ガウス分布に対する確率的範囲問合せのための索引手法の評価

董テイテイ† 郭　茜†† 肖　川† 石川　佳治†

† 名古屋大学情報科学研究科
†† 香港中文大学システム工学及び工学管理学科
E-mail: †dongtt@db.itc.nagoya-u.ac.jp, ††guoxi022@gmail.com, †††{chuanx,y-ishikawa}@nagoya-u.ac.jp

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 impreciseness 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.

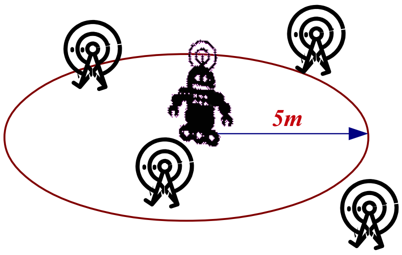


図 1 A motivating example

To remedy this kind of problem, the data records are typically represented by probability density functions instead of deterministic values. Typically, uncertain data is modeled by multi-dimensional *Gaussian distributions* [7]. The Gaussian distribution, also called the normal distribution, is a probability distribution widely used in various fields such as pattern recognition, statistical analysis, etc.. We discuss the problem based on the assumption that the uncertainties of target objects' location information in the database are described by multi-dimensional Gaussian distributions with different parameters for indicating their differences in uncertainty. We consider two cases for a query object: a certain point and an uncertain location represented by a Gaussian distribution.

Furthermore, queries here search for target objects within some specified range (i.e. the radius) from the query objects with high probabilities. Such a query is called a *probabilistic range query (PRQ)*, an extension of the standard range query in the traditional spatial database. In the case of the forgoing scenarios, an example query can be *"find customers currently located* 50 *meters around the shopping center with a probability more than* 40%*"* or *"find landmarks lying within* 5 *meters from my current location with a probability at least* 80%*"*. This kind of queries can provide more meaningful answers to users.

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 ($d \geqq 2$)-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 $o_i \in \mathcal{D}$ is located at $\boldsymbol{x}_i$ is defined by a d-dimensional Gaussian probability density function*

$$p_i(\boldsymbol{x}_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{o}_i)^t \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_i - \boldsymbol{o}_i)\right] \tag{1}$$

where $\mathcal{D}$ is a set of target objects, $\boldsymbol{o}_i$ is the mean of o_i and $\boldsymbol{\Sigma}_i$ is a $d \times d$ covariance matrix. $|\boldsymbol{\Sigma}_i|$ is the determinant of $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_i^{-1}$ is the inverse matrix of $\boldsymbol{\Sigma}_i$. $\boldsymbol{x}^t$ represents a transposition of a vector $\boldsymbol{x}$. □

3.2 Definition of Queries

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

(1) The query object is a fixed point, namely,

$$\boldsymbol{q} = (x_q^1, x_q^2, .., x_q^d)^t.$$

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

$$p_q(\boldsymbol{x}_q) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_q|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}_q - \boldsymbol{q})^t \boldsymbol{\Sigma}_q^{-1}(\boldsymbol{x}_q - \boldsymbol{q})\right]$$

where $\boldsymbol{q}$ is the mean, and $\boldsymbol{\Sigma}_q$ 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 $\boldsymbol{q}$, a distance threshold δ, and a probability threshold θ $(0 < \theta < 1)$, a* probabilistic range query with point query object *(*PRQ-P *for short) is defined as follows:*

$$\text{PRQ-P}(q, \delta, \theta) = \{o_i \mid o_i \in \mathcal{D}, \Pr(\|\boldsymbol{x}_i - \boldsymbol{q}\| \leqq \delta) \geqq \theta\}$$

where $\|\boldsymbol{x}_i - \boldsymbol{q}\|$ represents the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{q}$.

$\Pr(\|\boldsymbol{x}_i - \boldsymbol{q}\| \leqq \delta)$ *is defined as follows:*

$$\Pr(\|\boldsymbol{x}_i - \boldsymbol{q}\| \leqq \delta) = \int \chi_\delta(\boldsymbol{x}_i, \boldsymbol{q}) \cdot p_i(\boldsymbol{x}_i) d\boldsymbol{x}_i \tag{2}$$

where

$$\chi_\delta(\boldsymbol{x}_i, \boldsymbol{q}) = \begin{cases} 1 & \text{if } \|\boldsymbol{x}_i - \boldsymbol{q}\| \leqq \delta \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

