Evaluating an Index Method for Probabilistic Range Queries on Gaussian Distributions

Tingting DONG†, Xi GUO††, Chuan XIAO†, and Yoshiharu ISHIKAWA†

† Graduate School of Information Science, Nagoya University
†† Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong
E-mail: †dongtt@db.itc.nagoya-u.ac.jp, ††guoxi022@gmail.com, ††{chuanx,y-ishikawa}@nagoya-u.ac.jp

Abstract Recently uncertain data is attracting more and more attention in many fields and has become a major topic in the database research community. In this paper, we develop and evaluate our preliminary work, an index method for probabilistic range queries on Gaussian distributions. This method assumes that uncertain data items stored in the database are represented by multi-dimensional Gaussian distributions (Normal distributions), while the query object can be a point or a Gaussian distribution. We propose an index structure and several filtering techniques to support probabilistic range queries. In this work, we conduct experiments using both synthetic data and real data and examine the efficiency and effectiveness of this index method.

Key words uncertain data, probabilistic databases, range queries, spatial databases

1. Introduction

In recent years, uncertain data is gaining more and more attention in the database community and has involved a large variety of real-world applications, ranging from mobile robotics and sensor networks to location-based service. Uncertainty can be inherent properties of the data caused by measurement limitations and noises (e.g., Gaussian errors in GPS readings), or may be introduced to preserve the privacy of the source data.

For instance, in the area of location-based mobile advertising, operators typically provide services such as the delivery of mobile coupons or discounts to nearby mobile users using their location information (e.g., The Coupons App). The exact current location of users may not be available due to privacy preservation or delayed updates from users. In this case, a query like “find customers currently in the downtown area” cannot be fully evaluated and answered.

As another example, consider a self-navigated mobile robot moving in an environment as shown in Fig. 1. The robot builds a map of the environment by observing nearby landmarks using devices such as sonars and laser range finders. Due to the inherent limitation of measurement accuracy and unavoidable signal noises, the information acquired from measuring devices (e.g., the location of a landmark) is always not precisely correct. At the same time, the moving robot also conducts probabilistic localization [18] to estimate its location autonomously by integrating its movement history and the landmark information. This can result in imprecision in the location information of the robot, too.

During the movement, the robot may request information about nearby landmarks and issue a query such as “find landmarks within 5 meters from my current location”. In the traditional spatial database setting, this kind of query can be easily answered by performing a range query with the range specified as 5 meters. Nevertheless, it is difficult to process this query exactly in this situation, because locations of both the query object (i.e., the robot) and the target objects (i.e.,
nearby landmarks) are inexact. And if the obtained imprecise data is directly used to answer queries, it may result in erroneous answers and navigation failures.

![A motivating example](image)

2. Related Work

2.1 Uncertain Data Management

A number of approaches for managing uncertain data have been proposed. Early research primarily focuses on queries in a moving object database model [5], [13], [19], [21]. Cheng et al. classify several types of probabilistic queries including probabilistic range queries based upon uncertain data and present algorithms for solving them in [4]. In another study, Cheng et al. develop several solutions for probabilistic range queries [6]. However, they target the one-dimensional space only. Moreover, a range query processing method for the case where both target objects and a query object are imprecise is proposed in [3]. But they assume that each object exists within a rectangular area.

2.2 Spatial Data Indexing

The traditional spatial database has been well studied and many indexing methods have been proposed [1], [8], [12] to support spatial query processing. The well-known one is R-tree [8] and its extension R*-tree [1], which index objects by deriving their minimum bounding rectangle (MBR). TPR-tree [20] and TPR*-tree [17] are proposed to index moving objects. But none of them can be applied directly to index Gaussian objects directly for our problem.

2.3 Uncertain Data Indexing

In terms of probabilistic range queries in a multi-dimensional space, Tao et al. propose U-tree [16]. It is different from our tree here in that our tree indexes Gaussian distribution in the infinite space. As an index structure for Gaussian distributions, Gauss-tree is proposed for probabilistic identification query in [2]. What is problematic with the Gauss-tree lies in that it constructs its index structure based on the assumption that all Gaussian distributions are probabilistically independent in each dimension. In other words, each distribution axis of a Gaussian function should be parallel to a dimension axis. This imposes heavy restriction on the generality of the approach and the overall accuracy of the query result is limited.

