Probabilistic Range Monitoring of Streaming Uncertain Positions in GeoSocial Networks

Kostas Patroumpas[1], Marios Papamichalis[1], and Timos Sellis[1,2]

[1] School of Electrical and Computer Engineering
National Technical University of Athens, Hellas
[2] Institute for the Management of Information Systems, R.C. "Athena", Hellas
{kpatro, timos}@dbnet.ece.ntua.gr, papamixmarios@gmail.com

Abstract. We consider a social networking service where numerous subscribers consent to disclose their current geographic location to a central server, but with a varying degree of uncertainty in order to protect their privacy. We aim to effectively provide instant response to multiple user requests, each focusing at continuously monitoring possible presence of their friends or followers in a time-varying region of interest. Every continuous range query must also specify a cutoff threshold for filtering out results with small appearance likelihood; for instance, a user may wish to identify her friends currently located somewhere in the city center with a probability no less than 75%. Assuming a continuous uncertainty model for streaming positional updates, we develop novel pruning heuristics based on spatial and probabilistic properties of the data so as to avoid examination of non-qualifying candidates. Approximate answers are reported with confidence margins, as a means of providing quality guarantees and suppressing useless messages. We complement our analysis with a comprehensive experimental study, which indicates that the proposed technique offers almost real-time notification with tolerable error for diverse query workloads under fluctuating uncertainty conditions.

1 Introduction

Over this decade, we have been witnessing the rising popularity of social networks. Connecting people who share interests or activities has an all-increasing impact on communication, education, and business; their role even in politics and social movements is indisputable. One of the latest trends heads for *GeoSocial Networking Services* [13, 15], allowing location-aware mobile users to interact relative to their current positions. Platforms like Facebook Places, Google Latitude, or FireEagle[3], enable users to pinpoint friends on a map and share their whereabouts and preferences with the followers they choose. Despite their attraction, such features may put people's privacy at risk, revealing sensitive information about everyday habits, political affiliations, cultural interests etc. Hence, there has been strong legal and research interest on controlling the level of location precision, so as to prevent privacy threats and protect user anonymity.

[3] http://facebook.com/about/location; http://google.com/latitude; http://fireeagle.yahoo.net

Respecting privacy constraints, we turn our focus to real-time processing of *continuous range queries* against such imprecise user locations. In our proposed framework, a subscriber may receive instant notifications when a friend appears *with sufficient probability within her area of interest.* Mobile users are aware of their own exact location thanks to geopositional technologies (e.g., GPS, WiFi, Bluetooth), but they do not wish to disclose it to third parties. Instead, they consent to relay just a cloaked indication of their whereabouts [6] abstracted as an *uncertainty region* with Gaussian characteristics, enclosing (but apparently not centered at) their current position. Hence, the service provider accepts a geospatial stream of obfuscated, time-varying regions sent from numerous users at irregular intervals. Based on such massive, transient, imprecise data, the server attempts to give response to multiple search requests, which may also dynamically modify their spatial ranges and probability thresholds.

This query class may prove valuable to GeoSocial Networking and Location-based services (LBS). A typical request is "Notify me whenever it is highly likely (more than 75%) that any friends of mine are located somewhere in my neighborhood" just in case one wants to arrange a meeting. A micro-blogging enthusiast could be traveling or walking, and while on the move, may wish to post messages to followers nearby. Even virtual interactive games on smartphones could take advantage of such a service, e.g., assessing the risk of approaching "unfriendly territory" with several adversaries expectedly present in close proximity.

In a geostreaming context, identifying mobile users with varying degrees of uncertainty inside changing areas of interest poses particular challenges. Faced with strict privacy preferences and intrinsic positional inaccuracy, while also pursuing adaptivity to diverse query workloads for prompt reporting of results, we opt for an approximate evaluation scheme. We introduce optimizations based on inherent probabilistic and spatial properties of the uncertain streaming data. Thus, we can quickly determine whether an item possibly qualifies or safely skip examination of non-qualifying cases altogether. Inevitably, this probabilistic treatment returns approximate answers, along with confidence margins as a measure of their quality. Our contribution can be summarized as follows:

- We model uncertainty of incoming locations as a stream of moving regions with fluctuating extents under a Bivariate Gaussian distribution.
- We develop an online mechanism for evaluating range requests, employing lightweight, discretized verifiers amenable to Gaussian uncertainty.
- We introduce pruning criteria in order to avoid examination of objects most unlikely to fall inside query boundaries, with minimal false negatives.
- We empirically demonstrate that this methodology can provide approximate, yet timely response to continuous range queries with tolerable error margins.

The remainder of this paper proceeds as follows: Section 2 briefly reviews related work. Section 3 covers fundamentals of positional uncertainty and outlines application specifics. In Section 4, we develop an (ϵ, δ)-approximation algorithm for continuous range search against streaming Gaussian regions. In Section 5, we introduce heuristics for optimized range monitoring. Experimental results are reported in Section 6. Finally, Section 7 concludes the paper.

2 Related Work

Management of uncertain data has gained particular attention in applications like sensor networks, market surveillance, biological or moving objects databases etc. In terms of processing [14], besides range search, a variety of *probabilistic queries* have been studied: nearest-neighbors [10], reverse nearest neighbors [1], k-ranked [11], continuous inverse ranking [2], similarity joins [9, 12], etc.

In contrast to traditional range search, a probabilistic one requires its answer to be assessed for quality. Among related techniques in uncertain databases, the notion of x-bounds in [5] clusters together one-dimensional features with similar degrees of uncertainty in an R-tree-like index. U-tree [16] is its generalization for multiple dimensions and arbitrary probability distributions. U-tree employs probabilistically constrained regions to prune or validate an object, avoiding computation of appearance probabilities. U-tree can be further useful for "fuzzy" search, when the query range itself becomes uncertain [17]. We utilize a pruning heuristic with a similar flavor, but our proposed minimal areas correspond to distinct threshold values and are independent of uncertainty specifications for any object. For predicting the location distribution of moving objects, an adapted B^x-tree [18] has been used to answer range and nearest neighbor queries. Especially for inexact Gaussian data, the Gauss-tree [3] (also belonging to the R-tree family) models the means and variances of such feature vectors instead of spatial coordinates. Such policies may be fine for databases with limited transactions, but are not equally fit for geostreaming uncertain data; the sheer massiveness and high frequency of updates could overwhelm any disk-based index, due to excessive overhead for node splits and tree rebalancing.

Note that spatial ranges may be uncertain as well, e.g., modeled as Gaussians [8], or due to query issuer's imprecise location when checking for objects within some distance [4]. In our case, spatial ranges are considered typical rectangles, yet subject to potential changes on their placement, shape and extent.

