Journal of Computers Vol. 36 No. 3, June 2025, pp. 375-393
https://doi.org/10.63367/19911599202506360324

Approximate Aggregate Spatial Queries over Multidimensional Uncertain Databases

Yanqiu Wang[1], Chuanfei Xu[2*], Xiaojun Wen[1], Zhuonan Lai[2], Weipeng Cao[2], Xi Tao[2], Zhigang Wang[3], Jinxi Li[4], and Jia Xu[3]

[1] School of Artificial Intelligence, Shenzhen Polytechnic University, Shenzhen, China
{wangyanqiu, wxjun}@szpu.edu.cn

[2] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China
{xuchuanfei, laizhuonan, caoweipeng, taoxi}@nchu.gml.ac.cn

[3] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China
{wangzhigang, xujia}@gzhu.edu.cn

[4] College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China
lijinxi@sztu.edu.cn

Received 20 May 2025; Revised 4 June 2025; Accepted 9 June 2025

Abstract. Uncertain data is inherent in LBS, sensor network and so on. Although a considerable amount of research has been dedicated to answering queries on uncertain data, how to process spatial queries over multidimensional uncertain data is still a challenge due to the prohibitive time cost when the dimensionality is high. This work focuses on seeking efficient approximate answers for the spatial queries. Specifically, given a spatial query over multidimensional uncertain objects and an error bound, we propose effective synopses to return answers with a bounded error efficiently. The existing synopses for uncertain databases are mainly designed for one dimensional databases, which cannot offer efficient answers for queries over multidimensional data. Aiming to solve this problem, we formally define Possible World Synopses (PWS) which achieve data reduction on the original possible worlds for multidimensional uncertain objects. Furthermore, PWS retain the possible worlds semantics of uncertain objects, allowing for more accurate representation of original data and query results. Based on PWS, we present a framework to answer approximate spatial queries with different precision constraints. The effectiveness of our framework is illustrated by extensive experiments. The framework proposed can efficiently and approximately handle various types of complex queries in many applications like location selection for urban planning.

Keywords: multidimensional uncertain databases, aggregate spatial queries, histogram-based synopses, approximate queries

1 Introduction

Uncertain data management has received great attention due to the inherent uncertainty in sensor network monitoring [1, 2] and LBS [3, 4]. Most of these data management techniques can only handle one-dimensional uncertain data, while there are massive amounts of multidimensional uncertain data emerging along with the technical advances on data acquisition. For example, consider a meteorology monitoring system that monitors data including multiple attributes: temperature, humidity, and so on. Due to limitation of measurements, the monitoring system produces the multidimensional uncertain objects. The current queries on multidimensional uncertain objects, however, focus on answering specific queries such as probabilistic range queries [5] and do not offer a general framework to handle various types of complex queries. The time cost of these queries is extremely expensive, because in order to verify a multidimensional uncertain object as a result, we take into consideration not only

* Corresponding Author

each possible position of this object in a multidimensional space but also the probability of the object appearing at each possible position. In this work, we aim to develop a framework for modeling multidimensional uncertain objects approximately and efficiently. The attribute values for a data object in multidimensional uncertain databases are specified using a discrete probability distribution function in a multidimensional space. In the multidimensional uncertain database $\boldsymbol{D}$, each object is represented by a set of vector instances which are exclusive alternatives. We assume that these objects are independent of each other. The probability distributions of the uncertain objects are within the d-dimensional space $\boldsymbol{U}^d$ in which each instance of an object denotes a possible position of this object.

For spatial queries over multidimensional objects, similar to uncertainty of data objects, query objects are often uncertain. Aggregate spatial queries can be considered as a kind of spatial queries that aim to return answers to a set of multiple query objects [4], while traditional queries aim to return answers to only one query object. Consequently, traditional kNN or range queries can be regarded as special cases of an aggregate spatial query issued by one query object. To make our solution more generic, we study aggregate spatial queries, which have multiple query objects of multidimensional uncertainty such as aggregate kNN queries and aggregate range queries on multidimensional uncertain objects. In many real-world applications, aggregate spatial queries over multidimensional uncertain databases are important which is motivated by the following example.

Example 1. In the location selection application, we would like to search k candidate places from many places to build a botanical garden. We hope that the environment of these candidate places are most suitable for cultivating some rare plants. To ensure these plants to be alive, we desire that the environment of candidate places is most similar to that of original growth-places of all the plants. For the original growth-place of each kind of plants and all the given places, we mainly monitor three environmental factors, i.e., (*temperature*, *humidity*, *light*) via a sensor network. Due to packet loss or low battery power, sensor data are usual not precise. The imprecise sensor data are represented by probability distributions over exclusive instances, as shown in Table 1. In this case, we can develop an aggregate kNN query to find k places whose environmental factors are most likely to have the smallest aggregate deviation to environmental factors of all the plants' original growth-places. E.g., for $k = 1$, the probability of aggregate deviation between environment of Place 2 and original environment for Plant A and B to be smallest is equal to $(0.8\times0.4 + 0.8\times0.6 + 0.2\times0.4 + 0.2\times0.6) \times 0.7 = 0.7$. Likewise, the probability of that of Place 1 to be smallest is 0.3. Hence, Place 2 is more suitable for cultivating Plant A and B than Place 1.

Table 1. Uncertain objects and query objects

Environment	Probability distribution
Plant A	< (27, 35, 54), 0.8 >, < (25, 32, 51), 0.2 >
Plant B	< (28, 35, 57), 0.4 >, < (27, 34, 53), 0.6 >
...	...
Place 1	< (22, 30, 48), 1 >
Place 2	< (27, 34, 55), 0.7 >, < 21, 29, 48), 0.3 >
...	...

In some other applications, we desire aggregate range queries over high dimensional uncertain data. For instance, an uncertain database as described in [6] is composed of ten-dimensional environment sensor data (attributes): temperature, humidity, wind speed and so on. We wish to determine which uncertain objects are most likely to reside in the range of each query object. Suppose that there are two sensors collecting data denoting different environments, which are regarded as two uncertain query objects (i.e., q_1 and q_2). Each uncertain object o in the uncertain database describes the environment of some location. Our task is to find the object o which is most likely to belongs to the range of q_1 or q_2. Thereby, we define the minimum distance of o to q_1 and q_2 as the aggregate distance and use the aggregate distance to answer range queries. Given a range denoted by ε, a query is desired to retrieve the objects whose minimum distances to q_1 and q_2 are most likely to be in the range of q_1 or q_2 respectively. Assume that the object o has two instances u_1 and u_2. When $o = u_1$, the minimum distance to q_1 and q_2 is equal to the distance between u_1 and an instance of q_1. If the distance is not more than ε, o is most likely to be in the range of q_1. We will give the formal definition of such queries in the next section.

It is time-consuming to answer the above multidimensional aggregate spatial queries since these queries need to calculate a great number of aggregate distances (i.e., sum or minimum distances) in the high-dimensional space. Specially, when answering the query based on the possible world model in Example 1, we need to handle possible worlds of multidimensional environment data for each place and the original growth-places of all the plants. Since the number of possible worlds is exponential, how to improve the query efficiency for querying

over multidimensional databases is a big challenge. As proved later, the time cost of answering the above query is #P-hard, so that the query evaluation raises serious practicality concerns for uncertain database systems in a realistic setting [7]. To improve efficiency, approximate query processing is introduced to evaluate distances between any two objects approximately. Specifically, given an ε-approximate spatial query, the results will not be far away from the real ones by ε. The current solutions [8-10] aim to answer approximate query problems over exact data so that they are not suitable for multidimensional uncertain data.

In order to cope with multidimensional uncertain databases effectively, we aim to construct data synopses for the original uncertain databases with data reduction techniques and answer spatial queries based on the data synopses. Meanwhile, the total time cost of this process needs to be much less than that of answering these queries directly. Two uncertain data reduction techniques using probabilistic histograms have been proposed in [11, 12] to compress uncertain data effectively. However, the performance of these reduction techniques will decrease significantly when handling multidimensional uncertain data. The reason is that these techniques compact uncertain data into one-dimensional histograms which will produce a prohibitively large error for the multidimensional uncertain data. In addition, uncertain data clustering [13-15] can achieve data reduction as well. Using the current clustering methods, each uncertain object is represented by the center of a cluster approximately. This representation loses the possible world semantics of the original data. Hence, there may exist a large error when answering probabilistic spatial queries based on the clusters. Furthermore, based on probabilistic histograms or clusters, the precision of queries over multidimensional uncertain objects cannot be guaranteed.

This work focuses on answering aggregate spatial queries based on our data synopses, which addresses two main concerns. (i) How to build synopses for multidimensional uncertain databases according to the precision requirements of spatial queries. (ii) How to efficiently answer aggregate spatial queries on multidimensional uncertain objects based on the built data synopses. Aiming at addressing the previous concerns, we first propose a new kind of data synopses called Possible World Synopses (PWS) which is able to retain an approximate possible worlds semantics for original databases. Furthermore, based on PWS, we propose some methods for answering the multidimensional uncertain queries in polynomial time. During this process, the errors of queries are guaranteed by constructing the proper PWS under a given error bound. To the best of our knowledge, this is the first paper that aims to answer approximate aggregate spatial queries over multidimensional uncertain databases based on multidimensional data synopses. In detail, our contributions are summarized as follows.