In our preliminary work [10], we propose several query processing techniques for probabilistic range queries, assuming that the location of the query object is only uncertain and described by a Gaussian distribution, and target objects in the database are multi-dimensional points and are managed by a conventional spatial index such as an R-tree. Moreover, in our precedent work [11] of this research, we also present an index method for Gaussian distributions. The approach proposed in [11] is consistent theoretically, but not easy to implement practically and is greatly affected by computational errors. In this paper, we present stronger query processing techniques and a novel index structure to solve the problem.

3. Problem Definition

Uncertain target objects here are assumed to follow multi-dimensional Gaussian distributions with different parameters. A **probabilistic range query (PRQ)** is to retrieve objects among them located within some specific range from the query object (a certain point or an uncertain object represented by a Gaussian distribution) with probabilities greater than a probability threshold. We define the problem in a \( d \ (d \geq 2) \)-dimensional space. The one-dimensional case will not be discussed here since it is exceptional and can be solved easily.

3.1 Gaussian Distributions

**Definition 1 (Gaussian objects).** The probability that an object \( o_i \in D \) is located at \( x_i \) is defined by a \( d \)-dimensional Gaussian probability density function
\[ p_i(x_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2}(x_i - \mu_i)^T \Sigma_i^{-1}(x_i - \mu_i) \right] \tag{1} \]

where \( D \) is a set of target objects, \( \mu_i \) is the mean of \( o_i \) and \( \Sigma_i \) is a \( d \times d \) covariance matrix. \( |\Sigma_i| \) is the determinant of \( \Sigma_i \) and \( \Sigma_i^{-1} \) is the inverse matrix of \( \Sigma_i \). \( x_i^T \) represents a transposition of a vector \( x_i \).

### 3.2 Definition of Queries

In this paper, we consider two types of query objects:

1. The query object is a fixed point, namely, \( q = (x_q^1, x_q^2, \ldots, x_q^d)^t \).

2. The query object follows a \( d \)-dimensional Gaussian distribution, namely,

\[ p_q(x_q) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_q|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2}(x_q - \mu_q)^T \Sigma_q^{-1}(x_q - \mu_q) \right] \]

where \( \mu_q \) is the mean, and \( \Sigma_q \) is a \( d \times d \) covariance matrix.

#### 3.2.1 Probabilistic Range Query with Point Query Object (PRQ-P)

**Definition 2** (PRQ-P). Given a query object \( q \) represented by a vector \( q \), a distance threshold \( \delta \), and a probability threshold \( \theta \) (0 < \( \theta \) < 1), a probabilistic range query with point query object (PRQ-P for short) is defined as follows:

\[ \text{PRQ-P}(q, \delta, \theta) = \{ o_i | o_i \in D, \Pr(||x_i - q|| \leq \delta) \geq \theta \} \]

where \( ||x_i - q|| \) represents the Euclidean distance between \( x_i \) and \( q \).

\[ \Pr(||x_i - q|| \leq \delta) = \int \chi_d(x_i, q) \cdot p_i(x_i) dx_i \tag{2} \]

where \( \chi_d(x_i, q) = \begin{cases} 1 & \text{if } ||x_i - q|| \leq \delta \\ 0 & \text{otherwise} \end{cases} \tag{3} \)

is used to enforce the distance-based threshold.

![Fig. 2 - An illustration of PRQ-P query](image)

Fig. 2 shows an illustration of a PRQ-P query. The Gaussian object \( o_i \) exists in the space with a decreasing probability as spreading away from the center \( o_i \), i.e., the mean. The changing colors describe this diminishing trend of the probability. A PRQ-P query attempts to find Gaussian objects located near the query point with a high probability. Computing the probability using Eq. (2) corresponds to integrating the probability density function of \( o_i \) within the slash area around \( q \).