is used to enforce the distance-based threshold. □

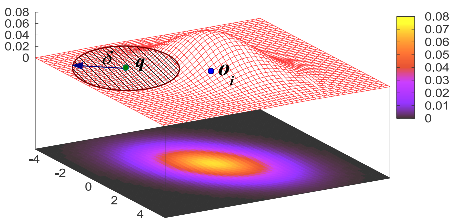


図 2 An illustration of PRQ-P query

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 $\boldsymbol{o}_i$, 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 $\boldsymbol{x}_i$ with a probability $p_i(\boldsymbol{x}_i)$, 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 δ, and a probability threshold θ $(0 < \theta < 1)$, a* probabilistic range query with Gaussian query object *(*PRQ-G*) is defined as follows:*

$$\text{PRQ-G}(q, \delta, \theta) = \{o_i \mid o_i \in \mathcal{D}, \Pr(\|\boldsymbol{x}_i - \boldsymbol{x}_q\| \leqq \delta) \geqq \theta\},$$

where $\Pr(\|\boldsymbol{x}_i - \boldsymbol{x}_q\| \leqq \delta$ *is defined as follows:*

$$\Pr(\|\boldsymbol{x}_i - \boldsymbol{x}_q\| \leqq \delta) = \iint \chi_\delta(\boldsymbol{x}_i, \boldsymbol{x}_q) \cdot p_i(\boldsymbol{x}_i) \cdot p_q(\boldsymbol{x}_q) d\boldsymbol{x}_i d\boldsymbol{x}_q \tag{4}$$

where

$$\chi_\delta(\boldsymbol{x}_i, \boldsymbol{x}_q) = \begin{cases} 1 & \text{if } \|\boldsymbol{x}_i - \boldsymbol{x}_q\| \leqq \delta \\ 0 & \text{otherwise} \end{cases}$$

is a thresholding function. □

To compute the numerical integration in Eq. (4), although we can simply generate random numbers for the two Gaussian distributions $p_i(\boldsymbol{x}_i)$ and $p_q(\boldsymbol{x}_q)$ 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 *ρ-Region*) and its approximation (called *bounding box*) of a Gaussian distribution to prune as many non-candidate objects as possible.

4.1 ρ-Region

Definition 4 (ρ-region)**.** *Consider the integration of the probability density function $p_i(\boldsymbol{x}_i)$ over an ellipsoidal region $(\boldsymbol{x}_i - \boldsymbol{o}_i)^t \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_i - \boldsymbol{o}_i) \leqq r^2$. Let r_ρ be the value of r for which the result of the integration exactly becomes ρ:*

$$\int_{(\boldsymbol{x}_i - \boldsymbol{o}_i)^t \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_i - \boldsymbol{o}_i) \leqq r_\rho^2} p_i(\boldsymbol{x}_i) d\boldsymbol{x}_i = \rho.$$

We call the ellipsoidal region

$$(\boldsymbol{x}_i - \boldsymbol{o}_i)^t \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_i - \boldsymbol{o}_i) \leqq r_\rho^2$$

defined by r_ρ the ρ-region*.* □

Nevertheless, it is costly to compute ρ-regions for arbitrary Gaussian distributions (with different $\boldsymbol{o}_i, \boldsymbol{\Sigma}_i$) 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 $\boldsymbol{o}_i = \mathbf{0}$ and $\boldsymbol{\Sigma}_i = \mathbf{I}$ in Eq. (1).

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

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

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

$$\int_{\|\boldsymbol{x}\|^2 \leqq \tilde{r}_\rho^2} p_{\text{norm}}(\boldsymbol{x}) d\boldsymbol{x} = \rho. \tag{5}$$

For a given ρ,

$$r_\rho = \tilde{r}_\rho \tag{6}$$

holds. ■

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

d	ρ	r_ρ
2	0.98	2.75
$\vdots$	$\vdots$	$\vdots$

図 3 (ρ, r_ρ)-Table

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

4.2 Deriving Bounding Box

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

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 o_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*

$$w_j = \sigma_j r_\rho \tag{7}$$

where σ_j corresponds to the standard deviation for the j-th dimension

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

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

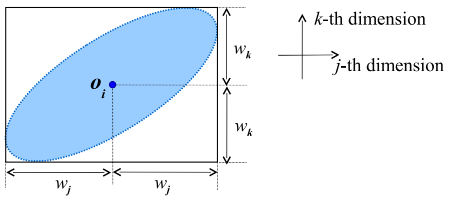


図 4 Image of Bounding Box bb_i

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: $\theta < 0.5$.

