) time to build, for any \( t \in [0, 1] \). The tunable parameter \( t \) gives the index a continuum of trade-offs between query time and its size. In particular, for \( d = 1 \) the index can achieve a query time of \( O(\sqrt{n}) \) with \( O(n^{3/2}) \) size and \( O(n^{3/2} \log n) \) preprocessing. We also present a different index for the MLM problem whose size is \( O(n \log^{d+1} n) \) and the query time for a query rectangle \( R \) is \( O(k \log^{d+2} n) \) where \( k \) is the size of the skyline of \( \{(v_i, a_i) \mid p_i \in P \cap R\} \). In the worst case \( k = \Omega(n) \) but it is often small. For example if the values of points are chosen according to a random permutation model (cf. Sec. 2.2), it is often small. For example if the values of points are chosen according to a random permutation model (cf. Sec. 2.2), it is often small.

(B) We prove a conditional lower bound for the most-likely range-max queries, based on the conjectured hardness of the set-intersection problem. In particular, this lower bound suggests that in the worst case, for \( d \geq 2 \), the product (query time)\(^2 \times \) (index size) is \( n^2 / \text{polylog}(n) \).

(C) We design a linear-size index for estimating \( \mu(P \cap R) \), for a query rectangle \( R \), within approximation factor \( 1/2 \) in \( O(\log^{d+1} n) \) time; that is, the query procedure returns a value \( \mu' \) such that \( \frac{1}{2} \mu(P \cap R) \leq \mu' \leq \mu(P \cap R) \).

(D) We develop extensions of our algorithms to several more general models in which the values and the locations of the points are described as probability distributions. In both location and value distributions, we propose generalizations of the sub-quadratic index in (A) to compute the EM or the MLM in sublinear time and generalizations of (C) to compute the \( \mu(P \cap R) \) within an approximation factor \( 1/2 \) in \( \text{polylog}(n) \) time. In case of location distribution, we extend the index of (A) for finding the MLM where the query time depends on the size of the skyline of \( \{(v_i, a_i) \mid p_i \in P \cap R\} \).

2 Exact Algorithms

Let \( P = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^d \) be a set of \( n \) uncertain points where each \( p_i \) is associated with a value \( v_i \in \mathbb{R} \) and an independent probability \( a_i \in [0, 1] \) of its existence; multiple points may have the same value. We present indexing schemes for computing the expected or the most-likely maximum value among the points lying in a query orthogonal rectangle in \( \mathbb{R}^d \). We begin with some definitions, and then describe our index structure.

Let \( X = \{x_1, \ldots, x_m\} \), for \( m \leq n \), be the set of (distinct) values associated with the points in \( P \). (Recall that the values of the points need not be all distinct.) Given a subset of points \( R \subseteq P \), we define \( \pi(R) \) as the probability that none of its points are present. That is, \( \pi(R) = \prod_{p_i \in R} (1 - a_i) \), where we assume \( \pi(R) = 1 \) if \( R = \emptyset \). For any value \( x \in X \), let \( \alpha(x, R) \) denote the probability that some point with associated value \( x \) exists:

\[
\alpha(x, R) = 1 - \pi(\{p_i \in P \mid v_i = x\}) = 1 - \prod_{p_i \in P : v_i = x} (1 - a_i),
\]

with the convention that \( \alpha(x, R) = 0 \) if there is no point with value \( x \) in \( R \). Define \( \beta(x, R) \) as the probability that no point with value larger than \( x \) is present in \( R \):

\[
\beta(x, R) = \pi(\{p_i \in P \mid v_i > x\}).
\]

It follows that the probability of \( x \) being the maximum value in \( R \), denoted \( \gamma(x, R) \), can be written as

\[
\gamma(x, R) = \alpha(x, R) \beta(x, R).
\]
If $\mu(R)$ denotes the expected maximum (EM) value in $R$, then we have

$$\mu(R) = \sum_{x \in X} x \gamma(x, R).$$

The most likely maximum (MLM) value, denoted by $\lambda(R)$, is

$$\lambda(R) = \operatorname{argmax}_{x \in X} \gamma(x, R).$$

### 2.1 Computing EM

We describe our index structure in detail for dimension $d = 1$, and then sketch its extension to higher dimensions.

**The indexing scheme.** We fix a parameter $t \in [0, 1]$, and partition the input $P$ into $k \leq 2^{\lfloor n^{1-t} \rfloor}$ subsets $P_1, \ldots, P_k$ such that (i) all points in $P_i$ have higher values than those in $P_{i+1}$, and (ii) either $|P_i| \leq 2^n t$ or all points in $P_i$ have the same value.

We call $P_i$ uniform if all of its points have the same value, and non-uniform otherwise. If $P_i$ is uniform, then we compute and store the values $\pi(P_i \cap \langle -\infty, p \rangle)$ and $\mu(P_i \cap \langle -\infty, p \rangle)$, for all $p \in P_i$. Otherwise (for non-uniform $P_i$) we compute and store $\pi(P_i \cap [a, b])$ and $\mu(P_i \cap [a, b])$, for all pairs $a, b \in P_i$ with $a < b$.

We also sort each $P_i$ and store the sorted lists $P_1, \ldots, P_k$ so that for any $q \in \mathbb{R}$ its predecessor and successor in each $P_i$, denoted by $\operatorname{pred}(q, P_i)$ and $\operatorname{succ}(q, P_i)$ respectively, can be computed efficiently. We use the so-called fractional-cascading scheme to expedite this process, where an augmented sequence $P^*_i$ corresponding to each $P_i$ is stored. More precisely, starting with the bottom-most set $P^*_k = P_k$, we construct the remaining augmented lists as follows. Suppose $P^*_i = \langle a_1, a_2, a_3, \ldots, a_n \rangle$, and let $\frac{1}{2}P^*_i = \langle a_2, a_4, a_6, \ldots \rangle$ be the subsequence of $P^*_i$ consisting of every other item. Then, we define $P^*_{i-1} = P_{i-1} \cup \frac{1}{2}P^*_i$, and with each item $a \in P^*_i$, we store the indices of $\operatorname{pred}(a, P_i)$, $\operatorname{succ}(a, P_i)$ and of $\operatorname{succ}(a, P^*_i)$ in $P^*_i + 1$. The total size of all the augmented lists is $\sum_{i=1}^k |P^*_i| \leq 2n$. With these augmented lists, we can find $\operatorname{pred}(q, P_i)$ and $\operatorname{succ}(q, P_i)$ for any $q \in \mathbb{R}$ in all the set $P_i$, $i = 1, 2, \ldots, k$, in total time $O(\log n + k)$, improving the obvious bound of $O(k \log n)$. See [8] for details. This completes the description of the index.

The index needs $O(|P|^2) = O(n^2)$ space when $P_i$ is non-uniform, and $O(|P|)$ space otherwise. Since $\sum_{i=1}^k |P_i| = n$, the total space for the index is $O(n) + O(n^{1-t}n^2) = O(n^{1+t})$. We now discuss how to answer a query using this index, and then discuss its construction details.

**Query procedure.** Given a query interval $I = [a, b]$, let $a_j = \operatorname{succ}(a, P_j)$ and $b_j = \operatorname{pred}(a, P_j)$, for $j = 1, \ldots, k$. Define $l_j = [a_j, b_j]$ as the projection of the query range onto the subset $P_j$, where we assume $l_j = \emptyset$ if $a_j > b_j$.