- We develop a general framework for solving approximate spatial query problems over multidimensional uncertain objects. Specially, we define a novel kind of data synopses called PWS, which compact possible worlds of the uncertain objects, to answer queries based on the data synopses.
- We present an approach to build a variety of optimal reduction-ratio PWS called OPT-RR under an error constraint. We also propose an error bound method to improve the construction efficiency by reducing search space safely.
- Given an error bound Δ, we can return approximate results for aggregate spatial queries over multidimensional uncertain databases (which are #P-hard problems) in polynomial time while guaranteeing the errors of the queries are smaller than Δ.
- We evaluate the effectiveness and the efficiency of our approaches through extensive experiments with real and synthetic datasets.

The rest of our work is organized as follows. Section 2 reviews related work. Section 3 depicts the preliminaries. Section 4 develops a framework for answering the approximate spatial queries. Section 5 proposes approaches to construct PWS. Section 6 provides solutions for answering approximate spatial queries based on the proposed synopses. Section 7 presents the experimental results. Section 8 concludes our paper.

2 Related Work

2.1 Query with Approximate Methods

Approximate Query Processing are essential for dealing with massive uncertain data. Many previous works [1, 3, 16-18] devoting to managing data approximately. C. Ré et al. [19] proposed sufficient lineage and polynomial lineage. In order to answer *k*-center and *k*-median problems efficiently, S. Alipour [16] propose approximation algorithms. Compared with traditional possible world, AU-DB contains more information about uncertainty.

Furthermore, S. Feng et al. [3] extend their work to support top-k, ranking and windowed aggregation queries over uncertain data.

However, these works are not natively designed for multidimensional uncertain data.

2.2 Query over Multidimensional Uncertain Data

With the prevalence of uncertain data, many work [20-22] investigate the problem of querying over uncertain databases, while others [23-27] focus on uncertain queries over distributed frameworks. Specifically, for queries on multidimensional uncertain data, Y. Tao et al. [28] address the range search. They first proposed the *probabilistically constrained rectangle* (*PCR*). In [5], J. Bernad et al. handle different uniform probabilistic distributions of different shapes. Also, L. Shou et al. [29] proposed a framework for efficient processing of probabilistic-threshold range query over multidimensional uncertain data. In addition, range-constrained probabilistic mutual furthest neighbor query is proposed in [21], where range query Q is a d-dimensional hyperrectangle. They also propose two effective pruning methods to help reduce the candidate pairs.

However, these works focus on answering specific queries and do not offer a general framework to handle various types of complex queries.

2.3 Uncertain Data Reduction

Compressing the probabilistic database is very important for queries and understanding data. G. Cormode et al. [12] present to build histogram based synopses on probabilistic data. They further explore how to build probabilistic histogram synopses [11], which can represent the uncertain characteristics of data and query results with higher accuracy. M. Tang et al. [30] investigate how to scale up histogram constructions in large uncertain data. Y. Zhang et al. [31] present a strategy of query selectivity estimation based on histogram, which focus on the entity-based relational database model instead of possible world instances. Besides, the clustering uncertain data method [13] is able to compact multidimensional uncertain data into a set of clusters. However, this method would lose the possible worlds semantics of the original data. The characteristics of the reduction methods is summarized in Table 2.

Table 2. Uncertain data reduction solutions

Solution	Reduction	Multi-dimensional	Possible world semantics
Histograms [12]	√	×	×
P-Histograms [11]	√	×	√
Clusters [13]	√	√	×
Proposed	√	√	√

3 Background

We first formalize a multidimensional uncertain data model to describe multidimensional uncertain objects. Then we define aggregate spatial queries over multidimensional uncertain objects. Finally, we propose a framework for our problem. Table 3 summarizes the commonly-used symbols in the remaining paper.

Table 3. Commonly used symbols

Symbol	Description
$\boldsymbol{D}$	d-dimensional uncertain database
n	cardinality of $\boldsymbol{D}$
$\boldsymbol{U}^d$	d-dimensional space
$\boldsymbol{V}$	set of all instances
$\boldsymbol{N}$	cardinality of $\boldsymbol{V}$
$\boldsymbol{X}$	*approximate set*

v_i	instance of an object
p_{v_i}	probability that an object appears at v_i
C	set of key instances
m	size of C
b	space bound
Δ	error bound

3.1 Problem Definition

Suppose that an uncertain database $\boldsymbol{D}$ consists of n objects $\{o_1, \ldots, o_i, \ldots, o_n\}$, where $o_i (1 \le i \le n)$ represents a d-dimensional uncertain data object located within the d-dimensional space $\boldsymbol{U}^d$. The query objects are also assumed to be a set of the d-dimensional uncertain objects $\boldsymbol{Q} = \{q_1, \ldots, q_i, \ldots, q_{|\boldsymbol{Q}|}\}$. In this paper, we assume independence across all the objects, and any object ($\forall o_i \in \boldsymbol{D}$, $\forall q_j \in \boldsymbol{Q}$) is associated with probability distributions over mutually exclusive alternatives. Each possible alternative of o_i is considered as an instance u. The value domain $\boldsymbol{V}$ is composed of the instances of all the uncertain objects in $\boldsymbol{D}$ and furthermore $|\boldsymbol{V}| = N$. In the d-dimensional space $\boldsymbol{U}^d$, our multidimensional uncertain data model is summarized as follows.

Definition 1. *(Multidimensional Uncertain Data Model) In an uncertain database $\boldsymbol{D}$, each multidimensional uncertain object is denoted as a discrete distribution within the d-dimensional space $\boldsymbol{U}^d$ in which an instance of the uncertain object is represented by a position v_i. The probability that this object appears at v_i is equal to p_{v_i}, and $\Sigma_i p_{v_i} \le 1$.*

Using the above model, the distance between a data object o_i and a query object q_j is denoted by $d(o_i, q_j)$. If $o_i = v_i$ and $q_j = w_j$ (i.e., o_i and q_j are located at positions v_i and w_j respectively), the probability that $d(o_i, q_j) = d(v_i, w_j)$ is equal to $p_{v_i} \cdot p_{w_j}$. Next, we formally define the aggregate spatial queries such as aggregate kNN queries and aggregate range queries over multidimensional uncertain objects as follows.

Definition 2. *(Aggregate kNN Queries over Multidimensional Uncertain Objects) Given a set of multidimensional uncertain query objects $\boldsymbol{Q}$, an aggregate kNN query retrieves k objects $\{o^*_r, r = 1, ..., k\} (o^*_r \in \boldsymbol{D})$ such that the probability of $\boldsymbol{h}(o^*_r, \boldsymbol{Q})$ to be smallest value is r-th. Here $\boldsymbol{h}(o^*_r, \boldsymbol{Q}) = \boldsymbol{h}(d(o^*_r, q_1), d(o^*_r, q_2), ..., d(o^*_r, q_{|\boldsymbol{Q}|}))$, where $\boldsymbol{h}()$ denotes an aggregate function (i.e., sum, min, max) and $d()$ denotes the Euclidean distance.*

Definition 3. *(Aggregate Range Queries over Multidimensional Uncertain Objects) Given a set of multidimensional uncertain query objects $\boldsymbol{Q}$, the range ε and the probability threshold τ, an aggregate range query returns any objects $o_i \in \boldsymbol{D}$ such that $Pr(\boldsymbol{h}(o_i, \boldsymbol{Q}) \le \varepsilon) > \tau$. Here $\boldsymbol{h}(o_i, \boldsymbol{Q}) = \boldsymbol{h}(d(o_i, q_1), d(o_i, q_2), ..., d(o_i, q_{|\boldsymbol{Q}|}))$, where $\boldsymbol{h}()$ denotes an aggregate function (i.e., sum, min, max) and $d()$ denotes the Euclidean distance.*

As mentioned in the above two definitions, if there is only one query object in $\boldsymbol{Q}$, both aggregate kNN and range queries become traditional spatial queries over multidimensional uncertain objects. Therefore, aggregate spatial queries include traditional spatial queries. From the problem statements, in order to answer aggregate spatial queries effectively, we need to analyze the time complexity of evaluating these queries over multidimensional uncertain objects in the following lemma.

Lemma 1. *The time cost for answering any aggregate spatial query over multidimensional uncertain objects is #P-hard.*

Proof. Given a set of multidimensional uncertain query objects $\boldsymbol{Q}$ and a multidimensional uncertain database $\boldsymbol{D}$, we need to evaluate the aggregate function $h(o_i, \boldsymbol{Q}) = h(d(o_i, q_1), d(o_i, q_2), \ldots, d(o_i, q|_{\boldsymbol{Q}}|))$ for $\forall o_i \in \boldsymbol{D}$ to answer the aggregate spatial query. Then we need to evaluate the minimum value of aggregate function with the maximum likelihood. Since there are the exponential size of groups of objects for the aggregate function $h(o_i, \boldsymbol{Q})$ (i.e., each group of objects represents $\{o_i, q_1, \ldots, q_{|\boldsymbol{Q}|}\}$), we must find a group of objects which is most likely to has the minimum value of $h(o_i, \boldsymbol{Q})$. We wish to select a group of instances to minimize $h(o_i, \boldsymbol{Q})$. Because we cannot verify whether any group of instances is best for minimizing $h(o_i, \boldsymbol{Q})$ in polynomial time, according to the reduction proof of #P-hard in [32], the time cost for answering the aggregate spatial query is #P-hard.

$$\sum_{v_i \in o, w_j \in q} p_{v_i} \cdot p_{w_j} \cdot d\left(v_i \cdot w_j\right) - \Delta \leq \widetilde{ED} \leq \sum_{v_i \in o, w_j \in q} p_{v_i} \cdot p_{w_j} \cdot d\left(v_i \cdot w_j\right) + \Delta \quad (1)$$

From the above inequality, the expected distance between o and q can be approximately evaluated in the approximate kNN query problem. Moreover, the error between the value of $\widetilde{ED}$ and the true value of the expected distance must be smaller than the error bound Δ. The precision of approximate kNN queries would be higher, if the error bound Δ is smaller (when $\Delta = 0$, the answers of an approximate kNN query are precise since all the expected distances between objects and the query object are evaluated exactly). Hence, the query precision is guaranteed by setting a proper value of Δ.

