Efficient Distance-based Outlier Detection on Uncertain Dataset of General Gaussian Distribution

Salman Ahmed Shaikh and Hiroyuki Kitagawa
Graduate School of Systems and Information Engineering
University of Tsukuba
Tennodai, Tsukuba, Ibaraki 305-8573, Japan
Email: salman@kde.cs.tsukuba.ac.jp and kitagawa@cs.tsukuba.ac.jp

***Abstract*—Uncertain data management, querying and mining have become important because the majority of real world data is accompanied with uncertainty these days. Uncertainty in data is often caused by the deficiency in underlying data collecting equipments or sometimes manually introduced to preserve the data privacy. This work discusses the problem of distance-based outlier detection on uncertain datasets of Gaussian distribution. We start with the Naive approach. We then present a Cell-based approach to increase the efficiency of outlier detection. The infinite nature of Gaussian distribution prevents us to devise very effective pruning methodologies. Therefore we propose an approximate approach using Bounded Gaussian distribution. Approximating Gaussian distribution by Bounded Gaussian distribution enables an approximate but more efficient Cell-based approach along with simple object-wise distance method. An extensive empirical study on synthetic data shows that our proposed approaches are scalable, efficient and accurate.**

***Index Terms*—Distance-based Outlier Detection; Cell Based Approach; Uncertain Data; Bounded Gaussian Distribution.**

I. Introduction

Outlier detection is one of the most important data mining techniques with a vital importance in many application domains including credit card fraud detection [18], network intrusion detection [16], environment monitoring [19], medical sciences [17] etc. Although there exists no any universally agreed upon definition of outliers, yet some definitions are general enough to give a basic idea of outliers. Hawkins [6] defines an outlier as an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism. In [7], Barnet and Lewis mentioned that an outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs.

Most of the earliest outlier detection techniques were given by statistics. In statistics over 100 outlier detection techniques have been developed for different circumstances, depending on the data distribution, whether or not the distribution parameters are known, the number of expected outliers and the type of expected outliers [7], [20]. However, most statistical techniques are univariate, and in the majority of techniques, the parameter of distribution may be difficult to determine. In order to overcome problems in statistical techniques several outlier detection approaches have been proposed in data mining [5],[13], [8], [9], [14], [15].

Due to the incremental usage of sensors, RFIDs and similar devices for data collection these days, data contains certain degree of inherent uncertainty. The causes of uncertainty may include limitation of equipments, absence of data and delay or loss of data in transfer. In order to get reliable results from such data, uncertainty needs to be considered in calculation. In this work we propose a notion of distance-based outliers on uncertain datasets.

In the following, uncertainty of data is modelled by the most commonly used PDF, i.e., Gaussian distribution. Since the actual distance function is very costly to compute, strong pruning techniques are required to prune as many objects as possible without the use of actual distance function. Therefore we propose a Cell-based approach of outlier detection on uncertain datasets of Gaussian distribution. The proposed approach prunes the majority of cells as inliers or outliers. For all the un-pruned objects from the Cell-based calculation, we follow Naive approach by utilizing Grid-file index. In some cases, when there are a lot of un-pruned objects from the Cell-based calculation, this approach becomes costly. Therefore we propose an approximate but very efficient Cell-based approach of outlier detection on uncertain datasets of Gaussian distribution. The basic idea is that the Gaussian distribution can be appropriately approximated by Bounded Gaussian distribution [2], and the outlier detection computation is less costly for Bounded Gaussian distribution since the degree of uncertainty is bounded. Actually this bounded distribution allows us to introduce strong pruning techniques at a small cost of accuracy.

The rest of the paper is organized as follows. Section II surveys the previous work related to ours. Section III deals with the basic concepts and formally defines distance-based outlier detection on uncertain datasets. Naive approach of distance-based outlier detection on uncertain datasets of Gaussian distribution is given in Section IV. The Cell-based approach is presented in Section V. In Section VI, an approximate approach of distance-based outlier detection based on Bounded Gaussian distribution is given. Section VII contains an extensive experimental evaluation that demonstrates the efficiency and accuracy of proposed techniques. Section VIII concludes our paper.

II. Related Work

Outlier detection is a well studied area of data mining. Different authors have classified this area differently. The problem of outlier detection has been classified into statistical approaches, depth-based approaches, deviation-based approaches, distance-based approaches, density-based approaches and high-dimensional approaches by [10].

Distance-based outliers detection approach on deterministic data was introduced by Knorr, et al. in [5]. They defined a point p to be an outlier if at most M points are within D-distance of the point. They also presented a Cell-based algorithm to efficiently compute the distance-based outliers. [11] formulated distance-based outliers based on the distance of a point from its kth nearest neighbour. The points were ranked on the basis of its distance to its kth nearest neighbour and the top n points were declared as outliers in this ranking. Recently in [8], the authors assessed and evaluated several distance-based outlier detection approaches and highlighted a family of state of the art distance-based outlier detection algorithms.

Recently a lot of research has focused on managing, querying and mining of uncertain datasets [12], [9]. The problem of outlier detection on uncertain datasets was first studied by Aggarwal, et al. in [12]. They represented an uncertain object by a PDF. They defined an uncertain object o to be a density-based (δ, η) outlier, if the probability of o existing in some subspace of a region with density at least η is less than δ. In [9], the authors proposed the distance-based outlier detection on uncertain datasets. In their approach, each tuple in the uncertain table is associated with an existential probability. Moreover in their work, possible world semantic was used to mine the outliers.

In our work, uncertainty of objects' attribute values is modelled by Gaussian distribution and we utilize Gaussian difference distribution to detect probabilistic outliers. This paper is an extended version of our previous paper [1]. Main contributions of this paper include an approximate approach to outlier detection using Bounded Gaussian distribution (Section VI). To the best of our knowledge, distance-based outlier detection on uncertain datasets of Gaussian distribution has not been studied by other researchers.

III. Distance-based Outliers in Uncertain datasets

The very first definition of distance-based outlier detection on certain data was given by Knorr, et al. in [5]. They defined distance-based outliers as follows.

Definition 1: An object o in a dataset DB is a distance-based outlier, if at least fraction p of the objects in DB lies greater than distance D from o.

In this work, our focus is uncertain datasets whose attribute values are uncertain. We assume that the uncertainty is given by Gaussian distribution. We choose the Gaussian distribution, because in statistics the Gaussian distribution *(or normal distribution)* is the most important and the most commonly used distribution. In the following, we consider k-dimensional uncertain objects o_i, with attribute $\overrightarrow{\mathcal{A}_i} = (x_{i,1}, ..., x_{i,k})^T$ following Gaussian PDF with mean $\overrightarrow{\mu_i} = (\mu_{i,1}, ..., \mu_{i,k})^T$ and co-variance matrix $\Sigma_i = diag(\sigma^2_{i,1}, ..., \sigma^2_{i,k})$, respectively. The complete database consists of a set of such objects, $\mathcal{GDB} = \{o_1, ..., o_N\}$, where $N = |\mathcal{GDB}|$ is the number of uncertain objects in $\mathcal{GDB}$. The vector $\overrightarrow{\mathcal{A}_i}$ is a random variable that follows Gaussian distribution $\overrightarrow{\mathcal{A}_i} \sim \mathcal{N}(\overrightarrow{\mu_i}, \Sigma_i)$.

We assume that the observed coordinates (attribute values) are vectors $\overrightarrow{\mu_i}$ of the objects which follow Gaussian distribution. Based on this assumption, in the rest of the paper we will use $\overrightarrow{\mu_i}$ to denote the real observed coordinates (attribute values) of object o_i.

We naturally extend Definition 1 for uncertain datasets as follows.

Definition 2: An uncertain object o in a database $\mathcal{GDB}$ is a distance-based outlier, if the expected number of objects $o_i \in \mathcal{GDB}$ (including o itself) lying within D-distance of o is less than or equal to threshold $\theta = N(1-p)$, where N is the number of uncertain objects in database $\mathcal{GDB}$, and p is the fraction of objects in $\mathcal{GDB}$ that lies farther than D-distance of o.

According to the definition above, the set of uncertain distance-based outliers in $\mathcal{GDB}$ is defined as follows.

$$\mathcal{GDBOutliers} = \{o_i \in \mathcal{GDB} | \sum_{j=1}^{|\mathcal{GDB}|} Pr(dist(o_i, o_j) \leq D) \leq \theta\} . \quad (1)$$

We call objects that lie within the D-distance of an object o as D-neighbours of o, and denote the set of D-neighbours of o as $DN(o)$. In order to find distance-based outliers in $\mathcal{GDB}$, the distance between uncertain objects need to be calculated. Fortunately the distance between two objects which follow Gaussian distribution is given by another distribution known as the Gaussian difference distribution [3].