Define $l_j = [a_j, b_j]$ as the projection of the query range onto the subset $P_j$, where we assume $l_j = \emptyset$ if $a_j > b_j$, see Figure 1 for an example. We use the shorthand notation $P_{<j} = \bigcup_{i=0}^{j-1} P_i$ to denote the union of all subsets with index smaller than $j$, with the convention that $P_{<1} = \emptyset$. The following lemma explains our main idea for computing the expected max value $\mu(P \cap I)$.

**Lemma 2.1** The partition $P_1, \ldots, P_k$ satisfies the following properties:

(i) $\mu(I \cap P) = \sum_{j=1}^k \mu(I \cap P_{<j}) \mu(I \cap P_j)$.

(ii) For $j \geq 1$, $\mu(I \cap P_{<j}) = \mu(I \cap N_{<j}) \mu(I \cap P_{j-1})$.

**Proof:** To prove (i), let $X_j \subseteq X$ be the set of distinct values among the points of $P_j$. Then the following holds:

$$\mu(I \cap P) = \sum_{x \in X} x \alpha(x, I \cap P) \beta(x, I \cap P) = \sum_{j=1}^k \sum_{x \in X_j} x \alpha(x, I \cap P) \beta(x, I \cap P).$$

The values in $X_{j-1}$ are all larger than those in $X_j$, and $x$ belongs to at most one such $X_j$, therefore $\alpha(x, I \cap P) = \alpha(x, I \cap P_j)$, and $\beta(x, I \cap P) = \pi(I \cap P_{<j}) \mu(I \cap \{p_i \in P_j \mid v_i > x\}) = \mu(I \cap P_{<j}) \beta(x, I \cap P_j)$. We also
have \( a(x, I \cap P_j) = a(x, I_j \cap P_j) \) and \( \beta(x, I \cap P_j) = \beta(x, I_j \cap P_j) \), which follows because \( P_j \cap (I \setminus I_j) = \emptyset \), by definition. Thus, we can conclude that

\[
\mu(1 \cap P) = \sum_{j=1}^{k} \sum_{x \in X_j} x \pi(1 \cap P_{<j}) a(x, I_j \cap P_j) \beta(x, I_j \cap P_j) = \sum_{j=1}^{k} \pi(I \cap P_{<j}) \mu(x, I_j \cap P_j).
\]

The proof of \((ii)\) follows from the fact that \( P_{<j} = P_{<j-1} \cup P_{j-1} \) and \( P_{<j-1} \cap P_{j-1} = \emptyset \).

If \( P_j \) is non-uniform, then the index stores the values of \( \mu(I \cap P_j) \) and \( \pi(I_j \cap P_j) \). For a uniform \( P_j \), we observe that \( \pi(I_j \cap P_j) = \pi(P_j \cap (-\infty, b_j]) / \pi(P_j \cap (-\infty, a_j)) \) and \( \mu(I_j \cap P_j) = x(1 - \pi(I_j \cap P_j)) \). Since we store \( \pi(P_j \cap (-\infty, p]) \) for all \( p \in P_j \), the values \( \pi(I_j \cap P_j) \) and \( \mu(I_j \cap P_j) \) can be computed in \( O(1) \) time. The pseudo-code in Algorithm 1 below describes the query procedure and Figure 1 shows the intervals in each group given the query interval \( I \).

Algorithm 1: EXACT_EM_QUERY

Input: \( I = [a, b] \)
Output: \( \mu(1 \cap P) \)
1: \( \text{for} \ j = 1 \text{ to } k \text{ do} \)
2: \( a_j = \text{succ}(a, P_j), \ b_j = \text{pred}(b, P_j) \)
3: \( \text{end for} \)
4: \( EM = 0, \ p = 1 \)
5: \( \text{for} \ j = 1 \text{ to } k \text{ do} \)
6: \( EM = EM + p \cdot \mu(I_j \cap P_j) \)
7: \( p = p \cdot \pi(I_j \cap P_j) \)
8: \( \text{end for} \)
9: return \( EM \)

To bound the query time, we note that \( a_j \) and \( b_j \) can be found for all \( j = 1, 2, \ldots, k \), in \( O(\log n + k) \) time using fractional-cascading. We spend \( O(1) \) time in each iteration of the second for loop, so that the total query time is \( O(\log n + k) = O(\log n + n^{1-\epsilon}) \).

Preprocessing. We now describe how to build the index structure. Given a subset \( S \subseteq P \), let \( S[x] = \{p_i \in S \mid \nu_i = x\} \) be the set of points in \( S \) that have value \( x \). We sort the set of distinct values \( X \) in the descending order and construct the partition of \( P \) by scanning the elements of \( X \) in sorted order. Suppose the sets \( P_1, \ldots, P_{j-1} \) have been constructed, we are building the set \( P_j \), and \( x_j \in X \) is the next value in the sorted sequence. If either \( |P_j| \geq n^\epsilon \) or \( |P[x_j]| \geq n^\epsilon \), we close \( P_j \) and start the new subset \( P_{j+1} \) by adding \( P[x_j] \) to it. Otherwise we have both \( |P_j| < n^\epsilon \) and \( |P[x_j]| < n^\epsilon \), in which case we add \( P[x_j] \) to \( P_j \), and continue to the next value of \( X \). The total time for constructing \( P_1, \ldots, P_k \) is clearly \( O(n \log n) \). Finally, the number of sets \( k \) is at most \( 2n^{1-\epsilon} \) because if \( |P_j| < n^\epsilon \), then \( |P_{j+1}| > n^\epsilon \). The construction of the fractional-cascaded sequences to expedite searching in these sets is straightforward, and takes \( O(n) \) time, as shown in [5].

The preprocessing cost of the uniform sets is only linear because computing \( \pi(P_j \cap (-\infty, a]) \), for all \( a \in P_j \), takes \( O(|P_j|) \) time if \( P_j \) is uniform. The main part of the construction is the preprocessing of the non-uniform sets, which we describe in detail below.

Suppose \( X_j = \{x_1, x_2, \ldots, x_s\} \) is the sequence of distinct values for the points of \( P_j \). Let \( T \) be a complete binary tree whose leaves store \( x_j \)'s in the sorted order and let \( X_u \subseteq X_j \) be the subset of values stored at the leaves of the subtree rooted at a node \( u \in T \). We will dynamically maintain a set \( S \subseteq P \) in \( T \) under insertion of points, and each node \( u \in T \) will store \( \pi_u = \pi(S_u) \) and \( \mu_u = \mu(S_u) \), where \( S_u = \{p_i \in S \mid \nu_i \in X_u\} \) is the set of points associated with values in \( X_u \). More specifically, if \( u \) is a leaf, with \( X_u = \{x\} \), then
We only need to compute the \(P\) values. O2.2 Computing MLM

MLM and another, an alternative skyline based approach. For any t

Theorem 2.4

Let \(P\) be a set of n uncertain points in \(\mathbb{R}^1\), where each point is associated with a real value and a probability of its existence. For any \(t \in [0,1]\), an index of size \(O(n^{1+t})\) can be constructed in \(O(n^{1+t} \log n)\) time so that the expected maximum value of \(P \cap I\) for a query interval I can be computed in \(O(n^{1+t} + \log n)\) time.

The parameter \(t\) can be chosen to strike desirable tradeoffs between query time and space complexity. For instance, by choosing \(t = 1/2\), we get the following result.

