World Wide Web
DOI 10.1007/s11280-013-0211-y

Efficient distance-based outlier detection on uncertain datasets of Gaussian distribution

Salman A. Shaikh · Hiroyuki Kitagawa

Received: 3 October 2012 / Revised: 1 March 2013 / Accepted: 1 April 2013

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 data privacy. This work discusses the problem of distance-based outlier detection on uncertain datasets of Gaussian distribution. The Naive approach of distance-based outlier on uncertain data is usually infeasible due to expensive distance function. Therefore a cell-based approach is proposed in this work to quickly identify the outliers. The infinite nature of Gaussian distribution prevents to devise effective pruning techniques. Therefore an approximate approach using bounded Gaussian distribution is also proposed. Approximating Gaussian distribution by bounded Gaussian distribution enables an approximate but more efficient cell-based outlier detection approach. An extensive empirical study on synthetic and real datasets show that our proposed approaches are effective, efficient and scalable.

Keywords Distance-based outlier detection · Cell-based approach · Uncertain data · Bounded Gaussian distribution

1 Introduction

Outlier detection is a key problem in data mining. It has applications in many domains including credit card fraud detection [5], network intrusion detection [17], environment monitoring [9], medical sciences [2] etc. Several definitions of outlier

S. A. Shaikh (✉) · H. Kitagawa
Graduate School of Systems and Information Engineering, University of Tsukuba,
Tennodai, Tsukuba, Ibaraki 305-8573, Japan
e-mail: salman@kde.cs.tsukuba.ac.jp

H. Kitagawa
e-mail: kitagawa@cs.tsukuba.ac.jp

Published online: 17 April 2013

have been given in past, but there exists no universally agreed upon definition. Hawkins [10] 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. Barnett and Lewis [6] mentioned that an outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs.

In statistics, one can find over 100 outlier detection techniques. These have been developed for different data distributions, parameters, desired numbers of outliers and types of expected outliers [6, 18]. However, most statistical techniques are not useful due to several reasons. For example, most statistical techniques are univariate, in some techniques parameters are difficult to determine, and in other techniques outliers cannot be obtained until the underlying data distribution is known. In order to overcome these problems, several distance-based outlier detection approaches have been proposed in data mining [14, 15, 21, 31].

Most of the outlier detection techniques proposed in data mining are suitable only for deterministic data. However, due to the increasing usage of sensors, RFIDs and similar devices for data collection these days, data contains certain degree of inherent uncertainty [8, 11, 25]. The causes of uncertainty may include but are not limited to limitation of equipments, absence of data, inconsistent supply voltage and delay or loss of data in transfer [25]. In order to get reliable results from such data, uncertainty needs to be considered in calculation. Therefore this work presents distance-based outlier detection technique on uncertain data.

Motivating example—(Identifying malfunctioning sensors): As a result of advancement in technology, wireless sensor networks (WSNs) are often deployed for environment monitoring, animal tracking, flood detection, and weather forecasting. Usually a WSN covers an area of interest where each sensor keeps reporting its measures. Due to calibration errors, short-circuited connections, damaged sensors and low battery voltage, sensor reported measurements may be different from true measurements [25]. In other words, such measurements are uncertain values. Therefore commercial sensor producers always mention accuracy (measurement error) on their products. Table 1 lists maximum measurement errors of some commercially available sensors. Detecting outliers from such uncertain values is helpful in identifying malfunctioning or isolated sensors in the WSN. This paper assumes that uncertainty in the values obtained from a sensor follows Gaussian distribution.

In order to obtain distance-based outliers from uncertain datasets, distance needs to be calculated between uncertain data objects. However, the computation of distance between uncertain data objects is very costly. Therefore a cell-based approach is proposed in this paper to quickly identify the outliers. The proposed cell-based approach can identify and prune the cells containing only inliers. Similarly it can also detect the cells containing outliers. Although the cell-based technique is very effective, yet it may leave some cells undecided, i.e., they are neither identified as inlier cells nor as outlier cells. For the uncertain data objects in undecided cell, an object-wise bounds pruning technique is proposed. Finally nested-loop method is used for the uncertain objects which remain undecided after two prunings. The infinite nature of Gaussian distribution prevents effective pruning. Therefore an approximate approach of outlier detection is also proposed. The basic idea is that

Table 1 Uncertainty in commercial sensor measurements

Company	Sensor type	Model	Parameter	Max. measurement error (%)*
Stevens [27]	Weather	WXT520	Air temperature	1
			Barometric pressure	0.2
			Relative humidity	5
			Wind speed	5
			Wind direction	3
	Pyranometer	LI200SA	Solar radiation	5
Vaisala [30]	Weather	HMP155	Air temperature	0.1
			Barometric pressure	0.05
		WMT700	Relative humidity	1.7
			Wind speed	2
		PTB110	Wind direction	0.55
	Pyranometer	CM6B	Solar radiation	2
Xylem [33]	Weather	WE100	Air temperature	1
		WE550	Barometric pressure	0.2
		WE570	Relative humidity	5
		WE600	Wind speed	5
		WE700	Wind direction	3
	Pyranometer	WE300	Solar radiation	5

*For some parameters, percentages are calculated from their respective maximum error values

the Gaussian distribution can be appropriately approximated by bounded Gaussian distribution [22], and the outlier detection computation is less costly for bounded Gaussian distribution since the degree of uncertainty is bounded. Actually this bounded distribution allows to introduce strong pruning techniques at a small cost of accuracy. Hence this work presents two cell-based approaches of distance-based outlier detection on uncertain data. The exact approach is denoted by *Conventional Gaussian* and the approximate approach is denoted by *Bounded Gaussian* in the rest of the paper. Since each approach handles different nature of distribution (i.e., unbounded and bounded), different pruning techniques are proposed for both. Table 2 lists the pruning techniques proposed for both the approaches.

The rest of the paper is organized as follows. Section 2 surveys the previous work related to ours. Section 3 discusses the basic concepts and formally defines distance-based outlier detection on uncertain datasets. The conventional Gaussian approach is presented in Section 4. In Section 5, the bounded Gaussian approach of distance-based outlier detection on uncertain data is given. Section 6 contains an extensive experimental evaluation that demonstrates the effectiveness, efficiency and scalability of the proposed techniques. Section 7 concludes our paper.

Table 2 Pruning techniques for distance-based outlier detection approaches

Pruning techniques	Conventional Gaussian approach	Bounded Gaussian approach
Cell-based pruning	√	√
Simple object-wise distance pruning	×	√
Object-wise bounds pruning	√	√

2 Related work

Distance-based outlier detection approach was introduced by Knorr et al. [15]. They defined a point p to be an outlier if at most M points are within D-distance of p. They also presented a cell-based approach to efficiently compute the distance-based outliers. Ramaswamy et al. [23] formulated distance-based outliers as the top-t data points whose distance to their κth nearest neighbour is largest. Angiulli et al. in [3] gave a slightly different definition of outliers than [23] by considering the average distance to their k nearest neighbours. Beside these, there are some works on the detection of distance-based outliers over stream data including [4, 16] and [13]. These works are based on the definition of distance-based outliers by Knorr et al. Furthermore, [4] gave an approximate algorithm to reduce the memory space required by its exact counterpart. Later on [16] extended [4] by adding the concepts of multi-query and micro-cluster based distance-based outlier detection. However all these approaches were given for deterministic data and cannot handle uncertain data.

Recently a lot of research has focused on managing, querying and mining of uncertain datasets [1, 31]. The problem of outlier detection on uncertain datasets was first studied by Aggarwal et al. [1]. According to [1], an uncertain object o is a density-based (δ, η) outlier, if the probability of existence of o in some subspace of a region with density at least η is less than δ. In order to compute (δ, η) outliers, firstly density of all subspaces needs to be computed and then the η-probability of each o in the dataset is computed to tell if o is an outlier. Since this computation is very expensive, a sampling procedure is used to approximate the η-probability. In contrast to [1], this paper addresses the detection of distance-based outliers in full space, where the distance between two uncertain objects is computed by Gaussian difference distribution [32]. Therefore, the problem definition is quite different from [1]. Moreover, the cell-based approach is used to prune larger portion of dataset objects. Experiments in Section 6 prove that only fraction of dataset objects require the evaluation of actual distance function, reducing the overall cost of computation.

Wang et al. [31] also proposed outlier detection on uncertain data. However their work focus on the uncertainty in the existence of a sensor at a certain location. In contrast, in this paper, the uncertainty lies in the measurements obtained from sensors. Each tuple in [31] is associated with the confidence of appearing at a corresponding location. In this work, each uncertain object is represented by a Gaussian PDF, with an assumption that sensor measurements may deviate from true values due to the reasons discussed in Section 1.

This paper is an extended version of our previous paper [24]. In [24], a cell-based approach of distance-based outlier detection on uncertain data is given. For the un-pruned objects from the cell-based approach, nested-loop is used. However, in this work two different cell-based approaches (an accurate and an approximate) are given. Moreover, pruning techniques are proposed for the un-pruned objects from respective cell-based pruning techniques. Main contributions of this paper include an approximate approach of outlier detection using bounded Gaussian distribution (see Section 5). The extension also includes experiments comparing the effectiveness of the proposed approach with the approach in Knorr et al. [15] work and an extensive empirical study on performance using larger real and synthetic datasets. To the best of our knowledge, distance-based outlier detection on uncertain datasets of Gaussian distribution has not been studied by other researchers.

3 Distance-based outliers in uncertain datasets

The distance-based outlier detection approach was introduced by Knorr et al. [15]. 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.

Definition 1 was given for deterministic datasets. However, the focus of this work is uncertain datasets whose attribute values are uncertain. This paper assumes that the uncertainty is given by Gaussian distribution. The Gaussian distribution is chosen for representing uncertainty, because in statistics the Gaussian distribution *(or the normal distribution)* is the most important and the most commonly used.

In this paper, k-dimensional uncertain objects o_i are considered, 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 $\Sigma_i = diag(\sigma_{i,1}^2, ..., \sigma_{i,k}^2)$, respectively. Namely, the vector $\overrightarrow{\mathcal{A}_i}$ is a random variable that follows Gaussian distribution $\overrightarrow{\mathcal{A}_i} \sim \mathcal{N}(\overrightarrow{\mu_i}, \Sigma_i)$. Note that $\overrightarrow{\mu_i}$ denotes the observed coordinates (attribute values) of object o_i. 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 $\mathcal{GDB}$. Hence Definition 1 can be extended naturally 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.