Privacy-aware query processing in LBS and GeoSocial networks has also attracted particular research interest. For exact and approximate search for nearest neighbors, the framework in [7] uses Private Information Retrieval protocols, thus eliminating the need for any trusted anonymizer. Shared processing for multiple concurrent continuous queries in [6] handles cloaked user areas independent of location anonymizers, offering tunable scalability versus answer optimality. A privacy-aware proximity detection service is proposed in [15], so that two users get notified whenever the vicinity region of each user includes the location of the other. Encryption and spatial cloaking methods enable the server to perform a blind evaluation with no positional knowledge. More sophisticated protocols [13] offer controllable levels of location privacy against the service provider and third parties, essentially trading off quality of service against communication cost. Nonetheless, such techniques principally address privacy concerns and assume uniform uncertainty distribution. Thus, they lack any probabilistic treatment of spatial queries and user whereabouts, as we attempt in this work. We stress that our approach is orthogonal to privacy preservation policies, focusing entirely on swift processing of continuous range requests at the service provider.

3 Managing Uncertain Moving Objects

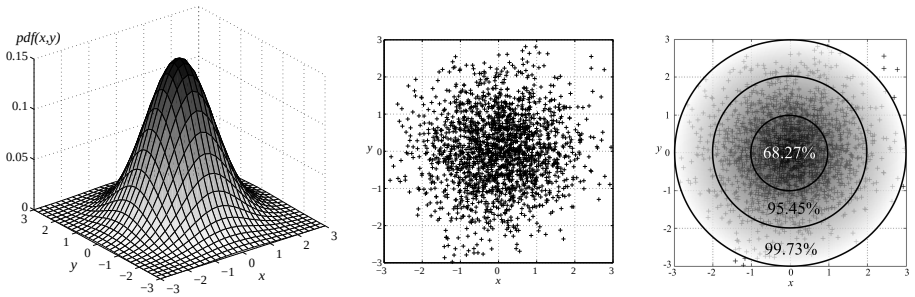
3.1 Capturing Positional Uncertainty

Typical causes of data uncertainty [14, 16] include communication delays, data randomness or incompleteness, limitations of measuring instruments etc. Apart from inherent imprecision of location representations, in this work we assume that mobile users have purposely sent an "inflated" positional update so as to conceal their precise coordinates from the server. Any anonymization technique that cloaks users' locations into *uncertainty regions* can be employed (e.g.,[6]). Typically, the larger the size of the region, the more the privacy achieved.

Positional uncertainty can be captured in a discrete or continuous fashion. A *discrete model* uses a probability mass function (*pmf*) to describe the location of an uncertain object. In essence, a finite number of alternative instances is obtained, each with an associated probability [10, 14]. In contrast, a *continuous model* uses a probability density function (*pdf*), like Gaussian, uniform, Zipfian etc., to represent object locations over the space. Then, in order to estimate the appearance probability of an uncertain object in a bounded region, we have to integrate its pdf over this region [16]. In a geostreaming scenario, a discrete model should be considered rather inappropriate, as the cost of frequently transmitting even a small set of samples per object could not be affordable in the long run. Hence, we adopt a continuous model, which may be beneficial in terms of communication savings, but it poses strong challenges in terms of evaluation. Table 1 summarizes the notation used throughout the paper.

Table 1. Primary symbols and functions.

Symbol	Description
ϵ	Error margin for appearance probability of qualifying objects
δ	Tolerance for reporting invalid answers
N	Total count of moving objects (i.e., users being monitored)
M	Total count of registered continuous range queries
μ_x, μ_y	Mean values of uncertainty pdf per object along axes x, y
σ_x, σ_y	Standard deviations of uncertainty pdf per object along axes x, y
Σ	Set of discrete uncertainty levels $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ for regulating location privacy
$\mathcal{N}(\mathbf{0}, \mathbf{1})$	Bivariate Gaussian distribution with mean (0,0) and standard deviation $\sigma_x = \sigma_y = 1$
r_q	Time-varying 2-d rectangular range specified by query q
r_o	Time-varying uncertainty area of moving object o
$MBB(r_o)$	Minimum Bounding Box of uncertainty area r_o with center at (μ_x, μ_y) and side 6σ
$V(r_o)$	Verifier for uncertainty area r_o, comprised of elementary boxes with known weights
L_q	Set of objects monitored by query q (i.e., *Contact list* of q)
C_q	Set of candidate objects that might qualify for query q
Q_q	Final set of objects estimated to qualify for query q
p_T, p_F, p_U	Total cumulative probability of elementary boxes marked with T, F, U respectively
λ	Granularity of subdivision for every discretized verifier along either axis x, y
$\beta(i, j)$	Weight (i.e., estimated cumulative probability) of elementary box $V(i, j)$
$P_{in}(o, q)$	Estimated probability that object o appears within range of query q
θ	Cutoff threshold for rejecting objects with insufficient appearance probability
Θ	A set $\{\theta_1, \theta_2, \ldots, \theta_m\}$ of m typical threshold values
$\tilde{\alpha}_\theta$	Minimal uncertainty area representing cumulative probability barely less than θ
$\mathcal{A}$	Set of minimal areas $\{\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_m\}$ corresponding to indicative thresholds $\theta_i \in \Theta$



(a) Probability density (b) Scatterplot of samples (c) Truncated region r_o

Fig. 1. Standard Bivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$.

3.2 Object Locations as Bivariate Gaussian Features

Locations of mobile users are modeled with Bivariate Gaussian random variables X, Y over the two dimensions of the Euclidean plane. Intuitively, the resulting uncertainty region implies higher probabilities closer to the mean values (i.e., the origin of the distribution), as illustrated with the familiar "bell-shaped" surface in Fig. 1a. For privacy preservation, the origin of the distribution should not coincide with the precise coordinates, known only by the user itself.

More specifically, let a Bivariate Gaussian (a.k.a. Normal) distribution with

$$\text{mean } \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \qquad \text{and} \qquad \text{covariance matrix } \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix},$$

where μ_x, μ_y are the mean values and σ_x, σ_y the standard deviations along axes x, y respectively, whereas ρ is the correlation of random variables X and Y. Assuming that objects are moving freely, X and Y are independent, hence $\rho = 0$. Of course, location coordinates may spread similarly along each axis, so $\sigma_x = \sigma_y = \sigma$. Thus, the joint probability density function (pdf) is simplified to:

$$pdf(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(x-\mu_x)^2+(y-\mu_y)^2}{2\sigma^2}} \tag{1}$$

As in the univariate case, we can define random variables $X' = \frac{X-\mu_x}{\sigma}$ and $Y' = \frac{Y-\mu_y}{\sigma}$ and derive the *standard bivariate Gaussian distribution* $\mathcal{N}(\mathbf{0}, \mathbf{1})$ with

$$pdf(x', y') = \frac{1}{2\pi} e^{-\frac{r^2}{2}} \tag{2}$$

where $r = \sqrt{x'^2 + y'^2}$ denotes the distance from the origin of the derived distribution at $(0, 0)$ with standard deviations $(1, 1)$. Figure 1b depicts a scatterplot of random samples under this distribution. As illustrated in Fig. 1c, there is 99.73% probability that the location is found within a radius 3σ from the origin. Depending on the variance, the density of a Gaussian random variable is rapidly diminishing with increasing distances from the mean. Thanks to its inherent simplicity, the uncertainty region can be truncated in a natural way *on the server side*, so the user itself does not need to specify a bounded area explicitly.