However, the integration in Eq. (2) is not in a closed-form and cannot be computed directly. To evaluate the probability, numerical integration (the Monte Carlo method) is employed actually. To be specific, the efficient importance sampling [14] approach can be used: generate \( x_i \) with a probability \( p_i(x_i) \), and increment the count when Eq. (3) is satisfied. Finally, we can get the integrated probability through dividing the count by the number of samples (e.g., 100000) generated. However, the Monte Carlo integration has an extremely high cost even though using the importance sampling approach. For this reason, we propose an effective approach to reduce the number of candidate objects.

#### 3.2.2 Probabilistic Range Query with Gaussian Query Object (PRQ-G)

**Definition 3** (PRQ-G). Given a query object \( q \) represented by a Gaussian distribution, a distance threshold \( \delta \), and a probability threshold \( \theta \) (0 < \( \theta \) < 1), a probabilistic range query with Gaussian query object (PRQ-G) is defined as follows:

\[ \text{PRQ-G}(q, \delta, \theta) = \{ o_i | o_i \in D, \Pr(||x_i - q|| \leq \delta) \geq \theta \} \]

where \( \Pr(||x_i - q|| \leq \delta) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_q| \Delta_d}} \int \exp \left[ -\frac{1}{2}(x_i - \mu_q)^T \Sigma_q^{-1}(x_i - \mu_q) \right] dx_i \tag{4} \)

is a thresholding function.

To compute the numerical integration in Eq. (4), although we can simply generate random numbers for the two Gaussian distributions \( p_i(x_i) \) and \( p_q(x_q) \) respectively, a more efficient method for handling this kind of numerical integration is shown in [11]. It constructs a \( 2d \)-dimensional Gaussian distribution by combining two \( d \)-dimensional Gaussian distributions together.

### 4. Filtering Based on Approximated Regions

Evaluating the two types of queries defined in Eq. (2) and Eq. (4) requires “expensive” numerical integration. To reduce query processing cost, it is essential to reduce the number of candidate Gaussian objects which need numerical integration. In this section, we propose several filtering techniques based on a probability region (called \( \rho \)-Region) and its approximation (called bounding box) of a Gaussian distribution to prune as many non-candidate objects as possible.

#### 4.1 \( \rho \)-Region

**Definition 4** (\( \rho \)-region). Consider the integration of the probability density function \( p_i(x_i) \) over an ellipsoidal region \( ||x_i - o_i||^2 \leq \rho^2 \). Let \( r_p \) be the value of \( r \) for which the result of the integration exactly becomes \( \rho \):

\[ \int_{||x_i - o_i||^2 \leq r_p^2} p_i(x_i) dx_i = \rho. \]

We call the ellipsoidal region \( ||x_i - o_i||^2 \leq r_p^2 \) defined by \( r_p \) the \( \rho \)-region.
Nevertheless, it is costly to compute $\rho$-regions for arbitrary Gaussian distributions (with different $\mathbf{a}_i, \mathbf{\Sigma}_i$) directly. To cope with this problem, an approach that transforms the integration over an ellipsoidal region to an integration over a $d$-dimensional sphere region is proposed in [10]. To begin with, let us introduce the normalized Gaussian distribution defined by assigning $\mathbf{a}_i = \mathbf{0}$ and $\mathbf{\Sigma}_i = \mathbf{I}$ in Eq. (1).

$$p_{\text{norm}}(\mathbf{x}) = N(0, \mathbf{I}) = \frac{1}{(2\pi)^{d/2}} \exp \left[ -\frac{1}{2} \| \mathbf{x} \|^2 \right]$$

Based on this probability density function, we can derive the following property.

Property 1. Consider integration of $p_{\text{norm}}(\mathbf{x})$ over the region $\| \mathbf{x} \|^2 \leq r^2$, which is a sphere with the origin as its center and the radius $r$. For the given $\rho (0 < \rho < 1)$, let $\hat{r}_\rho$ be the radius with which the integration result becomes $\rho$.

$$\int_{\| \mathbf{x} \|^2 \leq \hat{r}_\rho^2} p_{\text{norm}}(\mathbf{x}) d\mathbf{x} = \rho. \quad (5)$$

For a given $\rho$, $r_\rho = \hat{r}_\rho$ holds.