Hence the set of distance-based outliers in $\mathcal{GDB}$ is defined as follows.

$$\mathcal{GDBO}utliers = \left\{ o_i \in \mathcal{GDB} \,\middle|\, \sum_{j=1}^{|\mathcal{GDB}|} Pr(dist(o_i, o_j) \leq D) \leq \theta \right\}. \tag{1}$$

The objects that lie within the D-distance of an object o are called *D-neighbours* of o, and the set of D-neighbours of o is denoted by $DN(o)$. In order to find distance-based outliers in $\mathcal{GDB}$, the distance between uncertain objects needs to be calculated. Fortunately the distance between two objects which follow the Gaussian distribution is given by another distribution known as the Gaussian difference distribution [32].

Let $\overrightarrow{\mathcal{A}_i}$ and $\overrightarrow{\mathcal{A}_j}$ be two independent 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 $\Sigma_i = diag(\sigma_{i,1}^2, ..., \sigma_{i,k}^2)$ and $\Sigma_j = diag(\sigma_{j,1}^2,..., \sigma_{j,k}^2)$, respectively. Then

$\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j} = \mathcal{N}(\overrightarrow{\mu_i} - \overrightarrow{\mu_j}, \Sigma_i + \Sigma_j)$ [32]. Let $Pr(o_i, o_j, D)$ denotes the probability that $o_j \in DN(o_i)$. Then,

$$Pr(o_i, o_j, D) = \int_R \mathcal{N}(\overrightarrow{\mu_i} - \overrightarrow{\mu_j}, \Sigma_i + \Sigma_j) \mathrm{d}\overrightarrow{\mathcal{A}} \,, \tag{2}$$

where R is a sphere with centre $(\overrightarrow{\mu_i} - \overrightarrow{\mu_j})$ and radius D. Lemma 1 gives the 2-dimensional expression for $Pr(o_i, o_j, D)$. However, $Pr(o_i, o_j, D)$ expressions for higher dimensional cases can be derived using (2).

Lemma 1 *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 $\overrightarrow{\mu_i} = (\mu_{i,1}, \mu_{i,2})^T$, $\overrightarrow{\mu_j} = (\mu_{j,1}, \mu_{j,2})^T$, $\Sigma_i = diag(\sigma_{i,1}^2, \sigma_{i,2}^2)$ and $\Sigma_j = diag(\sigma_{j,1}^2, \sigma_{j,2}^2)$. The $Pr(o_i, o_j, D)$ is given as follows.*

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

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

Proof See Appendix. □

This paper assumes that $\sigma_{i,1} = \sigma_{j,1} = \sigma_{i,2} = \sigma_{j,2} = \sigma$, and let $\alpha^2 = \alpha_1^2 + \alpha_2^2$. Hence the (3) is simplified as follows.

$$Pr(o_i, o_j, D) = \frac{1}{4\pi\sigma^2} \int_0^D \int_0^{2\pi} \exp\left\{\frac{-1}{4\sigma^2}\left(r^2 - 2\alpha r\cos\theta + \alpha^2\right)\right\} r\, d\theta\, dr. \tag{4}$$

Note that $Pr(o_i, o_j, D)$ only depends on α^2 and not on the coordinates of o_i and o_j. Hence $Pr(o_i, o_j, D)$ is denoted by $Pr(\alpha, D)$ when there is no confusion. Computing this probability is usually very costly, and needs to be avoided as much as possible during the computation of outliers.

In the following part, the discussion focuses on 2-dimensional case. However, the discussion can be extended to higher dimensional cases without loss of generality. In addition, this work assumes $\sigma_{i,1} = \sigma_{j,1} = \sigma_{i,2} = \sigma_{j,2} = \sigma$ to keep discussion simple.

The Naive approach of distance-based outlier detection uses Nested-loop. For each object $o_i \in \mathcal{GDB}$, it requires computation of the expensive distance function for 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 $O(N^2)$ evaluations of the distance function, which is very expensive. For the Naive approach algorithm, please refer to our previous work [24].

4 Cell-based outlier detection

Due to computationally expensive distance function, the naive approach is infeasible. Hence two pruning techniques are proposed in this section. These pruning techniques are used for detecting distance-based outliers without the need of actual distance function.

4.1 Cell-based pruning

The cell-based technique is proposed to quickly identify and prune cells containing only inliers. Similarly, it can also detect cells containing outliers like the cell-based approach of Knorr et al. [15]. Since the cell-based approach by Knorr et al. deals with deterministic data only, they considered two cell layers that lie within certain distances from a target cell for its pruning. However, in this work, objects are infinitely uncertain, hence all the cell-layers in the Grid need to be considered for pruning of the target cell.

4.1.1 Grid structure

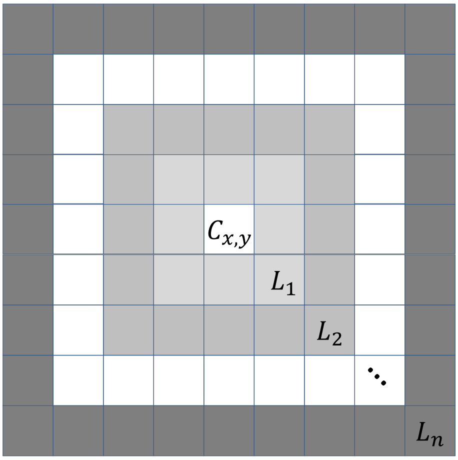
In order to identify distance-based outliers using cell-based technique, each object $o_i \in \mathcal{G}DB$ is mapped to a k-dimensional space that is partitioned into cells of length l. (The cell length is discussed in Section 4.1.4). Let $C_{x,y}$ be any cell in the Grid $\mathcal{G}$, where positive integers x and y denote the cell indices. The layers $(L_1, ..., L_n)$ of $C_{x,y} \in \mathcal{G}$ are the neighbouring cells of $C_{x,y}$ as shown in Figure 1 and are defined as follows.

$$L_1(C_{x,y}) = \{C_{u,v} | u = x \pm 1, v = y \pm 1, C_{x,y} \neq C_{u,v}\}.$$

$$L_2(C_{x,y}) = \{C_{u,v} | u = x \pm 2, v = y \pm 2, C_{u,v} \notin L_1(C_{x,y}), C_{x,y} \neq C_{u,v}\}.$$

$L_3(C_{x,y}), ..., L_n(C_{x,y})$ are defined in a similar way. The considerable maximum number of layers depends on the position of the target cell in the Grid. A cell $C_{x,y}$ in $\mathcal{G}$ can have the maximum number of layers if it exists at the corner of the Grid and

Figure 1 Cell layers.



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 $\lceil n/2 \rceil$.

4.1.2 Cell bounds

Like the cell-based approach by Knorr et al. [15], goal of the proposed cell-based technique is to identify and prune cells which are guaranteed to contain only inliers or outliers. A cell $C_{x,y}$ can be pruned as an "outlier cell" if the expected number of D-neighbours for any object in $C_{x,y}$ 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 the θ. Hence bounds on the expected number of D-neighbours of $C_{x,y} \in \mathcal{G}$ are defined to prune them. The upper and lower bounds bind the possible expected number 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 grid $\mathcal{G}$ for any object in cell $C_{x,y}$. Since the Gaussian distribution is infinite, two objects in the same cell may reside at the same coordinate. Hence the maximum expected number of D-neighbours in $C_{x,y}$ for any object in cell $C_{x,y}$ itself is equal to the number of objects in $C_{x,y}$, denoted by $N(C_{x,y})$.

Similarly, the maximum expected number of D-neighbours in cells in layer $L_m(C_{x,y})$ $(1 \leq m \leq n)$ for any object in $C_{x,y}$ can be obtained as follows.

$$\sum_{m=1}^{n} N(L_m(C_{x,y})) * Pr((m-1)l, D),$$

where $N(L_m(C_{x,y}))$ denotes the number of objects in layer $L_m(C_{x,y})$. Figure 2 shows how the $\alpha = (m-1)l$ values are obtained for the upper bounds. Hence $UB(C_{x,y})$ of $C_{x,y} \in \mathcal{G}$ is derived as follows.

$$UB(C_{x,y}) = N(C_{x,y}) + \sum_{m=1}^{n} N(L_m(C_{x,y})) * Pr((m-1)l, D).$$

Lower bound The lower bound of a cell $C_{x,y}$, $LB(C_{x,y})$, binds the minimum expected number of D-neighbours in grid $\mathcal{G}$ for any object in cell $C_{x,y}$. When two objects in the same cell reside at the opposite corners, the probability that they are D-neighbours takes the minimum value. Hence the minimum expected number of D-neighbours in $C_{x,y}$ for any object in cell $C_{x,y}$ itself is equivalent to $1 + (N(C_{x,y}) - 1) * Pr(\sqrt{2}l, D)$.

Similarly, the minimum expected number of D-neighbours in cells in layer $L_m(C_{x,y})$ $(1 \leq m \leq n)$ for any object in $C_{x,y}$ can be obtained as follows.

$$\sum_{m=1}^{n} N(L_m(C_{x,y})) * Pr((m+1)\sqrt{2}l, D).$$

Figure 2 shows how the $\alpha = (m+1)\sqrt{2}l$ values are obtained for the lower bounds. Hence $LB(C_{x,y})$ of $C_{x,y} \in \mathcal{G}$ is derived as follows.

$$LB(C_{x,y}) = 1 + (N(C_{x,y}) - 1) * Pr(\sqrt{2}l, D) + \sum_{m=1}^{n} N(L_m(C_{x,y})) * Pr((m+1)\sqrt{2}l, D).$$

Lookup table The bounds discussed above are required by each $C_{x,y} \in \mathcal{G}$ for pruning. Each bound computation requires evaluation of the costly distance function $Pr(\alpha, D)$ and the object counts of respective cell $C_{x,y}$ and its layers $L_m(C_{x,y})$. The number of distance function computations for the bounds calculation can be reduced by pre-computing $Pr(\alpha, D)$ values for $C_{x,y}$ bounds. Since the $Pr(\alpha, D)$ values are decided only by the α-values and are independent from the locations of $C_{x,y}$, $Pr(\alpha, D)$ values need to be computed only for $\alpha = m\sqrt{2}l$ $(1 \leq m \leq n+1)$ and $\alpha = ml$ $(0 \leq m \leq n-1)$. The pre-computed values are stored in a lookup table to be used by the cell-based pruning technique.