Let $\overrightarrow{\mathcal{A}_i}$ and $\overrightarrow{\mathcal{A}_j}$ be two k-dimensional normal random vectors with means $\overrightarrow{\mu_i} = (\mu_{i,1}, ..., \mu_{i,k})^T$ and $\overrightarrow{\mu_j} = (\mu_{j,1}, ..., \mu_{j,k})^T$ and diagonal covariance matrices $\Sigma_i = diag(\sigma^2_{i,1}, ..., \sigma^2_{i,k})$ and $\Sigma_j = diag(\sigma^2_{j,1}, ..., \sigma^2_{j,k})$, respectively. We use $Pr(o_i, o_j, D)$ to denote the probability that $o_j \in DN(o_i)$. Then,

$$Pr(o_i, o_j, D) = \int_R \mathcal{N}(\overrightarrow{\mu_{i-j}}, \Sigma_{i-j}) \mathrm{d}\overrightarrow{\mathcal{A}} , \quad (2)$$

where $\overrightarrow{\mu_{i-j}} = \overrightarrow{\mu_i} - \overrightarrow{\mu_j}$, $\Sigma_{i-j} = \Sigma_i + \Sigma_j$ and R is a sphere with centre $\overrightarrow{\mu_{i-j}}$ and radius D.

Here we only give the expression for 2-dimensional case. Let o_i and o_j be two 2-dimensional uncertain objects with attributes $\overrightarrow{\mathcal{A}_i} \sim \mathcal{N}(\overrightarrow{\mu_i}, \Sigma_i)$ and $\overrightarrow{\mathcal{A}_j} \sim \mathcal{N}(\overrightarrow{\mu_j}, \Sigma_j)$, where $\overrightarrow{\mu_i} = (\mu_{i,1}, \mu_{i,2})^T$, $\overrightarrow{\mu_j} = (\mu_{j,1}, \mu_{j,2})^T$, $\Sigma_i = diag(\sigma^2_{i,1}, \sigma^2_{i,2})$ and $\Sigma_j = diag(\sigma^2_{j,1}, \sigma^2_{j,2})$. $Pr(o_i, o_j, D)$ is given as follows.

$$Pr(o_i, o_j, D) = \frac{1}{2\pi\sqrt{(\sigma_{i,1}^2 + \sigma_{j,1}^2)(\sigma_{i,2}^2 + \sigma_{j,2}^2)}} \int_0^D \int_0^{2\pi} \exp\left\{-\left(\frac{(r\cos\theta - \alpha_1)^2}{2(\sigma_{i,1}^2 + \sigma_{j,1}^2)} + \frac{(r\sin\theta - \alpha_2)^2}{2(\sigma_{i,2}^2 + \sigma_{j,2}^2)}\right)\right\} r\, d\theta\, dr\,, \tag{3}$$

where $\alpha_1 = \mu_{i,1} - \mu_{j,1}$ and $\alpha_2 = \mu_{i,2} - \mu_{j,2}$.

Proof. See appendix.

In the following part, we focus our discussion to 2-dimensional case. However, the discussion can be extended to higher dimensional cases.

IV. Naive Approach

After giving the definition and basic concepts of distance-based outliers in uncertain datasets of Gaussian distribution in Section III, we now present the Naive approach of distance-based outlier detection.

The Naive approach of distance-based outlier detection uses Nested-loop. Algorithm 1 gives the Naive approach of distance-based outliers. The approach includes the evaluation of the distance function between each object $o_i \in \mathcal{GDB}$ and every other object in the $\mathcal{GDB}$ until o_i can be decided as an outlier or inlier. In the worst case this approach requires the evaluation of $O(N^2)$ distance functions. Usually it is very expensive.

Algorithm 1 The Naive Approach

Input: $\mathcal{GDB}$, D, p
Output: Distance-based Outliers

```
1:  N ← number of objects in GDB;
2:  θ ← N(1 − p);
3:  for each o_i in GDB do
4:      EN_oi ← 0;
        (EN_oi corresponds the expected number of objects that
        lie within the D-distance of o_i)
5:      for each o_j in GDB do
6:          EN_oi = EN_oi + Pr(o_i, o_j, D);
7:          if EN_oi > θ then
8:              mark o_i as inlier, GOTO next o_i;
9:          end if
10:     end for
11:     mark o_i as outlier;
12: end for
13: return set of outliers;
```

V. Cell-based Approach

The Naive approach requires a lot of computation time to detect outliers even from a small dataset due to the costly distance calculation. In the following we discuss the Cell-based approach of distance-based outlier detection, which can significantly reduce the number of distance function evaluations. The approach first maps database objects to a Cell-grid structure and then prunes the majority of objects by identifying the cells containing only inliers or outliers like the Cell-based approach of Knorr, et al. [5]. Since the original Cell-based approach by Knorr, et al. deals with certain data, they considered only two cell layers that lie within certain distances from a target cell for pruning. However, in our case, objects are infinitely uncertain, hence we have to consider all the cell-layers in the Grid for the pruning of the target cell. For un-pruned objects in un-pruned cells, Grid file indexing is utilized to further reduce the number of distance function computations.

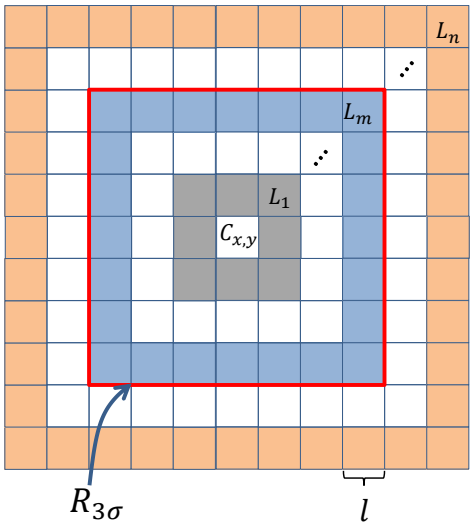


Fig. 1: Cell Layers

A. Grid Structure

In order to find distance-based outliers on an uncertain dataset, we first quantize each object $o_i \in \mathcal{GDB}$, to k-dimensional space that is partitioned into cells of length l. (The cell length is discussed in Section V-E.) Let $C_{x,y}$ be any cell in the Grid, then the neighbouring cells of $C_{x,y}$ form layers around it as shown in Fig.1. Layers of any cell $C_{x,y}$ in the Grid are defined as follows.

$$L_i(C_{x,y}) = \{C_{u,v} | u = x \pm i, v = y \pm i, C_{u,v} \notin L_{i-1}(C_{x,y})\},$$

where $1 \leq i \leq n$. A cell $C_{x,y}$ in the Grid can have the maximum number of layers if it exists at the corner of the Grid and the minimum number of layers if it exists at the centre of the Grid. Let n denotes the maximum number of layers, then the minimum number of layers is given by $\lceil n/2 \rceil$. $R_{3\sigma}$ is a region which consists of the cells which lie within $D+3\sigma^+$ distance of $C_{x,y}$, where $\sigma^+ = max(\sigma_1, ..., \sigma_k)$. Let $n_{R3\sigma} = \lceil \frac{D+3\sigma^+}{l} \rceil$, then we can define region $R_{3\sigma}$ of any cell $C_{x,y}$ in the Grid as follows.

$$R_{3\sigma}(C_{x,y}) = \{L_i(C_{x,y}) | 1 \leq i \leq n_{R3\sigma}\}.$$

B. Cells and Layers Bounds

Like the Cell-based approach on certain data by Knorr, et al. [5], our goal is to identify and prune cells which are guaranteed to contain only inliers or outliers. A cell $C_{x,y}$ can be pruned as