Corollary 2.3 Let \(P\) be a set of n uncertain points in \(\mathbb{R}^1\), where each point is associated with a real value and a probability of its existence. An index of size \(O(n^{3/2})\) can be constructed in \(O(n^{3/2} \log n)\) time so that the expected maximum value of \(P \cap I\) for a query interval I can be computed in \(O(n^{1/2} + \log n)\) time.

The 1-dimensional indexing algorithm can be extended to higher dimensions in a straightforward manner. We only need to compute the \(\mu(P_j \cap R)\) and \(\pi(P_j \cap R)\) values for \(O(n^{2d})\) “combinatorially distinct” rectangles \(R\). Omitting the details, we summarize the result.

Theorem 2.4 Let \(P\) be a set of n uncertain points in \(\mathbb{R}^d\), for some constant \(d \geq 1\), where each point is associated with a real value and a probability of its existence. For any \(t \in [0,1]\), an index of size \(O(n^{(2d-1)t+1})\) can be constructed in \(O(n^{(2d-1)t+1} \log n)\) time so that the expected maximum value of \(P \cap R\) for a query rectangle \(R\) can be computed in \(O(n^{1-t} + \log n)\) time.

2.2 Computing MLM

We now discuss our indexing schemes for computing the most likely maximum value of \(P\) in a query interval \(I\), namely, \(\lambda(P \cap I)\). We describe two approaches: one, an adaption of the preceding partition-based index, and another, an alternative skyline based approach.

MLM via the modified EM index. The index of the previous section can be adapted easily to compute \(\lambda(P \cap I)\), using the following analog of Lemma 2.1

Lemma 2.5 Given a query interval \(I = [a,b]\), let \(a_j = \text{succ}(a, P_j)\), \(b_j = \text{pred}(b, P_j)\) and \(I_j = [a_j, b_j]\). Suppose \(x_j^* = \lambda(P_j \cap I_j)\) is the most likely range-max value for the subset \(P_j\), \(j = 1, 2, \ldots, k\). Set \(X^* = \{x_1^*, \ldots, x_k^*\}\). Then the following holds:

\[
\lambda(P \cap I) = \arg \max_{x_j^* \in X^*} \pi(I \cap P_{<j}) \gamma(x_j^*, P_j \cap I_j)
\]
Therefore, for each non-uniform set \( P_i \) and for each pair \( p_u, p_v \in P_i \) with \( u < v \), we compute the value \( x_{uv} = \lambda(P_i \cap |p_u, p_v|) \), \( x_{uv} \), and \( \pi(P_i \cap |p_u, p_v|) \). The query algorithm is a simple adaptation of Algorithm 1 and takes time \( O(n^{1-t} + \log n) \). The index has size \( O(n^{1-t}) \), and it can be built in \( O(n^{1-t} + \log n) \) time. In higher dimensions \( d \geq 1 \), the query time, index size, and the preprocessing time are \( O(n^{1-t} + \log n), O(n (2d-1)t+1) \), and \( O(n (2d-1)t+1 \log n) \), respectively.

**Theorem 2.6** Let \( P \) be a set of \( n \) uncertain points in \( \mathbb{R}^d \), for some constant \( d \geq 1 \), where each point is associated with a real value and a probability of its existence. For any \( t \in [0, 1] \), an index of size \( O(n (2d-1)t+1 \log n) \) can be constructed in \( O(n^{(2d-1)t+1} \log n) \) time so that the most likely maximum value of \( P \cap R \) for a query rectangle \( R \) can be computed in \( O(n^{1-t} + \log n) \) time.

**A skyline based index.** For the MLM problem we can design another index using the idea of skylines. We assume that all points \( p_i \) have distinct values \( v_i \)—the scheme works with duplicate values also but requires extra space (similar to location uncertain model, Section 5). We map each point \( p_i \in P \) to a two-dimensional point \( \hat{p}_i = (a_i, v_i) \in \mathbb{R}^2 \). Recall that a point \( a = (a_1, a_2) \) is said to dominate another point \( b = (b_1, b_2) \) if \( a_1 > b_1 \) and \( b_2 > b_2 \) or \( a_1 \geq b_2 \) and \( b_1 > b_2 \). The **skyline** of a point set \( S \subseteq \mathbb{R}^2 \) is the subset of points in \( S \) that are not dominated by another point of \( S \) (see Figure 2). The following simple lemma is the basis of our index.

**Lemma 2.7** Let \( R \subseteq P \). If \( x_k = \lambda(P \cap R) \), then \( \hat{p}_k \) belongs to the skyline of \( \hat{R} = \{ \hat{p}_i \mid p_i \in R \} \).

**Proof:** If \( \hat{p}_k \) is not on the skyline of \( \hat{R} \), then it is dominated by some other point \( \hat{p}_j \in \hat{R} \setminus \{ \hat{p}_k \} \). Thus, we have \( a_j \geq a_k \) and \( v_j \geq v_k \). Since the values are distinct, the second inequality must be strict, namely, \( v_j > v_k \). It is easy to check that \( \beta(v_k, R) \leq (1 - a_j) \beta(v_j, R) \). Therefore, we have the following inequality:

\[
\gamma(v_k, R) = a(v_k, R) \beta(v_k, R) \leq a_k(1 - a_j) \beta(v_j, R) < a_j \beta(v_j, R) = \gamma(v_j, R)
\]

which contradicts the assumption that \( \gamma(v_k, R) = \arg\max_{x \in X} \gamma(x, R) \). This completes the proof.

Using Lemma 2.7 \( \lambda(P \cap R) \) for a query rectangle can be computed as follows. Let \( \hat{P}_R = \{ \hat{p} \mid p \in P \cap R \} \), let \( S_R \subseteq P \cap R \) be the set of points \( p_i \) such that \( \hat{p}_i \) belongs to the skyline of \( \hat{P}_R \). For each \( p_i \in S_R \), we compute \( \gamma(v_i, P) \). The MLM value will be the value of the point with the maximum \( \gamma(v_i, P) \), i.e., \( \arg\max_{p_i \in S_R} \gamma(v_i, P) \).

Once we have \( S_R \), \( \gamma(v_i, P) \) for all \( p_i \in S_R \) can be computed in \( O(|S_R| \log^d n) \) time using a \((d + 1)\)-level range tree of \( O(n \log^d n) \) space. For \( d > 1 \), Saladi and Janardan [27] have presented an index for storing \( P \) and \( \hat{P} \) of size \( O(n \log^{d+1} n) \), that can be constructed in time \( O(n \log^{d+1} n) \) so that for any query rectangle \( R \), \( S_R \) can be computed in time \( O(|S_R| \log^{d+2} n) \). For \( d = 1 \) the index is of size \( O(n) \), can be built in \( O(n \log^2 n) \) time, and the query time is \( O(|S_R| + \log n) \).

**Theorem 2.8** Let \( P \) be a set of \( n \) uncertain points in \( \mathbb{R}^d \), for a constant \( d \geq 1 \), where each point of \( P \) is associated with a real value and a probability of its existence. Assuming all points have distinct values, an index of size \( O(n \log^{d+1} n) \) can be built in \( O(n \log^{d+1} n) \) time so that for a query rectangle \( R \), the most likely range-max value \( \lambda(P \cap R) \) can be computed in time \( O(k \log^{d+2} n) \), where \( k = |S_R| \). For \( d = 1 \), the size and query time can be improved to \( O(n) \) and \( O(k \log n) \), respectively.