4.1.3 Cell pruning

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

If $LB(C_{x,y})$ is greater than θ, $C_{x,y}$ cannot contain outliers. Hence it can be pruned as an inlier cell. On the other hand if $UB(C_{x,y})$ is less than or equal to θ, $C_{x,y}$ is identified as an outlier cell. Lines 1–12 in Algorithm 1 show the cell-based pruning technique.

4.1.4 Cell length l

Due to complexity of the distance function used in this paper, it is not possible to derive a single cell length l suitable for all the combinations of D and standard

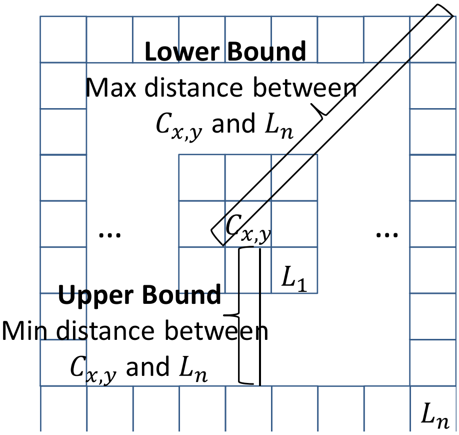


Figure 2 Cell and layers bounds

Algorithm 1 Cell-based outlier detection

```
Input: $\mathcal{G}DB, D, p, l$
Output: Set of distance-based outliers $\mathcal{O}$
Create cell grid $\mathcal{G}$ depending upon dataset $\mathcal{G}DB$ values and cell length $l$;
Initialize $Count_k$ of each cell $C_k \in \mathcal{G}$;
Map each object $o$ in $\mathcal{G}DB$ to an appropriate $C_k$, and increment $Count_k$ by 1;
$\theta \leftarrow |\mathcal{G}DB|(1-p)$, $\mathcal{O} = \{\}$; ($\theta$ correspond to the threshold)
/*Bounds computation*/
Compute $Pr(\alpha, D)$ values for the computation of bounds as discussed in Section 4.1.2;
/*Pruning cells using bounds*/
for each non-empty $C_k$ in $\mathcal{G}$ do
    if $LB(C_k) > \theta$ then
        $C_k$ is an inlier cell, mark $C_k$ green. GOTO Next $C_k$;
    else if $UB(C_k) \leq \theta$ then
        $C_k$ is an outlier cell, add objects of $C_k$ to $\mathcal{O}$, mark $C_k$ black. GOTO Next $C_k$;
    end if
end for
/*Object-wise pruning*/
$\mathcal{O} = \mathcal{O} \cup ObjectWisePruning(\mathcal{G}, D, \theta)$;
/*Unpruned objects processing*/
for each object $o_i$ in non-empty, uncoloured $C_k \in \mathcal{G}$ do
    if $o_i$ is uncoloured then compute $EN_{oi}$ (expected number of $D$-neighbours of $o_i$) using objects in $C_k$ and higher layers of $C_k \in \mathcal{G}$;
    if $EN_{oi} \leq \theta$ then $o_i$ is outlier. Add $o_i$ to $\mathcal{O}$;
end for
return $\mathcal{O}$;
```

deviations. Very small cell-length increases the number of cells in the grid exponentially and hence the execution time of the algorithm. On the other hand, larger cell length reduces the pruning capability of cell-based technique. Therefore a few cell lengths need to be checked before reaching the appropriate cell length. A good starting point of l that is found through experiments is the standard deviation, i.e., $l = \sigma$.

4.2 Object-wise bounds pruning

Although the cell-based technique is very effective, yet it may leave some cells undecided, i.e., they are neither pruned as inlier cells nor are identified as outlier cells. For the pruning of uncertain data objects in such cells an object-wise bounds pruning technique is proposed. This technique helps in the computation of expected D-neighbours for the un-pruned objects from the cell-based approach. Using this approach, a lot of expensive distance function computations may be avoided.

In this technique, $Pr(\alpha, D)$ is pre-computed for some α values. In this work, $Pr(\alpha, D)$ is computed for several α values between 0 and $D + 3\sigma$. α is chosen in this range because $Pr(\alpha, D)$ values for $\alpha > D + 3\sigma$ are negligibly small and are usually not effective in pruning. The set of pre-computed $Pr(\alpha, D)$ values is denoted by ψ and $n_{\text{bounds}} = |\psi|$. These pre-computed values serve as the upper and the lower bounds of the probability $Pr(\alpha, D)$ for objects o_i and o_j. These pre-computed lower and upper bounds are denoted by $Pr(\alpha, D)_{LB}$ and $Pr(\alpha, D)_{UB}$, respectively. Procedure 1 shows the object-wise bounds pruning.

Procedure 1 ObjectWisePruning

```
Input: $\mathcal{G}$, $D$, $\theta$
Output: Set of distance-based outliers $\mathcal{O}$
$\mathcal{O} = \{\}$;
for each non-empty uncoloured $C_k$ in $\mathcal{G}$ do
    for each $o_i$ in $C_k$ do
        for each $o_j$ in $D3\sigma(C_k)$
        ($D3\sigma(C_k)$ corresponds to the cells within $D + 3\sigma$ distance of cells $C_k$) do
            if $0 < \alpha \leq D + 3\sigma$ then
                Update $EN_{oiLB}$ and $EN_{oiUB}$ using precomputed bounds;
                ($EN_{oiLB}$ & $EN_{oiUB}$ corresponds the lower and upper bounds of expected number of $D - neighbours$ of $o_i$ respectively.)
            end if
        end for
        if $EN_{oiLB} > \theta$ then $o_i$ is inlier, mark $o_i$ green. GOTO next $o_i$;
        else if $EN_{oiUB} \leq \theta$ then $o_i$ is outlier, mark $o_i$ black. Add $o_i$ to $\mathcal{O}$;
    end for
end for
return $\mathcal{O}$;
```

For example, let $D = 90$ and $\sigma = 10$, then $D + 3\sigma = 120$. Therefore $Pr(\alpha, D)$ values need to be computed for $0 < \alpha \leq 120$. Assuming that $Pr(\alpha, D)$ is pre-calculated for $\alpha = 20, 40, 60, 80, 100, 120$ then $\psi = \{0.99, 0.9, 0.75, 0.5, 0.2, 0.001\}$. If $\alpha = 70$ for o_i and o_j then $Pr(70, D)_{UB} = 0.75$ and $Pr(70, D)_{LB} = 0.5$.

4.3 Un-pruned objects processing and grid file index

There may be some undecided objects, i.e., they are neither pruned as inliers nor identified as outliers, even after the cell-based pruning and the object-wise bounds pruning. For all such uncertain objects, nested-loop computation follows. Usually the number of such objects is very small, yet it can be expensive due to the costly distance function. According to the distance function, $Pr(o_i, o_j, D)$ is higher when o_i and o_j are close. Hence for an undecided object o_i, if $o_j \in \mathcal{G}DB$ nearer to o_i are chosen earlier for the computation of expected D-neighbours of o_i, the number of $Pr(o_i, o_j, D)$ computations can be reduced. If an un-pruned object o_i is inlier, it will be pruned by considering only nearer objects. Since the objects are already in grid structure, it can be utilized as grid-file index [20] with no additional indexing cost to retrieve nearer objects for the undecided objects. This helps in deciding the un-pruned objects faster than using no index at all. Lines 14–17 of Algorithm 1 shows the processing of such objects.

5 Cell-based outlier detection using bounded Gaussian uncertainty

Despite the techniques presented in Section 4, unbounded nature of the Gaussian distribution prevents from computing outliers very efficiently. Hence in this section an approximate distance-based outlier detection approach using the bounded Gaussian distribution is presented. Approximating the Gaussian distribution by the bounded Gaussian distribution enables an approximate but more efficient cell-based pruning technique along with simple object-wise distance and bounds pruning techniques. According to this paper's assumption, attributes of uncertain objects follow the Gaussian distribution. Therefore according to the 3-sigma rule there is a 95.45 % chance that uncertain objects' attribute values lie within two standard deviations of the observed values and 99.73 % chance that the values lie within three standard deviations of the observed values [22]. Hence the conventional unbounded Gaussian distribution can be normalized within certain boundaries to increase efficiency of outlier detection at a small cost of accuracy.

Given a conventional Gaussian function $g_{\overrightarrow{\mathcal{A}}}(x_1, x_2)$ with mean $\overrightarrow{\mu} = (\mu_1, \mu_2)^T$ and co-variance matrix $\Sigma = diag(\sigma^2, \sigma^2)$, the bounded Gaussian distribution $f_{\overrightarrow{\mathcal{A}}}(x_1, x_2)$ can be defined following the practise of [28], as follows.

$$f_{\overrightarrow{\mathcal{A}}}(x_1, x_2) = \begin{cases} \frac{g_{\overrightarrow{\mathcal{A}}}(x_1,x_2)}{\int_{(x_1,x_2)\in o.ur} g_{\overrightarrow{\mathcal{A}}}(x_1,x_2)\mathrm{d}x_1\mathrm{d}x_2} & (x_1, x_2) \in o.ur \\ 0 & otherwise \end{cases} \tag{5}$$

where $o.ur$ denotes the uncertainty region of the bounded Gaussian distribution. This paper assumes that the uncertainty region is a sphere with centre at $(\mu_1, \mu_2)^T$ and radius rad. Note that,

$$\int_{o.ur} f_{\overrightarrow{\mathcal{A}}}(x_1, x_2)\mathrm{d}x_1\mathrm{d}x_2 = 1.$$

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

$$Pr(o_i, o_j, D) = \int_0^{rad}\int_0^{2\pi}\int_0^{D}\int_0^{2\pi} f_{\overrightarrow{\mathcal{A}_i}}(r_1\cos\theta_1, r_1\sin\theta_1) \times f_{\overrightarrow{\mathcal{A}_j}}(r_1\cos\theta_1 + r_2\cos\theta_2, r_1\sin\theta_1 + r_2\sin\theta_2) r_1 r_2 \mathrm{d}\theta_2\mathrm{d}r_2\mathrm{d}\theta_1\mathrm{d}r_1. \tag{6}$$

Hence an uncertain dataset $\mathcal{G}DB = \{o_1, ..., o_N\}$ with the conventional Gaussian distribution can be approximated by the bounded Gaussian distribution. Namely, $\mathcal{G}DB_b = \{o_1, ..., o_N\}$ denotes a set of objects whose attributes $\overrightarrow{\mathcal{A}_i} = (x_{i,1}, x_{i,2})^T$ follow the bounded Gaussian distribution with mean $\overrightarrow{\mu_i} = (\mu_{i,1}, \mu_{i,2})^T$ and co-variance matrix $\Sigma_i = diag(\sigma^2, \sigma^2)$ respectively and radius rad.