We also study approximate answers for range queries. For any uncertain query object q, accurate range queries calculate the probability that any uncertain object o is within the ε range to q. If $Pr(d(o, q) \leq \varepsilon) > \tau$, o is a result. Similar to approximate kNN queries, we need to evaluate the probability of $d(o, q) \leq \varepsilon + \Delta$, which is the Δ-approximate range query problem. For $\forall o \in \boldsymbol{D}$, if $Pr(d(o, q) \leq \varepsilon + \Delta) > \tau$, o is a result of the Δ-approximate range query. Consequently, approximate range queries over multidimensional uncertain objects can be answered under the error bound Δ.

3.2 General Framework for Approximate Spatial Queries

To guarantee that the error for an approximate query is not more than the given error bound Δ, we propose a solution to build data synopses for multidimensional uncertain objects according to Δ. Then we answer aggregate spatial queries based on the synopses. In this subsection, we develop a general framework for achieving this process. The framework is composed of three phases: building data synopses, retrieving candidate results and refinement, which is illustrated in Fig. 1.

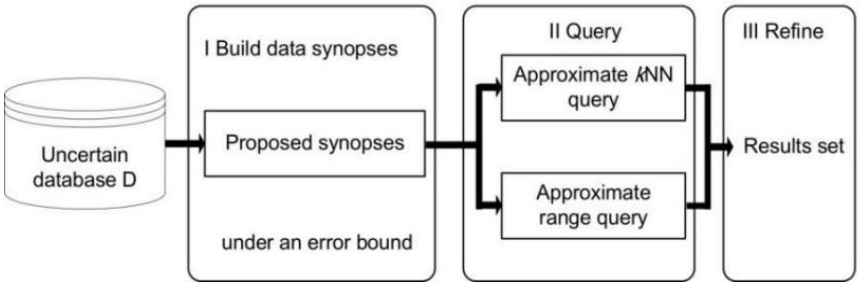

Fig. 1. The approximate query framework

In detail, we highlight our framework for answering approximate kNN and range queries (even for the queries including multiple uncertain query objects) based on data synopses in the three phases.

- **Building data synopses phase:** According to query requirements, we construct a new kind of data synopses for modeling multidimensional uncertain databases.
- **Retrieving candidate results phase:** Based on the constructed synopses, we retrieve the approximate results of aggregate spatial queries while guaranteeing that the errors for these queries are not more than Δ.
- **Refinement phase:** We refine the candidate results to improve precision of results in the refinement phase. For example, given an approximate kNN query, if the probabilities of some objects that are nearest to the query object q are equal, we can further compare them in the refinement phase.

4 Possible World Synopses

4.1 Possible World Synopses Definition

One instance with its appearance probability from each object in the multidimensional uncertain database $\boldsymbol{D}$ generates a possible world of the database $\boldsymbol{D}$. In this way, the exponential size of possible worlds is generated. For example, four uncertain objects $o_1 = \{u_1, u_2, u_3, u_4\}$, $o_2 = \{v_1, v_2, v_3\}$, $o_3 = \{w_1, w_2, w_3, w_4\}$ and $o_4 = \{r_1, r_2\}$ are shown in Fig. 2 whose instances are marked as different shapes in $\boldsymbol{U}^d$. For a possible world $\{o_1 = u_1, o_2 = v_2, o_3 = w_4, o_4 = r_1\}$ (e.g., select u_1 for o_1), its probability is computed as $p_{u_1} \cdot p_{v_2} \cdot p_{w_4} \cdot p_{r_1} = 0.0072$. There are only four objects but 4×3×4×2 = 96 possible worlds generated by the instances of the objects. Hence, it is rather time-consuming to answer queries over all these possible worlds. If these instances can be replaced by the lower number of new instances approximately, the possible world model would be simplified. In the following definition, we formally define a novel type of data synopses for possible worlds, called Possible World Synopses (PWS) which can compress original possible worlds greatly while retaining the possible worlds semantics (i.e., any object is associated with a probability distribution over different alternatives, which describes the likelihood that the object appears at each position).

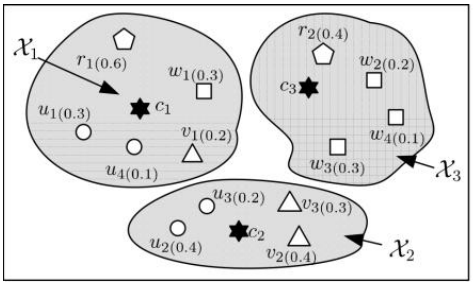

Fig. 2. An example of PWS

Definition 4. (*Possible World Synopses*) *Possible World Synopses* (*PWS*) *is composed of reduced possible worlds which can represent the approximate possible worlds semantics of the uncertain database* $\boldsymbol{D}$. *Under PWS, each possible world is generated by a series of new instances which are the representative values for the instances of uncertain objects in the database* $\boldsymbol{D}$.

Then we define *key instances* by which all the possible worlds for PWS are generated as follows.

Definition 5. (*Key instance*) *For a set of instances* $\boldsymbol{V}$, *any instance* $v \in \boldsymbol{V}$ *is approximately represented by a new representative instance* $\boldsymbol{\sigma}(\boldsymbol{v})$ *that is called key instances* $c = \boldsymbol{\sigma}(\boldsymbol{v})$ $(\boldsymbol{c} \in \boldsymbol{C})$, *where* $\boldsymbol{\sigma}()$ *denotes a synopses mapping function by which V is mapped into a key instances set C.*

Since multiple instances are represented by a *key instances*, the number of instances is more than that of *key instances*. The instances mapped into the same *key instances* form an *approximate set* in which all the instances are represented by this *key instances*, and all the *key instances* form *C*.

To construct proper PWS, we need to decide which instances can be represented by a *key instances* and to find each *key instances's* position in the space $\boldsymbol{U}^d$, which will be discussed in detail in Section 5. As illustrated in Fig. 2, we assume that a synopses mapping function $\boldsymbol{\sigma}()$ maps the instances in $\boldsymbol{X}_1$, $\boldsymbol{X}_2$ or $\boldsymbol{X}_3$ (i.e., assume that $\boldsymbol{X}_1 = \{u_1, u_4, v_1, w_1, r_1\}$, $\boldsymbol{X}_2 = \{v_2, v_3, u_2, u_3\}$, $\boldsymbol{X}_3 = \{w_2, w_3, w_4, r_2\}$) into *key instances* c_1, c_2 or c_3 respectively. Furthermore, these instances are represented by the three *key instances* c_1, c_2 and c_3. Using PWS, each uncertain object can solely appear at positions of *key instances*. For example, o_1 can appear at c_1 or c_2 (i.e., the instances of o_1 are included by $\boldsymbol{X}_1$ or $\boldsymbol{X}_2$), and the probabilities to appear at the *key instances* are denoted by $Pr(o_1 = c_1)$ and $Pr(o_1 = c_2)$ respectively. Since o_1's instances u_1 and u_4 are assumed to be grouped into $\boldsymbol{X}_1$, they are represented by c_1 approximately. The probability that o_1 is located at c_1 equals to the summation of the likelihood that o_1 appears at u_1 and u_4, namely $Pr(o_1 = c_1) = p_{u_1} + p_{u_4} = 0.4$. Likewise, $Pr(o_1 = c_2) = p_{u_2} + p_{u_3} = 0.6$. Because each object is represented by a probability distribution, the possible worlds semantics are retained by PWS.

Table 4. Comparing PWS with possible world model

	Possible world	PWS
Number of instances	$\|V\|$=13	$\|C\|$=3
Number of possible worlds	96	16
A distribution (e.g., o_1)	$\{(u_1, 0.3), (u_2, 0.4), (u_3, 0.2), (u_4, 0.1)\}$	$\{(c_1, 0.4), (c_2, 0.6)\}$
A possible world	$\{o_1 = u_1, o_2 = v_1, o_3 = w_2, o_4 = r_1\}$	$\{o_1 = c_1, o_2 = c_1, o_3 = c_3, o_4 = c_1\}$

Compared to other data reduction methods without retaining the possible worlds semantics such as clustering multidimensional uncertain data, the precision of probabilistic queries is higher under PWS. Given a probabilistic query, if an object is replaced by a cluster center, the probability that the object can be the query result is one or zero, which is not meaningful for probabilistic queries in most cases. However, compared to cluster representation, PWS have more accurate representation of the uncertainty characteristics of data and query results.