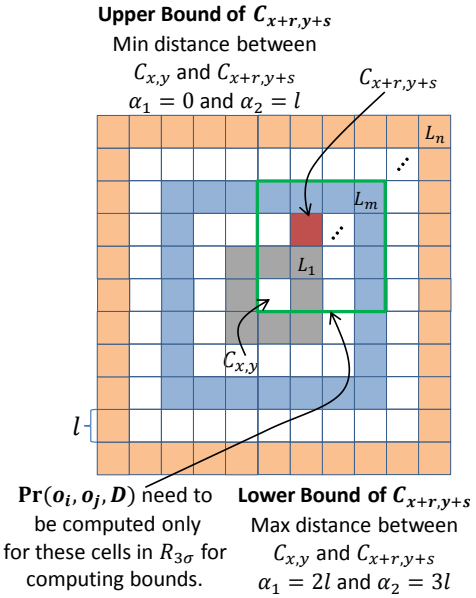


Fig. 2: Cell Bounds

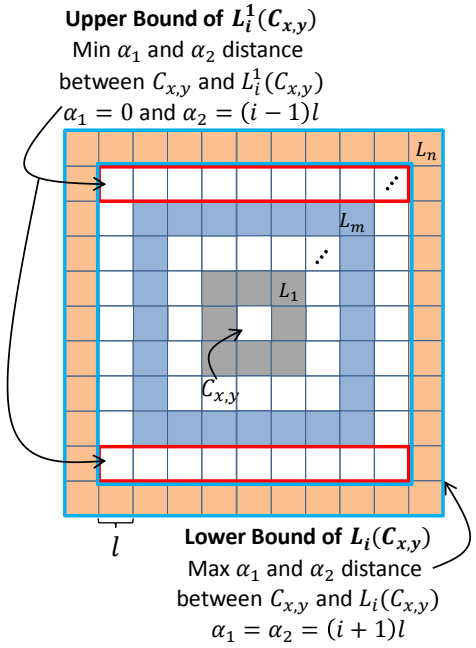


Fig. 3: Layer Bounds

an "outlier cell" if the expected number of D-neighbours for any object in cell $C_{x,y}$ according to Definition 2 is less than or equal to the threshold θ. Similarly a cell can be pruned as an "inlier cell" if the expected number of D-neighbours for any object in cell $C_{x,y}$ is greater than θ. We define upper and lower bounds to prune the cells. The upper and lower bounds bind the possible expected numbers of D-neighbours without expensive object-wise distance computation.

Upper Bound. The upper bound of a cell $C_{x,y}$, $UB(C_{x,y})$, binds the maximum expected number of D-neighbours in $C_{x,y}$ for any object in cell $C_{x,y}$ itself. Since the Gaussian distribution is infinite, two objects in the same cell may reside at the same coordinate with some non-zero probability. Hence $UB(C_{x,y}) = N(C_{x,y})$, where $N(C_{x,y})$ is the number of objects in $C_{x,y}$.

The upper bound of a cell $C_{x+r,y+s}$ $(1 \leq r, s \leq n_{R3\sigma})$ in region $R_{3\sigma}(C_{x,y})$, $UB(C_{x+r,y+s})$, gives the maximum expected number of D-neighbours in $C_{x+r,y+s}$ for any object in cell $C_{x,y}$. It can be obtained by the product of the number of objects in $C_{x+r,y+s}$ and $Pr(o_i, o_j, D)$ with $\alpha_1 = (r-1)l$ and $\alpha_2 = (s-1)l$. Note that, for calculating cell bounds we do not need to evaluate distance function, $Pr(o_i, o_j, D)$, for all the cells in $R_{3\sigma}(C_{x,y})$. Since the Gaussian function is symmetric, $Pr(o_i, o_j, D)$ computed for quarter of the cells in $R_{3\sigma}(C_{x,y})$ can be used for computing the bounds for all the cells in $R_{3\sigma}(C_{x,y})$ as shown in Fig.2.

The upper bound of a cell layer $L_m(C_{x,y})$ $(n_{R3\sigma} < m \leq n)$, $UB(L_m(C_{x,y}))$, gives the maximum expected number of D-neighbours in $L_m(C_{x,y})$ for any object in cell $C_{x,y}$. We calculate the upper bound for each dimension of the layer. Let the cells in dimension 1 of layer m are given by $L_m^1(C_{x,y}) = \{C_{u,v} | u = x \pm m, v = y + m \; OR \; v = y - m, C_{u,v} \notin L_{m-1}(C_{x,y})\}$. Then the upper bound of $L_m^1(C_{x,y})$, $UB(L_m^1(C_{x,y}))$, can be obtained by the product of the number of objects in $L_m^1(C_{x,y})$ cells and $Pr(o_i, o_j, D)$ with $\alpha_1 = 0$ and $\alpha_2 = (m-1)l$. Similarly $UB(L_m^2(C_{x,y}))$ can be obtained by the product of the number of objects in $L_m^2(C_{x,y})$ cells and $Pr(o_i, o_j, D)$ with $\alpha_1 = (m-1)l$ and $\alpha_2 = 0$, as shown in Fig.3.

Lower Bound. The lower bound of the cell $C_{x,y}$, $LB(C_{x,y})$, binds the minimum expected number of D-neighbours for any object in cell $C_{x,y}$ itself. When two objects in the same cell reside at the opposite corners, the probability that they are D-neighbours takes the minimum value. Hence $LB(C_{x,y}) = 1 + (N(C_{x,y} - 1))Pr(o_i, o_j, D)$, where $\alpha_1 = \alpha_2 = l$.

The lower bound of a cell $C_{x+r,y+s}$ $(1 \leq r, s \leq n_{R3\sigma})$ in region $R_{3\sigma}(C_{x,y})$, $LB(C_{x+r,y+s})$, gives the minimum expected number of D-neighbours in $C_{x+r,y+s}$ for any object in cell $C_{x,y}$. It can be obtained by the product of the number of objects in $C_{x+r,y+s}$ and $Pr(o_i, o_j, D)$ with $\alpha_1 = (r+1)l$ and $\alpha_2 = (s+1)l$, as shown in Fig.2.

The lower bound of a cell layer $L_m(C_{x,y})$ $(n_{R3\sigma} < m \leq n)$, $LB(L_m(C_{x,y}))$, gives the minimum expected number of D-neighbours in $L_m(C_{x,y})$ for any object in cell $C_{x,y}$. It can be obtained by the product of the number of objects in the layer $L_m(C_{x,y})$ and $Pr(o_i, o_j, D)$ with $\alpha_1 = \alpha_2 = (m+1)l$, as shown in Fig.3.

Lookup Table. The bounds discussed above are required by each cell $C_{x,y}$ for pruning. Each bound computation requires the evaluation of a costly distance function $Pr(o_i, o_j, D)$ and the object count of respective cell $C_{x,y}$, $R_{3\sigma}(C_{x,y})$ cells or cell layer $L_m(C_{x,y})$. We can reduce the number of distance function computations for the bounds calculation by pre-computing $Pr(o_i, o_j, D)$ values for $C_{x,y}$, $R_{3\sigma}(C_{x,y})$ cells and cell layer $L_m(C_{x,y})$ bounds. Since the $Pr(o_i, o_j, D)$ values are decided only by the α_1 and α_2 values and are independent from the locations of $C_{x,y}$. We need to compute $Pr(o_i, o_j, D)$ values only for $\alpha_1 = (r-1)l$, $\alpha_2 = (s-1)l$ $(1 \leq r, s \leq n_{R3\sigma})$ for $R_{3\sigma}(C_{x,y})$ cells, $\alpha_1 = 0$, $\alpha_2 = (m-1)l$ and $\alpha_1 = (m-1)l$, $\alpha_2 = 0$

$(n_{R3\sigma} < m \leq n)$ for $UB(L_m(C_{x,y}))$ and $\alpha_1 = \alpha_2 = (m+1)l$ $(n_{R3\sigma} < m \leq n)$ for $LB(L_m(C_{x,y}))$. The pre-computed values are stored in a lookup table to be used by the Cell-based pruning method.

C. Pruning of Outlier and Inlier Cells

Having defined bounds and lookup table, a cell $C_{x,y}$ in the Grid can be pruned as an outlier or inlier cell as follows.

If $LB(C_{x,y}) + \sum_{r=1}^{n_{R3\sigma}} \sum_{r=1}^{n_{R3\sigma}} LB(C_{x+r,y+s}) + \sum_{m=n_{R3\sigma}+1}^{n} LB(L_m(C_{x,y}))$ is greater than θ, $C_{x,y}$ is pruned as an inlier cell. On the other hand if $UB(C_{x,y}) + \sum_{r=1}^{n_{R3\sigma}} \sum_{r=1}^{n_{R3\sigma}} UB(C_{x+r,y+s}) + \sum_{m=n_{R3\sigma}+1}^{n} UB(L_m(C_{x,y}))$ is less than or equal to θ, $C_{x,y}$ is pruned as an outlier cell. Algorithm 2 shows the Cell-based pruning method.