5.1 Cell-based pruning for bounded Gaussian

Approximating the Gaussian distribution by the bounded Gaussian distribution enables better cell-based pruning. The proposed technique prunes cells containing only inliers and identify outlier cells just like the cell-based approach for conventional Gaussian distribution. In contrast to computing bounds using many layers in the conventional Gaussian cell-based approach, cell size is set in such a way that a target cell can be pruned by just counting objects within the target cell and its neighbouring layers. Moreover, no pre-computation is required for the cell-based technique using the bounded Gaussian uncertainty.

5.1.1 Grid structure for bounded Gaussian

In order to identify distance-based outliers using the cell-based technique, each object $o_i \in \mathcal{G}DB_b$ is mapped to a k-dimensional space that is partitioned into cells of length l (The cell length is discussed in Section 4.1.4). Let $C_{x,y}$ be a cell in the Grid $\mathcal{G}$, then cells in region $R_1(C_{x,y})$ are those which completely lie within $D - 2rad$ distance of the $C_{x,y}$, including the $C_{x,y}$ itself, as shown in Figure 3. Let $n_{R1} = \left\lfloor \frac{D-2rad}{l} \right\rfloor - 1$, then the region $R_1(C_{x,y})$ is derived as follows.

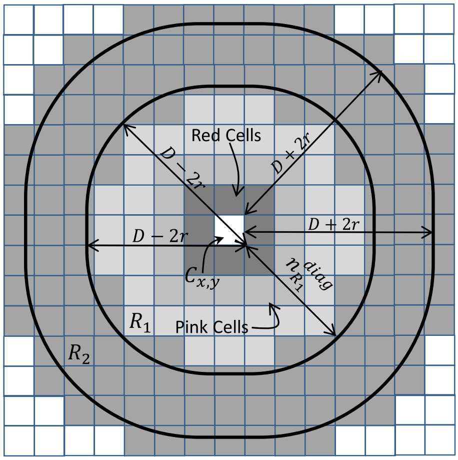
$$R_1(C_{x,y}) = \Bigg\{ C_{u,v} | u = x \pm n_{R1}, v = y \pm n_{R1}, \\ \sqrt{((|u| + 1)l)^2 + ((|v| + 1)l)^2} < D - 2rad, C_{u,v} \neq C_{x,y} \Bigg\}.$$

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

Property 1 If $C_{u,v} \in R_1(C_{x,y})$, then the objects $o_i \in C_{x,y}$ and $o_j \in C_{u,v}$ are at most $D - 2rad$ distance apart.

From Property 1, the $o_i \in C_{x,y}$ and the $o_j \in R_1(C_{x,y})$ are guaranteed to be D-neighbours mutually, hence the $Pr(o_i, o_j, D)$ is always equal to 1. Cells in region

Figure 3 Bounded Gaussian cell grid



$R_2(C_{x,y})$ are those which fall within $D+2rad$ distance of the $C_{x,y}$. Let $n_{R2} = \lceil \frac{D+2rad}{l} \rceil$, then the region $R_2(C_{x,y})$ is derived as follows.

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

Note that the $R_1(C_{x,y})$ and the $R_2(C_{x,y})$ satisfy 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 the objects $o_i \in C_{x,y}$ and $o_j \in C_{u,v}$ are greater than $D + 2rad$ distance apart.

From Property 2, it can be guaranteed that the $o_i \in C_{x,y}$ and $o_j \in C_{u,v}$ are greater than $D + 2rad$ distance apart, hence the $Pr(o_i, o_j, D)$ is always equal to 0. Two more types of cells help in pruning. These cells are named "red cells" and "pink cells" and are denoted by $R_r(C_{x,y})$ and $R_p(C_{x,y})$ respectively. Let $n_{R_1}^{\text{diag}} = \left\lfloor \frac{D-2rad}{l\sqrt{2}} \right\rfloor - 1$ denotes number of diagonals within $D - 2rad$ distance of a cell $C_{x,y}$. Then the red and the pink cells are defined as follows.

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

$$n_r = \min \left\{ n \mid N(C_{x,y}) \cup \sum_{i=1}^{n} N(L_i(C_{x,y})) > \theta \ , 0 < n \leq \left\lfloor \frac{n_{R_1}^{\text{diag}}}{2} \right\rfloor \right\}.$$

where $N(L_i(C_{x,y}))$ denotes number of objects in $C_{x,y}$ and its layer $L_i(C_{x,y})$ $(1 \leq i \leq n)$. The n_r value which meets above condition may not exist. If it exists, $R_p(C_{x,y})$ is defined as follows.

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

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

For example, in Figure 3, $n_{R_1}^{\text{diag}} = 3$, and assume that $n_r = 1$. Moreover, assume that the total number of objects in $C_{x,y}$ and $L_1(C_{x,y})$ are greater than θ. Then, $n_p = 2$, and all the objects in the $C_{x,y}$, the $L_1(C_{x,y})$ and the $L_2(C_{x,y})$ are inliers. Note that combined thickness of $C_{x,y}$ and $R_r(C_{x,y})$ is always less than $n_{R_1}^{\text{diag}}$, 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 $C_{x,y}$ are 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 $C_{x,y} \cup R_1(C_{x,y})$ are greater than θ, none of the objects in $C_{x,y}$ is an outlier.
(c) If the number of objects in $C_{x,y} \cup R_r(C_{x,y})$ are 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 $C_{x,y} \cup R_1(C_{x,y}) \cup R_2(C_{x,y})$ are less than or equal to θ, every object in $C_{x,y}$ is an outlier.

Algorithm 2 shows the cell-based calculation for the bounded Gaussian distribution.

Algorithm 2 Cell-based outlier detection on bounded Gaussian

```
Input: $\mathcal{G}DB_b$, $D$, $p$, $l$, $rad$, $n_{\text{bounds}}$
Output: Set of distance-based outliers $\mathcal{O}$
Compute $n_{\text{bounds}}$ bounds between $D - 2rad$ and $D + 2rad$;
Create cell Grid $\mathcal{G}$ depending upon dataset $\mathcal{G}DB_b$ values and cell length $l$ and initialize the count of each cell $C_k \in \mathcal{G}$, $Count_k \leftarrow 0$;
Map each object $o \in \mathcal{G}DB_b$ to an appropriate cell $C_k$, and increment $Count_k$ by 1;
$\theta \leftarrow |\mathcal{G}DB_b|(1-p), \mathcal{O} = \{\}$; ($\theta$ correspond to the threshold)
/* Cell-based Pruning */
For each non-empty $C_k \in \mathcal{G}$, If $Count_k > \theta$, $C_k$ is an inlier cell, mark $C_k$ green;
For each green cell $C_k \in \mathcal{G}$, mark $R_1(C_k)$ cells $blue$, provided the neighbour has not already been marked green;
For each non-empty, uncoloured cell $C_k \in Grid$, If $Count_k + \sum_{m \in R_r(C_k)} Count_m > \theta$, then mark $C_k$, $R_r(C_k)$ and $R_p(C_k)$ $blue$;
for each non-empty, uncoloured cell $C_k$ in $\mathcal{G}$ do
    $Count_{k2} \leftarrow Count_k + \sum_{m \in R_1(C_k)} Count_m$;
    if $Count_{k2} > \theta$ then
        $C_k$ is an inlier cell, mark $C_k$ $blue$. GOTO next $C_k$;
    else if $Count_{k2} + \sum_{m \in R_2(C_k)} Count_m \leq \theta$ then
        $C_k$ is an outlier cell, add objects of $C_k$ to $\mathcal{O}$. GOTO next $C_k$;
    end if
end for
/*Object-wise pruning*/
$\mathcal{O} = \mathcal{O} \cup ObjectWisePruning(\mathcal{G}, D, \theta)$;
/*Unpruned objects processing*/
for each object $o_i$ in non-empty, uncoloured $C_k \in \mathcal{G}$ do
    if $o_i$ is uncoloured then compute $EN_{oi}$ (expected number of $D$-neighbours of $o_i$) using objects in $C_k$ and higher layers of $C_k \in \mathcal{G}$;
    if $EN_{oi} \leq \theta$ then $o_i$ is outlier. Add $o_i$ to $\mathcal{O}$;
end for
return $\mathcal{O}$;
```

5.2 Simple object-wise distance pruning

Although the cell-based technique is very effective, yet it may leave some cells undecided, i.e., they are neither pruned as the inlier cells nor are identifies as the outlier cells. For pruning of uncertain data objects in such cells, an object-wise distance pruning technique is proposed. A similar technique was used in [19] for finding distance between an object and a cluster representative.

Since the uncertainty of objects in $\mathcal{G}DB_b$ is bounded, it can be found weather two objects are within the D-distance only by calculating distance between their observed coordinates. The distance computation in this case is just ordinary Euclidean and is cheap. Since α denotes an ordinary Euclidean distance between observed coordinates of two objects. Let the objects be $o_i, o_j \in \mathcal{G}DB_b$. If $\alpha \leq D - 2rad$, it is guaranteed that object o_j lies within the D-distance of o_i. In other words, $Pr(o_i, o_j, D) = 1$. On the other hand, if $\alpha > D + 2rad$, 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 Figure 4, $\alpha < D - 2rad$ for o_i and o_p and $\alpha > D + 2rad$ for o_i and o_q. Therefore $Pr(o_i, o_p, D) = 1$ and $Pr(o_i, o_q, D) = 0$.