In the worst-case, of course, we can have \( k = \Omega(n) \), but typically the size of the skyline is quite small. In fact, under some mild assumptions, one can even prove that the size of the skyline is polylogarithmic. We discuss one such setting namely, the random permutation model. Suppose the probabilities \( a_i \)'s of points are
chosen by an adversary but their values $v_i$‘s are assigned randomly as follows: Let $x_1 \geq x_2 \geq \ldots \geq x_n$ be a set of values chosen by an adversary. We choose a random permutation $\sigma$ of $[1 : n]$ and set $v_i = x_{\sigma(i)}$. Under this model, the size of the skyline of a subset $S \subseteq P$ is $k = O(\log n)$ with high probability [18]. We, therefore, have the following result:

Theorem 2.9 Let $P$ be a set of $n$ uncertain points in $\mathbb{R}^d$ where $d \geq 1$, where each point is associated with an existential probability and a real value, where the values of $P$ are all distinct and chosen according to a random permutation model. An index of size $O(n \log^{d+1} n)$ on $P$ can be built in $O(n \log^{d+1} n)$ time so that for a query rectangle $R$, the most likely range-max value $\lambda(P \cap R)$ can be computed in time $O(\log^{d+3} n)$, with high probability. For $d = 1$, the size and query time can be improved to $O(n)$ and $O(\log^2 n)$, respectively.

3 A Hardness Result for the MLM Problem

The range-maximum query problem for uncertain data seems significantly harder than its deterministic counterpart. Indeed, the latter can be solved easily in $O(\log^{d-1} n)$ query time and $O(n \log^{d-1} n)$ space using orthogonal range trees with fractional cascading, for $d \geq 2$ [13]. This stands in sharp contrast to the best we are able to achieve for the uncertain data model, namely, $O(n^{1-1/2d})$ query time and $O(n^{2-1/2d})$ space. In this section, we offer theoretical evidence for the hardness of uncertain range-maximum problem through a reduction from the set-intersection problem, which suggests that indexes with near-linear space and poly-logarithmic query time are unlikely to exist even for $d = 2$.

The set-intersection problem is defined as follows. Given a family of sets $S_1, S_2, \ldots, S_m$, with real-valued items, preprocess them so that the intersection queries of the following form can be answered efficiently: given indices $i, j$, do the sets $S_i$ and $S_j$ have a non-empty intersection? Suppose an index can answer each query in worst-case time $O(t)$. Then, it is widely believed that, ignoring polylog factors, the index must use $\Omega((n/t)^2)$ space [12,26,27], where $n = \sum |S_i|$ is the total size of all the sets. We show in the following that any index for solving the MLM problem in dimension 2 (or higher) can also solve the set-intersection problem, and is therefore subject to the same query-space tradeoff.

The reduction transforms an instance of the set-intersection problem of size $n$ into an instance of the MLM problem with $2n$ points in 2 dimensions. Specifically, let $S_1, S_2, \ldots, S_m$ be the input to the set-intersection problem, where $\sum_{i=1}^m |S_i| = n$. Let $n_0 = 0$ and $n_i = n_{i-1} + |S_i|$ for $i = 1, \ldots, m$. We use $s_{ik}$ to denote the (value of) the $k$th item of $S_i$, where $1 \leq k \leq |S_i|$. We create a 2-dimensional instance of the MLM problem corresponding to these sets as follows. All the points in the MLM lie on two parallel lines, $L : y = x + n$ and $L' : y = x - n$. For each member of the set $S_i$, we create two points, one on each of these lines. Specifically, the points corresponding to the $k$th item $s_{ik} \in S_i$ are the following: $(-(k + n_{i-1}), -(k + n_{i-1} + n))$, which lies on $L$, and $((k + n_{i-1}), (k + n_{i-1} - n))$, which lies on $L'$. The “probability-value” tuple associated with each of these points is $(a, s_{ik})$, where $a$ will be fixed later and $s_{ik}$ is the value of the $k$th item in $S_i$. Let $P_i$ and $P'_i$ denote the set of points corresponding to $S_i$ that lie on $L$ and $L'$, resp. Finally, let $P = \bigcup_i (P_i \cup P'_i)$ denote the set of all the $2n$ points that form the input to the MLM problem.

Clearly the construction takes $O(n)$ time, and one can easily verify that (1) all points on $L$ lie in the northwest quadrant while those on $L'$ lie in the southeast quadrant, and (2) the points corresponding to $S_i$ are placed in consecutive order on both $L$ and $L'$, and they lie between the points corresponding to $S_{i-1}$ and $S_{i+1}$. 

\[8\]
The placement of the points in opposite quadrants ensures the following geometric property: given any two indices \( i, j \), there is a rectangular range \( R_{ij} \) that includes only \( P_i \) and \( P'_j \), namely, \( R_{ij} \cap P = P_i \cup P'_j \). See Figure 3.

Given a set-intersection query: “Does \( S_i \) intersect \( S_j \)?” we compute the MLM for the query range \( R_{ij} \). If the answer is value \( v \), then we check if \( v \) belongs to both \( S_i \) and \( S_j \). If it does, then we return “Yes,” meaning that the sets intersect; otherwise, we answer “No.”

We answer a set-intersection query using one MLM range query and two set membership queries, to decide if \( v \in S_i \cap S_j \). The membership queries are easily performed in \( O(\log n) \) time, using any standard balanced search tree, and so the set-intersection query time is dominated by the MLM query time, as long as the latter is \( \Omega(\log n) \). The correctness follows from the following lemma.

**Lemma 3.1** There exists a choice of \( \alpha \) such that the probability of the MLM is strictly greater than \( \alpha \) if and only if \( S_i \cap S_j \neq \emptyset \), and the item realizing this probability is in the intersection \( S_i \cap S_j \).

*Proof*: We show that the probability of the MLM is at most \( \alpha \) if the sets \( S_i \) and \( S_j \) are disjoint, and strictly larger than \( \alpha \) otherwise, which proves the claim. First consider the case when the sets are disjoint. In this case, the probability that a particular item, say, the \( k \)-th smallest element of \( S_i \cup S_j \) is the MLM, is \( \alpha(1-\alpha)^{|S_i|+|S_j|-k} \): the first term accounts for the probability that \( v_k \) is present and the second for the probability that none of the larger items in the range are present. Clearly, \( \alpha(1-\alpha)^{|S_i|+|S_j|-k} \leq \alpha \), and therefore in this case the MLM has probability at most \( \alpha \).

On the other hand, if \( S_i \cap S_j \neq \emptyset \), then let \( v_k \in S_i \cap S_j \) be any item common to the two sets. The probability that \( v_k \) is the MLM is \( (2\alpha - \alpha^2)(1-\alpha)^r \), where \( r \) is the number of values larger than \( v_k \) in \( S_i \cup S_j \): the first term accounts for the probability that at least one copy of \( v_k \), either in \( P_i \) or in \( P'_j \), is present, and the second term ensures that no larger value in the range is present. Clearly, \( r < n \), and therefore this probability is strictly larger than \( (2\alpha - \alpha^2)(1-\alpha)^n \). We can choose any value of \( \alpha \) for which \( (2\alpha - \alpha^2)(1-\alpha)^n > \alpha \) holds, to satisfy the claim, and in particular, the following loose bound suffices \( \alpha = 1 - 2^{-\frac{1}{\eta}} \).

We have established the following result,

**Theorem 3.2** If there is an index to solve the MLM problem for \( n \) points in \( \mathbb{R}^2 \) with \( S(n) \) space and \( Q(n) \) query time, for \( Q(n) \geq \Omega(\log n) \), then we can solve the set-intersection problem in \( Q(2n) \) time with space \( S(2n) \).