3.3 System Model

We consider a social networking service with a large number N of location-aware subscribers, each moving over the Euclidean plane and communicating with the provider. Messages transmitted from mobile users concern either cloaked positions or spatial requests. By convention, the former (termed *objects*) are being searched by the latter (*queries*). So, objects and queries alike represent mobile users of the service, but with distinct roles (passive or active) in terms of monitoring. All messages are timestamped according to a global clock at distinct instants τ (e.g., every minute).

Every object o relays to the centralized server its *uncertainty region* r_o, i.e., an imprecise indication of its current location. The server is not involved in the cloaking process, but passively receives vague positional information according to a privacy preserving protocol. Updates may be sent over at irregular intervals, e.g., when an object has gone far away from its previously known position or upon significant change at its speed. Although the server knows nothing about the exact (x, y) coordinates of a given object o, it can be sure that o is definitely found somewhere within its uncertainty region until further notice.

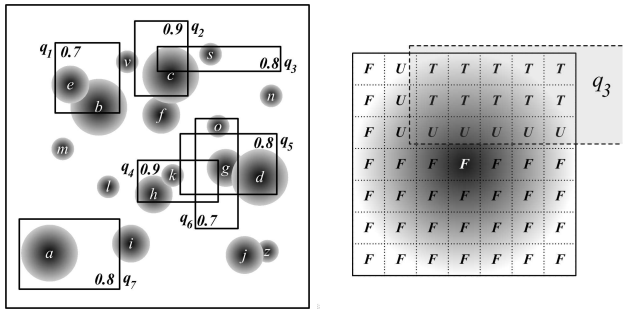
Each uncertainty region r_o follows a Bivariate Gaussian distribution, so an object o must sent the origin (μ_x, μ_y) of its own pdf and the standard deviation σ (common along both dimensions). Upon arrival to the server from numerous objects, these items constitute a unified *stream* of tuples $\langle o, \mu_x, \mu_y, \sigma, \tau \rangle$, ordered by their timestamps τ. Note that μ, σ are expressed in distance units of the coordinate system (e.g., meters). Larger σ values indicate that an object's location can be hidden in a greater area around its indicated mean (μ_x, μ_y). As object o is moving, it relays (μ_x, μ_y) updates. We prescribe k *uncertainty levels* $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, so any object can adjust its degree of privacy dynamically.

A set of M continuous queries are actually registered at the server, each specifying a *rectangular extent* r_q and a cutoff *threshold* $\theta \in (0, 1)$. During their lifetime, ranges r_q may be moving and also vary in size, whereas a query may arbitrarily change its own θ. Therefore, the server accepts query updates specifying $\langle q, r_q, \theta, \tau \rangle$, replacing any previous request with identifier q. As is typical in social networking [13], each query issuer states its *contact list* L_q declaring its friends, fans, or followers. Hence, the server retains a table with entries $\langle q, o \rangle$, which specifies that query q has an interest on monitoring object o, provided that the latter is consenting. Evaluation takes place periodically with execution cycles at each successive τ, upon reception of the corresponding updates. Query q identifies any object o from its contact list L_q currently within specified range r_q with *appearance probability* $P_{in}(o, q)$ at least θ. Analogously to [16, 18]:

Definition 1. *A* probabilistic range query q*, at any timestamp* τ *reports objects* $\{o \in L_q \mid P_{in}(o, q) \geq \theta\}$ *with:*

$$P_{in}(o, q) = \int_{r_q \cap r_o|y} \int_{r_q \cap r_o|x} pdf(x, y) dx dy \tag{3}$$

where $r_q \cap r_o|x$ denotes the interval along x-axis where areas r_q and r_o spatially overlap (notation similar for the y-axis). For the example setting in Fig. 2a,



(a) Example setting (b) Verifier for $P_{in}(c, q_3)$

Fig. 2. Probabilistic range search over uncertain objects.

object c qualifies for query q_2, but not for q_3 since $P_{in}(c, q_3) < 0.8$, assuming that c belongs to the contact list of both queries.

The problem is that Gaussian distributions cannot be integrated analytically, so we need to resort to numerical methods like Monte-Carlo to get a fair estimation for Eq. (3). Yet, Monte-Carlo simulation incurs excessive CPU cost as it requires a sufficiently large number of samples (at the order of 10^6 [16]). Given the mobility and mutability of objects and queries, such a solution is clearly prohibitive for processing range requests in online fashion.

4 Approximation with Discretized Uncertainty Regions

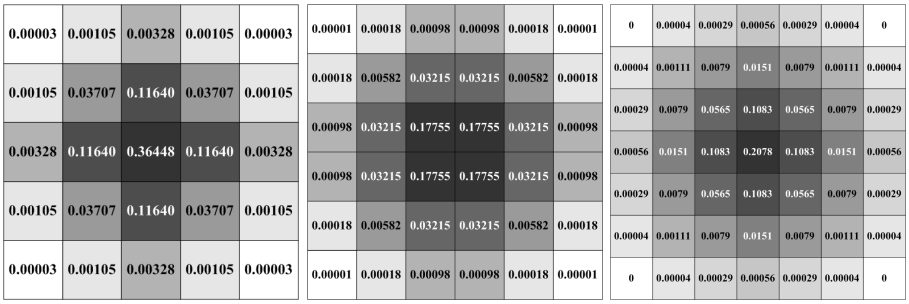
4.1 Probing Objects through Probabilistic Verifiers

An object never specifies a bounded uncertainty region; still, the server may safely conjecture that its location is within a truncated density area of radius 3σ around its mean (μ_x, μ_y), as exemplified in Fig. 1c. To simplify computations, instead of such a circle, its *rectilinear circumscribed square* of side 6σ can stand for uncertainty region just as well. In fact, the cumulative probability of this Minimum Bounding Box (MBB) is greater than 99.73% and tends asymptotically to 1, although its area is $\pi/4$ times larger than the circle of radius 3σ.

Now suppose that for a known σ, we subdivide this MBB uniformly into $\lambda \times \lambda$ *elementary boxes*, $\lambda \in \mathbb{N}^*$. Boxes may have the same area, but represent diverse cumulative probabilities, as shown in Fig. 3. Once precomputed (e.g., by Monte-Carlo), these probabilities can be retained in a lookup table V. If λ is odd, the central box $V(\lceil \frac{\lambda}{2} \rceil, \lceil \frac{\lambda}{2} \rceil)$ is the one with the highest density. Anyway:

Lemma 1. *The cumulative probability in each of the $\lambda \times \lambda$ elementary boxes is independent of the parameters of the applied Bivariate Gaussian distribution.*

In other words, for a fixed λ, the contribution of each particular box in Fig. 3 remains intact for any σ value. The spatial area of a box is $(6\sigma/\lambda)^2$, so it expands quadratically with increasing σ. Yet, as a measure of its probability density, each



(a) Box weights for $\lambda = 5$ (b) Box weights for $\lambda = 6$ (c) Box weights for $\lambda = 7$

Fig. 3. Diverse subdivisions of the same uncertainty region into $\lambda \times \lambda$ elementary boxes.

box $V(i, j)$ maintains its own characteristic *weight* $\beta(i, j)$, which depends entirely on λ and fluctuates with the placement of $V(i, j)$ in the MBB.