This object-wise distance pruning helps in computing expected D-neighbours of the un-pruned objects. Using this approach, a lot of expensive distance function computations may be avoided.

5.3 Object-wise bounds pruning

Using simple object-wise distance pruning of Section 5.2, $Pr(o_i, o_j, D)$ can be computed only for the objects whose $D + 2rad < \alpha \leq D - 2rad$. However in order to compute expected D-neighbours for an object o_i in $\mathcal{G}DB_b$, the $Pr(o_i, o_j, D)$ needs to be computed for all $o_j \in \mathcal{G}DB_b$ within the regions $R_1(C_{x,y})$ and $R_2(C_{x,y})$. Here $C_{x,y}$ is the cell containing o_i. Hence a technique similar to the object-wise bounds pruning of Section 4.2 can be used to compute bounds of the $Pr(o_i, o_j, D)$ for $D - 2rad < \alpha \leq D + 2rad$. The object-wise bounds pruning for the bounded Gaussian is exactly similar to that of the conventional Gaussian with an exception that in the bounded Gaussian, the bounds need to be computed for $D - 2rad < \alpha \leq D + 2rad$ only.

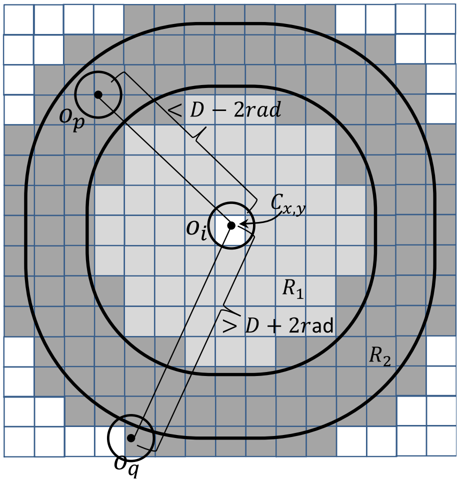


Figure 4 Simple object-wise distance pruning

5.4 Un-pruned objects processing for bounded Gaussian

Un-pruned objects processing for the bounded Gaussian distribution is same as the one presented in Algorithm 1. The only difference between the un-pruned objects processing of the conventional Gaussian distribution and the bounded Gaussian distribution is that the later needs to consider objects in only 2 regions, i.e., $R_1(C_{x,y})$ and $R_2(C_{x,y})$ for the computation of expected D-neighbours of an un-pruned object $o_i \in C_{x,y}$. In contrast, the conventional Gaussian needs to consider objects in the complete Grid $\mathcal{G}$.

6 Experiments

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

6.1 Datasets

In this paper two synthetic and three real datasets are used for experiments. Synthetic datasets, unimodal Gaussian (UG) and trimodal Gaussian (TG) are 2-dimensional and are generated using BoxMuller method [29]. This method generates pair of independent, standard, normally distributed (zero mean, unit variance) random numbers, given a source of uniformly distributed random numbers. A 3-dimensional trimodal Gaussian (TG3D) dataset is also generated for the evaluation of our proposed approaches on 3-dimensional data. Unless specified, 2-dimensional and 3-dimensional datasets consist of 10,000 and 1,000 tuples respectively.

As for real-world data, three datasets are used: ADAPTE, SDSS and ISPD. ADAPTE and ISPD are obtained from CISL Research data archive [7] and SDSS is obtained from Sloan Digital Sky Survey [26]. ADAPTE consists of about 1,851 maximum and minimum temperature values collected from the National Polytechnic Institute of Mexico and National Meteorological System. SDSS dataset contains 10,136 Right Ascension and Declination coordinates of stars and galaxies. SDSS dataset used in the experiments is a subset of SDSS Data Release 7 (DR7), which includes a huge collection of more than 6 million stars, 8 million galaxies, and 4,500 quasars [26]. The International Surface Pressure Databank (ISPD) dataset consists of 108,015 values of sea level pressure and surface pressure, which is the world's largest collection of pressure observations [12].

All the datasets are normalized to have a domain of [0, 1,000] on every dimension. For each point z in any dataset, an uncertain object o is created, whose uncertainty is given by the Gaussian distribution with mean z and standard deviation σ in all the dimensions. Pre-computation time is not included in the measurements. Unless specified, the following parameter values are used in experiments: $D = 100$, $\sigma = 10$, $l = 5 (l = 10$ for 3D dataset), $n_{\text{bounds}} = 10$, $rad = 3\sigma$, and $p = 0.995$. In figures, the

Knorr et al. [15] algorithm is denoted by *Knorr* and the algorithms proposed in this paper are denoted by *Conventional Gaussian* and *Bounded Gaussian* respectively.

6.2 Effectiveness of proposed approaches

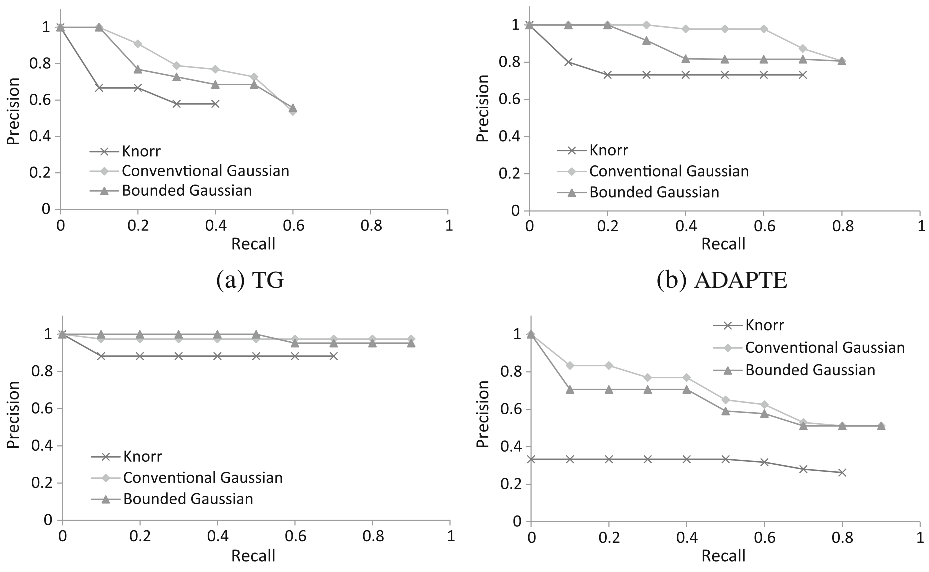
Firstly, experiments are performed to evaluate the effectiveness of our proposed approaches. Since there are no known algorithms for distance-based outlier detection on uncertain data, the deterministic algorithm for distance-based outlier detection given by Knorr et al. [15] is used as a baseline. Since the outliers are not known, for both synthetic and real datasets, baseline algorithm is used to determine the outliers on the original datasets. The results obtained from the baseline algorithm are used as the ground truth. In order to judge the effectiveness of the proposed algorithms, the precision and recall are measured on the perturbed dataset for the baseline algorithm and the proposed algorithms. The perturbed dataset is obtained by adding normal random numbers with zero mean and standard deviation σ_p to each of the tuple values of the original dataset. The σ_p was varied from 10 to 30 (with a step of five) to generate perturbed datasets of five different levels. Experiments will show that the proposed algorithms are superior than the baseline algorithm, since they do not degrade quite as much with increasing uncertainty.

The quality of the results are measured in terms of the precision and recall compared to the ground truth. The precision was defined as the ability of the algorithm to present only true outliers. The recall was defines as the ability of the algorithm to present all true outliers.

Unless specified, the following parameter values are used for the experiments in this subsection: $D = 100$, $\sigma = 10$, $\sigma_p = 20$, $l = 10$, $n_{\text{bounds}} = 10$, $rad = 3\sigma$, and p is selected in such a way that the baseline algorithm and the proposed algorithms return approximately 0.5 % outliers.

Firstly the precision-recall trade-off curves are presented for the different datasets. In all the graphs in Figure 5, the trade-off curves are higher for the proposed approaches. In Figure 5a, the precision-recall curves are somewhat closer for all three algorithms due to the sparsity of synthetic dataset TG. Addition of perturbation to the dataset TG, changed a lot of outliers to inliers and vice versa. Yet the recall, i.e. the number of true outliers retrieved, of the proposed algorithms is better than the baseline algorithm. The trade-off curves in Figure 5b and c show very good precision and recall for all the algorithms. This is due to the obvious outliers in the real datasets ADAPTE and SDSS. Even then the baseline algorithm cannot obtain high recall. The number of false positive outliers produced by the baseline algorithm in Figure 5d is very large. This can be observed from the very low precision of the trade-off curve. On the other hand, the trade-off curves for the proposed approaches are far better.

The effectiveness of the proposed algorithms is also tested with increasing level of uncertainty. From Figure 6, it is clear that the precision falls with increasing uncertainty level. Moreover, the precision results for the proposed approaches are superior to the results of the baseline approach for all uncertainty levels in Figure 6. Furthermore, the difference between the baseline approach and the proposed approaches increases with increasing uncertainty. Please note the sharp decrease in precision in Figure 6d. Since the dataset of Figure 6d is very large and dense, addition of even low level of perturbation produced a lot of false positive outliers in case of the baseline algorithm. On the other hand, the proposed algorithms produced better