Since the set-intersection problem is believed to require space \( \Omega((n/t)^2) \) for query time \( t = \Omega(\log n) \), where we ignore polylog factors, Theorem 3.2 rules out the possibility of a near-linear size index and polylog query time. In fact, the theorem suggests that in the worst case, ignoring polylog factors, the product (query time)\(^2 \times \) (index size) is \( \Omega(n^2) \), for \( d \geq 2 \).

## 4 Approximation Algorithm

In this section we describe an \( O(n \text{ polylog}(n)) \)-size index, that for a query rectangle \( R \), quickly computes a value \( \eta_R \) such that \( \frac{1}{2}\mu(P \cap R) \leq \eta_R \leq \mu(P \cap R) \). The index is based on the following lemma.

**Lemma 4.1** Let \( Y_1, \ldots, Y_m \) be Bernoulli random variables such that \( Y_i \) assumes value \( v_i \) with probability \( \alpha_i \) and 0 otherwise. Then, \( E \left[ \max_i Y_i \right] \geq w \geq \frac{1}{2} E \left[ \max_i Y_i \right] \), where \( w \) is a solution of the equation \( w = \sum_j E \left[ \max \{ Y_j - w, 0 \} \right] \).
A proof of this lemma is given in Appendix [A]. The second part of this inequality is the well-known Prophet’s inequality [29], but our index requires an upper bound on the returned value as well, which is new. We now discuss how to use this lemma to devise an index for the range-max queries.

For a point \( p_j \in P \), let \( Y_j \) be a random variable that has value \( v_j \) with probability \( a_j \) and 0 otherwise. For a subset \( R \subseteq P \) and a real-valued parameter \( t > 0 \), we define

\[
F_R(t) = \sum_{p_j \in R} E \left[ \max \{ Y_j - t, 0 \} \right].
\]

(3)

Let \( X = \langle x_1, \ldots, x_m \rangle \) be the sequence of distinct values of points of \( P \) in decreasing order. The proof of the following lemma is straightforward.

**Lemma 4.2** For any subset \( R \subseteq P \),

(i) \( F_R(t) \) is a monotonically decreasing, piecewise-linear continuous function.

(ii) For any \( 1 \leq i \leq m \), \( F_R(t) \) is a linear function in the interval \( [x_i, x_{i-1}] \) with the following form:

\[
F_{R,i}(t) = \sum_{p_j \in R : v_j \geq x_{i-1}} a_j(v_j - t).
\]

(4)

The following corollary is an easy consequence of Lemma 4.2(i).

**Corollary 4.3** There is a unique value \( t \), denoted by \( \tau(R) \), for which \( F_R(t) = t \).

Given a subset \( R \subseteq P \), \( \tau(R) \) can be computed in two stages. The first stage computes the index \( i := i(R) \) such that \( \tau(R) \in [x_i(R), x_{i(R)}] \) by performing a binary search on \( X \). The second stage computes \( \tau(R) \) by solving the linear equation \( F_{R,i}(t) - t = 0 \) (cf. [4]).

**The index.** We now describe our index structure for estimating the value of \( \mu(P \cap R) \) within factor 1/2. Define \( P = \{ \bar{p}_i \mid 1 \leq i \leq n \} \), where \( \bar{p}_i = (p_i, v_i) \in \mathbb{R}^{d+1} \), for \( p_i \in P \). We build an index on \( P \) so that for any query \( (d + 1) \)-dimensional rectangle \( \rho \), quantities \( b_{\rho} = \sum_{p_i \in \rho} a_i v_i \) and \( a_{\rho} = \sum_{p_i \in \rho} a_i \) can be computed quickly, using an instance of a \( (d + 1) \)-dimensional range query. In particular, an index of size \( O(n \log^d n) \) can be built in time \( O(n \log^d n) \) so that \( a_{\rho}, b_{\rho} \) can be computed in \( O(\log^d n) \) time [13].

**Query procedure.** Let \( R \) be a query rectangle in \( \mathbb{R}^d \), and we wish to estimate \( \mu(P \cap R) \). We perform a binary search on \( X \) to compute \( i := i(P \cap R) \) such that \( \tau(P \cap R) \in [x_i, x_{i-1}) \). Each step of the binary search choose a value \( x_r \in X \) and queries the index with the rectangle \( \rho_r = \rho \times [x_r, \infty) \), which returns

\[
a_{\rho_r} = \sum_{p_i \in \rho} a_i = \sum_{p_i \in \rho : v_i \geq x_r} a_i \quad \text{and} \quad b_{\rho_r} = \sum_{p_i \in \rho : v_i \geq x_r} a_i v_i.
\]

(5)

By Lemma 4.2

\[
F_{P \cap R}(x_r) = b_{\rho_r} - a_{\rho_r} x_r.
\]

(6)

Therefore, by comparing \( F_{P \cap R}(x_r) \) with \( x_r \), we can determine in \( O(1) \) time whether \( \tau(P \cap R) = x_r \), \( \tau(P \cap R) > x_r \), or \( \tau(P \cap R) < x_r \). In the first case, we have the value of \( \tau(P \cap R) \) and return \( F_{P \cap R}(x_r) \) as an estimate of \( \mu(P \cap R) \). Otherwise, we proceed with the binary search.

After having computed the index \( i(P \cap R) \), we can compute \( \tau(P \cap R) \) in another \( O(1) \) time. Putting everything together we obtain the following:

**Theorem 4.4** Let \( P \) be a set of \( n \) uncertain points in \( \mathbb{R}^d \) where each point of \( P \) is associated with an existential probability \( \alpha \) and a value \( v \). An index of size \( O(n \log^d n) \) can be built on \( P \) in time \( O(n \log^d n) \) that for a query rectangle \( R \) returns in \( O(\log^{d+1} n) \) time, a value \( \eta_R \) such that \( \frac{1}{2} \mu(P \cap R) \leq \eta_R \leq \mu(P \cap R) \).
5 Extensions

In the interest of exposition and to highlight the conceptual framework, we have described our range-max index structures for a simple model of data uncertainty. The methodology, however, extends naturally to more complex uncertainty models, albeit at the expense of added technical details. In this section, we briefly discuss some of those extensions. The first extension deals with location uncertainty where the position of each point also has a probability distribution. The second deals with the value uncertainty case where the uncertainty is associated with the value of each point which is modeled as a probability distribution.

Location uncertainty model. In the location uncertainty model, we are given a set $P$ of $n$ uncertain points, where an uncertain point is a probability distribution over locations whose support size is at most $f$. We can assume that every distribution has support size $f$ by adding locations with null probability. Thus, an uncertain point $P_i$ is specified by a finite set of points $\{p_{i1}, \ldots, p_{if}\}$, with associated probabilities $\{a_{i1}, \ldots, a_{if}\}$. That is, $a_{ij} = \Pr[P_i \text{ occurs at } p_{ij}]$, $\sum_{j=1}^{f} a_{ij} \leq 1$, and $P_i$ does not appear with probability $1 - \sum_{j=1}^{f} a_{ij}$. The value associated with $P_i$ is a real number $v_i$. Given a range $R$, one can define range-max EM and range-max MLM for $P \cap R$ in an analogous manner to the basic model. In particular, the probability that $P_i$ does not lie in the range $R$ is $(1 - \sum a_{ij})$, where the sum is over all those locations $p_{ij}$ that fall within $R$. Omitting the technical details, due to space constraints, we simply summarize our main results for the location model, as follows.