The rationale behind this subdivision is that it may be used as a *discretized verifier* V when probing uncertain Gaussians. Consider the case of query q_3 against object c, shown in detail in Fig. 2b. Depending on its topological relation with the given query, each elementary box of V can be easily characterized by one of three possible *states*: (i) T is assigned to elementary boxes totally within query range; (ii) F signifies disjoint boxes, i.e., those entirely outside the range; and (iii) U marks boxes partially overlapping with the specified range.

Then, summing up the respective cumulative probabilities for each subset of boxes returns three *indicators* p_T, p_F, p_U suitable for object validation:

(i) In case that $p_T \geq \theta$, there is no doubt that the object qualifies.
(ii) If $p_F \geq 1 - \theta$, then the object may be safely rejected, as by no means can its appearance probability exceed the query threshold. This is the case for object c in Fig. 2, since its $p_N = 0.72815 \geq 1 - 0.8$.
(iii) Otherwise, when $p_T + p_U \geq \theta$, eligibility is ambiguous. To avoid costly Monte-Carlo simulations, the object could be regarded as *reservedly qualifying*, but along with a confidence margin $[p_T, 1 - p_F)$ as a degree of its reliability.

Because $p_T + p_F + p_U \simeq 1$, only indicators p_T, p_F need be calculated. Still, in case (iii) the magnitude of the confidence margin equals the overall cumulative probability of the U-boxes, which depends entirely on granularity λ. The finer the subdivision into elementary boxes, the less the uncertainty in the emitted results. In contrast, a small λ can provide answers quickly, which is critical when coping with numerous objects. As a trade-off between timeliness and answer quality, next we turn this range search problem into an (ϵ, δ)-approximation one.

4.2 Towards Approximate Answering with Error Guarantees

Let $\bar{p}$ the exact[4] $P_{in}(o, q)$ appearance probability that object o of uncertainty region r_o lies within range r_q of query q. Also, let $\hat{p}$ the respective approximate

[4] $\bar{p}$ cannot be computed analytically, but can be estimated with numerical methods.

probability derived after probing the elementary boxes of verifier $V(r_o)$. Given parameters ϵ and δ, we say that object o *qualifies* for q, if approximate estimation $\hat{p}$ deviates less than ϵ from exact $\bar{p}$ with probability at least $1 - \delta$. Formally:

$$P(|\bar{p} - \hat{p}| \leq \epsilon) \geq 1 - \delta. \tag{4}$$

Intuitively, $\epsilon \in (0, \theta)$ is the *error margin* of the allowed overestimation in $P_{in}(o, q)$ when reporting a qualifying object. In fact, ϵ relates to the size of the elementary boxes and controls the granularity of $V(r_o)$. On the other hand, $\delta \in (0, 1)$ specifies the *tolerance* that an invalid answer may be given (i.e., a false positive). But in practice, given the arbitrary positions and extents of objects and queries, as well as the variability of threshold θ which determines qualifying results, it is hard to verify whether (4) actually holds for specific ϵ, δ values.

As it is difficult to tackle this problem, we opt for a relaxed approach with heuristics. Without loss of generality, we assume that extent r_q is never fully contained within uncertainty region r_o of any object o. According to the abstraction of uncertainty with MBB's, it suffices that any side of rectangle r_q is never less than 6σ, which is quite realistic. Thus, a query range either contains or intersects or is disjoint with an uncertainty region. In cases of full containment or clear separation, there is no ambiguity; the object is qualified or rejected with 100% confidence, respectively. As for intersections, among all cases discussed in Section 4.1, the trouble comes from partial overlaps of type (iii) that may lead to considerable overestimation. Indeed, rectangle r_q may only cover a tiny slice of elementary boxes marked as U, as it occurs with the three vertical U-boxes in Fig. 2b. The redundancy in the estimated cumulative probability owed to the non-covered area of U-boxes is evident.

Let us take a closer look at partial overlaps of type (iii) between a query rectangle r_q and the elementary boxes of an uncertainty region r_o, assuming a fixed λ. As illustrated in Fig. 4, there are three possible cases that r_q may intersect $V(r_o)$ and leave uncovered a particular stripe of the verifier. Of particular concern are *horizontal*, *vertical* or L-*shaped stripes*, comprised of consecutive slices of U-boxes (the red hatched bars in Fig. 4), which amplify the confidence margin. There are eight combinations in total, classified into two groups: (a) four cases concern a straight (horizontal or vertical) stripe, depending on which side of the verifier remains uncovered, and (b) other four create an L-shaped stripe touching the enclosed query corner ($\lrcorner, \llcorner, \urcorner, \ulcorner$). Due to the square shape of verifiers and the underlying symmetry of Gaussian features, the horizontal and vertical cases are equivalent; it also does not matter which corner of $MBB(r_o)$ is enclosed in range r_q. Hence, it suffices to examine an indicative combination from either group.

The worst case happens when r_q has just a tiny overlap with each U-box, hence overestimation $|\bar{p} - \hat{p}|$ becomes almost p_U. In contrast, when just a small area of all U-boxes is left uncovered, overestimation is minimized and upper bound $p_T + p_U$ of the margin is fairly reliable. In between, since objects and queries are not expected to follow any specific mobility pattern, there are infinitely many chances for such partial overlaps, leading to a variety of stripes

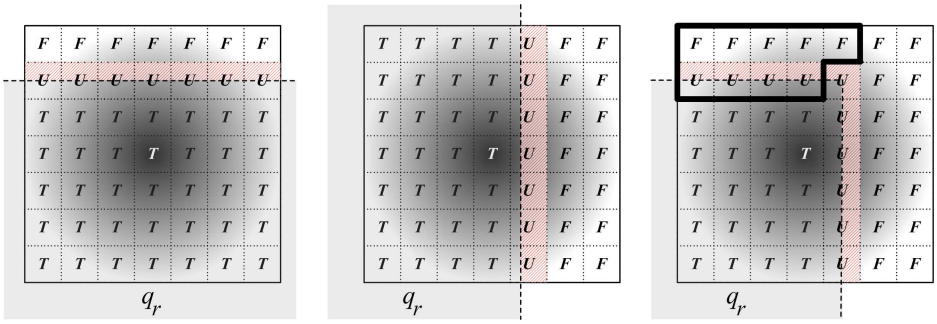


Fig. 4. Horizontal, Vertical and L-shaped stripes of U-box slices beside query boundary.

with diverse cumulative probability. Each case has equal likelihood to occur, but incurs varying errors in probability estimation. Nevertheless, for increasing λ values, each elementary box of the verifier steadily gets less and less weight, so the overestimation effect weakens drastically. In the average case, and for sufficiently large λ, we may approximately consider that each U-box contributes half of its density to the confidence margin. In other words, we assume that the query boundary crosses each U-box in the middle (especially for a corner box, it encloses a quarter of its area), as exemplified in Fig. 4c.