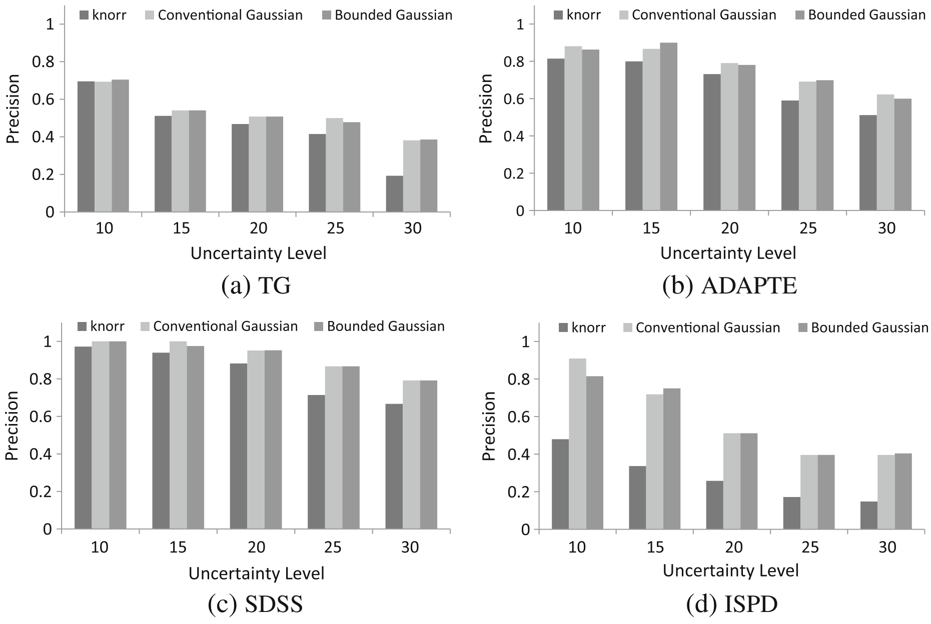
(a) TG

(b) ADAPTE

(c) SDSS

(d) ISPD

Figure 5 Precision-recall trade-off curves ($D = 100, \sigma = 10, \sigma_p = 20, l = 10, n_{\text{bounds}} = 10, rad = 3\sigma$, and p is selected in such a way that approximately 0.5 % outliers are returned by all the algorithms)



(a) TG

(b) ADAPTE

(c) SDSS

(d) ISPD

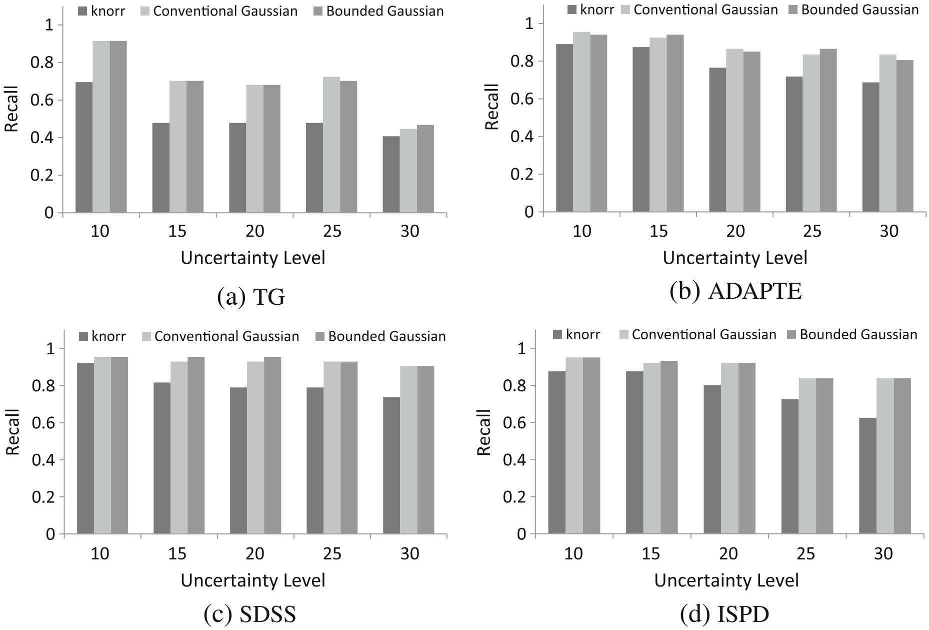
Figure 6 Precision with increasing σ_p ($D = 100$, $\sigma = 10$, $l = 10$, $n_{\text{bounds}} = 10$, $rad = 3\sigma$, and p is selected in such a way that approximately 0.5 % outliers are returned by all the algorithms)

precisions for all the datasets. Similar results are illustrated for recall in Figure 7. In all four plots of Figure 7, the recall is somewhat consistent with increasing uncertainty level for the proposed algorithms. Which proves that the proposed algorithms are capable of retrieving true outliers, even from the noisy data. Although the baseline algorithm seems to be competitive in Figure 7, but the precision (too many false positive outliers) of the baseline algorithm hide the real outliers.

6.3 Efficiency

In this subsection, experiments are conducted to evaluate efficiency of the proposed outlier detection algorithms presented in Sections 4 and 5. The time taken by the Naive algorithm is too high. It takes several hours even on the smallest sample (of 5,000 tuples) of the synthetic dataset. Hence the results of the Naive algorithm are not included.

Graphs in Figure 8 show the effect of varying cell lengths on execution times. It is obvious from the graphs that the smaller cell lengths require the lower execution times. However very small cell length increases the number of cells exponentially, increasing the execution time of the algorithms, as can be observed for $l = 1$. Therefore we recommend the use of cell length equal to the standard deviation of uncertain objects in the dataset as discussed in Section 4.1.4. Moreover it can be observed that the execution time did not increase for the bounded Gaussian algorithm as sharply as that for the conventional Gaussian for $l = 1 (l = 5$ for



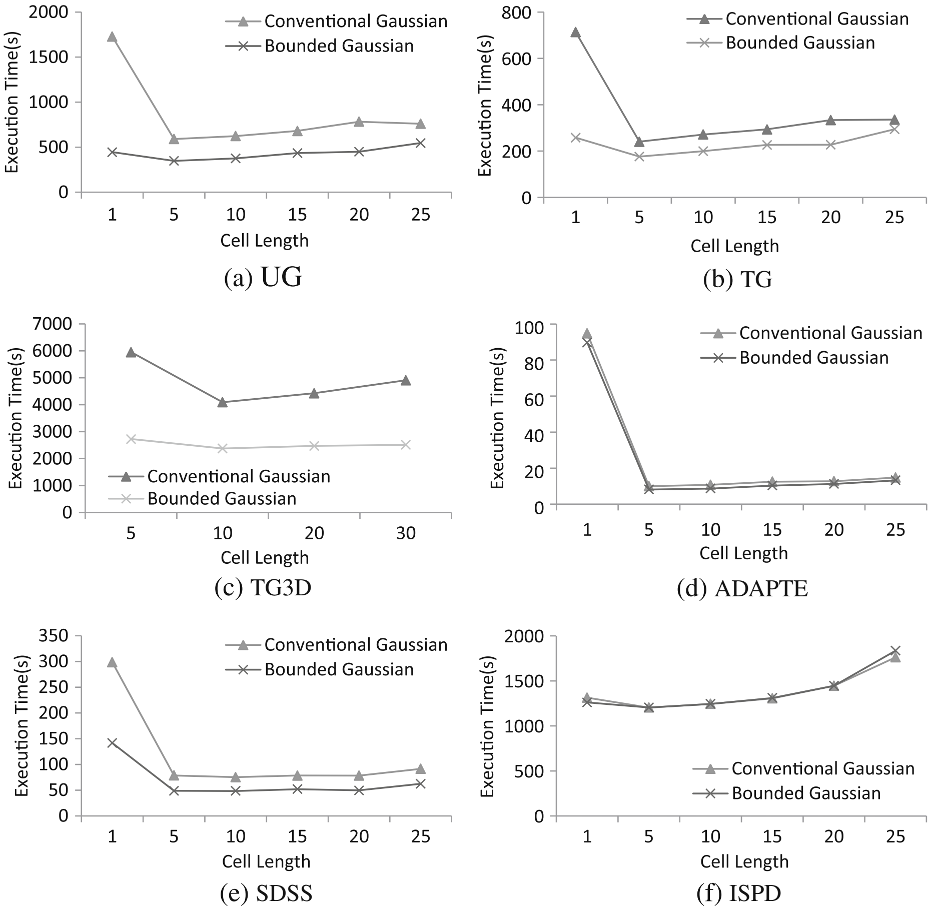
(a) TG

(b) ADAPTE

(c) SDSS

(d) ISPD

Figure 7 Recall with increasing σ_p ($D = 100$, $\sigma = 10$, $l = 10$, $n_{\text{bounds}} = 10$, $rad = 3\sigma$, and p is selected in such a way that approximately 0.5 % outliers are returned by all the algorithms)



(a) UG

(b) TG

(c) TG3D

(d) ADAPTE

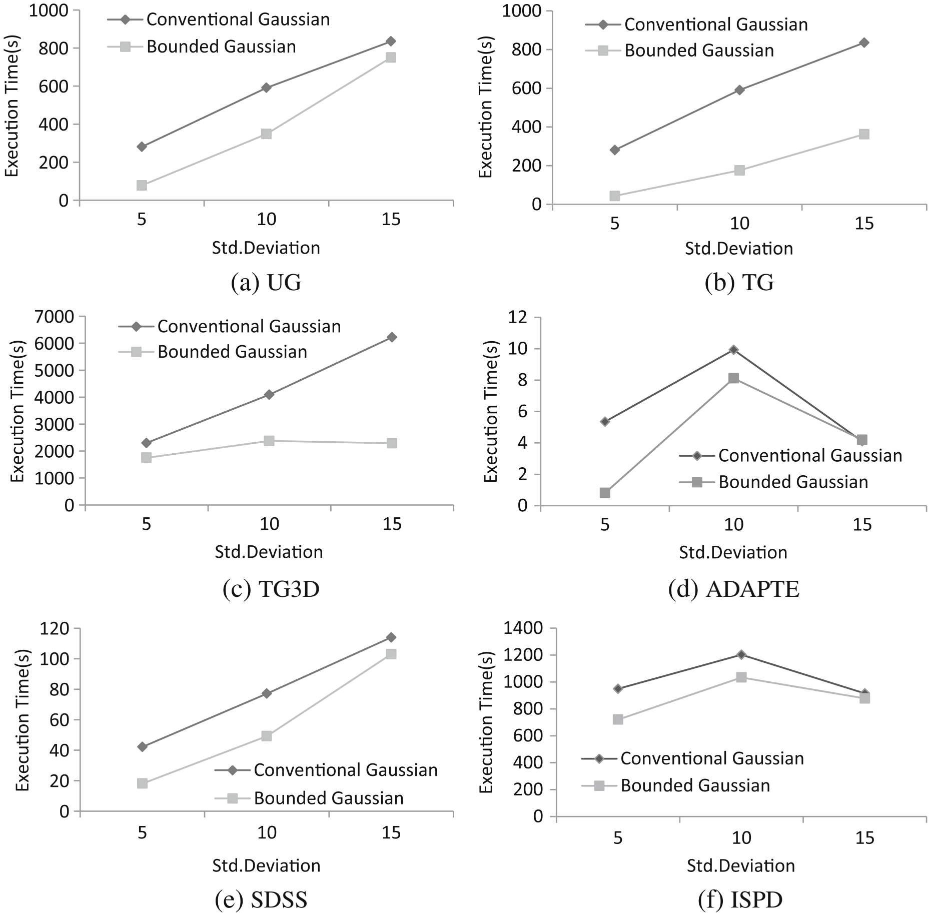
(e) SDSS

(f) ISPD

Figure 8 Varying l ($D = 100, \sigma = 10, n_{\text{bounds}} = 10, rad = 3\sigma, p = 0.995$)

3-dimensional dataset). This is due to the expensive bounds computation time in the conventional Gaussian algorithm for cell-based pruning, which is not required in the bounded Gaussian algorithm. Furthermore if can be observed from the graphs in Figure 8 that the bounded Gaussian algorithm is more efficient than the conventional Gaussian algorithm, except in Figure 8d and f. The dataset in Figure 8d is so small that the difference between the two algorithms is not obvious. In Figure 8f, the number of un-pruned objects requiring nested-loop computation were large, due to very large size of dataset. Hence the un-pruned objects processing time dominates in Figure 8f, which is almost same for both the algorithms.