The proof is shown in [10]. The property indicates that if the $r_\rho (= \hat{r}_\rho)$ value is calculated for a given $\rho$ value using Eq. (5), we can use it for our context using the equality in Eq. (6).

$$\begin{array}{|c|c|c|}
\hline
d & \rho & r_\rho \\
\hline
2 & 0.98 & 2.75 \\
\vdots & \vdots & \vdots \\
\hline
\end{array} \quad (\rho, r_\rho)-\text{Table}$$

That is, if a table like Fig. 3 is constructed beforehand (numerical integration is necessary), we can easily obtain the corresponding $r_\rho$ for the $\rho$ value in this table and hence derive the $\rho$-region. However, due to the ellipsoidal shape of the $\rho$-region, it is not suitable to be used for filtering processing. We will derive the bounding box which tightly bounds the $\rho$-region.

4.2 Deriving Bounding Box

Definition 5 (Bounding Box). Given the parameter $\rho (0 < \rho < 1)$, the rectangular region which tightly bounds the $\rho$-region of Gaussian object $\mathbf{a}_i$ is called the $\rho$-bounding box of $\mathbf{a}_i$, and represented by $bb_i(\rho)$. For simplicity, we sometimes omit $\rho$ and call it bounding box directly and abbreviate it to $bb_i$. \(\Box\)

Fig. 4 shows the image of the bounding box $bb_i$ in $j$-th dimension and $k$-th dimension. Let the width of the box from the object center $\mathbf{a}_i$ along the $j$-th dimension and $k$-th dimension be $w_j$ and $w_k$ respectively. The following property holds [10].

Property 2. The value of $w_j$ $(j = 1, 2, \ldots, d)$ is given as

$$w_j = \sigma_j r_\rho \quad (7)$$

where $\sigma_j$ corresponds to the standard deviation for the $j$-th dimension

$$\sigma_j = \sqrt{(\mathbf{\Sigma}_i)_{jj}}$$

where $(\mathbf{\Sigma}_i)_{jj}$ represents the $(j, j)$ entry of $\mathbf{\Sigma}_i$.

4.3 Filtering for PRQ-P Queries

4.3.1 Strategy 1: RR Method

Here we detail the idea of bounding box-based filtering techniques for the PRQ-P query. The first filtering processing approach is an extension of the rectlinear-region-based approach (RR) proposed in our paper [10], except that in [10] the target objects are certain points and the query object is a Gaussian distribution.

Case 1: $\theta < 0.5$.

Consider four kinds of target objects $\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, \mathbf{o}_4$ as shown in Fig. 5. First, let’s consider $\mathbf{o}_4$. Since the probability that $\mathbf{o}_4$ is located inside its ellipsoidal $\rho$-region is $\rho$, the probability that $\mathbf{o}_4$ is located outside $bb_i(\rho)$ region is definitely less than $1 - \rho$. Furthermore, given the line symmetry of a Gaussian distribution, the probability that $\mathbf{o}_4$ is located inside the sphere region of $q$ is at most $(1 - \rho)/2$. For example, suppose that $\rho = 20\%$, then $(1 - \rho)/2 = 40\%$ is the upper-bound probability. Hence, if $(1 - \rho)/2 = \theta$; i.e.,

$$\rho = 1 - 2\theta$$

is true, when $bb_i(\rho)$ and the sphere are disjoint (that is, connected or separated), the probability that the target object $\mathbf{o}_4$ is within the $\delta$ range of query object $q$ will be less than $\theta$. On the contrary, if $bb_i(\rho)$ and the sphere query region have intersection, this probability is possible to reach $\theta$.

For $\mathbf{o}_1$ and $\mathbf{o}_3$, since their mean locations are inside the spherical query region, it is obvious that their bounding boxes will intersect with the query region. Therefore, we can add them to the candidate list without deriving their bounding boxes. On the other hand, we have to derive the bounding box $bb_i(\rho)$ of $\mathbf{o}_2$ to check whether it intersects with the spherical region. If they have intersection, then $\mathbf{o}_2$ will be selected as a candidate object.

Moreover, for all candidate objects, we also derive their bounding boxes of $\theta$-regions by letting $\rho = \theta$. If the query
region contains the $\theta$-valued bounding box as $o_3$, this object is undoubtedly a query result. We will return this kind of target objects as result objects directly without "expensive" numerical integration.