D. Grid File Index

Cell-based pruning may leave some of the cells undecided, i.e., they are neither pruned as inliers nor as outliers. For all the uncertain objects in such cells, we have to follow the Naive computation. Our distance function has a property that it produces higher probability for the nearer objects than the farther objects. We can utilize our Grid structure as Grid-file index [4] with no additional indexing cost to retrieve the nearer objects before the farther objects for the computation of the expected number of D-neighbours of un-pruned objects. This helps us decide the un-pruned objects faster than using no index at all, since it can reduce the number of distance function evaluations required. Algorithm 3 shows the processing of un-pruned objects in un-pruned cells after Algorithm 2.

E. Cell Length l

Due to the complexity of our distance function, it is not possible to derive a single cell length l suitable for all the combinations of D and variances. Therefore we conducted several experiments to come up with a suitable cell length.

A general observation from several experiments is that smaller the cell-length, shorter the execution time. However very small cell length may also increase the execution time for Cell-based algorithm as very small cell length increases the number of cells in the Grid exponentially and the time required to compute the lookup table. Therefore we need to check a few cell lengths before reaching the appropriate cell length. A good starting point of the cell length that we found through experiments is the average of standard deviations, i.e., $l = \frac{\sigma_1,\ldots,\sigma_k}{k}$.

Algorithm 2 The Cell Based Pruning

Input: $\mathcal{GDB}$, D, p
Output: Distance-based Outliers

1: Create cell grid *Grid* depending upon dataset values and cell length l;
2: Initialize $Count_k$ of each cell $C_k \in Grid$;
3: Map each object o in $\mathcal{GDB}$ to an appropriate cell C_k, and increment $Count_k$ by 1;
4: $\theta \leftarrow N(1-p)$;
(θ and N correspond to the threshold and the number of objects in $\mathcal{GDB}$, respectively.)
/*$Pr(o_i, o_j, D)$ computation for bounds*/
5: $LB(C_k) = Pr(o_i, o_j, D)$ with $\alpha_1 = \alpha_2 = l$;
6: **for** $r = 1$ to $n_{R3\sigma}$ **do**
7: **for** $s = 1$ to $n_{R3\sigma}$ **do**
8: $LB(C_{r,s}) = Pr(o_i, o_j, D)$ with $\alpha_1 = (r+1)l$ and $\alpha_2 = (s+1)l$;
9: $UB(C_{r,s}) = Pr(o_i, o_j, D)$ with $\alpha_1 = (r-1)l$ and $\alpha_2 = (s-1)l$;
10: **end for**
11: **end for**
12: **for** $m = n_{R3\sigma} + 1$ to n (where n denotes the max number of cell layers) **do**
13: $LB(L_m(C_k)) = Pr(o_i, o_j, D)$ with $\alpha_1 = \alpha_2 = l$;
14: $UB(L_m^1(C_k)) = Pr(o_i, o_j, D)$ with $\alpha_1 = 0$ and $\alpha_2 = (m-1)l$;
15: $UB(L_m^2(C_k)) = Pr(o_i, o_j, D)$ with $\alpha_1 = (m-1)l$ and $\alpha_2 = 0$;
16: **end for**
/*Pruning cells using bounds*/
17: For each non-empty $C_k \in Grid$, If $Count_k * Pr(o_i, o_j, D)_{LB_0} > \theta$, C_k is an inlier cell, mark C_k green;
18: **for each** non-empty, uncoloured C_k in *Grid* **do**
19: **if** $LB(C_k).Count_k + \sum_{r=1}^{n_{R3\sigma}} \sum_{r=1}^{n_{R3\sigma}} LB(C_{r,s}).Count_{r,s} + \sum_{m=n_{R3\sigma}+1}^{n} LB(L_m(C_k)).Count_{Lm} > \theta$ ($Count_{r,s}$ and $Count_{Lm}$ denote the object count of cell $C_{r,s}$ and layer $L_m(C_k)$ respectively) **then**
20: C_k is an inlier cell, mark C_k green. GOTO Next C_k;
21: **else if** $UB(C_k).Count_k + \sum_{r=1}^{n_{R3\sigma}} \sum_{r=1}^{n_{R3\sigma}} UB(C_{r,s}).Count_{r,s} + \sum_{m=n_{R3\sigma}+1}^{n} UB(L_m(C_k)).Count_{Lm} \leq \theta$ **then**
22: C_k is an outlier cell, mark C_k black. GOTO Next C_k;
23: **end if**
24: **end for**
25: **return** Objects in black cells $\cup$ Outliers from Algorithm 3;

VI. Outlier Detection using Bounded Gaussian Approximation

Despite the techniques presented in Section V, the un-bounded nature of Gaussian distribution may prevent us from computing outliers very efficiently. The Cell-based method tends to take much time when there are a lot of un-pruned objects. Hence we propose an approximate distance-based outlier detection approach using Bounded Gaussian distribution. Approximating Gaussian distribution by Bounded Gaussian distribution enables an approximate but more efficient Cell-based approach along with simple object-wise distance pruning method. Since in the Gaussian distribution, about 95.45% of the values lie within 2 standard deviations of the mean and 99.73% of the values lie within 3 standard deviations of the mean [2], infinite Gaussian distribution can be normalized within certain boundaries to increase the efficiency of outlier detection at a small cost of accuracy.

Algorithm 3 Processing Un-pruned Objects

Input: $Grid$, D, θ
Output: Distance-based Outliers

for each non-empty, uncoloured C_k in $Grid$ **do**
  for each o_i in C_k **do**
    $EN_{oi} \leftarrow 0$;
    (EN_{oi} corresponds the expected number of objects that lie within the D-distance of o_i)
    for each o_j in C_k and higher layers of C_k in $Grid$ **do**
      $EN_{oi} \leftarrow EN_{oi} + Pr(o_i, o_j, D)$;
      if $EN_{oi} > \theta$ **then**
        o_i can not be outlier, GOTO next o_i;
      end if
    end for
    mark o_i as outlier;
  end for
end for
return set of outliers;

Given a conventional Gaussian function $g(\overrightarrow{\mathcal{A}})$ with mean $\overrightarrow{\mu} = (\mu_1, ..., \mu_k)^T$ and co-variance matrix $\Sigma = diag(\sigma_1^2, ..., \sigma_k^2)$, we define the Bounded Gaussian distribution $f(\overrightarrow{\mathcal{A}})$ following the practise of [21], as follows.

$$f(\overrightarrow{\mathcal{A}}) = \begin{cases} \frac{g(\overrightarrow{\mathcal{A}})}{\int_{\overrightarrow{\mathcal{A}} \in o.ur} g(\overrightarrow{\mathcal{A}}) \mathrm{d}\overrightarrow{\mathcal{A}}} & \overrightarrow{\mathcal{A}} \in o.ur \\ 0 & otherwise \end{cases} \tag{4}$$

where $o_i.ur$ denotes the uncertainty region of the Bounded Gaussian distribution. Here we assume that the uncertainty region is the ellipsoid with the centre at $(\mu_1, ..., \mu_k)^T$ and the radius $(r_1, ..., r_k)$. Note that

$$\int_{o.ur} f(\overrightarrow{\mathcal{A}}) \mathrm{d}\overrightarrow{\mathcal{A}} = 1.$$

When two objects o_i and o_j follow the Bounded Gaussian distribution, the probability $Pr(o_i, o_j, D)$ is given as follows.

$$Pr(o_i, o_j, D) = \int_{-r_k}^{r_k} \int_{x_{j,k}-D}^{x_{j,k}+D} ... \int_{-r_1}^{r_1} \int_{x_{j,1}-D}^{x_{j,1}+D} f(\overrightarrow{\mathcal{A}_i}).f(\overrightarrow{\mathcal{A}_j}) \mathrm{d}x_{i,1} \mathrm{d}x_{j,1} ... \mathrm{d}x_{i,k} \mathrm{d}x_{j,k}. \tag{5}$$

The basic idea of our approximation approach here is to approximate an uncertain dataset $\mathcal{GDB} = \{o_1, ..., o_N\}$ with the conventional Gaussian distribution by $\mathcal{GDB}_b = \{o_1, ..., o_N\}$ with the corresponding Bounded Gaussian distribution with radius $(r_1, ..., r_k)$. Namely, $\mathcal{GDB}_b = \{o_1, ..., o_N\}$ denotes a set of objects whose attributes $\overrightarrow{\mathcal{A}_i} = (x_{i,1}, ..., x_{i,k})^T$ follow the Bounded Gaussian distribution with mean $\overrightarrow{\mu_i} = (\mu_{i,1}, ..., \mu_{i,k})^T$, co-variance matrix $\Sigma_i = diag(\sigma_1^2, ..., \sigma_k^2)$, respectively and radius $(r_1, ..., r_k)$.