Next experiments are performed by varying the standard deviation (σ) of uncertain objects in the datasets. As σ increases, the uncertainty of the dataset objects also increases. This increase in uncertainty results in smaller $Pr(o_i, o_j, D)$ values even if o_i and o_j are located nearby. Hence the number of distance function evaluations required increases for un-pruned objects during their nested-loop evaluation, which results in higher execution times as can be observed from Figure 9. However in



(a) UG

(b) TG

(c) TG3D

(d) ADAPTE

(e) SDSS

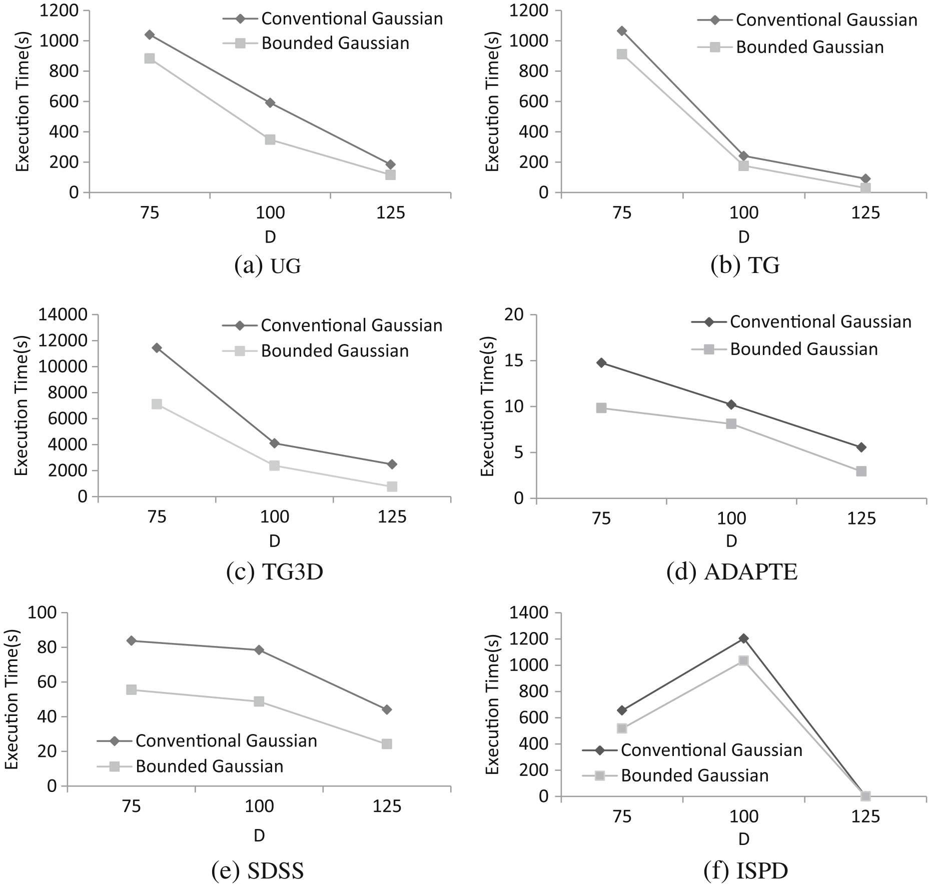
(f) ISPD

Figure 9 Varying σ ($D = 100, l = 5$($l = 10$for 3D dataset), $n_{\text{bounds}} = 10$, $rad = 3\sigma$, $p = 0.995$)

Figure 9d and f, execution times of the algorithms fell sharply for $\sigma = 15$. This is due to the less number of un-pruned objects for $\sigma = 15$ than $\sigma = 10$ from pruning algorithms. This phenomenon is discussed in Figure 12a.

Figure 10 shows the effect of varying the distance parameter D. Increase in D results in an increase in D-neighbours of dataset objects. As a result, objects are more easily pruned as inliers, bringing down execution time of the overall algorithm for larger values of D. Unusual behaviour of curves in Figure 10f is again due to the large number of un-pruned objects for $D = 100$ and no un-pruned object for $D = 125$.

In Figure 11, the number of outliers are varied by varying the parameter p. As can be observed from the graphs in Figure 11, increase in p results in decrease in execution times of the algorithms. This is due to the fact that increase in p, results in decrease of the threshold value θ. Hence the dataset objects are pruned more easily, bringing down the algorithm execution time.



(a) UG

(b) TG

(c) TG3D

(d) ADAPTE

(e) SDSS

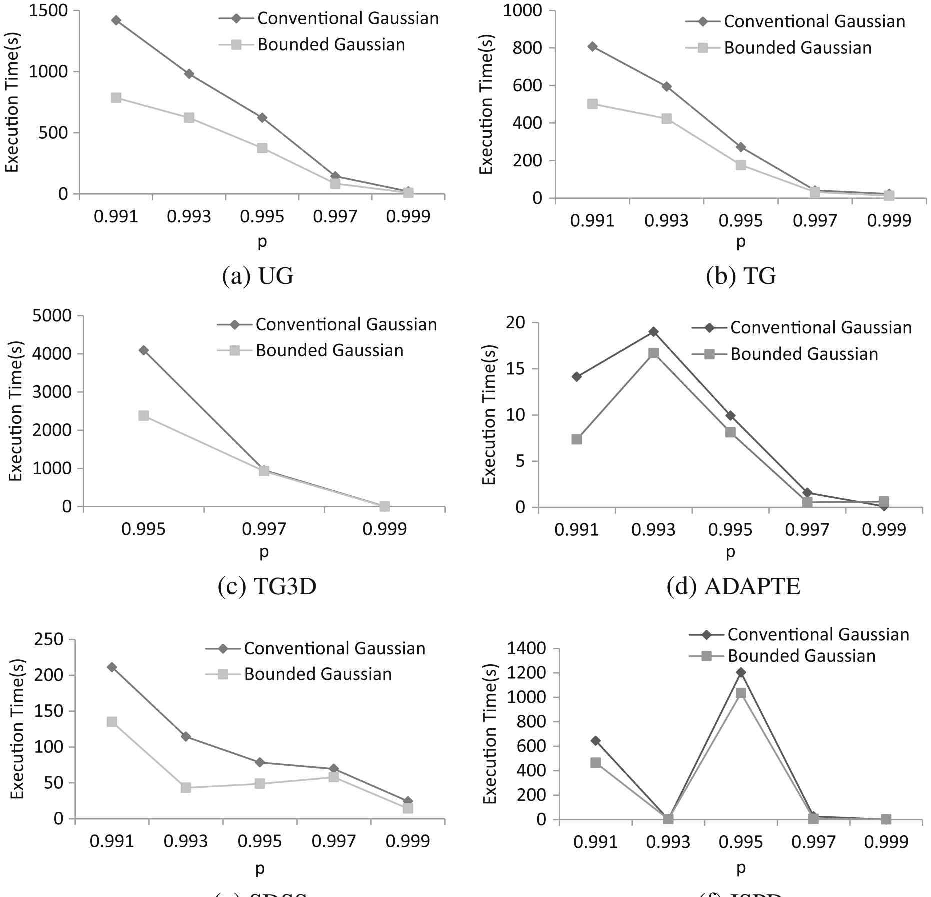
(f) ISPD

Figure 10 Varying $D(\sigma = 10, l = 5(l = 10 \text{for 3D dataset}), n_{\text{bounds}} = 10, rad = 3\sigma, p = 0.995)$

The abnormal curves in Figure 11d and f are due to the number of un-pruned objects. In some experiments, the number of un-pruned objects are quite small or even zero, resulting in dramatic decrease in execution times of the algorithms. In the proposed algorithms, un-pruned objects effect severely on execution times, as can be observed from Figure 12a.

Figure 12b compares the pre-computation times of the conventional Gaussian and the bounded Gaussian algorithms. Since the conventional Gaussian algorithm needs to compute two types of bounds, i.e., $Pr(\alpha, D)$ values for cell bounds and object-wise bounds, the pre-computation times of the conventional Gaussian are higher than the bounded Gaussian algorithm (note the logarithmic scale in Figure 12b). Furthermore, the pre-computation times for 3-dimensional datasets are far higher than 2-dimensional datasets. This is due to the fact that the distance function becomes costlier with the increase in dimensions.