Theorem 5.1 Let $P$ be a set of $n$ uncertain points in $\mathbb{R}^d$ in the location uncertain model, where each point is associated with a real value and a location distribution of support size $f$. Set $N = nf$. For any constant $t$ with $0 < t \leq 1$, an index of size $O(N(2f-1)t^{1+1})$ can be constructed in $O(N(2f-1)t^{1+1} \log N)$ time so that the expected maximum or the most likely maximum value of $R \cap P$ for a query rectangle $R$ can be computed in $O(N^{1-t})$ time.

Theorem 5.2 Let $P$ be a set of $n$ points in $\mathbb{R}^d$ in location uncertain model, where each point is associated with a real value and a location distribution of support size $f$. Set $N = nf^2d$. (i) An index of size $O(N \log^2d N)$ can be computed in time $O(N \log^2d N)$ so that for a query rectangle $R$ it returns in $O(\log^{2d+1} N)$ time a value $\eta_R$ such that $\frac{1}{2} \mu(R \cap P) \leq \eta_R \leq \mu(R \cap P)$, and, (ii) An index of size $O(N \log^{2d+1} N)$ can be built in $O(N \log^{2d+1} N)$ time so that for a query rectangle $R$, $\lambda(R \cap P)$ can be computed in time $O(r \log^{2d+2} N)$, where $r$ is the number of points on the skyline (when values are chosen according to a random permutation model $r = O(\log n)$ with high probability).

The location uncertain model clearly generalizes the earlier existential uncertainty model, and therefore the lower bound of Section 3 holds for this version as well. In fact, for this model, we can also establish the lower bound for the EM problem, as shown in the following theorem. (In the lower bound, we use sets with integer-valued items, as opposed to real-valued items. However, the set-intersection problem over reals is easily reduced to an integer-valued instance, and so the former is just as hard [12, 26, 27].) Please see Appendix [B] for the proof of the theorem.

Theorem 5.3 Suppose there exists an index with size $S(n, f)$ and query time $Q(n, f)$ for solving the EM problem for $n$ points in $\mathbb{R}^2$ under the location uncertain model, where each point has a distribution over at most $f$ possible locations, where $S(x, y)$, $Q(x, y)$ are monotonically increasing functions in both $x$ and $y$. Then we can solve an instance of the set-intersection problem for sets $S_1, \ldots, S_m$, with $\sum_{i=1}^{m} |S_i| = n$, in $O(Q(n, m))$ query time using space $O(S(n, m))$.

Value uncertainty model. In the value uncertainty model, we are given a set $P = \{P_1, \ldots, P_n\}$ of $n$ uncertain points, where an uncertain point $P_i$ has a unique location (that we also denote by $P_i$), and a discrete distribution over values, of support size at most $f$. Again, we assume each distribution has support size exactly $f$, by adding null probability values. Thus, each point is associated with finite set of values $V_i = \{v_{i1}, \ldots, v_{if}\} \subset \mathbb{R}$ and probabilities $\{a_{i1}, \ldots, a_{if}\} \subset [0, 1]$, where the point assumes value $v_{ij}$ with probability $a_{ij}$ and $\sum_{j=1}^{f} a_{ij} = 1$. The range-max EM and MLM are defined for the value uncertainty model,
as natural extensions of the simple model considered in Section 2. Our main result for this model is the following theorem.

**Theorem 5.4** Let $P$ be a set of $n$ points in $\mathbb{R}^d$ in the value uncertainty model, where each point is associated with a value distribution, of support size $f$. Set $N = nf$. (i) For any constant $t$ with $0 < t \leq 1$, an index of size $O(N(2d-1)t+1)$ can be constructed in $O(N^{2dt})$ time so that the expected maximum or the most likely maximum value of $P \cap R$ for a query rectangle $R$ can be computed in $O(N^{1-t})$ time. (ii) An index of size $O(N \log^d N)$ can be built in time $O(N \log^d N)$ so that for a query rectangle $R$, it returns in $O(\log^{d+1} N)$ time a value $\eta_R$ such that $\frac{1}{2}\mu(P \cap R) \leq \eta_R \leq \mu(P \cap R)$.

**Top-$k$ queries.** Efficient processing of top-$k$ queries is crucial requirement for many applications (e.g. databases, web search engine) so we briefly show how our proposed algorithms can be extended in this setting. Instead of the most likely maximum value we are interested in the most likely top-$k$ maximum values. Recall that in the first index described in Section 2, for each non-uniform set $P_j$ and for each pair $p_u, p_v$ with $p_u < p_v$, we precompute the most likely value, its probability to be the maximum in $P_j$, and the probability that no point in $[p_u, p_v]$ is present. For the top-$k$ query we need to keep sorted the $k$ most likely values (let $x_{iuj}$ denote the $i$-th most likely maximum value in $P_j \cap [p_u, p_v]$), their probabilities $\gamma(x_{iuj}, P_j \cap [p_u, p_v])$, and the probability $\tau(P_j \cap [p_u, p_v])$ to continue the query in the next group. The size of our index will be $O(kn(2d-1)t+1)$ and can be constructed in $O(kn(2d-1)t+1 \log n)$ time. The query procedure can be adapted to report the most likely top-$k$ maximum values in time $O(n^{1-t} + k \log n)$. Similarly the skyline based index can be adapted to answer top-$k$ queries. Omitting all the details we conclude the following:

**Theorem 5.5** Let $P$ be a set of $n$ uncertain points in $\mathbb{R}^d$, for some constant $d \geq 1$, where each point is associated with a real value and a probability of its existence. (i) For any $t \in [0, 1]$, an index of size $O(kn(2d-1)t+1)$ can be constructed in $O(kn(2d-1)t+1 \log n)$ time so that the most likely $k$-maximum values of $P \cap R$ for a query rectangle $R$ can be computed in $O(n^{1-t} + k \log n)$ time. (ii) Assuming that the values of $P$ are all distinct and chosen according to a random permutation model, an index of size $O(n \log^{d+1} n)$ on $P$ can be built in $O(n \log^{d+1} n)$ time so that for a query rectangle $R$, the most likely $k$-maximum values can be computed in amortized time $O(k \log^{d+3} n)$, with high probability.

6 Conclusion

In this paper, we considered the design of index structures for answering range-max queries over uncertain data. We showed that in any fixed dimension orthogonal range queries can be answered in sublinear time using a sub-quadratic size index, both for the expected maximum (EM) or the most likely maximum (MLM) value. We also give evidence that range-max queries over uncertain data are more difficult that many other types of orthogonal range queries, via a lower bound based on the conjectured difficulty of the set-intersection problem. In particular, our lower bound suggests that in the worst case the product (query time)$^2 \times$ (index size) is $n^d / \text{polylog}(n)$ for range-max queries in dimension $d \geq 2$. Our results hold for a number of data uncertainty models including the location and the value uncertainty models. We conclude by mentioning some problems for future research.

(I) What is the query-space tradeoff for the 1-dimensional range-max problem? Is the problem hard even in one dimension, or does there exists a linear-size index with polylog query?

(II) In higher dimensions, close the gap between our lower and upper bounds.

**Acknowledgments.** We thank Kamesh Munagala for pointing out the prophet inequality and Wuzhou Zhang for helpful discussions.
References


[34] G. Xiao, K. Li, K. Li, and X. Zhou. Efficient top-(k, l) range query processing for uncertain data based on multicore architectures. Distributed and Parallel Databases, pages 1–33, 2014.