A. Cell-based Approach for Bounded Gaussian

Approximating Gaussian distribution by Bounded Gaussian distribution enables better Cell-based pruning. The proposed approach first maps dataset objects to a Cell-grid structure and then prunes the majority of objects by identifying the cells containing only inliers or outliers just like the Cell-based approach for conventional Gaussian distribution. However, in contrast to computing the bounds using potentially many layers, we set the cell size in such a way that the target cell can be pruned by just counting objects within the target cell and the neighbouring layers.

1) Grid Structure for Bounded Gaussian: In order to find distance-based outliers on uncertain datasets of Bounded Gaussian distribution, we map each object $o_i \in \mathcal{GDB}_b$ to the k-dimensional space that is partitioned into cells of length l_b. (The cell length is discussed in Section VI-A2.) Let $C_{x,y}$ be any cell in the Grid and $r^+ = max(r_1, ..., r_k)$, then $R_1(C_{x,y})$ cells are the cells which completely lie within $D - 2r^+$ distance of $C_{x,y}$, including $C_{x,y}$ itself, as shown in Fig. 4. Let $n_{R1} = \lfloor \frac{D-2r^+}{l_b} \rfloor - 1$, then we can define $R_1(C_{x,y})$ as follows.

$$R_1(C_{x,y}) = \{C_{u,v} | u = x \pm n_{R1}, v = y \pm n_{R1}, \sqrt{((|u|+1)l_b)^2 + ((|v|+1)l_b)^2} < D - 2r^+, C_{u,v} \neq C_{x,y}\}.$$

The number of $R_1(C_{x,y})$ cells vary depending upon n_{R1}. Note that $R_1(C_{x,y})$ satisfies the following property.

Property 1. If $C_{u,v} \in R_1(C_{x,y})$, then any object $o_j \in C_{u,v}$ and $o_i \in C_{x,y}$ are at most $D - 2r^+$ distance apart.

From property 1, we can say that objects in $C_{x,y}$ and $R_1(C_{x,y})$ are guaranteed to be D-neighbours mutually, hence the $Pr(o_i, o_j, D)$ is always equal to 1.

The $R_2(C_{x,y})$ cells are those which fall within $D + 2r^+$ distance of $C_{x,y}$. Let $n_{R2} = \lceil \frac{D+2r^+}{l_b} \rceil$, then we can define $R_2(C_{x,y})$ cells as follows.

$$R_2(C_{x,y}) = \{C_{u,v} | u = x \pm n_{R2}, v = y \pm n_{R2}, \sqrt{((|u|-1)l_b)^2 + ((|v|-1)l_b)^2} < D + 2r^+, C_{u,v} \notin R_1(C_{x,y}), C_{u,v} \neq C_{x,y}\}.$$

Note that $R_1(C_{x,y})$ and $R_2(C_{x,y})$ satisfy the following property.

Property 2. If $C_{u,v}$ is neither in $R_1(C_{x,y})$ nor in $R_2(C_{x,y})$ and $C_{u,v} \neq C_{x,y}$, then any object $o_j \in C_{u,v}$ and $o_i \in C_{x,y}$ have distance greater than $D + 2r^+$.

From property 2, we can guarantee that o_i and o_j are greater than $D + 2r^+$ distance apart, hence the $Pr(o_i, o_j.D)$ is always equal to 0.

We define two more types of cells that can help in pruning cells. We name these cells as "red cells" and "pink cells" and are denoted by $R_r(C_{x,y})$ and $R_p(C_{x,y})$, respectively. Let $n_{R_1}^{diag} = \lfloor \frac{D-2r^+}{l_b \sqrt{2}} \rfloor - 1$ denotes the number of diagonals within

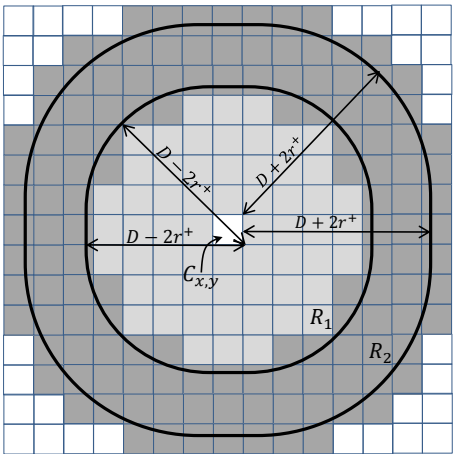


Fig. 4: Bounded Gaussian Cell Grid

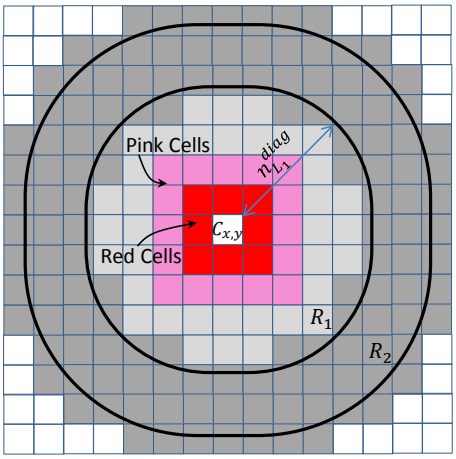


Fig. 5: Red and Pink Cells

$D-2r^+$ distance of any cell $C_{x,y}$. Then we can define the red cells and the pink cells as follows.

$$R_r(C_{x,y}) = \{C_{u,v} | u = x \pm n_r, v = y \pm n_r, C_{u,v} \neq C_{x,y}\};$$

$$n_r = min\{n \mid N(C_{x,y}) \cup \sum_{i=1}^{n} N(L_i(C_{x,y})) > \theta\,,$$

$$0 < n \leq \lfloor \frac{n_{R_1}^{diag}}{2} \rfloor\}.$$

where $L_i(C_{x,y}) = \{C_{u,v} | u = x \pm i, v = y \pm i\}$ and $N(C_{x,y})$ and $N(L_i(C_{x,y}))$ denote the number of objects in $C_{x,y}$ and $L_i(C_{x,y})$, respectively. The n_r value which meets the above condition may not exist. If it exists, we can define $R_p(C_{x,y})$ as follows.

$$R_p(C_{x,y}) = \{C_{u,v} | u = x \pm n_p, v = y \pm n_p, C_{u,v} \notin R_r(C_{x,y}), C_{u,v} \neq C_{x,y}, n_r < n_p < n_{R_1}^{diag}\}.$$

$R_r(C_{x,y})$ is chosen in such a way that if the total number of objects in $C_{x,y}$ and $R_r(C_{x,y})$ is greater than threshold θ, then they can prune all objects in cell $C_{x,y}$, $R_r(C_{x,y})$ and $R_p(C_{x,y})$ as inliers. n_r is smaller the better, since smaller n_r results in larger n_p, hence more cells can be pruned as inliers.

For example, in Fig. 5, $n_{R_1}^{diag} = 3$, and we assume $n_r = 1$. Namely, we assume the total number of objects in $C_{x,y}$ and $L_1(C_{x,y})$ is greater than θ. Then, $n_p = 2$, and all the objects in $C_{x,y}$, $L_1(C_{x,y})$ and $L_2(C_{x,y})$ are inliers. Note that the combined thickness of $C_{x,y}$ and $R_r(C_{x,y})$ is always less than $n_{R_1}^{diag}$, hence they can prune cells in $R_p(C_{x,y})$, just like $C_{x,y}$ can prune $R_1(C_{x,y})$ cells according to Property 3a.

Property 3.

a) If the number of objects in $C_{x,y}$ is greater than θ, none of the objects in $C_{x,y}$ and $R_1(C_{x,y})$ is an outlier.

b) If the number of objects in $C_{x,y} \cup R_1(C_{x,y})$ is greater than θ, none of the objects in $C_{x,y}$ is an outlier.

c) If the number of objects in $C_{x,y} \cup R_r(C_{x,y})$ is greater than θ, none of the objects in $C_{x,y}$, $R_r(C_{x,y})$ and $R_p(C_{x,y})$ is an outlier.

d) If the number of objects in $C_{x,y} \cup R_1(C_{x,y}) \cup R_2(C_{x,y})$ is less than or equal to θ, every object in $C_{x,y}$ is an outlier.