We build PWS for all possible worlds of the objects in Fig. 2. Table 4 compares PWS with the possible world model. From Table 4, the number of possible worlds is 96 under the possible world model. On the other hand, the size of PWS are only 16 (i.e., each object has two exclusive alternatives). Under PWS, each possible world is denoted as a series of *key instances* that are selected for each uncertain object (e.g., $\{o_1 = c_1, o_2 = c_1, o_3 = c_3, o_4 = c_1\}$). Compared to the possible world model, the size of PWS is evidently reduced. Moreover, our synopses represent each object with a probability distribution, which retains an approximate possible worlds semantics for the original data.

The deviations (i.e., distances) between *key instances* and the instances of the original uncertain objects produce errors of PWS. Clearly, if we process a query based on PWS, the query results may not be accurate because of the errors produced by our synopses. Next, we study the trade off between the query precision and efficiency based on PWS.

4.2 Optimal Reduction-Ratio

Under PWS, each original object is represented by a probability distribution over exclusive alternatives, each denoting a *key instances* at which the object appears possibly. Intuitively, the time cost of answering spatial queries based on PWS will rise directly with the size of *key instances* set C. The larger the size of C, the lower the query efficiency. Thus, the size of C is another important concern for constructing PWS. To improve query efficiency, we desire synopses with smaller size of C while ensuring that the error bound of any object is not more than a given threshold. We summarize a type of optimal PWS as follows.

Definition 6. (*The Optimal Reduction-Ratio for PWS, OPT-RR*) *OPT-RR denotes a type of PWS with the minimum amount of key instances while ensuring that the maximum deviation between key instances and the instances of any object is not more than a given error bound.*

As described in the above definition, due to the minimum number of *key instances*, OPT-RR has the optimal reduction-ratio for the instances of original objects under the given error bound. Given an error bound Δ, we address the approximate aggregate kNN query problem based on OPT-RR.

Lemma 2. *If OPT-RR is constructed under the error bound* Δ*, then the error of any approximate aggregate kNN query based on OPT-RR is not more than* Δ*.*

Proof. According to Eq. (1), for each object o and each query object q, the upper bound of error between estimated value and real value is denoted by Δ. Due to triangular inequality, the upper bound of $|d(v_i, q_j) - d(c_s, q_j)|$ is not more than $|d(v_i, c_s)|$. Consequently, $|\Sigma_{v_i \in o,\, q_j \in q} p_{v_i} \cdot p_{q_j} \cdot d(v_i, q_j) - \Sigma_{c_s \in C,\, q_j \in q} p_{c_s} \cdot p_{q_j} \cdot d(c_s, q_j)| \leq \Delta$. Using the bound to build OPT-RR, thereby, the expected distance between any object and the query object is not more than the error bound Δ. Based on the above description, the error of approximate aggregate kNN query based on OPT-RR is not more than Δ.

Then we study approximate aggregate range queries which can be answered by setting an error bound Δ. Under the error constraint, we also propose an efficient solution for approximate aggregate range query problem, which is summarized in the following lemma.

Lemma 3. *If OPT-RR is constructed under the error bound* $\mathbf{\Delta}$*, then the error of any approximate aggregate range query based on OPT-RR is not more than* Δ*.*

Proof. Based on OPT-RR, we set the maximum deviation (i.e., distance) between *key instances* and the instances of of any uncertain object as the error bound **Δ**. For any object o and each query object q, the expected distance for evaluating $d(o, q)$ is computed as $|\Sigma_{v_i \in o,\, q_j \in q} p_{v_i} \cdot p_{q_j} \cdot d(v_i, q_j) - \Sigma_{c_s \in C,\, q_j \in q} p_{c_s} \cdot p_{q_j} \cdot d(c_s, q_j)| \leq \Delta$, where c_s denotes one of *key instances* for the object o. Therefore, the probability that o is located within the ε range to q for the approximate range query is equal to $P_r(d(o, q) - |\Sigma_{v_i \in o,\, q_j \in q} p_{v_i} \cdot p_{q_j} \cdot d(v_i, q_j) - \Sigma_{c_s \in C,\, q_j \in q} p_{c_s} \cdot p_{q_j} \cdot d(c_s, q_j)| \leq \varepsilon)$. Based on OPT-RR, since the maximum deviation between *key instances* and instances of o is at most **Δ**, the above probability is computed as $Pr(d(o, q) \leq \varepsilon + \mathbf{\Delta})$. Note that o is a result for this approximate query based on OPT-RR when the value of $Pr(d(o, q) \leq \varepsilon + \mathbf{\Delta})$ is more than the probability threshold. Based on the above error computing, the error of the approximate aggregate range query based on OPT-RR is not more than Δ.

5 OPT-RR Construction

Due to precision constraints, we need to guarantee that the error of approximate aggregate spatial queries is not more than the given error bound Δ. According to Lemma 2 and Lemma 3, we can build OPT-RR, a type of optimal PWS which ensures that the maximum deviation between *key instances* and the instances of the uncertain object for $\forall o \in \boldsymbol{D}$ is not more than Δ to solve the approximate spatial query problem. As query-based synopses, the construction of PWS depend on the specific queries. In this section, we propose the naive solution for constructing OPT-RR according to different query requirements. And then we also present a more efficient algorithm by reducing the search space.

5.1 Naive Solution

Given an error bound Δ of an approximate aggregate spatial query, OPT-RR can be constructed to achieve the best query efficiency. To build OPT-RR, there are two key issues to be addressed: (i) how to decide which instances are represented by *key instances*; (ii) how to represent each uncertain object with *key instances*. At first, we desire to find each *approximate set* $\boldsymbol{X}_i (1 \leq i \leq m)$ in which the instances are mapped into the same *key instances* for OPT-RR. We discuss the relation between the deviations (i.e., the distances between instances and their *key instances*) in any *approximate set* and the error bound Δ as follows.

Lemma 4. *For each approximate set* $\boldsymbol{X}_i (1 \leq i \leq m)$*, if the maximum distance between instances and their key instance is not more than the given error bound* Δ*, then the deviations between key instances and the instances of any object are also not more than* Δ*.*

Proof. Assume that $d_{\max}$ denotes the maximum distance between instances and their *key instances* in the d-dimensional Euclidean space $\boldsymbol{U}^d$. For $\forall o \in \boldsymbol{D}$, its deviation is computed as $\Sigma_{i=1}^{I} p_{v_i} \cdot d(v_i, \boldsymbol{\sigma}(v_i))$ where v_i is the *i-th* instance of $o (1 \leq i \leq I)$. Due to $d(v_i, \boldsymbol{\sigma}(v_i)) \leq d_{\max}$, it follows that as $\Sigma_{i=1}^{I} p_{v_i} \cdot d(v_i, \boldsymbol{\sigma}(v_i)) \leq \Sigma_{i=1}^{I} p_{v_i} \cdot d_{\max} = d_{\max}$. According to the condition of this lemma, $d_{\max} \leq \Delta$, the deviations for any object are not more than Δ. That is so say, the deviations between *key instances* and the instances of any object are not more than Δ.

Note that, OPT-RR can be constructed by *approximate sets* which can ensure the maximum distance between *key instances* and the instances of any object is more than the given error bound Δ. Since aggregate spatial queries need to compare the distances between query objects and data objects, we build multidimensional R-tree to index multidimensional uncertain objects in order to answer approximate spatial queries efficiently. As illustrated in Fig. 3, the nodes of R-tree are represented by d-dimensional rectangles, called *Minimum Bounding Rectangles* (e.g., the *MBR* of node N_3 is shown in Fig. 3).

To satisfy the error bound for constructing OPT-RR, we desire some *approximate sets* whose maximum distance between instances and their *key instance* is not more than the given error bound Δ. Thus, we only need to find a set of instances that are mapped into a proper *key instance*. For the R-tree, we construct each *approximate sets* that can be formed by a *MBR* in which the distances between all the instances and the center are not more than Δ. Also, the center of *MBR* is set to be the *key instance* for these instances.

Next, we propose the naive solution for constructing OPT-RR. In this solution, the foremost step of building OPT-RR is evaluating the distances between any two instances in the *MBR*. When Δ is less than the maximum distance among instances in the *MBR*, this *MBR* should be unfolded and its children nodes are searched until satisfying the error bound constraint. In this way, the *MBRs* satisfying the error bound constraint are retrieved and the centers of these *MBRs* are set to be the *key instance*. According to the definition of PWS, each uncertain

object is represented by a probability distribution over these *key instance*.

The naive solution first search the *MBRs* that have the minimum distances to the query objects and then form the *approximate sets* using the *MBRs* in which the maximum distance among instances is not more than Δ. If any node's *MBR* satisfies the error bound, its center is set to be a *key instance*. Its child nodes are not necessary for constructing OPT-RR, because OPT-RR wishes to reduce the original data with the optimal reduction ratio. Hence, OPT-RR is built by this kind of *MBRs*. Using the *MBRs*, the lowest number of *key instance* are searched to answer approximate spatial queries based on OPT-RR so that the reduction-ratio optimization is guaranteed. Furthermore, the queries based on OPT-RR can be solved under the precision constraints.