A Proof of Lemma 4.1

Lemma 4.1 Let $Y_1, \ldots, Y_m$ be Bernoulli random variables such that $Y_i$ assumes value $v_i$ with probability $\alpha_i$ and 0 otherwise. Then, $E \left[ \max_j Y_j \right] \geq w \geq \frac{1}{2} E \left[ \max_j Y_j \right]$, where $w$ is a solution of the equation $w = \sum_j E \left[ \max \{ Y_j - w, 0 \} \right]$.

Proof: The easy half of the inequality is the second part, which follows simply from the following chain of implications:

$$2w = w + E \left[ \sum_{j=1}^{m} \max \{ Y_j - w, 0 \} \right] \geq w + E \left[ \max \{ \sum_{j=1}^{m} \max \{ Y_j - w, 0 \} \} \right] \geq E \left[ \max_j Y_j \right].$$

The more complex part is the first part, which we now proceed to establish. For notation simplicity we prove it for distinct values, $v_1 \geq v_2 \geq \ldots \geq v_m$, but the same result holds for non distinct values defining an arbitrary order among points with same values.

From Lemma 4.2, we know that $F_R(t) = t$ has a unique solution. We examine the function in the intervals $v_1 \geq t \geq v_2 \geq t \geq v_3 \geq \ldots \geq v_m \geq t \geq 0$. Clearly, there cannot be a solution larger than $v_1$ as is easy to check. Now, when $t$ satisfies $v_j \geq t \geq v_{j+1}$ we have that,$$
F_R(t) = \sum_{i=1}^{j} \alpha_i (v_i - t)$$

Thus, if the solution $w$ were to satisfy $v_j \geq w \geq v_{j+1}$ it would equal precisely$$w = \frac{\sum_{i=1}^{j} \alpha_i v_i}{1 + \sum_{i=1}^{j} \alpha_i},$$

as can be easily verified. This holds even for $j = m$ if we define $v_{m+1} = 0$. We let $W(j)$, for $j = 1, \ldots, m$ denote the expression as above, i.e.,$$W(j) = \frac{\sum_{i=1}^{j} \alpha_i v_i}{1 + \sum_{i=1}^{j} \alpha_i}$$

and we let $\mu_j$ denote $\mu(\{ p_i \in P \cap R \mid v_i \geq v_j \})$, i.e., the EM value only of the points $p_1, \ldots, p_j$. It is easy to see that $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_m = \mu(\bar{P} \cap R)$. We will prove that $W(j) \leq \mu_j$ for $j = 1, \ldots, m$.

Recall, $\beta(v_j, R) = \prod_{v_i > v_j} (1 - \alpha_i)$, and that $\mu_j = \sum_{i=1}^{j} v_i \gamma(v_i, R) = \sum_{i=1}^{j} v_i \alpha_i \beta(v_i, R)$ when values are distinct. We prove the result by induction on $j$. For $j = 1$, the inequality holds true because $W(1) = \alpha_1 v_1 / (1 + \alpha_1)$ while $\mu_1 = \alpha_1 v_1$ and so $W(1) \leq \mu_1$. Assume that the proposition holds true for all $j$ up to $k$. Now,$$W(k+1) = \frac{\sum_{i=1}^{k+1} \alpha_i v_i}{1 + \sum_{i=1}^{k+1} \alpha_i} \leq \frac{1 + \sum_{i=1}^{k} \alpha_i}{1 + \sum_{i=1}^{k+1} \alpha_i} \left( \sum_{i=1}^{k} \alpha_i \beta(v_i, R) v_i \right) + \alpha_{k+1} v_{k+1},$$

by using the induction assumption. So, it is sufficient to prove that,$$
\frac{1 + \sum_{i=1}^{k} \alpha_i}{1 + \sum_{i=1}^{k+1} \alpha_i} \left( \sum_{i=1}^{k} \alpha_i \beta(v_i, R) v_i \right) + \alpha_{k+1} v_{k+1} \leq \sum_{i=1}^{k+1} \alpha_i \beta(v_i, R) v_i$$
This is equivalent to,
\[
(1 + \sum_{i=1}^{k+1} a_i) \beta(v_{k+1}, R) \leq (1 + \sum_{i=1}^{k+1} a_i) \beta(v_{k+1}, R) + v_{k+1} \left( \sum_{i=1}^{k+1} a_i \right),
\]
where the last implication follows as \(a_{k+1} > 0\). So it is sufficient to prove the last inequality. Now,
\[
(1 + \sum_{i=1}^{k+1} a_i) \beta(v_{k+1}, R) v_{k+1} + (\sum_{i=1}^{k} a_i \beta(v_i, R) v_i) \geq (1 + \sum_{i=1}^{k+1} a_i) \beta(v_{k+1}, R) v_{k+1} + v_{k+1} \left( \sum_{i=1}^{k+1} a_i \beta(v_i, R) \right)
\]
\[
= (1 + \sum_{i=1}^{k+1} a_i) \beta(v_{k+1}, R) v_{k+1} + v_{k+1} (1 - \beta(v_{k+1}, R))
\]
\[
= v_{k+1} + v_{k+1} \beta(v_{k+1}, R) \sum_{i=1}^{k+1} a_i \geq v_{k+1},
\]
where we have used the easy identity, \(\sum_{i=1}^{k} a_i \beta(v_i, R) = 1 - \beta(v_{k+1}, R)\). To finish the proof, notice that \(w\) is one of the \(W(i)\) for some \(i \in [1, m]\) and so \(w = \mu_m = \mu(P \cap R)\).

\[\]

\section{Proof of Theorem 5.3}

\textbf{Theorem 5.3} Suppose there exists an index with size \(S(n, f)\) and query time \(Q(n, f)\) for solving the EM problem for \(n\) points in \(\mathbb{R}^d\) under the locational model, where each point has a distribution over at most \(f\) possible locations, where \(S(x, y), Q(x, y)\) are monotonically increasing functions in both \(x\) and \(y\). Then we can solve an instance of the set-intersection problem for sets \(S_1, \ldots, S_m\), with \(S_i = n\), in \(O(Q(n, m))\) query time using space \(O(S(n, m))\).

We first show the following result, which is required later.

\textbf{Lemma B.1} Let \(x_1, x_2, \ldots, x_n\) be the set of non-negative values assigned to a set of points \(S\), and let \(a_i, i = 1 \ldots n\) be the corresponding probabilities such that \(\sum_{i=1}^{n} a_i \leq 1\). Suppose, \(A\) is the EM value when these are considered as points in the existential model, and \(B\) is the EM value when these points are considered as points in the locational model where all points with the same value belong to one uncertain point. Then \(A \leq B\). The inequality is strict if \(a_i > 0\) for \(i = 1 \ldots n\).