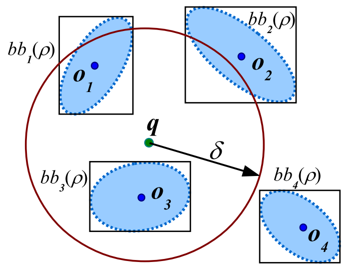


図 5 RR Method $(\rho = 1 - 2\theta, \theta < 0.5)$

Consider four kinds of target objects o_1, o_2, o_3, o_4 as shown in Fig. 5. First, let's consider o_4. Since the probability that o_4 is located inside its ellipsoidal ρ-region is ρ, the probability that o_4 is located outside $bb_4(\rho)$ region is definitely less than $1 - \rho$. Furthermore, given the line symmetry of a Gaussian distribution, the probability that 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 $bb_4(\rho)$ and the sphere are disjoint (that is, connected or separated), the probability that the target object o_4 is within the δ range of query object q will be less than θ. On the contrary, if $bb_4(\rho)$ and the sphere query region have intersection, this probability is possible to reach θ.

For o_1 and 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 $bb_2(\rho)$ of o_2 to check whether it intersects with the spherical region. If they have intersection, then o_2 will be selected as a candidate object.

Moreover, for all candidate objects, we also derive their bounding boxes of θ-regions by letting $\rho = \theta$. If the query region contains the θ-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: $\theta \geqq 0.5$. Let $\rho = \theta$.

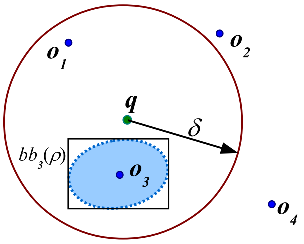


図 6　RR Method ($\rho = \theta, \theta \geqq 0.5$)

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.

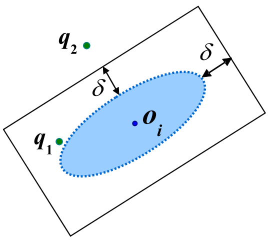


図 7　OR Method

Consider the rectangle paralleled to the axes of the ρ-region ellipsoid. The distance between the rectangle and the ρ-region is at least δ. Obviously when o_i is located inside the ρ-region (with a probability ρ), the distance between o_i and q_2 will be more than δ. 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 δ 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 $\boldsymbol{y}_i$ for a given d-dimensional point $\boldsymbol{x}_i$ which satisfies $\boldsymbol{x}_i = \boldsymbol{E}\boldsymbol{y}_i$. Here $\boldsymbol{E}$ is a diagonal matrix consisting of the eigenvectors of $\boldsymbol{\Sigma}_i^{-1}$. 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 ρ-regions. As shown in Fig. 8, consider the situation that the distance between the bounding boxes of two ρ-region is exactly δ. Since q and o_i are located inside their ρ-regions respectively both with probability ρ, the probability that each of them exists within individual ρ-region at the same time is ρ^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 δ. Specifically, the distance between q and o_i becomes less than δ with a probability at most $1 - \rho^2$.

For a given probability threshold θ of the query, letting $1 - \rho^2 = \theta$, that is, $\rho = \sqrt{1-\theta}$, we can compute ρ. For example, if $\theta = 5\%$, then $\rho = \sqrt{1-0.05} = 0.9747$. Construct the bounding boxes of ρ-regions dynamically for q, o_i with the ρ 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 δ.

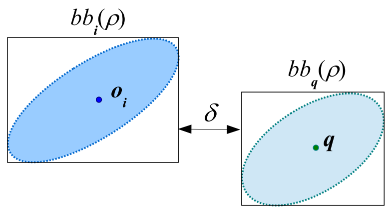


図 8　Filtering for PRQ-G Queries

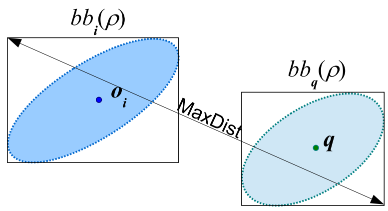


図 9　Filtering for PRQ-G Queries: validation

Moreover, if the maximum distance of $bb_i(\rho)$ and $bb_q(\rho)$ is less than δ, assigning $\rho^2 = \theta$ (i.e., $\rho = \sqrt{\theta}$) will guarantee that the target object o_i is located inside the δ range from the query object q with a probability greater than θ. 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 $o_i = (id_i, \boldsymbol{o}_i, \boldsymbol{\Sigma}_i, bb_i)$, where id_i is the object id, $\boldsymbol{o}_i, \boldsymbol{\Sigma}_i$ are the mean value (average location) and the covariance matrix of the Gaussian distribution, and bb_i is the bounding box of ρ-region for o_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 o_i centered at $\langle x_i^1, \ldots, x_i^d \rangle$ in the d-dimensional space, the bounding box bb_i of o_i is a rectangle parameterized with r_ρ. Its extent (i.e., left bound and right bound) in j-th dimension can be represented as