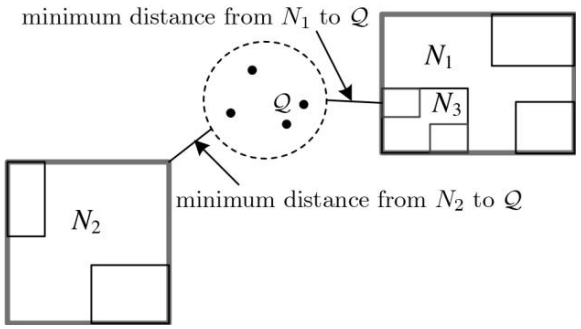

Fig. 3. An example of multidimensional R-tree

5.2 Error Bounds for MBRs Based Algorithm

As mentioned previously, we have discussed the naive solution for building OPT-RR, a type of optimal PWS. This solution evaluates the maximum distances between instances and their *key instance* for all *MBRs* until the precision constraints are satisfied. The cost of evaluating the maximum distance is rather high since we must compute the distance between each instance and its *key instance*. To improve the efficiency, we reduce the search space with the error bounds for these *MBRs*. The upper and lower error bounds of a *MBR* limit the distances between instances and *key instance* within the bounds. If we can evaluate *MBRs'* error bounds, the number of evaluated *MBRs* would decrease and thus improve the efficiency of building OPT-RR. We summarize the method for calculating error bounds of *MBRs* in the following lemma.

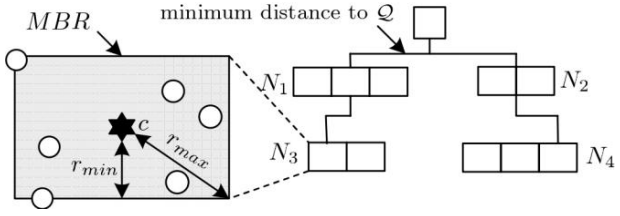

Fig. 4. Uncertain objects indexed by an R-tree

Lemma 5. *For a d-dimensional MBR, the upper error bound can be formed by the maximum distance between its center and the farthest vertex of this MBR.*

Proof. One *MBR* for a node of the *d*-dimensional R-tree is assumed to be an *approximate set* whose *key instance* is denoted by the center point *c* (as illustrated by N_3 in Fig. 4). The maximum error of this *approximate set* is formed by the maximum distance between the center point and the instance within the *d*-dimensional *MBR*. Due to the constraints on *MBR*, the above distance is not more than the maximum distance (e.g., r_{max}) between the center point and vertexes of the *MBR*. Thus, upper error bound is formed by the maximum distance between its center and the farthest vertex of this *MBR*.

Lemma 6. *For a d-dimensional MBR, the lower error bound can be formed by the minimum distance between its center and the nearest edge of this MBR.*

Proof. One *MBR* for a node of the *d*-dimensional R-tree is assumed to be an *approximate set* whose *key instance* is denoted by the center point *c*. The minimum error of this *approximate set* is formed by the minimum distance between the center point and the instance within the *d*-dimensional *MBR*. Due to the constraints on *MBR*, the above distance is not more than the minimum distance (e.g., $r_{\min}$) between the center point and vertexes of the *MBR*. Thus, lower error bound is produced by this distance. Thus, the lower error bound not more than the minimum distance between its center and the nearest edge.

As indicated by Lemma 5 and Lemma 6, error bounds of each *MBR* can be evaluated without calculating the maximum distance between instances and their *key instance*. We propose error bounds of the *MBR* based algorithm which unfolds the *MBRs* whose lower error bounds are higher than the given error bound Δ, and search their child *MBRs*. In another case, if the upper error bound of any *MBR* is less than Δ, the instances in this *MBR* generate an *approximate set*. The key steps of this algorithm are given in Algorithm 1.

Algorithm 1. Error bounds of *MBR* based algorithm

Input:
uncertain database *D* indexed by R-tree, the query object *q*, and error bound Δ
Output:
approximate sets $\{X_1, X_2, \ldots, X_m\}$ and *key instances* $\{c_1, c_2, \ldots, c_m\}$

```
1   initialize all nodes of R-tree
2   i ← 1 and j ← 1
3   for the center cen_j of MBR for j-th node N_j do
4      evaluate the maximum distance r_max_j and minimum distance r_min_j of any instance to cen_j
5      if r_max_j ≤ Δ then
6         take all instance of N_j into X_i
7         c_i ← cen_j
8         delete all child nodes of N_j from the R-tree
9      if r_min ≤ Δ < r_max then
10        for each instance of N_j do
11           compute the distance d(v, cen_j)
12           if d(v, cen_j) ≤ Δ then
13              take instances of N_j into X_i
14              c_i ← cen_j
15              delete all child nodes of N_j from the R-tree
16     else
17        i ← i + 1 and j ← i + 1
```

This algorithm can efficiently construct each *approximate set* for OPT-RR by reducing the number of evaluated *MBRs* based on their bounds $r_{\max}$ and $r_{\min}$. If the upper bounds of some nodes' *MBRs* are lower than Δ, these nodes can be pruned without complex computation (lines 4-6). On the other hand, the algorithm searches the child nodes when the lower bounds of the nodes' *MBRs* are higher than Δ. With this algorithm, OPT-RR is constructed efficiently by reducing search space.

In an *approximate set*, all the instances are replaced by a *key instance*. Therefore, the original probability distribution of an object is represented by a new probability distribution over *key instance* that this object has likelihood to appear at. In particular, the possible worlds semantics are retained by OPT-RR so that the synopses are quite suitable for probabilistic spatial queries. We answer the spatial queries over multidimensional uncertain databases based on OPT-RR, which will be discussed in the next section.

6 Approximate Aggregate Spatial Query Processing

As mentioned previously, a nice property of OPT-RR is that the approximate aggregate spatial queries can be naturally answered while insuring the best efficiency for a given error bound. In this section, we discuss how to answer approximate spatial queries based on OPT-RR.

6.1 Approximate Aggregate *k*NN Query

Due to multidimensional uncertainty of data objects and query objects, we are unable to exactly answer aggregate *k*NN queries over multidimensional uncertain objects (As analyzed in Lemma 1, the time complexity of aggregate *k*NN queries over multidimensional uncertain objects is #P-hard) in polynomial time with respect to data size. Aiming to solve the problem, we acquire approximate answers of the query with our approximate query framework.

In the first phase of the framework, we construct OPT-RR under the error bound for an approximate aggregate *k*NN query. Based on OPT-RR, all the instances of uncertain objects are mapped into *key instance* that are located at the centers of *MBRs* for the nodes of R-tree. Using OPT-RR, each multidimensional uncertain object is represented by a probability distribution for these *key instance*.

Before answering the approximate aggregate *k*NN query, we are able to compute the upper (or lower) bound of the aggregate distance of o_r^* to all the query objects in $\boldsymbol{Q}$ when the distance of o_r^* to each query object is maximum (or minimum). We prune some objects that are not the answers for the query definitely, which is summarized as follows.

Lemma 7. *For* $\forall \boldsymbol{o} \in \boldsymbol{D}$*, if* $\exists\{o_r^*, r = 1, ..., k\} h^{\top}(o_r^*, \boldsymbol{Q}) \le h^{\perp}(o, \boldsymbol{Q})$*, the multidimensional uncertain object o can be safely pruned in any aggregate kNN query, where* $\boldsymbol{h}^{\top}(o_r^*, \boldsymbol{Q})$ (*or* $\boldsymbol{h}^{\perp}(o, \boldsymbol{Q})$) *denotes the upper (or lower) bound of aggregate distance of an object to all the query object in* $\boldsymbol{Q}$.

Proof. Since *if* $\exists\{o_r^*, r = 1, ..., k\} h^{\top}(o_r^*, \boldsymbol{Q}) \le h^{\perp}(o, \boldsymbol{Q})$, $Pr(h(o_r^*, \boldsymbol{Q}) > h(o, \boldsymbol{Q})) = 0$. According to definition of aggregate *k*NN queries, there is no likelihood that the object *o* is ranked *i-th* ($1 \le i \le k$) for approximate *k*NN queries. Hence, the probability that *o* is nearer to query objects than o_i is equal to zero. Note that, *o* is not the query result so that it can be safely pruned without effect of the final result.

From the above lemma, we compute the bounds of the aggregate distance, and then achieve pruning with these bounds. Next, we find the *key instances* from which the minimum aggregate distance to all the query objects in $\boldsymbol{Q}$ is the smallest. The minimum aggregate distance function between Q and any *key instances c*, namely $h_{\min}(c, \boldsymbol{Q})$ is defined as

$$h_{\min}(c, \boldsymbol{Q}) = h\left(\min\left\{d(c, q_1)\right\}, \ldots, \min\left\{d\left(c, q_{|\mathbf{Q}|}\right)\right\}\right), \tag{2}$$

where $\min\{d(o_i, q_j)\}$ denotes the minimum distance between *c* and the *j-th* query object in $\boldsymbol{Q}$ ($1 \le j \le |\boldsymbol{Q}|$). For the Euclidean distance, $h_{\min}(c, \boldsymbol{Q})$ is equal to the aggregate distance from the *key instance c* to all the query objects. Using an R-tree to index objects, the *key instance* which are near to query objects are first retrieved so that any approximate *k*NN query can be answered by searching *key instance* of OPT-RR efficiently. Using OPT-RR, the number of possible positions of uncertain objects are only the number of these *key instance* while ensuring that the error is not more than Δ. As such, it is much easier to evaluate approximate aggregate distances with respect to query objects with our data synopses.