Under this discretized relaxation of the problem, we could evaluate the expected superfluous density for all possible arrangements of straight or L-shaped overlaps and estimate the chances that Formula (4) gets fulfilled. For a fixed subdivision of MBB's, there are 4λ possible instantiations for a straight stripe, considering that each side of the query rectangle r_q may be crossing a horizontal or vertical series of U-boxes (Fig. 4a, 4b). Similarly, any corner of r_q may be centered in any elementary box, giving $4\lambda^2$ potential instantiations of L-shaped stripes (Fig. 4a). In total, we consider $4\lambda + 4\lambda^2$ equiprobable instantiations, yet each one causes a varying overestimation. Suppose that for a given λ, it turns out that ν out of those $4\lambda(\lambda+1)$ cases incur an error less than ϵ. Then, if

$$P(|\bar{p} - \hat{p}| \leq \epsilon) = \frac{\nu}{4\lambda(\lambda+1)} \geq 1 - \delta, \tag{5}$$

we may *accept* that the object approximately qualifies under the aforementioned assumptions.

Since the quality of the approximate answer strongly depends on granularity λ of verifier $V(r_o)$, we wish to select the *minimal* λ^* value so that the resulting probabilities could fulfill inequality (5). In a brute-force preprocessing step based on Monte-Carlo simulation, we can estimate the cumulative probabilities of problematic stripes, starting from a small λ and steadily incrementing it until (5) eventually holds. Then, for the given ϵ, δ values, these fine-tuned $\lambda^* \times \lambda^*$ elementary boxes are expected to provide reliable results that only rarely digress from the given confidence margins, as we experimentally verify in Section 6.

Algorithm 1 Probabilistic Range Monitoring

Procedure *RangeMonitor* (timestamp τ)
Input: Stream items $\langle o^j, \mu_x^j, \mu_y^j, \sigma^j, \tau \rangle$ from $j = 1..N$ Bivariate Gaussian objects.
Input: Specification updates $\langle q^i, r_q^i, \theta^i, \tau \rangle$ from $i = 1..M$ continuous range queries.
Output: Qualifying results $Q = \{\langle q^i, o^j, \theta^{min}, \theta^{max}, \tau \rangle : \bigcap(r_o^j, r_q^i) \neq \emptyset$ with confidence $(\theta^i \leq \theta^{min} < \theta^{max}) \vee (\theta^{min} < \theta^i \leq \theta^{max})\}$.
$Q \leftarrow \{\}$; *//Inital result set for all queries at execution cycle τ*
for each q^i **do**
  $\tilde{\alpha}^* \leftarrow$ minimal area looked up from $\mathcal{A}$, corresponding to maximal $\theta^* \in \Theta, \theta^* \leq \theta^i$
  $C_q^i \leftarrow \{o^j \in L_q^i \mid MBB(r_o^j) \cap r_q^i \neq \emptyset\}$; *//Candidates only from contact list of q^i*
  for each $o^j \in C_q^i$ **do**
    if (o^j is unchanged $\wedge$ q^i is unchanged) **then**
      continue; *//Skip evaluation for unmodified entities*
    else if $MBB(r_o^j) \subset r_q^i$ **then**
      $Q \leftarrow Q \cup \langle q^i, o^j, 1, 1, \tau \rangle$; *//Certain object due to full containment*
    else if $\|MBB(r_o^j) \cap r_q^i\| < \sigma^j \cdot \sigma^j \cdot \tilde{\alpha}^*$ **then**
      continue; *//Pruning with respective minimal area of overlap*
    else
      $\langle \theta^{min}, \theta^{max} \rangle \leftarrow$ *ProbeVerifier* $(r_q^i, MBB(r_o^j), \theta^i)$; *//Approximate indicators*
      if $\theta^i \leq \theta^{min}$ **then**
        $Q \leftarrow Q \cup \langle q^i, o^j, \theta^{min}, \theta^{max}, \tau \rangle$; *//Object qualifies, margin $[\theta^{min}, \theta^{max})$*
      else if $\theta^{min} < \theta^i \leq \theta^{max}$ **then**
        $Q \leftarrow Q \cup \langle q^i, o^j, \theta^{min}, \theta^{max}, \tau \rangle$; *//Reservedly qualifying object*
      end if
    end if
  end for
end for
Report Q; *//Disseminate results to each query for execution cycle τ*
End Procedure

Function *ProbeVerifier* (query range r_q^i, object region $MBB(r_o^j)$, threshold θ^i)
$V(r_o^j) \leftarrow$ verifier with symbols $\{T, F, U\}$ stating any overlap of r_q^i over $MBB(r_o^j)$;
$p_T \leftarrow 0$; $p_F \leftarrow 0$; *//Initialize indicators for appearance probability $P_{in}(o^j, q^i)$*
for each box $b_k \in V(r_o^j)$ **by** spiroid (or ripplewise) visiting order **do**
  if $b_k =$ 'T' **then**
    $p_T \leftarrow p_T + \beta(k)$; *//$k^{th}$ elementary box of verifier V is completely inside r_q^i*
  else if $b_k =$ 'F' **then**
    $p_F \leftarrow p_F + \beta(k)$; *//$k^{th}$ elementary box of verifier V is completely outside r_q^i*
  end if
  if $p_F \geq 1 - \theta_i$ **then**
    return $\langle 0, 0 \rangle$; *//Eager rejection for non-qualifying objects*
  end if
end for
return $\langle p_T, 1 - p_F \rangle$; *//Bounds for appearance probability $P_{in}(o^j, q^i)$*
End Function

5 Online Range Monitoring over Streaming Gaussians

5.1 Evaluation Strategy

The pseudocode for the core range monitoring process is given in Algorithm 1. Implicitly, query q does not wish to find every object in range r_q; searching concerns only those enrolled in its contact list L_q. During query evaluation at timestamp τ, the spatial predicate is examined first against items from contact list L_q, offering a set of *candidate objects* $C_q(\tau) = \{o \in L_q \mid MBB(r_o) \cap r_q \neq \emptyset\}$ with uncertainty regions currently overlapping with r_q (Line 8). At a second stage described next, candidates with a likelihood above θ to lie within range should be returned as *qualifying objects* $Q_q(\tau) = \{o \in C_q(\tau) \mid P_{in}(o, q) \geq \theta\}$.

In case that $MBB(r_o)$ is fully contained in rectangle r_q, object o clearly qualifies with confidence 100%, irrespective of any threshold θ the query may stipulate (Lines 12-13). Similarly, if $MBB(r_o)$ and r_q are spatially disjoint, then object o is rejected also with confidence 100%. Both cases involve no probabilistic reasoning, as simple geometric checks can safely determine eligible objects.