**Case 2:** $\theta \geq 0.5$. Let $\rho = \theta$.

![Fig. 6 RR Method ($\rho = \theta, \theta \geq 0.5$)](image)

We show our idea in Fig. 6. For the integrated probability that a target object exists within the spherical query region to reach 0.5, the mean location of a target object should locate inside the query region. Otherwise, the probability is definitely less than 0.5. In this way, $o_2$ and $o_4$ can be pruned. Similarly, $o_3$ can be returned as a result object without numerical integration.

**4.3.2 Strategy 2: OR Method**

In [10], the oblique-region-based approach (OR) method is proposed besides the RR method. This method can also be extended for our query processing. The idea is shown in Fig. 7.

![Fig. 7 OR Method](image)

Consider the rectangle paralleled to the axes of the $\rho$-region ellipsoid. The distance between the rectangle and the $\rho$-region is at least $\delta$. Obviously when $o_i$ is located inside the $\rho$-region (with a probability $\rho$), the distance between $o_i$ and $q_2$ will be more than $\delta$. If $\rho = 90\%$, the probability that $o_i$ will be located outside the rectangle is at most 10%. Furthermore, given the line symmetry, the probability that $o_i$ exists within $\delta$ range of $q_2$ is no more than 5%. This means that we can obtain the filtering condition by letting $(1-\rho)/2 = \theta$, i.e., $\rho = 1 - 2\theta$. On the other hand, $o_i$ becomes a candidate object for query $q_1$.

Since it is difficult to determine the relation between a point and an oblique rectangular region, the oblique region is transformed into an axis-parallel rectangle actually. The transformation is implemented by deriving the corresponding point $y_i$ for a given $d$-dimensional point $x_i$ which satisfies $x_i = Ey_i$. Here $E$ is a diagonal matrix consisting of the eigenvectors of $\Sigma^{-1}$. The proof of this property is shown in [10]. In this work, the OR method is applied to further refine candidate objects returned by the RR method.

**4.4 Filtering for PRQ-G Queries**

For PRQ-G queries, we obtain both of their bounding boxes of $\rho$-regions. As shown in Fig. 8, consider the situation that the distance between the bounding boxes of two $\rho$-region is exactly $\delta$. Since $q$ and $o_i$ are located inside their $\rho$-regions respectively both with probability $\rho$, the probability that each of them exists within individual $\rho$-region at the same time is $\rho^2$, assuming that they are independent in the space. In this case, obviously the distance between $q$ and $o_i$ is very likely to be larger than $\delta$. Specifically, the distance between $q$ and $o_i$ becomes less than $\delta$ with a probability at most $1 - \rho^2$.

For a given probability threshold $\theta$ of the query, letting $1 - \rho^2 = \theta$, that is, $\rho = \sqrt{1 - \theta}$, we can compute $\rho$. For example, if $\theta = 5\%$, then $\rho = \sqrt{1 - 0.05} = 0.9747$. Construct the bounding boxes of $\rho$-regions dynamically for $q, o_i$ with the $\rho$ value. And we can exclude $o_i$ from the candidate list if the minimum distance between $bb_i(\rho)$ and $bb_q(\rho)$ is more than $\delta$.

![Fig. 8 Filtering for PRQ-G Queries](image)

Moreover, if the maximum distance of $bb_i(\rho)$ and $bb_q(\rho)$ is less than $\delta$, assigning $\rho' = \theta$ (i.e., $\rho = \sqrt{1 - \theta}$) will guarantee that the target object $o_i$ is located inside the $\delta$ range from the query object $q$ with a probability greater than $\theta$. Hence, $o_i$ can be validated as a result object directly.

To efficiently process queries over databases consisting thousands of or millions of target objects, we propose a dynamic index structure which stores bounding boxes of all objects instead of deriving their bounding boxes on-the-fly. We will describe this index structure in the next section.

**5. Index Structure**

**5.1 Overall Index Structure**

The overall index structure is a balanced hierarchical tree. Entries in leaf nodes contain target Gaussian objects in the form of $o_i = (id_i, o_i, \Sigma_i, bb_i)$, where $id_i$ is the object id, $o_i, \Sigma_i$ are the mean value (average location) and the covariance
matrix of the Gaussian distribution, and \( \text{bb}_i \) is the bounding box of \( \rho \)-region for \( \alpha_i \). In a non-leaf node, an entry contains a pointer to a subtree and a bounding box that encloses the leaf bounding boxes or other internal bounding boxes in that subtree.