As mentioned previously, OPT-RR compacts all the instances (i.e., possible positions) of uncertain objects into the constant number of *key instance*} (i.e., *C* approximate positions). The aggregate distances between the *key instance* and query objects can be evaluated in constant time when the set of query objects $\boldsymbol{Q}$ is given. We then find a *key instance c* to which the aggregate distance to query objects is ranked *r-th* for each *r* ($1 \le r \le k$). The probability of *c* to have the minimum aggregate distance is computed as $p_c \prod_{i=1}^{r-1} (1 - p_i)$, where p_c denotes the probability that an object is located at this *key instance c* and p_i denotes the probability of other *key instance* whose aggregate distance with respect to query objects are smaller than that of *c* ($1 \le i \le r-1$). Thus, we can calculate the probability of the *key instance c* whose aggregate distance with respect to query objects is ranked *r-th* as $\frac{p_r}{1-p_r} \cdot Pr[rank(c) = r-1]$, where p_r represents any other *key instance* whose aggregate distance with respect to query objects is smaller than *c*, and $Pr[rank(c) = r-1]$ denotes the probability of *c* whose aggregate distance to query objects is ranked $(r-1)$-*th*.

Based on OPT-RR, we calculate the probability of any *key instance's* aggregate distance with respect to query objects to be ranked *r*-th. From the information, the objects which are most likely to be nearest to all uncertain query objects can be returned by evaluating the probabilities that these objects are ranked *r-th* for each *r* (i.e., $1 \le r \le k$ the probabilities are computed by U*k*ranks query [33]). In detail, the above process for answering approximate aggregate *k*NN queries based on our synopses is illustrated in Algorithm 2.

Algorithm 2. Approximate aggregate *k*NN query algorithm

Input:
number of returned objects k, error bound Δ, uncertain database $\boldsymbol{D}$ and the set of uncertain query objects $\boldsymbol{Q}$
Output:
result set $\boldsymbol{R}_{kNN}$

```
build OPT-RR under the bound Δ using Algorithm 1
r ← 1
prune objects in D
while r ≤ k do
    for each key instance c in C do
        compute d(c, q_j) for ∀q_j ∈ Q
        evaluate the probabilities of c to be ranked at r
        find c whose probability to be ranked at r is largest
        for each object o in D do
            if o is located at c with the highest probability then
                take o into R_kNN
r ← r + 1
```

Lastly, we analyze the time cost for answering the approximate aggregate *k*NN queries over uncertain databases. As illustrated in the above algorithm, evaluating the probability of an instance of any uncertain object to be closest to the query objects ranked at first k positions requires $O(k)$ time complexity. Because the number of instances is N, the computing cost is solved in $O(kN)$. In addition, the time complexity of retrieving k objects from n uncertain objects is $O(k \log n)$. Thus, the query cost is $O(k(N + \log n))$ based on OPT-RR. Since the time complexity for building OPT-RR is $O(N^2)$, evaluating the approximate *k*NN query problem can be solved in time $O(k(N + \log n) + N^2)$. Due to $k \ll n$, approximate aggregate *k*NN queries over multidimensional uncertain databases can be answered in polynomial time with respect to the size of database under the given error bound Δ.

6.2 Approximate Aggregate Range Query

Given an error bound Δ, OPT-RR is constructed to guarantee any approximate range query is answered under Δ according to Lemma 3.

For multiple query objects, the aggregate queries will retrieve the set of uncertain objects whose aggregate distances are most likely to be within distance ε (i.e., the range) with respect to the set of query objects in $\boldsymbol{Q}$. Compared to the case of one query object, we need to calculate the probability that any uncertain object is located within the range for each query object in $\boldsymbol{Q}$ so that it is more complex to answer the aggregate range queries. According to the definition of aggregate range queries, the query retrieves the objects whose aggregate distances with respect to all the query objects in $\boldsymbol{Q}$, denoted that the distances are not more than ε with the highest probabilities.

Given the error bound Δ, the approximate aggregate range queries are answered with the lowest number of *key instances* so that the efficiency of queries can be optimized based on OPT-RR. Thus, we build OPT-RR to represent positions of uncertain objects in $\boldsymbol{U}^d$ under the given Δ. Any uncertain object o's possible positions are represented by the *key instances* approximately. For the aggregate function of aggregate range queries,

$$h\left(o,\boldsymbol{Q}\right)=h\left(d\left(o,q_1\right),d\left(o,q_2\right),\ldots,d\left(o,q_{|\boldsymbol{Q}|}\right)\right). \tag{3}$$

We evaluate the aggregate distance of any uncertain object o to all the query objects in $\boldsymbol{Q}$. When the aggregate distances between *key instances* of o are located within the ε range of a query object q, there exists likelihood that objects are results for this query object. Then we calculate the probability of any object o that is a result for one of uncertain query objects q,

$$\begin{aligned} &Pr[d(o,q)\le\varepsilon+\Delta]=\Sigma_{p_{ij}}\\ &\left(p_{ij}=\begin{cases}p_i\cdot p_j & iff\ d\left(c_i,w\right)\le\varepsilon+\Delta\\ 0 & iff\ d\left(c_i,w\right)>\varepsilon+\Delta\end{cases}\right), \end{aligned} \tag{4}$$

where c_i and w denote *key instances* and the instance of q_j whose probabilities are p_i and p_j respectively. Since $d(c_i, w) \leq \varepsilon + \Delta$ is the approximate condition of range queries, the precision is guaranteed by setting a proper error bound Δ. In the case of multiple query objects, we will compute the probability that o is a result for each query object by Eq.(4) and find one query object q such that $Pr\,(d\,(o, q) \leq \varepsilon + \Delta)$ is highest. Then we take the object into the result set of q. The key steps are illustrated in Algorithm 3.

Algorithm 3. Approximate aggregate range query algorithm

Input:
query range ε, error bound Δ, uncertain database $\boldsymbol{D}$ and the set of uncertain query objects $\boldsymbol{Q}$
Output:
result set R_{range}

```
1  build OPT-RR under the bound Δ using Algorithm 1
2  for each query object q in Q do
3     for each uncertain object o based on OPT-RR do
4        if h (o, q) ≤ ε then
5           take o into R_range of q
6     else
7        compute Pr [h (o, q) ≤ ε + Δ]
8        find the highest value of Pr [h (o, q) ≤ ε + Δ]
9        take o into R_range of q
```

As described in the above algorithm, approximate aggregate range queries can be answered in polynomial time with respect to the size of *key instances* for OPT-RR. Furthermore, the queries can satisfy the precision requirement by setting a proper error bound Δ. Note that, we can achieve efficiency optimization while insuring the high query precision by constructing OPT-RR under Δ.

As mentioned previously, aggregate kNN and range queries over multidimensional uncertain objects with our data synopses can be answered in polynomial time while insuring the high precision. Besides, in the refinement phase, the candidate results should be verified with task-specific requirements so we do not discuss this in detail.

7 Experiments

7.1 Experiment Settings

We used two real datasets (‘‘Los Angeles’’ dataset and scientific real-world dataset) and one synthetic dataset with different distributions and dimensionality. Moreover, we used an R-tree to index these datasets respectively. We next describe the three datasets in detail.

***LA* Dataset**: ‘‘Los Angeles’’ (*LA*) dataset, a two-dimensional real dataset available from [34], with 60*K* geographical objects described by ranges of longitudes and latitudes, where each object is represented by a set of 20-50 unequal instances located within the corresponding range.

***SCI* Dataset**: This dataset which was used in previous work [6] is a set of 1,600 objects, where each object consists of 48 ten-dimensional instances. Each instance corresponds to a set of environmental sensor measurements of one single day (once per 30 minutes) that consists of ten dimensions (attributes): temperature, humidity, speed and direction of wind w.r.t. degree and sector, as well as concentrations of *CO*, SO_2, *NO*, NO_2, and O_3. These attributes are normalized within the interval [0, 1] to give each attribute the same weight.

***SYN* Dataset**: The synthetic dataset is composed of 100*K* objects, each consisting of 20 instances uniformly distributed in the uncertain region which is specified by a three-dimensional hyperrectangle.

We calculate the distance between two arbitrary points in *LA* dataset, *SCI* dataset and *SYN* dataset, and distance between two objects can be measured using Euclidean distance. For each dataset, we standardize the distances between any points within the d-dimensional space $\boldsymbol{U}^d$ (d represents dimensionality of each dataset) into [0, 1000].

Alternative Techniques Considered. We compare our techniques of constructing data synopses with the other data reduction methods, namely probabilistic histograms [12], which compacts interactive uncertain data into a bucket and choose the expectation of these uncertain data as the representative value, and the Uk-means method [13], which is able to reduce the number of uncertain data by clustering similar uncertain data into a cluster.

Using these methods, each uncertain object is represented by a bucket or a center of the cluster approximately. Given an error bound Δ, however, probabilistic histograms and U*k*-means methods are unable to ensure the maximum error for all the objects is less than Δ. Furthermore, we compare the performance of approximate spatial queries based on synopses constructed by these reduction methods.