$$bb_i^j = [x_i^j - w_i^j, x_i^j + w_i^j] = [x_i^j - \sigma_i^j r_\rho, x_i^j + \sigma_i^j r_\rho].$$

We denote bb_i^j as the *bounding interval* of the bounding box bb_i in the j-th dimension. Specifically, bb_i is represented as

$$bb_i = (\langle x_i^1, \sigma_i^1 \rangle, \ldots, \langle x_i^d, \sigma_i^d \rangle).$$

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 covariance 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_ρ values.

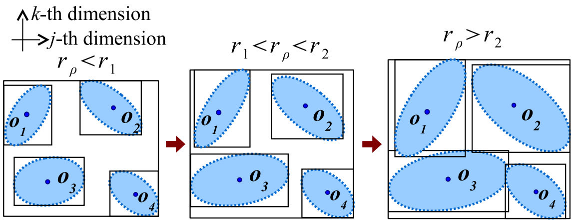


図 10 Bounding boxes of o_1, o_2, o_3 and o_4

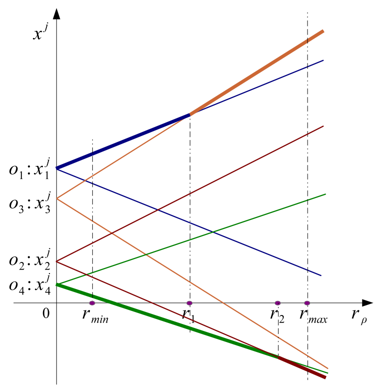


図 11 Bounding intervals of o_1, o_2, o_3 and o_4

Fig. 10 illustrates this problem. This figure shows the changing bound objects of the bounding box for four objects o_1, o_2, o_3 and o_4 as r_ρ increases. Fig. 11 shows their corresponding bounding intervals in j-th dimension. For each object, the pairs of symmetrical lines in Fig. 11 describe the extension of the left and right bounds as r_ρ increases. Lines have different slopes because the standard deviations of o_1, o_2, o_3 and o_4 in j-th dimension differ from each other.

If r_ρ is within the range $(0, r_1]$, the left bound of bounding box bb in j-th dimension is determined by the object o_1, while the object o_3 turns out to be the left bound object of bb if r_ρ is in $(r_1, +\infty]$. Also, when r_ρ increases over r_2, the right bound object of bb changes from o_4 to o_2. In Fig. 11, the upper bold polyline illustrates the left side of the bounding interval of the bounding box 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_ρ, 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_ρ is actually within a range $[r_{min}, r_{max}]$. 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

$$bb_l^j = (\langle x_1^j, \sigma_1^j, [r_{min}, r_1^j] \rangle, \ldots, \langle x_i^j, \sigma_i^j, (r_k^j, r_{max}] \rangle)$$

Now we can obtain the corresponding j-th dimensional bounding interval of the bounding box for objects o_1, o_2, o_3 and o_4 illustrated in Fig. 10 and Fig. 11, and derive the entry in j-th dimension bb^j as

$$bb_l^j = (\langle x_1^j, \sigma_1^j, [r_{min}, r_1] \rangle, \langle x_3^j, \sigma_3^j, (r_1, r_{max}] \rangle)$$

$$bb_r^j = (\langle x_4^j, \sigma_4^j, [r_{min}, r_2] \rangle, \langle x_2^j, \sigma_2^j, (r_2, r_{max}] \rangle)$$

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 $\boldsymbol{q}$ with the radius δ is exactly contiguous with $bb(\rho)$. As discussed in Section 4.3, if $\rho = 1 - 2\theta$, among objects o_1, o_2 and o_3 inside $bb(\rho)$, none of them can satisfy the query condition. In other words, $bb(\rho)$ can be removed from the searching list if its distance from $\boldsymbol{q}$ is more than δ.