But as already pointed out, evaluation is mainly complicated because of partial overlaps between $MBB(r_o)$ and r_q. Since this is expectedly a very frequent case, employing indicators over discretized verifiers with precomputed cumulative probabilities can provide a tolerable approximation, instead of unaffordable Monte-Carlo simulations as analyzed in Section 4. Even so, probing a few hundred or maybe thousand elementary boxes per candidate object may still incur excessive CPU time. Moreover, such a task must be repeatedly applied at each execution cycle τ against changing query specifications and mutable uncertainty regions. Next, we propose heuristics that may substantially reduce processing cost, effectively filtering out improbable candidate objects (i.e., true negatives) and avoiding exhaustive investigation of discretized verifiers.

5.2 Pruning Candidates using Indicative Minimal Areas

Suppose that we could identify the smallest possible area $\tilde{\alpha}_\theta$ inside an uncertainty region, such that $\tilde{\alpha}_\theta$ represents a cumulative probability barely less than threshold θ of a given query q. If $\|r_q \cap MBB(r_o)\| < \tilde{\alpha}_\theta$, object o cannot qualify for query q, as its appearance probability $P_{in}(o, q)$ is definitely below θ. Then, estimating $P_{in}(o, q)$ is not necessary at all, because $\tilde{\alpha}_\theta$ indicates the *minimal area of overlap* between object o and query q in order for o to qualify.

Ideally, this observation could eliminate candidate objects substantially, as no further examination is required for those of overlapping areas less than $\tilde{\alpha}_\theta$ with the given query q. However, applying such a pruning criterion, necessitates precomputation of the respective $\tilde{\alpha}_\theta$ values for every possible threshold $\theta \in (0, 1)$ a query could specify. A second issue relates to the density of uncertainty regions. Let query q equally overlap two objects o_1, o_2 with uncertainty regions r_{o_1}, r_{o_2} of diverse standard deviations $\sigma_1 \neq \sigma_2$. Notwithstanding that $\|r_q \cap MBB(r_{o_1})\| = \|r_q \cap MBB(r_{o_2})\|$, it does not necessarily hold that these overlaps represent equal appearance likelihood, since $P_{in}(o_1, q)$ and $P_{in}(o_2, q)$ are derived from Eq. (3)

according to different pdf parametrization. Thus, even for a fixed θ, a single minimal value $\tilde{\alpha}_\theta$ cannot be used against every uncertainty region.

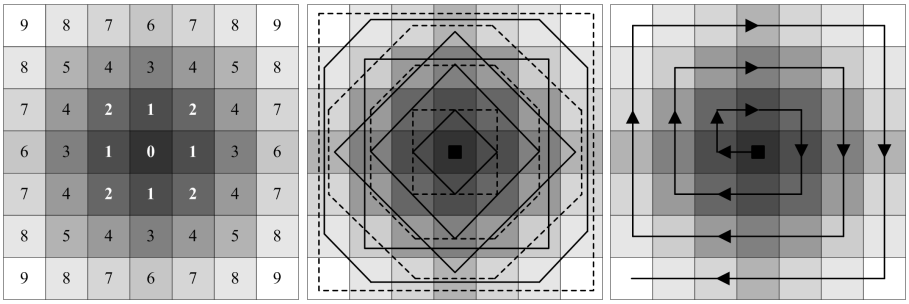
To address issues concerning computation of such minimal areas $\tilde{\alpha}_\theta$, let us start with a specific threshold θ, stipulating a Standard Bivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$ for the uncertainty region. Because the density of Gaussians is maximized around the mean and then decreases rapidly for increasing distances across all directions (Fig. 1a), the sought minimal area is always a circle centered at the origin (μ_x, μ_y) of the pdf with a radius $R \in (0, 3)$ that depends on the given θ. To discover that R value and hence compute $\tilde{\alpha}_\theta = \pi R^2$, we can perform successive Monte-Carlo simulations, increasing R by a small step until the cumulative probability inside the circle becomes only just below θ. For other $\sigma \neq 1$, it turns out that the respective area is $\tilde{\alpha}_\theta = \pi(\sigma R)^2$, as standard deviation σ actually dictates the spread of values, and hence the magnitude of the circle.

Due to the variety of possible thresholds specified by user requests, it makes sense to discover minimal areas only for a small set $\Theta = \{\theta_1, \theta_2, \ldots, \theta_m\}$ of m typical values, e.g., $\Theta = \{10\%, 20\%, \ldots, 100\%\}$. Catalogue $\mathcal{A} = \{\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_m\}$ of respective area magnitudes can be computed offline by the aforesaid Monte-Carlo process assuming a distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$. Having Θ readily available during online evaluation, when a query specifies an arbitrary threshold $\theta \notin \Theta$, we can easily identify the maximal $\theta_i^* \in \Theta, \theta_i^* \leq \theta$ and safely choose its corresponding minimal area $\tilde{\alpha}_i^*$ from the precomputed set $\mathcal{A}$ (Line 7). For the pruning condition, it suffices to compare whether $\|MBB(r_o) \cap r_q\| < \sigma^2 \cdot \tilde{\alpha}_i^*$, so as to account for the magnitude of an uncertainty region r_o with any particular σ (Lines 14-15).

5.3 Optimized Examination of Elementary Boxes

As pointed out in Section 4.1, a discretized verifier can provide a fairly reliable approximate answer in case of partial overlaps between query rectangles and circumscribed uncertainty regions. Essentially, after iterating through each elementary box b_i and having updated indicators p_T, p_F, we can safely determine whether an object qualifies (if $p_T \geq \theta$) or must be rejected (when $p_F \geq 1 - \theta$). In addition, reservedly qualifying objects could be reported with a confidence margin $[p_T, 1 - p_F)$, in case that $1 - p_F \geq \theta$ (Lines 16-21).

However, for finer subdivisions of verifiers, probing at each execution cycle τ an increasing number $\lambda^* \times \lambda^*$ of elementary boxes could incur considerable cost, especially for objects having little chance to qualify. Note that elementary boxes towards the center have much more weight (i.e., greater cumulative probabilities) than peripheral ones. Therefore, we had better start visiting boxes from the center and progressively inspect others of less and less importance. Updating p_T and p_F accordingly, we can resolve object qualification much faster, since peripheral boxes have practically negligible weight (Fig. 3). Such *eager rejections* can be decided as soon as a (yet incomplete) p_F exceeds $1 - \theta$, thus avoiding an exhaustive investigation of the entire verifier, particularly when $\theta > 0.5$. Since p_F for a given object could never decrease with further box examinations at current τ, continuing calculation of indicators is pointless as the result cannot be altered. Considering that a query range might overlap many uncertainty areas only by a



(a) Box rankings (b) Ripplewise inspection (c) Spiroid inspection

Fig. 5. Visiting order of elementary boxes for $\lambda = 7$.

small fringe, the savings can be enormous, as rejections could be resolved soon after inspecting just a few boxes around the center. A similar argument holds when issuing qualifying objects, especially for relatively lower thresholds.