Algorithm 4 shows the Cell-based calculation for Bounded Gaussian distribution.

Algorithm 4 The Cell Based Pruning for Bounded Gaussian Dist.

Input: $\mathcal{GDB}_b$, D, p

Output: Distance-based Outliers on Bounded Gaussian ($\mathcal{GDB}_b Outliers$)

1: Create cell $Grid$ depending upon dataset values and cell length l_b and initialize the count of each cell $C_k \in Grid$, $Count_k \leftarrow 0$;
2: Map each object o in $\mathcal{GDB}_b$ to an appropriate cell C_k, and increment $Count_k$ by 1;
3: $\theta \leftarrow N(1-p)$;
(θ and N correspond to the threshold and the number of objects in $\mathcal{GDB}_b$, respectively.)
/* Cell-based Pruning */
4: For each non-empty $C_k \in Grid$, If $Count_k > \theta$, C_k is an inlier cell, mark C_k *green*;
5: For each *green* cell $C_k \in Grid$, mark $R_1(C_k)$ cells *blue*, provided the neighbour has not already been marked *green*;
6: For each non-empty, uncoloured cell $C_k \in Grid$, If $Count_k + \sum_{m \in R_r(C_k)} Count_m > \theta$, mark C_k, $R_r(C_k)$ and $R_p(C_k)$ *blue*;
7: **for each** non-empty, uncoloured cell C_k in $Grid$ **do**
8: $Count_{k2} \leftarrow Count_k + \sum_{m \in R_1(C_k)} Count_m$;
9: **if** $Count_{k2} > \theta$ **then**
10: Mark C_k *blue*, all the objects in C_k are inliers;.
11: **else if** $Count_{k2} + \sum_{m \in R_2(C_k)} Count_m \leq \theta$ **then**
12: Mark C_k *black*, all the objects in C_k are outliers;
13: **end if**
14: **end for**
15: $\mathcal{GDB}_b Outliers$ = Objects in black cells $\cup$ Outliers from Algorithm 5;
16: **return** $\mathcal{GDB}_b Outliers \cup$ Outliers from Algorithm 3;

2) Cell Length l_b for Bounded Gaussian: Due to the complexity of our distance function, it is not possible to find a single cell length l_b suitable for all the combinations of D and radius. We are interested in finding the expected number of D-neighbours for the target cell $C_{x,y}$. This D-distance forms a sphere type shape around $C_{x,y}$. If cell size is small, pruning can be done in finer details and more accurately. However, very small cell length can also increase the execution time of our algorithm due to the exponential increase in the number of cells. Therefore we need a cell length compromising both, i.e., pruning power and execution time. A good approximate of cell length that we found through experiments is the average of standard deviations, as was used in the Cell-based algorithm of the conventional Gaussian distribution, i.e., $l = \frac{\sigma_1,\ldots,\sigma_k}{k}$.

B. Simple Object-wise Distance Pruning

The Cell-based calculation prunes cells as inlier or outlier cells depending upon the count of objects in cell $C_{x,y}$ and neighbouring layers. It may leave some cells undecided, i.e., they are neither pruned as inlier cells nor as outlier cells. If the number of un-pruned objects is large, they can degrade the performance. Hence we propose a technique based on minimum bounding rectangles (MBRs)[25] and inexpensive distance to reduce the number of costly distance function computations for un-pruned objects. This can increase the efficiency of our algorithm.

Since we assume the Gaussian distribution is bounded, we can use an MBR to represent the uncertain region of an object. Using MBR we can tell whether two objects are within the D-distance only by calculating the distance between their MBRs. The distance computation in this case is just ordinary Euclidean distance calculation and cheap. Let $mindist(o_i, o_j)$ and $maxdist(o_i, o_j)$ denote the minimum and maximum ordinary Euclidean distances between MBRs of two objects $o_i, o_j \in \mathcal{G}DB_b$ respectively. If $maxdist(o_i, o_j) \leq D$, it is guaranteed that object o_j lies within D-distance of o_i. In other words, $Pr(o_i, o_j, D) = 1$. On the other hand, if $mindist(o_i, o_j) > D$, then object o_j is guaranteed to lie outside the D distance of o_i. In this case, $Pr(o_i, o_j, D) = 0$. For example, in Fig. 6, $maxdist(o_i, o_p) < D$ and $mindist(o_i, o_q) > D$. Therefore $Pr(o_i, o_p, D) = 1$ and $Pr(o_i, o_q, D) = 0$.

C. Grid File Index for Bounded Gaussian

Just like the Cell-based algorithm for the conventional Gaussian distribution, we make use of the Grid file index for the un-pruned objects in the Cell-based approach. While using pruning technique of Section VI-B, Grid file index helps in retrieving the nearer objects before the farther objects, hence helping in early pruning of the objects. For un-pruned objects from Algorithm 4 and Algorithm 5 we need to follow the Naive computation. Grid file index can help in retrieving nearer objects before the farther objects. This reduces the number of evaluations required for distance function, hence reducing the overall cost of computation.

The Naive computation for the Bounded Gaussian distribution is same as the one presented in Algorithm 3. The only difference between the Naive computation for the Bounded Gaussian and conventional Gaussian distribution is that the former requires only 2 regions, i.e., R_1 and R_2 for Naive outlier detection. In contrast, the conventional Gaussian requires all *Grid* layers for Naive computation.

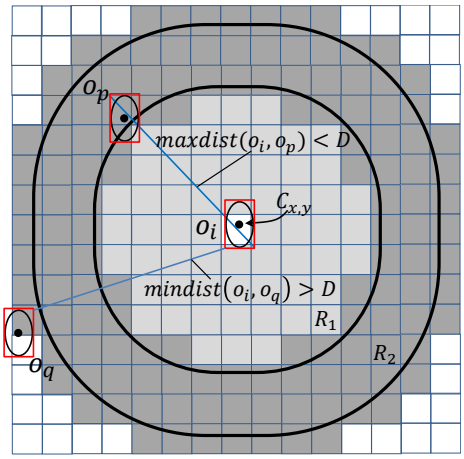


Fig. 6: Simple Object-wise Distance Pruning

Algorithm 5 Further Pruning

Input: *Grid* with $Count_{k2}$ of each cell, D, θ
Output: Distance-based Outliers on Bounded Gaussian

```
1:  for each non-empty uncoloured C_k in Grid do
2:    for each o_i in C_k do
3:      EN_oiLB ← Count_k2 of C_k, EN_oiUB ← Count_k2 of C_k;
        (EN_oiLB & EN_oiUB corresponds the lower and upper
        bound expected number of objects respectively, that
        lie within the D-distance of o_i.)
4:      for each o_j in R_2(C_k) do
5:        if maxdist(o_i, o_j) ≤ D then
6:          EV_oiLB ++ and EV_oiUB ++.
7:        end if
8:      end for
9:      if EN_oiLB > θ then
10:       Mark o_i as inlier.
11:     else if EN_oiUB ≤ θ then
12:       Mark o_i as outlier.
13:     end if
14:   end for
15: end for
16: return set of outliers;
```

VII. Experiments

We conducted extensive experiments on synthetic data to evaluate the effectiveness and accuracy of our proposed algorithms. All algorithms were implemented in C#, Microsoft Visual Studio 2008. All experiments were performed on a system with an Intel Core 2 Duo E8600 3.33GHz CPU and 2GB main memory running Windows 7 Professional OS. All programs run in main memory and no I/O cost is considered.

We have used BoxMuller method [23] for generating pairs

of independent, standard, normally distributed (zero mean, unit variance) random numbers, given a source of uniformly distributed random numbers as shown in Fig 7. The generated dataset $\mathcal{G}DB$ is then normalized to have a domain of [0,1000] on every dimension. For each point z in DB, we create an uncertain object o, whose uncertainty is given by Gaussian distribution with mean z and standard deviation σ in both the dimensions.

Unless specified, the following parameter values are used. $D = 100$, $\sigma_1 = 10$, $\sigma_2 = 15$, $r_1 = 20$, $r_2 = 30$, $l = l_b$ = average of standard deviations, and $p = 0.99$. Pre-computation time is not included in the measurements unless otherwise stated.