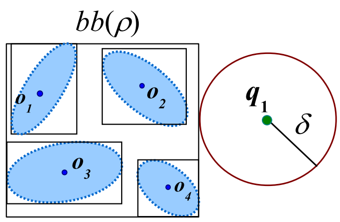


図 12 Filtering for PRQ-P Queries on Non-Leaf Nodes

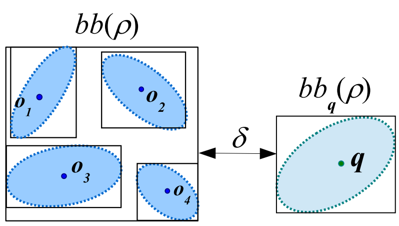


図 13 Filtering for PRQ-G Queries on Non-Leaf Nodes

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 $\rho = \sqrt{1-\theta}$, if the distance between the node bounding box and the query bounding box is larger than δ, this node can be deleted from the searching list of the query.

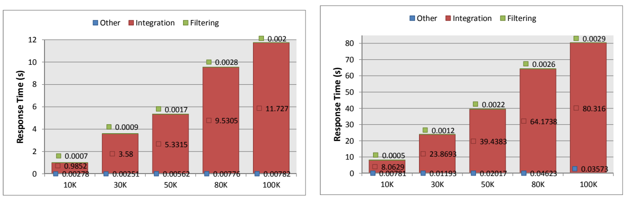
6. Experiments

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 1000×1000 space with size 10000, 30000, 50000, 80000, and 100000, 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 1000×1000 space.

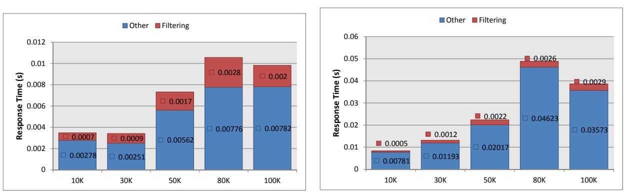
The query dataset is also generated randomly within the same data space. The query range is a random value within $[5, 25]$ and the query probability threshold lies within $[0.01, 0.99]$ for both PRQ-P and PRQ-G queries. We run 100 queries for each experimental setting and use the average result to evaluate the performance.

6.1 Performance Analysis



(a) PRQ-P (b) PRQ-G

図 14 Response Time



(a) PRQ-P (b) PRQ-G

図 15 Response Time: filtering vs. other

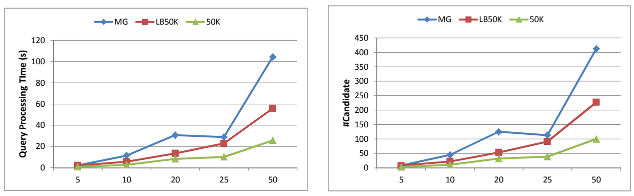
The average response time of 100 random PRQ-P and PRQ-G queries is shown in Fig. 14. The overall runtime (i.e., wall clock time) consists of three components: integral computation, filtering processing, and the rest for other operations (e.g., reading data from files).Clearly in all cases integral computation occupies most of the runtime. This implies that it is important to find effective query processing techniques and to avoid computing the exact probability by numerical integration as much as possible. Time used for candidate filtering is only about one to three milliseconds on average. And it does not show any sharp increase as the dataset size enlarges. This demonstrates the efficiency and scalability of our approach. The similar trend can also be observed in the case of PRQ-G queries.

Excluding the time part of integral computation, we show the time comparison of filtering processing and other operations in Fig. 15. For a PRQ-P query, the time used for filtering processing is no more than half of that used for other operations in all cases. This difference is greater and more evident in a PRQ-G query.

In the following experiments, we will use both the two real datasets and the synthetic dataset 50K. While 50K is randomly generated and its data actually follows the uniform distribution, The data in MG and LB is bias-distributed, and many points are concentrated within an area. For the convenience of performance comparison, we utilize the first 50K of LB (50,747) and call it LB50K.

6.2 Range Trend

The average result of 10 queries is used for performance evaluation. The experimental result of PRQ-G queries by varying δ is very similar to that of PRQ-P queries. So we just take the result of PRQ-P queries for explanation. The average query processing time (including integral computation and filtering processing) of three datasets for PRQ-P queries is shown in Fig. 16(a).