Based on this important observation, we could take advantage of the inherent ranking of elementary boxes for visiting them by decreasing weight, so as to progressively update indicators p_T and p_F. In Fig. 5, graduated gray color reflects a ranking (Fig. 5a) of elementary boxes classified by their inferred cumulative probability for the verifier depicted in Fig. 3c. Intuitively, we may opt for a *ripplewise order* regarding box inspections, distantly reminiscent of raindrops rippling on the water surface. But instead of forming circles, groups of perhaps nonadjacent, yet equi-ranked boxes are arranged as vertices of squares, diamonds, octagons etc. Inspection starts from the central box and continues in rippling waves (depicted with alternating solid and dashed lines in Fig. 5b) rushing outwards across the underlying uncertainty region. An alternative choice is the simplified visiting order illustrated in Fig. 5c, which takes a *squarish spiroid* pattern. Occasionally violating the strict succession of rankings, it follows a continuous *meander* line that again starts from the central box (or next to the center, in case of even subdivisions) and traverses the rest in rings of increasing radius and gradually diminishing weight.

Both orderings aim to give precedence to boxes with potentially significant contribution to appearance probabilities. Assuming a $\theta = 0.6$ and following a spiroid visiting order for the example shown in Fig. 4b, we can easily conclude that the object qualifies for query q after examining the nine central boxes only, which account for a $p_T \simeq 0.6457 \geq \theta$ based on cumulative probabilities (Fig. 3c). Function *ProbeVerifier* in Algorithm 1 outlines this optimized verification step.

6 Experimental Evaluation

6.1 Experimental Setup

Next, we report results from an empirical validation of our framework for probabilistic range monitoring against streams of Gaussian positional uncertainty. We

generated synthetic datasets for objects and queries moving at diverse speeds along the road network of greater Athens in a spatial Universe $\mathcal{U}$ of 625 km^2. By calculating shortest paths between nodes chosen randomly across the network, we were able to create samples of 200 concurrent timestamps from each such route. In total, we obtained a point set representing mean locations for $N = 100\ 000$ objects, and similarly, the centroids of $M = 10\ 000$ query ranges. Spatial range of queries is expressed as percentage (%) of the entire $\mathcal{U}$. However, ranges are not necessarily squares with the given centroid, because we randomly modify their width and height in order to get arbitrarily elongated rectangles of equal area. Each object updates its uncertainty area regularly at every timestamp. Concerning query ranges, their *agility* of movement is set to 0.1, so a random 10% of them modify their specification at each timestamp.

However, contact list L_q per user must not be specified at random; otherwise, probabilistic search would hardly return any meaningful results with synthetic datasets. Thus, for each query we computed a preliminary list of all objects in its vicinity for the entire duration (200 timestamps). Only for data generation, we considered exact locations of N objects within circular areas of 1% of $\mathcal{U}$ centered at each query centroid. After calculating object frequencies for each query, we created two sets of contact lists nicknamed *MOD* and *POP*, respectively retaining the top 50% and top 75% of most recurrent objects per query. Indicatively, a query in *MOD* on average has a modest number of 87 subscribers (i.e., monitored objects) with a maximum of 713, whereas a *POP* list typically has 693 and at most 5911 members, i.e., is almost an order of magnitude more popular.

Evaluation algorithms were implemented in C++ and executed on an Intel Core 2 Duo 2.40GHz CPU running GNU/Linux with 3GB of main memory. Typically for data stream processing, we adhere to online in-memory computation, excluding any disk-bound techniques. We ran simulations using different parameter settings for each experiment. Due to space limitations, we show results just from some representative ones. All results are averages of the measured quantities for 200 time units. Table 2 summarizes experimentation parameters and their respective ranges; the default value (in bold) is used for most diagrams.

6.2 Experimental Results

Verifiers for uncertainty areas should strike a balance between approximation quality and timely resolution of appearance probabilities. So, we first attempt to

Table 2. Experiment parameters.

Parameter	Values
Number N of objects	**100 000**
Number M of range queries	**10 000**
Range area (% of universe $\mathcal{U}$)	0.01, 0.1, **1**, 2, 5, 10
Standard deviation σ (meters)	50, 100, **200**, 300, 500
Cutoff threshold θ	0.5, 0.6, 0.7, **0.75**, 0.8, 0.9, 0.99
Error margin ϵ	0.02, 0.03, **0.05**, 0.1
Tolerance δ	0.01, 0.02, **0.03**, 0.05, 0.1

Table 3. Fine-tuning λ^*.

ϵ	δ	λ^*	ϵ	δ	λ^*
0.02	0.01	103	0.05	0.02	38
0.02	0.02	97	**0.05**	**0.03**	**37**
0.03	0.01	67	0.05	0.05	35
0.03	0.02	**65**	0.1	0.02	19
0.03	0.03	63	0.1	0.05	**18**
0.05	0.01	41	0.1	0.1	17

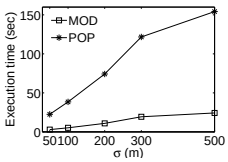


Fig. 6.

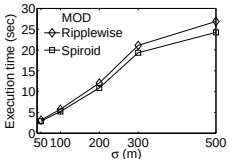


Fig. 7.

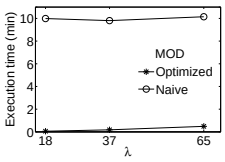


Fig. 8.

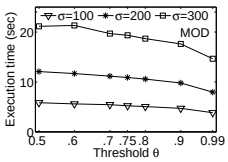


Fig. 9.

determine a fine-tuned subdivision according to the desired accuracy of answers. Table 3 lists the minimal granularity λ^* of verifiers so as to meet the bounds for tolerance δ and error ϵ, using a brute-force preprocessing step (Section 4.2). But a large λ around 100, would create verifiers with 10 000 tiny elementary boxes of questionable practical use, considering the numerous spatial arrangements of queries and objects. For our experiments, we have chosen three moderate values (in bold in Table 3) that represent distinct levels of indicative accuracy. Unless otherwise specified, we mostly set $\lambda = 37$, which dictates that qualifying objects must not deviate above $\epsilon = 5\%$ from their actual appearance likelihood, and these results can be trusted with $1 - \delta = 97\%$ probability at least.

Next, we examine the total query evaluation cost per timestamp for each of the two query workloads *MOD* and *POP*, assuming diverse sizes of Gaussian uncertainty regions. In fact, standard deviation σ controls the density as well as the extent of the region; e.g., a $\sigma = 200$ meters prescribes a square area of side $6\sigma = 1200$ meters, which is large enough for urban settings. Quite predictably, execution cost deteriorates with σ as plotted in Fig. 6, because larger uncertainty regions intersect more frequently with multiple query ranges. Despite the increasing number of such overlapping cases, the pruning heuristics can quickly discard improbable candidates, hence the total cost for all queries mostly remains at reasonable levels, particularly for the *MOD* workload. It only exacerbates for larger uncertainty regions with the *POP* dataset, but mainly due to the disproportionate size of its contact lists.