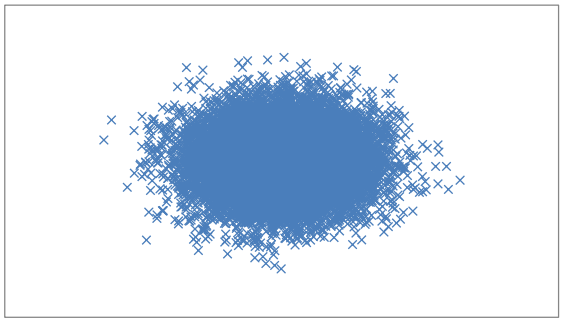


Fig. 7: Synthetic Data Distribution

1) Efficiency. We first conduct experiments to evaluate the efficiency of our proposed outlier detection algorithms presented in Sections V and VI. Fig. 8 compares the time taken by the Naive algorithm and two of our proposed algorithms. The algorithm for the conventional Gaussian distribution produces accurate results but is costly. On the other hand, the Bounded Gaussian distribution algorithm is very efficient but it may contain small amount of errors as discussed in later experiments.

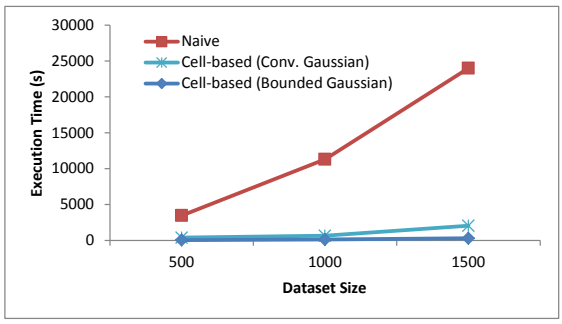
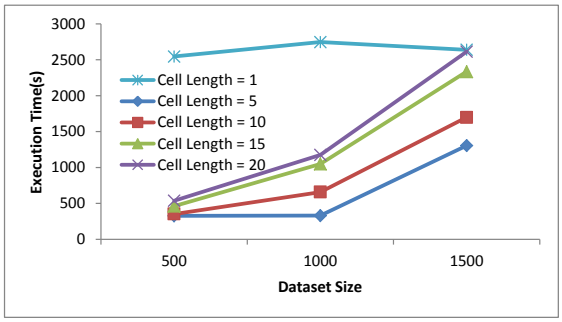
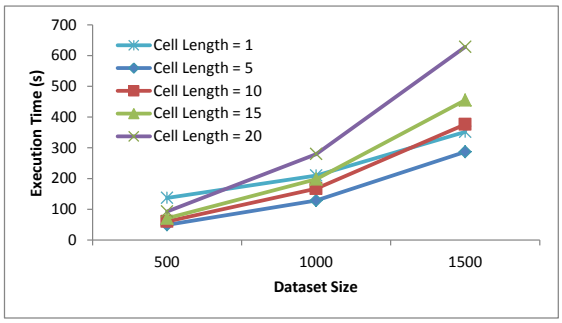


Fig. 8: Naive vs. Cell-based

Fig. 9 shows the effect of varying cell length on the execution time. It is obvious from the figures that smaller cell length produces shorter execution time. However very small cell length increases the number of cells exponentially and therefore the time required to calculate the bounds. Fig. 10 shows the bounds pre-computation time of Cell-based algorithm of conventional Gaussian distribution. When cell length decreases to 1, number of cells and thus the cells and layers bounds computation time increases dramatically. Therefore we recommend the use of cell length equal to the average of standard deviations as discussed in Sections V-E and VI-A2.



(a) Conventional Gaussian Distribution



(b) Bounded Gaussian Distribution

Fig. 9: Varying Cell Length

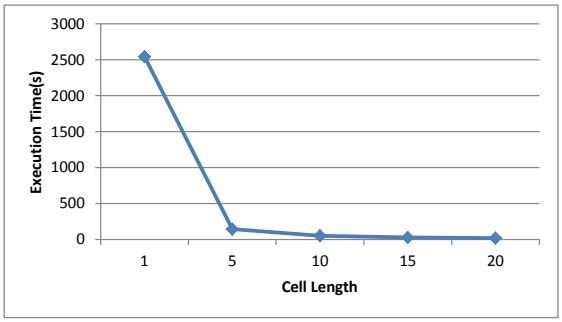
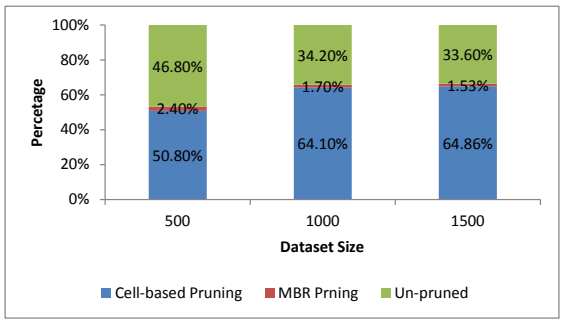


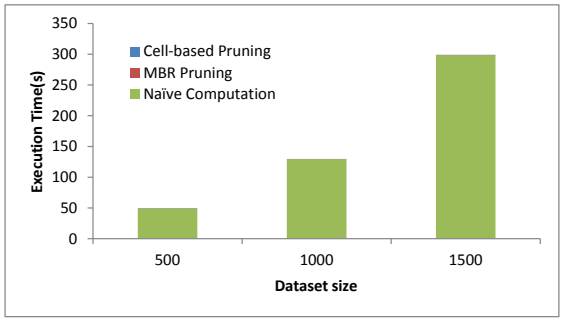
Fig. 10: Bounds Computation for Conventional Gaussian

Now, we focus on the algorithm in Section VI for the Bounded Gaussian distribution. Fig.11a shows the percentages of objects pruned by different pruning methods for the Bounded Gaussian distribution. Cell-based pruning is effective and pruned more than 50% of objects in all the experiments. However object-wise distance pruning could only prune very small percentages. Fig.11b shows the time taken by the different phases of Bounded Gaussian algorithm. It is obvious from the figure that the time taken for un-pruned objects processing is the largest.

2) Accuracy. Our algorithm for the conventional Gaussian distribution is 100% accurate, however there is a trades-



(a) Percentage of Objects Pruned



(b) Execution Time by Pruning Techniques

Fig. 11: Bounded Gaussian Pruning

off between efficiency and accuracy in case of the Bounded Gaussian algorithm. Fig.12 shows that our Bounded Gaussian algorithm is very accurate. The accuracy is measured using the following formula.

$$\text{Accuracy} = \frac{\text{Outliers from the Bounded Gaussian}}{\text{Outliers from the conventional Gaussian}} \times 100.$$

The accuracy here suggests that the approximation by the Bounded Gaussian distribution brings some "false negatives". We measured "false positives" too. But no false positive was detected in all cases.

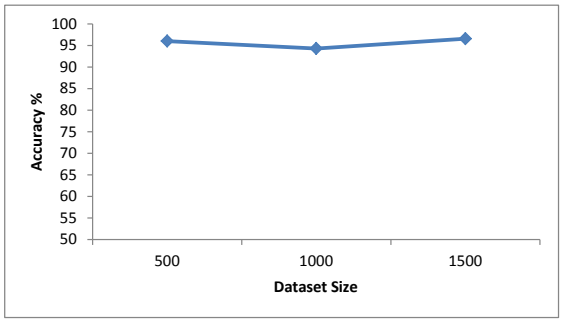


Fig. 12: Bounded Gaussian Accuracy

VIII. Conclusion and Future Work

In this work, we proposed two approaches of distance-based outlier detection on uncertain datasets of Gaussian distribution. Firstly we proposed a Cell-based approach of distance-based outlier detection on uncertain datasets of Gaussian distribution. Secondly we proposed an approximate approach using Bounded Gaussian distribution to increase the efficiency of outlier detection. Approximating Gaussian distribution by Bounded Gaussian distribution enables more effective Cell-based pruning along with simple object-wise distance pruning and bounds pruning methods. An extensive empirical study on real and synthetic data demonstrate the efficiency, accuracy and scalability of our proposed approaches.
In future, we are planning to work on distance-based outlier detection on uncertain datasets of arbitrary distribution.

Acknowledgement

This work has been partly supported by "New generation network R&D program for innovative network virtualization platform and its application(s)", the Commissioned Research of National Institute of Information and Communication Technology (NICT).