As discussed in Section 4.2, for an object \( \alpha_i \) centered at \( \langle x'_1, \ldots, x'_d \rangle \) in the \( d \)-dimensional space, the bounding box \( \text{bb}_i \) of \( \alpha_i \) is a rectangle parameterized with \( r_p \). Its extent (i.e., left bound and right bound) in \( j \)-th dimension can be represented as

\[
\text{bb}_i^j = [x'_j - w'_j, x'_j + w'_j] = [x'_j - \sigma'_j r_p, x'_j + \sigma'_j r_p].
\]

We denote \( \text{bb}_i^j \) as the bounding interval of the bounding box \( \text{bb}_i \) in the \( j \)-th dimension. Specifically, \( \text{bb}_i \) is represented as

\[
\text{bb}_i = (\langle x'_1, \sigma'_1 \rangle, \ldots, \langle x'_d, \sigma'_d \rangle).
\]

In order to achieve best filtering performance, a leaf bounding box should tightly enclose its child bounding boxes. The challenge is that target objects always have different co-variance matrices, and their bounding boxes can scale up or down in different rates (i.e., different standard deviations) according to Eq. (7). So the left bound or right bound of the bounding box of a leaf node is determined by different child target objects in different \( r_p \) values.

If \( r_p \) is within the range \((0, r_1]\), the left bound of bounding box \( \text{bb} \) in \( j \)-th dimension is determined by the object \( \alpha_1 \), while the object \( \alpha_2 \) turns out to be the left bound object of \( \text{bb} \) if \( r_p \) is in \([r_1, +\infty) \). Also, when \( r_p \) increases over \( r_2 \), the right bound object of \( \text{bb} \) changes from \( \alpha_1 \) to \( \alpha_2 \). In Fig. 11, the upper bold polyline illustrates the left side of the bounding interval of the bounding box \( \text{bb} \), while the lower bold polyline shows the right side of that. For this purpose, a bounding box is represented by several combined segments, each of which has a different left bound or right bound on certain interval value of \( r_p \), corresponding to the polyline in Fig. 11.

In our implementation setting, we allow users to specify a query probability range \([\theta_{\min}, \theta_{\max}]\). Then \( r_p \) is actually within \([\theta_{\min}, \theta_{\max}]\). In this way, the overall index structure can be more compact and more efficient for query processing. In \( j \)-th dimension, the left bound of a (both leaf and non-leaf) node bounding box is in the form of

\[
\text{bb}_i^j = (\langle x'_1, \sigma'_1, [r_{\min}, r_1]\rangle, \ldots, \langle x'_j, \sigma'_j, (r_1, r_{\max}]\rangle)
\]

Now we can obtain the corresponding \( j \)-th dimensional bounding interval of the bounding box for objects \( \alpha_1, \alpha_2, \alpha_3 \) and \( \alpha_4 \) illustrated in Fig. 10 and Fig. 11, and derive the entry in \( j \)-th dimension \( \text{bb}_i^j \) as

\[
\text{bb}_i^j = (\langle x'_1, \sigma'_1, [r_{\min}, r_1]\rangle, \langle x'_3, \sigma'_3, (r_1, r_{\max}]\rangle)
\]

\[
\text{bb}_i^j = (\langle x'_4, \sigma'_4, [r_{\min}, r_2]\rangle, \langle x'_2, \sigma'_2, (r_2, r_{\max}]\rangle)
\]

5.2 Filtering at Non-Leaf Level

5.2.1 Processing PRQ-P Queries

Consider filtering on the non-leaf node for a PRQ-P query as shown in Fig. 12 (\( \theta < 0.5 \)). Assume that the sphere centered at \( q \) with the radius \( \delta \) is exactly contiguous with \( \text{bb}(\rho) \). As discussed in Section 4.3, if \( \rho = 1 - 2\theta \), among objects \( \alpha_1, \alpha_2 \) and \( \alpha_3 \) inside \( \text{bb}(\rho) \), none of them can satisfy the query condition. In other words, \( \text{bb}(\rho) \) can be removed from the searching list if its distance from \( q \) is more than \( \delta \).
5.2.2 Processing PRQ-G Queries

Filtering on a non-leaf node for a PRQ-G query is very similar with that of a PRQ-P query. Fig. 13 illustrates the idea. Let $\rho = \sqrt{1 - \delta}$, if the distance between the node bounding box and the query bounding box is larger than $\delta$, this node can be deleted from the searching list of the query.