The choice of inspection order for elementary boxes is not critical, provided that they are visited by descending weight. Thanks to its simplicity, a spiroid ordering gives response slightly faster than its ripplewise counterpart (Fig. 7). Still, Fig. 8 demonstrates that such optimizations economize enormously by first examining important boxes as opposed to a naïve strategy. With more restrictions on accuracy (i.e., larger λ), execution time escalates linearly, but always remains under 30 sec per cycle for answering all queries. In contrast, blindly examining all boxes and employing expensive Monte-Carlo simulations for ambiguous cases incurs execution times utterly incompatible with online monitoring.

With respect to threshold values, Fig. 9 shows the effectiveness of pruning for diverse uncertainty levels. Clearly, the higher the threshold, the more frequent the cases of eager rejections, as examination of objects terminates very early. This trend gets even more pronounced with greater uncertainty ($\sigma = 300$m).

When specifying diverse areas of query range, execution cost fluctuates, as illustrated in Fig. 10. However, this phenomenon depends on the extent and

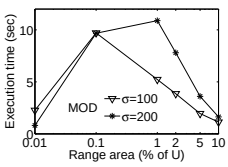


Fig. 10.

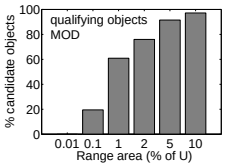


Fig. 11.

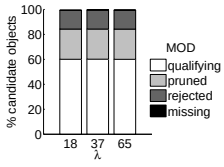


Fig. 12.

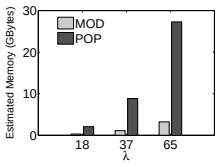


Fig. 13.

spread of the uncertainty regions that may cause a mounting number of partial overlaps with the query rectangles, which require verification. For smaller query areas, such intersections are rare, so they incur negligible cost. Similarly, with ranges equal to 10% of the entire universe $\mathcal{U}$, many more objects fall completely within range and get directly qualified with less cost. This is also confirmed with statistical results in Fig. 11 regarding the fraction of candidate objects that finally get qualified for $\sigma = 200$m. For ranges with extent 1% of $\mathcal{U}$, about 60% of candidates are reported, so a lot many of the rest 40% have been disqualified after verification, which explains the respective peak in Fig. 10.

Concerning the quality of the reported results, Fig. 12 plots a breakdown of the candidates for varying accuracy levels. Compared with an exhaustive Monte-Carlo evaluation, about 15% of candidates are eagerly rejected, while another 25% is pruned. Most importantly, false negatives are less than 0.1% at all cases, which demonstrates the efficiency of our approach. Although qualitative results are similar for varying λ, they still incur differing execution costs (Fig. 8).

The astute reader may have observed that our approach is not incremental; at every cycle, each candidate must be examined from scratch for any query, no matter its previous state regarding the given query. This is a deliberate choice, if one considers the extreme mutability of both objects and queries. Apart from their continuous free movement and features that change in probabilistic fashion, there are also practical implications. Figure 13 plots the estimated memory consumption for maintaining states of every verifier for all combinations of queries and members of their contact lists. This cost may seem reasonable for fair accuracy constraints ($\lambda = 18$), but becomes unsustainable with stricter quality requirements, especially for query workloads with excessively large membership. Considering its maintenance overhead, a stateful approach would clearly become more a burden rather than an assistance in terms of probabilistic evaluation.

7 Conclusion

In this work, we proposed a probabilistic methodology for providing online response to multiple range requests over streams of Gaussian positional uncertainty in GeoSocial networks. Abiding to privacy preserving protocols, we introduced an (ϵ, δ)-approximation framework, as a trade-off between quality guarantees and timeliness of results. We also developed optimizations for effective pruning and eager rejection of improbable answers. Our evaluation strategy drastically

reduces execution cost and offers answers of tolerable error, confirmed by an extensive experimental study over massive synthetic datasets.

References

1. T. Bernecker, T. Emrich, H.-P. Kriegel, M. Renz, S. Zankl, and A. Züfle. Efficient Probabilistic Reverse Nearest Neighbor Query Processing on Uncertain Data. *PVLDB*, 4(10):669-680, 2011.
2. T. Bernecker, H.-P. Kriegel, N. Mamoulis, M. Renz, and A. Züfle. Continuous Inverse Ranking Queries in Uncertain Streams. In *SSDBM*, pp. 37-54, 2011.
3. C. Böhm, A. Pryakhin, and M. Schubert. Probabilistic Ranking Queries on Gaussians. In *SSDBM*, pp. 169-178, 2006.
4. J. Chen and R. Cheng. Efficient Evaluation of Imprecise Location-Dependent Queries. In *ICDE*, pp. 586-595, 2007.
5. R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J.S. Vitter. Efficient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data. In *VLDB*, pp. 876-887, 2004.
6. C.-Y. Chow, M.F. Mokbel, and W.G. Aref. Casper*: Query Processing for Location Services without Compromising Privacy. *ACM TODS*: 34(4):24, 2009.
7. G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private Queries in Location Based Services: Anonymizers are not Necessary. In *SIGMOD*, pp. 121-132, 2008.
8. Y. Ishikawa, Y. Iijima, J. Xu Yu. Spatial Range Querying for Gaussian-Based Imprecise Query Objects. In *ICDE*, pp. 676-687, 2009.
9. H.-P. Kriegel, P. Kunath, M. Pfeifle, and M. Renz. Probabilistic Similarity Join on Uncertain Data. In *DASFAA*, pp. 295-309, 2006.
10. H.-P. Kriegel, P. Kunath, and M. Renz. Probabilistic Nearest-Neighbor Query on Uncertain Objects. In *DASFAA*, pp. 337-348, 2007.
11. X. Lian and L. Chen. Ranked Query Processing in Uncertain Databases. *IEEE TKDE*, 22(3):420-436, 2010.
12. X. Lian and L. Chen. Similarity Join Processing on Uncertain Data Streams. *IEEE TKDE*, 23(11):1718-1734, 2011.
13. S. Mascetti, D. Freni, C. Bettini, X.S. Wang, and S. Jajodia. Privacy in Geo-social Networks: Proximity Notification with Untrusted Service Providers and Curious Buddies. *VLDB Journal*, 20(4):541-566, 2011.
14. J. Pei, M. Hua, Y. Tao, and X. Lin. Query Answering Techniques on Uncertain and Probabilistic Data: Tutorial Summary. In *SIGMOD*, pp. 1357-1364, 2008.
15. L. Šikšnys, J.R. Thomsen, S. Šaltenis, and M.L. Yiu. Private and Flexible Proximity Detection in Mobile Social Networks. In *MDM*, pp. 75-84, 2010.
16. Y. Tao, R. Cheng, X. Xiao, W. Ngai, B. Kao and S. Prabhakar. Indexing Multi-Dimensional Uncertain Data with Arbitrary Probability Density Functions. In *VLDB*, pp. 922-933, 2005.
17. Y. Tao, X. Xiao, and R. Cheng. Range Search on Multidimensional Uncertain Data. *ACM TODS*, 32(3):15, 2007.
18. M. Zhang, S. Chen, C.S. Jensen, B.C. Ooi, and Z. Zhang. Effectively Indexing Uncertain Moving Objects for Predictive Queries. *PVLDB*, 2(1):1198-1209, 2009.