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

図 16 Range Trend: PRQ-P vs. δ

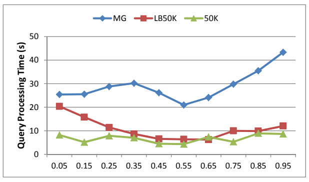
Generally, as δ increases, more query processing time is needed. That's because more and more target objects will become candidates, leading to more integral computation cost. When δ reaches so large (20 in Fig. 16(a)) that many potential candidates can be identified as result objects directly without 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 δ 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.

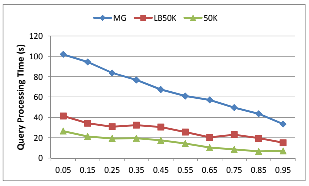
Although the real dataset MG has less data items (about 39K) than LB50K and 50K, it retrieves more data objects and thus results in more query processing time, because its data distribution is highly biased, and many data objects are located around the central area.

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.

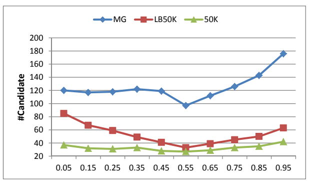


(a) PRQ-P

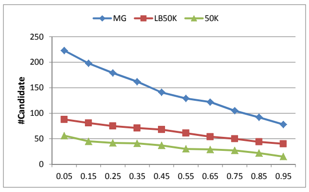


(b) PRQ-G

図 17 Average Query Processing Time vs. θ



(a) PRQ-P



(b) PRQ-G

図 18 Average Candidate Number vs. θ

For PRQ-P queries, the query processing time (candidate number) decreases as the probability threshold θ increases first, then the time (number) increases when $\theta \geqq 0.5$. It precisely matches the computing policy of parameter ρ as discussed in Section 4.3. This is because that ρ decides the size of all bounding boxes. When ρ is large, bounding boxes are very large, so they have weaker filtering power on objects. If ρ 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 ρ is different.

7. Conclusion and Future Work

In this paper, by modeling uncertain data with multi-dimensional 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 献

[1] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In *Proc. ACM SIGMOD*, 1990.

[2] C. Böhm, A. Pryakhin, and M. Schubert. The Gauss-tree: Efficient object identification in databases of probabilistic feature vectors. In *Proc. ICDE*, 2006.

[3] J. Chen and R. Cheng. Efficient evaluation of imprecise location-dependent queries. In *Proc. ICDE*, 2007.

[4] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proc. ACM SIGMOD*, 2003.

[5] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *IEEE TKDE*, 16(9):1112–1127, 2004.

[6] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In *Proc. VLDB*, 2004.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2nd edition, 2000.

[8] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proc. ACM SIGMOD*, pages 47–57, 1984.

[9] M. Hadjieleftheriou, E. Hoel, and V. J. Tsotras. Sail: A spatial index library for efficient application integration. *GeoInformatica*, 9:367–389, 2005.

[10] Y. Ishikawa, Y. Iijima, and J. X. Yu. Processing spatial range queries for objects with imprecise Gaussian-based location information. In *Proc. ICDE*, pages 676–687, 2009.

[11] K. Kodama, T. Dong, and Y. Ishikawa. An index structure for spatial range querying on Gaussian distributions. In *Proc. Fifth International Workshop on Management of Uncertain Data (MUD 2011)*, pages 1–7, 2011.

[12] Y. Manolopoulos, A. Nanopoulos, A. N. Papadopoulos, and Y. Theodoridis. *R-Trees: Theory and Applications*. Springer, 2005.

[13] D. Pfoser and C. S. Jensen. Capturing the uncertainty of moving-object representations. In *Proc. 6th Intl. Symp. on Advances in Spatial Databases (SSD'99)*, 1999.

[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipies: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007.

[15] Spatial Index Library. http://libspatialindex.github.com/.

[16] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao, and S. Prabhakar. Indexing multi-dimensional uncertain data with arbitrary probability density functions. In *Proc. VLDB*, 2005.

[17] Y. Tao, D. Papadias, and J. Sun. The TPR*-tree: An optimized spatio-temporal access method for predictive queries. In *Proc. VLDB*, pages 790–801, 2003.

[18] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[19] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM TODS*, 29(3):463–507, 2004.

[20] S. Šaltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the positions of continuously moving objects. In *Proc. ACM SIGMOD*, pages 331–342, 2000.

[21] O. Wolfson, A. P. Sistla, S. Chamberlain, and Y. Yesha. Updating and querying databases that track mobile units. *Distributed and Parallel Databases*, 7(3):257–287, 1999.