References

[1] Salman Ahmed Shaikh and Hiroyuki Kitagawa,: "Distance-based Outlier Detection on Uncertain Data of Gaussian Distribution", The 14th Asia-Pacific Web Conference (APWeb), 2012.
[2] Pukelsheim, Friedrich.: "The Three Sigma Rule", The American Statistician, 1994.
[3] Weisstein, Eric W.:"Normal Difference Distribution", From MathWorld - A Wolfram Web Resource. http://mathworld.wolfram.com/NormalDifferenceDistribution.
[4] J. Nievergelt, H. Hinterberger and K.C. Sevick.:"The Grid File: An Adaptable, Symmetric multikey File Structure", ACM Transaction on Database Systems, 1984.
[5] Edwin M. Knorr , Raymond T. Ng , Vladimir Tucakov.: "Distance-Based Outliers: Algorithms and Applications", The VLDB Journal, 2000.
[6] Hawkins D., "Identication of Outliers", Chapman and Hall, 1980.
[7] Barnett V., Lewis T., "Outliers in Statistical Data", John Wiley, 1994.
[8] Gustavo H. Orair , Carlos H. C. Teixeira , Wagner Meira.: "Distance-Based Outlier Detection: Consolidation and Renewed Bearing", Proc. of the VLDB Endowment, 2010.
[9] Bin Wang, Gang Xiao, Hao Yu and Xiaochun Yang.: "Distance-Based Outlier Detection on Uncertain Data", IEEE 9th International Conference on Computer and Information Technology, 2009.
[10] Hans-Peter Kriegel, Peer Krger, Arthur Zimek.: "Outlier Detection Techniques", Tutorial at 16th ACM SIGKDD Conference, 2010.
[11] Sridhar Ramaswamy, Rajeev Rastogi , Kyuseok Shim.: "Efficient Algorithms for Mining Outliers from Large Data Sets", Proceedings International Conference on Management of Data, ACM, SIGMOD, 2000.
[12] Aggarwal, C.C., Yu, P.S.: "Outlier Detection with Uncertain Data", SIAM International Conference on Data Mining, 2008.
[13] Edwin M. Knorr, Raymond T. Ng.: "Algorithms for Mining Distance-Based Outliers in Large Datasets", In Proceedings of 24rd International Conference on Very Large Data Bases, 1998.
[14] Zengyou He, Xiaofei Xu, and Shengchun Deng.: "Discovering cluster-based local outliers", Pattern Recognition Letters, Vol. 24, Issues 910, 2003.
[15] Markus Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jrg Sander.: LOF: Identifying Density-Based Local Outliers, ACM SIGMOD, 2000.
[16] M. Mahoney and P. Chan.: "Learning rules for anomaly detection of hostile network traffic", In Proceedings of the Third IEEE International Conference on Data Mining, 2003.
[17] Noor Alaydie, Farshad Fotouhi, Chandan K. Reddy, Hamid Soltanian-Zadeh.: "Noise and Outlier Filtering in Heterogeneous Medical Data Sources", Workshops on Database and Expert Systems Applications, DEXA, 2010.
[18] Arturo Elas, Alberto Ochoa-Zezzatti, Alejandro Padilla and Julio Ponce.: "Outlier Analysis for Plastic Card Fraud Detection a Hybridized and Multi-Objective Approach", Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, 2011.

[19] Hugo Garces, Daniel Sbarbaro.: "Outliers detection in environmental monitoring databases", Engineering Applications of Artificial Intelligence, 24(2), 2011.
[20] Maimon O. and Rockach L.: "Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers", Kluwer Academic Publishers, 2005.
[21] Yufei Tao, Xiaokui Xiao, and Reynold Cheng.: "Range search on multidimensional uncertain data", ACM Transactions on Database Systems, 32(3), 2007.
[22] Wang Kay Ngai, Ben Kao, Chun Kit Chui, Reynold Cheng, Michael Chau, and Kevin Y. Yip. 2006. Efficient Clustering of Uncertain Data. ICDM, 2006.
[23] W. Thistleton, J.A. Marsh, K. Nelson and C. Tsallis, Generalized Box-Muller method for generating q-Gaussian random deviates, IEEE Transactions on Information Theory, 2007.
[24] Bodik, P., Hong, W., Guestrin, C., Madden, S., Paskin, M. and Thibaux, R., "Intel lab data", 2004. Available at: http://db.csail.mit.edu/labdata/labdata.html.
[25] Minimum bounding rectangle. In Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Minimum_bounding_rectangle.

Appendix

*

Let o be a k-dimensional uncertain object with attributes $\overrightarrow{\mathcal{A}} = (x_1, ..., x_k)$, mean $\overrightarrow{\mu} = (\mu_1, ..., \mu_k)^T$ and a diagonal covariance matrix $\Sigma = diag(\sigma_1^2, ..., \sigma_k^2)$. The probability density function of o can be expressed as follows.

$$f(\overrightarrow{\mathcal{A}}) = \frac{1}{\sqrt{(2\pi)^k det\Sigma}} exp\left\{ -\frac{(\overrightarrow{\mathcal{A}} - \overrightarrow{\mu})^T \Sigma^{-1} (\overrightarrow{\mathcal{A}} - \overrightarrow{\mu})}{2} \right\}.$$

Since Σ is diagonal, the distribution functions are independent in coordinates. Hence the k-dimensional normal distribution function is given by the product of k 1-dimensional normal distribution functions.

$$f(\overrightarrow{\mathcal{A}}) = \prod_{1 \le i \le k} \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left\{ -\frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right\}.$$

Let o_i and o_j are two 2-dimensional uncertain objects with attributes $\overrightarrow{\mathcal{A}_i} = (x_{i,1}, x_{i,2})$ and $\overrightarrow{\mathcal{A}_j} = (x_{j,1}, x_{j,2})$, means $\overrightarrow{\mu_i} = (\mu_{i,1}, \mu_{i,2})^T$ and $\overrightarrow{\mu_j} = (\mu_{j,1}, \mu_{j,2})^T$ and diagonal covariance matrices $\Sigma_i = diag(\sigma_{i,1}^2, \sigma_{i,2}^2)$ and $\Sigma_j = diag(\sigma_{j,1}^2, \sigma_{j,2}^2)$ respectively. The difference between normal random vectors of o_i and o_j is given by $\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j} \sim \mathcal{N}(\overrightarrow{\mu_{i-j}}, \Sigma_{i-j})$, where $\overrightarrow{\mu_{i-j}} = \mu_i - \mu_j$ and $\Sigma_{i-j} = \Sigma_i + \Sigma_j$ [3].

Since Σ_i and Σ_j are diagonal matrices, the distribution functions are independent in coordinates. Hence the 2-dimensional normal difference distribution of uncertain objects o_i and o_j is given as follows.

$$f(\overrightarrow{\mathcal{A}_i - \mathcal{A}_j}) = \frac{1}{2\pi\sqrt{(\sigma_{i,1}^2 + \sigma_{j,1}^2)(\sigma_{i,2}^2 + \sigma_{j,2}^2))}} exp\left\{ -\left(\frac{(x-\alpha_1)^2}{2(\sigma_{i,1}^2 + \sigma_{j,1}^2)} + \frac{(y-\alpha_2)^2}{2(\sigma_{i,2}^2 + \sigma_{j,2}^2)} \right) \right\}, \quad (6)$$

where $\alpha_1 = \mu_{i,1} - \mu_{j,1}$ and $\alpha_2 = \mu_{i,2} - \mu_{j,2}$ are the differences between the means of objects o_i and o_j and $\sigma_{i,1}^2$, $\sigma_{j,1}^2$, $\sigma_{i,2}^2$ and $\sigma_{j,2}^2$ are the variances of the uncertain objects o_i and o_j in dimensions 1 and 2 respectively.

Hence the probability that the uncertain object o_i lies within D-distance of uncertain object o_j, denoted by $Pr(o_i, o_j, D)$, is given as follows.

$$Pr(o_i, o_j, D) = \frac{1}{2\pi\sqrt{(\sigma_{i,1}^2 + \sigma_{j,1}^2)(\sigma_{i,2}^2 + \sigma_{j,2}^2)}} \int_0^D \int_0^{2\pi} \exp\left\{ -\left(\frac{(r\cos\theta - \alpha_1)^2}{2(\sigma_{i,1}^2 + \sigma_{j,1}^2)} + \frac{(r\sin\theta - \alpha_2)^2}{2(\sigma_{i,2}^2 + \sigma_{j,2}^2)} \right) \right\} r \, d\theta \, dr \ \blacksquare \quad (7)$$