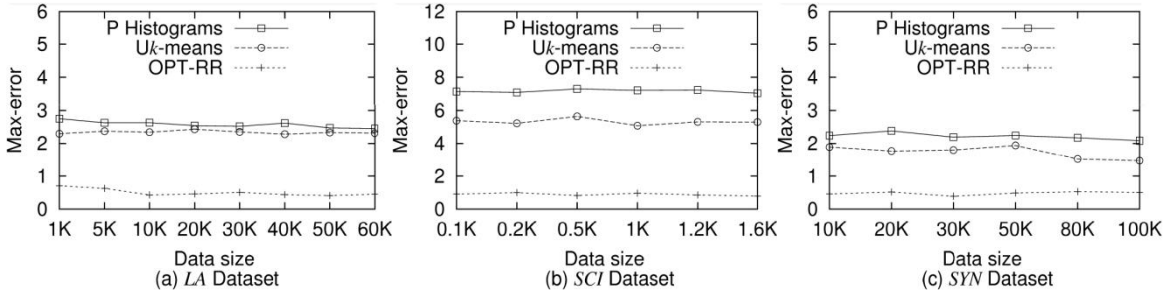

Fig. 5. The maximum error of different data reduction methods for three datasets

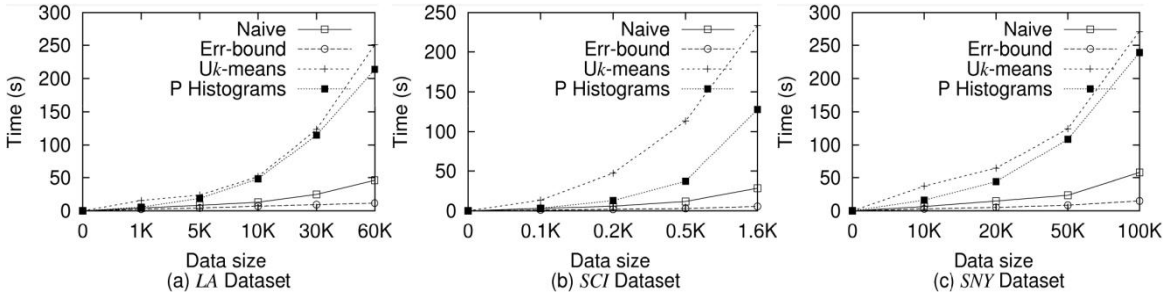

Fig. 6. The execution time of different construction methods for three datasets

Table 5. The scalability of different data synopses

	Number/Size			Reduction ratio		
	Clusters	OPT-RR	Histograms	Clusters	OPT-RR	Histograms
10K	673	368	460	6.57%	3.6%	4.48%
20K	647	372	861	3.16%	1.81%	4.2%
30K	626	344	1404	2.03%	1.12%	4.57%
40K	679	352	1981	1.66%	0.86%	4.84%
50K	618	372	2414	1.2%	0.73%	4.71%
60K	605	362	3015	0.98%	0.59%	4.9%
70K	644	334	3419	0.9%	0.53%	4.77%
80K	615	378	3849	0.75%	0.46%	4.7%
90K	628	317	4261	0.68%	0.34%	4.62%
100K	643	327	4604	0.63%	0.32%	4.5%

7.2 Experiments on Synopses

We evaluate the error and efficiency of different data reduction methods for building data synopses. We analyze the error model of data reduction techniques (i.e., probabilistic histograms [12], U*k*-means [13], and OPT-RR) for maximum error function on different datasets. We set the error bound Δ to be 1 for constructing OPT-RR. We vary the size of dataset and contrast the maximum error (Max-error, i.e., the deviation between any uncertain object and the representation of data synopses) for all objects in each dataset based on OPT-RR with those based on probabilistic histograms and U*k*-means methods. Max-error is defined as

$$\text{Max} - \text{error} = max\left\{\sum_{i=1}^{I} p_{v_i} \cdot d\left(v_i, c\right)\right\}, \tag{5}$$

where I denotes the number of instances for any object. As shown in Fig. 5, the Max-error based on OPT-RR is much lower than those based on others for any dataset. The reason is that we ensure that Max-error based on OPT-RR is not more than the given error bound τ. Furthermore, we cannot ensure that the error for representing objects based on probabilistic histograms or Uk -means is lower than τ. Thereby, OPT-RR is more suitable for reducing multidimensional uncertain data.

We also investigate the effect of the dataset size n on the time cost of the different algorithms to construct different data synopses. We set the error bound Δ to be 1 for constructing OPT-RR. Fig. 6 illustrates the results by varying n for different datasets. The time cost of building OPT-RR is much smaller than those of building other synopses. Besides, error bounds for *MBR* (Err-bound) based algorithm is more efficient than the naive solution algorithm (Naive). And this algorithm increases steady along with increasing n because it can safely reduce the search space.

Next, we study the effect of the dimensionality on errors for the various methods. We vary dimensionality of *SCI* dataset from 1 to 10. We generate the probability of each instance for any uncertain object with uniform or Zipf distribution, which is shown in Fig. 7(a) or Fig. 7(b) respectively. The OPT-RR is more precise than other synopses for representing uncertain objects. Especially, for the case of high-dimensionality, OPT-RR performs much better than the others. The reason is that OPT-RR is constructed for multidimensional data synopses.

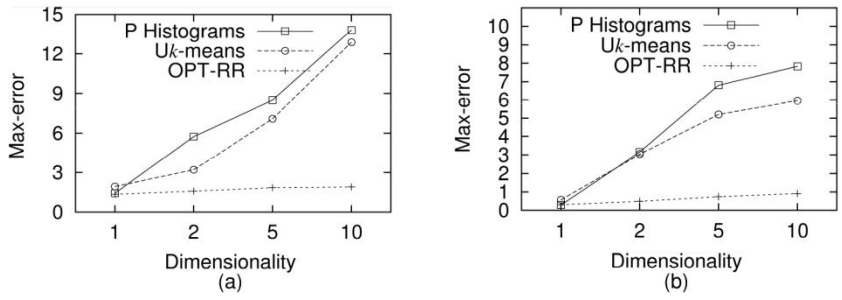

Fig. 7. Effect of dimensionality on maximum error

Finally, we test the scalability of different data synopses. We set the error bound Δ to be 1. Table 5 demonstrates the size of different data synopses and their compression ratio. Columns I and II show the number of clusters and the size of OPT-RR, and column III gives the size of probabilistic histograms. It is obvious that the size of OPT-RR is smaller than the others. The reason is that OPT-RR achieves the optimal reduction ratio. Moreover, the scalability of OPT-RR representation is better since the size of OPT-RR is stable with the increasing size of the dataset. Scalability is of great importance when managing large-scale uncertain databases. Therefore, our proposed synopses is highly suitable for large-scale uncertain databases.

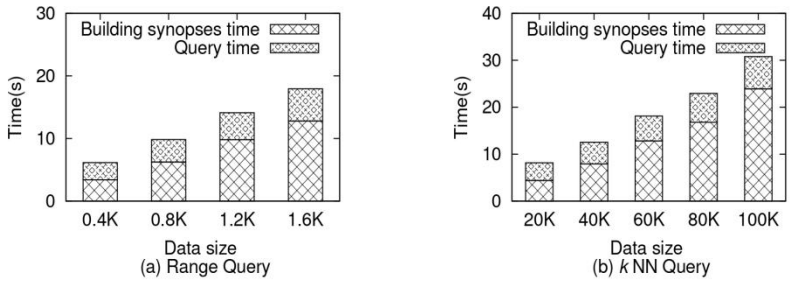

Fig. 8. Comparing building synopses time with query time

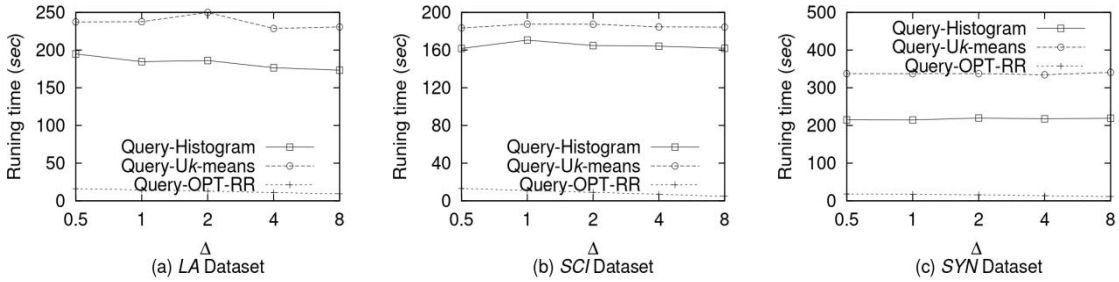

Fig. 9. The time efficiency of approximate spatial queries based on different synopses

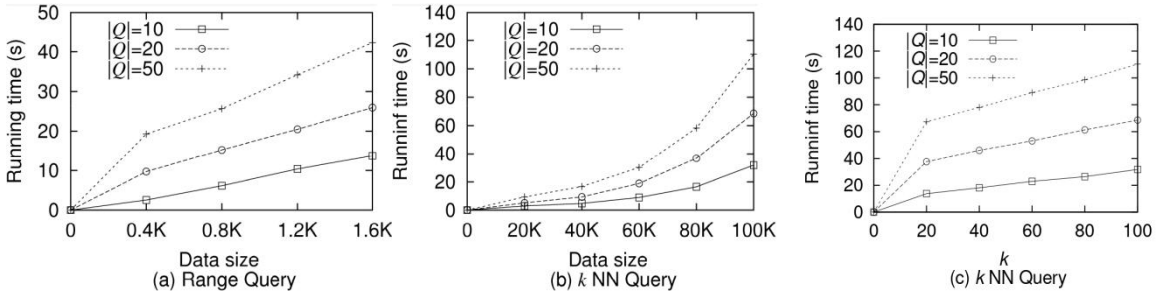

Fig. 10. The time efficiency of approximate spatial queries queries on varying $\boldsymbol{Q}$

7.3 Experiments on Queries

In the subsection, we evaluate the performance of answering approximate aggregate *k*NN queries and range queries based on different synopses that are built with probabilistic histograms, U*k*-means and OPT-RR over different datasets respectively. We set k to 50 for *k*NN queries, set ε to be 2 for range queries and set $|\boldsymbol{Q}|$ to be 20. Moreover, we run aggregate *k*NN queries over the *LA* dataset or the *SYN* dataset and run aggregate range queries over the *SYN* dataset.