6. Experiments

We implemented the index structure by extending the spatial index library SaiL [9]. This C++ library can be downloaded from [15] for free. We conducted experiments using a PC with Intel Core 2 Duo CPU E8500 (3.16GHz), RAM 4GB and OS Fedora 12.

We generate 5 two-dimensional synthetic datasets in a $1000 \times 1000$ space with size 10K, 30K, 50K, 80K, and 100K respectively, referred as 10K, 30K, 50K, 80K and 100K respectively. For the real data, we used road line segment data of Long Beach, California and Montgomery, Maryland. We extracted the midpoint of each line segment as the mean and generate the corresponding covariance matrix randomly. The two extracted real datasets (called “LB” and “MG”) contain 39,226 and 50,747 items respectively and are normalized to the $1000 \times 1000$ space.

The query dataset is also generated randomly within the same data space. The query range is a random value within [5\%\text{5}] and the query probability threshold lies within $[0.01, 0.99]$ for both PRQ-P and PRQ-G queries. We run 100 queries for each experimental setting and use the average result to evaluate the performance.

6.1 Performance Analysis

(a) Average Query Processing Time
(b) Average Candidate Number

图 16 Range Trend: PRQ-P vs. $\delta$

Generally, as $\delta$ increases, more query processing time is needed. That’s because more and more target objects will become candidates, leading to more integral computation cost. When $\delta$ reaches so large (20 in Fig. 16(a)) that many potential candidates can be identified as result objects directly without integration. Then the query processing time turns from a gradually rising trend into a steady state and may decrease slightly. This reveals the great power of our result-validation techniques. But if $\delta$ continues to increase, the integral computation cost dominates over all factors and the query processing time raises rapidly. An interesting thing is that the line chart in Fig 16(b) of candidate number almost precisely matches that of query processing time in Fig 16(a).
This again demonstrates that probability integration dominates overall query processing cost.

Although the real dataset MG has less data items (about 39K) than LB50K and 50K, it retrieves more data objects and thus results in more query processing time, because its data distribution is highly biased, and many data objects are located around the central area.

6.3 Probability Trend

We use 20 as a default query range in this subsection to study the probability trend. We run the same query 10 times and use the average result for performance evaluation. Fig. 17 (Fig. 18) shows the query processing time (candidate number) for PRQ-P and PRQ-G queries respectively.

![Graphs showing query processing time and candidate number for PRQ-P and PRQ-G](a) PRQ-P (b) PRQ-G

For PRQ-P queries, the query processing time (candidate number) decreases as the probability threshold \( \theta \) increases first, then the time (number) increases when \( \theta \geq 0.5 \). It precisely matches the computing policy of parameter \( \rho \) as discussed in Section 4.3. This is because that \( \rho \) decides the size of all bounding boxes. When \( \rho \) is large, bounding boxes are very large, so they have weaker filtering power on objects. If \( \rho \) is small, bounding boxes are very small, and they have strong power of filtering. PRQ-G queries work in the similar way with PRQ-P queries, except that their policy of computing \( \rho \) is different.

7. Conclusion and Future Work

In this paper, by modeling uncertain data with multidimensional Gaussian distributions, we propose query processing techniques for two types of probabilistic range queries: PRQ-P and PRQ-G. We further propose a novel index structure to support queries for Gaussian distributions. We implement the index structure and examine its efficiency and effectiveness with experiments. In the future, we will extend its generality and enhance it to be applicable to other types of uncertainty models and queries.

8. Acknowledgements

This research was partly supported by KAKENHI (23650047) and Funding Program for World-Leading Innovative R&D on Science and Technology (First Program).

文 獻