\textbf{Proof:} Let \(X_1, \ldots, X_n\) be independent Bernoulli random variables, where \(X_i = x_i\) with probability \(a_i\) and \(X_i = 0\) otherwise. Then \(A = E[\max_i X_i]\). Without loss of generality let \(x_1, x_2, \ldots, x_t\) be all the distinct values and let \(Z_i, i = 1 \ldots t\), be Bernoulli random variables where \(Z_i\) assumes value \(x_i\) with probability \(\sum_{j \neq i} a_j\). It can be verified that \(B = E[\max_i Z_i]\). Now, we claim that for any given \(\lambda\) it is true that, \(E[Y] = \int_0^\infty P[Y > \lambda] \, d\lambda\), it will be true that \(E[\max_i Z_i] \geq E[\max_i X_i]\). To prove our claim it suffices to see that, \(P[\max_i Z_i > \lambda] = \prod_{j > \lambda} (1 - \prod_{i \neq j} a_i)\). In the corresponding expression for \(P[\max_i Z_i > \lambda]\) a term \(1 - \prod_{i \neq j} (1 - a_i)\) for some \(j\) with \(x_j > \lambda\) is replaced with \(\sum_{i \neq j} a_i\). Since for any \(0 \leq y_i \leq 1\) we have that \((1 - y_1) \ldots (1 - y_s) \geq (1 - (y_1 + \ldots + y_s))\), where the inequality is strict if \(y_i > 0\), it follows that, \(1 - \prod_{i \neq j} (1 - a_i) \leq 1 - \left(1 - \sum_{i \neq j} a_i\right) = \sum_{i \neq j} a_i\), and the inequality is strict when \(a_i > 0\). As such, \(P[\max_i Z_i > \lambda] \geq P[\max_i X_i > \lambda]\). The claim about strictness also follows if \(a_i > 0\).\[\]
Suppose we are given a family of sets $S_1, \ldots, S_m$ of integers with $\sum_{i=1}^m |S_i| = n$. Without loss of generality, we may assume the elements in any $S_i$ are divisible by 3. If not, we can work with the instance $S'_i = \{3x \mid x \in S_i\}$, without a change to the answer of any set-intersection query. Let $n_0 = 0$ and $n_i = n_{i-1} + |S_i|$, for $i = 1 \ldots m$. We order the elements of each set $S_i$ arbitrarily. Let $s_{i1}, \ldots, s_{ik}$ be the points in the set $S_i$ as per this ordering. We now specify our instance to the EM problem. For the set $S_i$ we consider the set of $|S_i|$ points $U_i \cup U'_i$ defined as follows. For an element $s_{ik}$ of $S_i$ we have the point $(- (k + n_{i-1}), - (k + n_{i-1}) + n)$ on the line $L: y = x + n$ in $U_i$ and the point $((k + n_{i-1}), (k + n_{i-1}) - n)$ on $L': y = x - n$ in $U'_i$. Each such point is associated the value $s_{ik}$. Let $U_i = \bigcup U_{ij}$ and $U'_i = \bigcup U'_{ij}$. We shift this entire construction horizontally by $3n$ to get the points $U_2, U'_2$ and by $6n$, to get $U_3, U'_3$ see Figure 4. The point corresponding to $s_{ik}$ in $U_2, U'_3$ are assigned the value $s_{ik}$ as before but $s_{ik} + 1$ in $U'_2$ and $s_{ik} + 2$ in $U'_3$. Let $U$ denote the union of all the points, i.e., $U = \bigcup_{k=1}^3 (U_k \cup U'_k)$. For given indexes $i, j$ we have rectangles $R_{kij}$ for $k = 1, 2, 3$ where $R_{kij} \cap U = U_{ij} \cup U'_{ij}$, see Figure 4. We let all points with the same value belong to a single uncertain point — thus all uncertain points can be computed in $O(n \log n)$ time, by sorting all the values assigned in the above construction, and identifying the common values. Each of the points has probability $\alpha$ where $\alpha < 1/10m$ is so small that every such uncertain point is still deficient (since a value can be repeated at most $6m$ times), i.e., the sum of probabilities of its “constituent points” is less than 1. Since we have at most $n$ uncertain points from $U_1 \cup U'_1 \cup U_2 \cup U_3$ and at most $2n$ other points from $U'_2 \cup U'_3$ the number of uncertain points is bounded by $3n$. Further, the maximum support size for each uncertain point can be easily bounded by $4m$. This completes our description of the input to the EM problem.

**Answering a query.** Suppose we want to answer a query: “Does $S_i$ intersect $S_j$?” where $i \neq j$. We consider the ranges $R_{1ij}, R_{2ij}, R_{3ij}$ and seek the corresponding EM values. Suppose these are $\Psi_1, \Psi_2, \Psi_3$. If $\Psi_2 = (\Psi_1 + \Psi_3)/2$ we answer “Yes”, else we answer “No”.

**Correctness.** The correctness follows from the following lemma.

**Lemma B.2** $\Psi_2 = (\Psi_1 + \Psi_3)/2$ if and only if the sets $S_i, S_j$ do not intersect.

**Proof:** Suppose $S_i \cap S_j = \emptyset$. Consider the elements of $S_i \cup S_j$ ordered by value as $x_1, \ldots, x_r$ where $r = |S_i \cup S_j|$. For the range $R_{ij}$, the points in $U_{ij} \cup U'_{ij}$ assume the values in $S_i \cup S_j$, and as these are all different the EM value is,

$$\Psi_1 = \sum_{i=1}^{r} x_i \alpha (1-\alpha)^{r-i} = \sum_{x_i \in S_i} x_i \alpha (1-\alpha)^{r-i} + \sum_{x_j \in S_j} x_j \alpha (1-\alpha)^{r-i}.$$  

For range $R_{2ij}$ the points in $U_{2i} \cup U'_{2j}$ have the values in $S_i \cup (S_j+1)$ where by $S_j + 1$ we denote the values of $S_j$ increased by 1 each. As each value in $S_j$, $S_j$ is a multiple of 3 it is easy to see that all values in $S_i \cup (S_j+1)$...
are again all distinct, and moreover the ordering among them is the same as the corresponding values from \( S_i \cup S_j \). An easy calculation shows that, \( \Psi_2 = \sum_{x_l \in S_i} x_l a(1 - a)^{r - l} + \sum_{x_l \in S_j} (x_l + 1) a(1 - a)^{r - l} \). Similarly, \( \Psi_3 = \sum_{x_l \in S_i} x_l a(1 - a)^{r - l} + \sum_{x_l \in S_j} (x_l + 2) a(1 - a)^{r - l} \). It can be easily verified that \( \Psi_2 = (\Psi_1 + \Psi_3)/2 \).

We denote the value of \( \Psi_1 \) as above by \( A \). Now, suppose \( S_i \cap S_j \neq \emptyset \). In this case, the sorted sequence of values \( x_1, \ldots, x_r \) from \( S_i \cup S_j \) does contain repeated values, but the sorted sequence of values from points in \( S_i \cup (S_j + 1) \) or \( S_i \cup (S_j + 2) \) contains no repeated values, as the numbers from \( S_i \cup (S_j + 1) \) or \( S_i \cup (S_j + 2) \) are distinct modulo 3. Thus as before, \( \Psi_2 = \sum_{x_l \in S_i} x_l a(1 - a)^{r - l} + \sum_{x_l \in S_j} (x_l + 1) a(1 - a)^{r - l} \), and \( \Psi_3 = \sum_{x_l \in S_i} x_l a(1 - a)^{r - l} + \sum_{x_l \in S_j} (x_l + 2) a(1 - a)^{r - l} \). It follows from what has been shown that, \( \Psi_2 = (A + \Psi_3)/2 \). The range \( R_{\text{ij}} \) contains points from the same uncertain point. Since \( \alpha > 0 \), Lemma B.1 implies the strict inequality \( \Psi_1 > A \). Therefore, \( \Psi_2 < (\Psi_1 + \Psi_3)/2 \).