At first, we compare the building synopses time with the query time. Along with increasing the dataset size, the building synopses time of OPT-RR and the time of answering our queries based on the synopses increase for different datasets as shown in Fig. 8.

We also analyze the time efficiency of approximate aggregate queries based on different synopses for three datasets. As illustrated in Fig. 9, the efficiency of the approximate spatial queries based on OPT-RR is the best for all the datasets. It can be explained that OPT-RR ensures the efficiency for answering approximate aggregate queries under a given error bound, which is in accordance with the analysis of Lemma 2 and Lemma 3.

Next, we evaluate the performance of our data synopses for answering approximate range queries and *k*NN queries. Aiming at it, we first vary the size of query object set $\boldsymbol{Q}$ and test the effect of the dataset size on the running time. The results are illustrated in Fig. 10(a) and Fig. 10(b). It is obvious that more running time for both *k*NN queries and range queries will be cost with increasing the dataset size under different $|\boldsymbol{Q}|$ values. This demonstrates that these queries are answered based on OPT-RR in polynomial time with respect to the size of datasets.

Lastly, we study the effect of k for answering approximate aggregate *k*NN queries on the running time over the *SYN* dataset. We vary k from 20 to 100 and set different $|\boldsymbol{Q}|$ values. As shown in Fig. 10(c), the time cost approximates to the linear growth along with increasing k. The results confirm that we can solve the problem in polynomial time using our proposed synopses OPT-RR while guaranteeing that the error is not more than the given error bound.

8 Conclusion

In this paper, we have proposed a framework for answering approximate aggregate spatial queries over multidimensional uncertain objects. In this framework, we first have presented a new sort of data synopses, called PWS for multidimensional uncertain objects. Our data synopses have reduced the complexity of the possible world model and retained the possible world semantics for the original data objects. Then we have answered approximate spatial queries based on OPT-RR which is a kind of optimal PWS. Using OPT-RR to compact original databases, we have returned the approximate results for aggregate spatial problems over multidimensional uncertain objects in polynomial time while the error bound is not more than a given error bound Δ. Finally, we have identified the efficiency and precision of the proposed methods through the theoretical analysis and extensive experiments by comparing the existing methods with real and synthetic datasets. Our work can be used to efficiently solve aggregated spatial queries for multidimensional uncertain data in many domains, such as Location based services.

References

[1] S. Feng, B. Glavic, A. Huber, O. Kennedy, Efficient Uncertainty Tracking for Complex Queries with Attribute-level Bounds, in: Proc. 2021 The ACM Special Interest Group on Management of Data (SIGMOD), 2021.
[2] Y. Chen, H. Duan, D. Wang, Y. Liu, C. Chang, SERDA: Secure Enhanced and Robust Data Aggregation Scheme for Smart Grid, Journal of Computers 35(5)(2024) 73-90.
[3] S. Feng, B. Glavic, O. Kennedy, Efficient Approximation of Certain and Possible Answers for Ranking and Window Queries over Uncertain Data, in: Proc. 2023 the 49th International Conference on Very Large Data Bases (VLDB), 2023.
[4] M. Chung, S. J. Hyun, W-K Loh, Efficient exact *k*-flexible aggregate nearest neighbor search in road networks using the M-tree, The Journal of Supercomputing (78)(2022) 16286–16302.
[5] J. Bernad, C. Bobed, E. Mena, Uncertain Probabilistic Range Queries on Multidimensional Data, Information Sciences (537)(2020) 334-367.
[6] T. Bernecker, H.P. Kriegel, N. Mamoulis, M. Renz, A. Zuefle, Scalable Probabilistic Similarity Ranking in Uncertain Databases, IEEE Transactions on Knowledge and Data Engineering (22)(9)(2010) 1234-1246.
[7] N. Dalvi, D. Suciu, Efficient query evaluation on probabilistic databases, The VLDB Journal (16)(2007) 523-544.
[8] E. Kushilevitz, R. Ostrovsky, Y. Rabani, Efficient search for approximate nearest neighbor in high dimensional spaces, in: Proc. 1998 ACM Symposium on Theory of Computing (STOC), 1998.
[9] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, A. Wu, An optimal algorithm for approximate nearest neighbor searching, in: Proc. 1994 ACM-SIAM Symposium of Discrete Algorithms, 1994.
[10] N. Koudas, B. Ooi, T. Lee, R. Zhang, Approximate NN queries on streams with guaranteed error/performance bounds, in: Proc. 2004 International Conference on Very Large Data Bases (VLDB), 2004.
[11] G. Cormode, A. Deligiannakis, M. Garofalakis, A. McGregor, Probabilistic histograms for probabilistic data, in: Proc. 2009 International Conference on Very Large Data Bases (VLDB), 2009.
[12] G. Cormode, M. Garofalakis, Histograms and wavelets on probabilistic data, in: Proc. 2009 IEEE International Conference on Data Engineering (ICDE), 2009.
[13] B. Kao, S. D. Lee, F.K.F. Lee, D.W. Cheung, W.S. Ho, Clustering uncertain data using voronoi diagrams and r-tree index, IEEE Transactions on Knowledge and Data Engineering (22)(9)(2010) 1219-1233.
[14] H. Liu, X. Zhang, X. Zhang, Q. Li, X.-M. Wu, RPC: Representative possible world based consistent clustering algorithm for uncertain data, Computer Communications (176)(2021) 128-137.
[15] Y. Liu, Z. Liu, S. Li, Y. Guo, Q. Liu, G. Wang, Cloud-Cluster: An uncertainty clustering algorithm based on cloud model, Knowledge-Based Systems (263)(2023) 110261.
[16] S. Alipour, Approximation Algorithms for Probabilistic k-Center Clustering, in: Proc. 2020 IEEE International Conference on Data Mining (ICDM), 2020.
[17] S. Feng, A. Huber, B. Glavic, O. Kennedy, Uncertainty Annotated Databases - A Lightweight Approach for Approximating Certain Answers, in: Proc. 2019 The ACM Special Interest Group on Management of Data (SIGMOD), 2019.
[18] S. Alipour, Improvements on approximation algorithms for clustering probabilistic data, Knowledge and Information Systems 63(10)(2021) 2719-2740.
[19] C. Ré, D. Suciu, Approximate lineage for probabilistic databases, in: Proc. 2008 International Conference on Very Large Data Bases (VLDB), 2008.
[20] Q. Liu, X. Lian, L. Chen, Probabilistic Maximum Range-Sum Queries on Spatial Database, in: Proc. 2019 International Conference on Advances in Geographic Information Systems (SIGSPATIAL), 2019.
[21] K. Bavi, X. Lian, Range-constrained probabilistic mutual furthest neighbor queries in uncertain databases, Knowledge and Information Systems 65(2023) 2375-2402.

[22] L. Chen, Y. Gao, A. Zhong, C.S. Jensen, G. Chen, B. Zheng, Indexing metric uncertain data for range queries and range joins, The VLDB Journal 26(4)(2017) 585-610.
[23] Y. Ma, X. Meng, Set similarity join on massive probabilistic data using MapReduce, Distributed and Parallel Databases 32(2014) 447-464.
[24] G. Xiao, K. Li, K. Li, X. Zhou, Efficient top-(*k*,l) range query processing for uncertain data based on multicore architectures, Distributed and Parallel Databases 33(2015) 381-413.
[25] X. Wang, D. Shen, G. Yu, Uncertain top-*k* query processing in distributed environments, Distributed and Parallel Databases 34(2016) 567-589.
[26] D. Amagata, Y. Sasaki, T. Hara, S. Nishio, Probabilistic nearest neighbor query processing on distributed uncertain data, Distributed and Parallel Databases 34(2016) 259-287.
[27] P. Moutafis, F. García-García, G. Mavrommatis, M. Vassilakopoulos, A. Corral, L. Iribarne, Algorithms for processing the group K nearest-neighbor query on distributed frameworks, Distributed and Parallel Databases 39(2021) 733-784.
[28] Y. Tao, X. Xiao, R. Cheng, Range search on multidimensional uncertain data, ACM Transactions on Database Systems (TODS) 32(3)(2007) 15.
[29] L. Shou, X. Zhang, G. Chen, Y. Gao, K. Chen, MUD: Mapping-based query processing for high-dimensional uncertain data, Information Sciences 198(2012) 147–168.
[30] M. Tang, F. Li, Scalable Histograms on Large Probabilistic Data, in: Proc. 2014 Proceedings of the 20th ACM international conference on Knowledge discovery and data mining (SIGKDD), 2014.
[31] Y. Zhang, H. Wang, L. Yang, J. Li, Efficient histogram-based range query estimation for dirty data, Frontiers of Computer Science 12(5)(2018) 984-999.
[32] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to algorithms, The MIT Press, 2009.
[33] M.A. Soliman, I.F. Ilyas, Top-*k* query processing in uncertain databases, in: Proc. 2006 IEEE International Conference on Data Engineering (ICDE), 2006.
[34] Topologically integrated geographic encoding and referencing (tiger) system. http://www.census.gov/geo/www/tiger/ (accessed 28.09.2020).