Cell-based approach is capable of pruning majority of objects. This can be observed from plots in Figure 13. In most of the experiments, more than 90 % of the



(a) UG

(b) TG

(c) TG3D

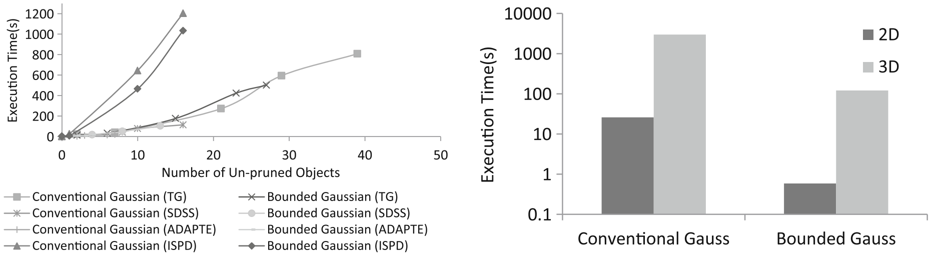
(d) ADAPTE

(e) SDSS

(f) ISPD

Figure 11 Varying p ($D = 100, \sigma = 10, l = 5 (l = 10$ for 3D dataset$), n_{\text{bounds}} = 10, rad = 3\sigma$)

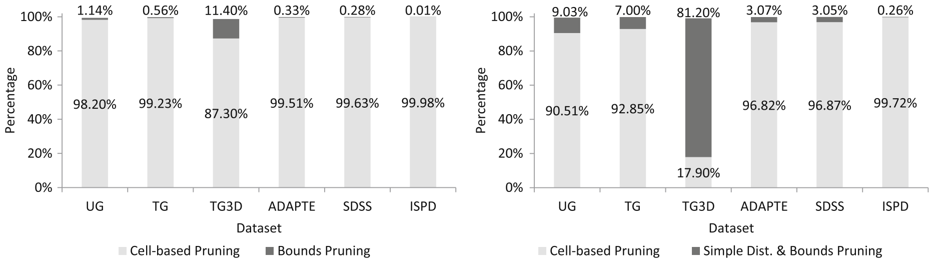
objects are pruned by the cell-based approach. However, in case of 3D experiments, small percentages of objects are pruned by the cell-based approach. This is due to the fact that in higher dimensions, objects are sparse and therefore cell-based pruning



(a) Effect of Un-pruned Objs. on Exec. Time

(b) Pre-computation Time

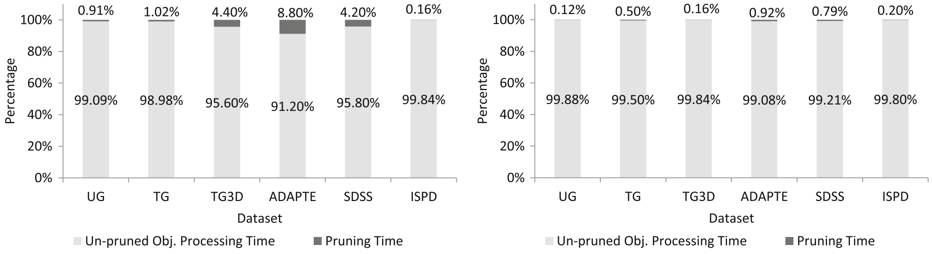
Figure 12 Effect of un-pruned objects on exec. time and pre-computation time



(a) Conventional Gaussian

(b) Bounded Gaussian

Figure 13 Percentage of pruned objects



(a) Conventional Gaussian

(b) Bounded Gaussian

Figure 14 Split of execution times

is not very much effective. Here the importance of object-wise pruning becomes evident as can be observed from Figure 13a and b.

Finally, the split of execution times for different phases of the proposed algorithms are shown in graphs of Figure 14. The time taken by un-pruned objects (which makes use of nested loop) is highest due to the costly distance function, although the percentages of un-pruned objects are very small. On the other hand, the cell-based pruning and the bounds pruning techniques require only fraction of the overall algorithm time.

7 Conclusion

In this work, two approaches of distance-based outlier detection on uncertain datasets are proposed. Firstly a cell-based approach of distance-based outlier detection on uncertain objects following the Gaussian distribution is proposed. Secondly an approximate cell-based approach of outlier detection using the bounded Gaussian distribution is proposed to increase the efficiency of outlier detection. Approximating the Gaussian distribution with the bounded Gaussian distribution enables more effective pruning. An extensive empirical study on real and synthetic datasets demonstrate the effectiveness, efficiency and scalability of the proposed approaches.

Acknowledgements 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).

Appendix: Proof of Lemma

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.

$$f_{\overrightarrow{\mathcal{A}}}(x_1, ..., x_k) = \frac{1}{\sqrt{(2\pi)^k det\Sigma}} \exp\left\{ -\frac{(\overrightarrow{\mathcal{A}} - \overrightarrow{\mu})^T \Sigma^{-1} (\overrightarrow{\mathcal{A}} - \overrightarrow{\mu})}{2} \right\}.$$

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.

$$f_{\overrightarrow{\mathcal{A}}}(x_1, ..., x_k) = \prod_{1 \le i \le k} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{ -\frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right\}. \tag{7}$$

Let o_i and o_j are two k-dimensional uncertain objects with attributes $\overrightarrow{\mathcal{A}_i} = (x_{i,1}, ..., x_{i,k})^T$ and $\overrightarrow{\mathcal{A}_j} = (x_{j,1}, ..., x_{j,k})^T$, 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 $\Sigma_i = diag(\sigma_{i,1}^2, ..., \sigma_{i,k}^2)$ and $\Sigma_j = diag(\sigma_{j,1}^2, ..., \sigma_{j,k}^2)$, respectively. Assuming that $\overrightarrow{\mathcal{A}_i}$ and $\overrightarrow{\mathcal{A}_j}$ are independent random vectors, then $\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j} = \mathcal{N}(\overrightarrow{\mu_i} - \overrightarrow{\mu_j}, \Sigma_i + \Sigma_j)$ [32]. Since Σ_i and Σ_j are diagonal matrices, the k-dimensional normal difference distribution function can be given by the product of k 1-dimensional normal distribution functions as follows.

$$f_{\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j}}(x_1, ..., x_k) = \prod_{1 \le m \le k} \frac{1}{\sqrt{2\pi(\sigma_{i,m}^2 + \sigma_{j,m}^2)}} \exp\left\{ -\frac{(x_m - (\mu_{i,m} - \mu_{j,m}))^2}{2(\sigma_{i,m}^2 + \sigma_{j,m}^2)} \right\}. \tag{8}$$

Since this lemma focus on 2-dimensional $Pr(o_i, o_j, D)$. The normal difference distribution of 2-dimensional uncertain objects o_i and o_j is given by,

$$f_{\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j}}(x_1, x_2) = \frac{1}{2\pi\sqrt{(\sigma_{i,1}^2 + \sigma_{j,1}^2)(\sigma_{i,2}^2 + \sigma_{j,2}^2))}} \exp\left\{ -\left(\frac{(x_1 - \alpha_1)^2}{2(\sigma_{i,1}^2 + \sigma_{j,1}^2)} + \frac{(x_2 - \alpha_2)^2}{2(\sigma_{i,2}^2 + \sigma_{j,2}^2)} \right) \right\}, \tag{9}$$

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 respectively. Hence the probability that $o_j \in DN(o_i)$ denoted by $Pr(o_i, o_j, D)$, is given as follows.

$$\begin{aligned} Pr(o_i, o_j, D) = & \frac{1}{2\pi\sqrt{\left(\sigma_{i,1}^2 + \sigma_{j,1}^2\right)\left(\sigma_{i,2}^2 + \sigma_{j,2}^2\right)}} \\ & \times \int_0^D \int_0^{2\pi} \exp\left\{ -\left(\frac{(r\cos\theta - \alpha_1)^2}{2\left(\sigma_{i,1}^2 + \sigma_{j,1}^2\right)} + \frac{(r\sin\theta - \alpha_2)^2}{2\left(\sigma_{i,2}^2 + \sigma_{j,2}^2\right)} \right) \right\} r \, d\theta \, dr \end{aligned} \tag{10}$$

□

References

1. Aggarwal, C.C., Yu, P.S.: Outlier detection with uncertain data. In: SIAM International Conference on Data Mining, pp. 483–493 (2008)
2. Alaydie, N., Fotouhi, F., Reddy, C.K., Soltanian-Zadeh, H.: Noise and outlier filtering in heterogeneous medical data sources. In: Workshops on Database and Expert Systems Applications, DEXA, pp. 115–119 (2010)
3. Angiulli, F., Pizzuti, C.: Fast outlier detection in high dim. spaces. In: PKDD, pp. 15–26 (2002)
4. Angiulli, F., Fassetti, F.: Detecting distance-based outliers in streams of data. In: CIKM, pp. 811–820 (2007)
5. Arturo, E., Alberto, O.Z., Alejandro, P., Julio, P.: Outlier analysis for plastic card fraud detection a hybridized and multi-objective approach. In: Hybrid Artificial Intelligent Systems, LNCS, pp. 1–9 (2011)
6. Barnett, V., Lewis, T.: Outliers in Statistical Data. Wiley, New York (1994)
7. CISL Research Data Archive. http://rda.ucar.edu. Accessed 16 July 2012
8. Diao, Y., Li, B., Liu, A., Peng, L., Sutton, C., Tran, T., Zink, M.: Capturing data uncertainty in high-volume stream processing. In: CIDR (2009)
9. Garces, H., Sbarbaro, D.: Outliers detection in environmental monitoring databases. Eng. Appl. Artif. Intell. **24**(2), 341–349 (2011)
10. Hawkins, D.: Identification of Outliers. Chapman and Hall, London (1980)
11. Helm, I., Jalukse L., Leito I.: Measurement uncertainty estimation in amperometric sensors: a tutorial review. Sensors **10**(5), 4430–4455 (2010)
12. International Surface Pressure Databank (ISPDv2) 1768–2010. http://rda.ucar.edu/datasets/ds132.0/index.html. Accessed 16 July 2012
13. Ishida, K., Kitagawa, H.: Detecting current outliers: continuous outlier detection over time-series data streams. In: DEXA, pp. 255–268 (2008)
14. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: Proceedings of 24th VLDB, pp. 392–403 (1998)
15. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. VLDB J. **8**(3–4), 237–253 (2000)
16. Kontaki, M., Gounaris, A., Papadopoulos, A.N., Tsichlas, K., Manolopoulos, Y.: Continuous monitoring of distance-based outliers over data streams. In: ICDE, pp. 135–146 (2011)
17. Mahoney, M., Chan, P.: Learning rules for anomaly detection of hostile network traffic. In: Proceedings of the 3rd ICDM, pp. 601–604 (2003)
18. Maimon, O., Rockach, L.: Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers. Kluwer Academic, Norwell (2005)
19. Ngai, W.K., Kao, B., Chui, C.K., Cheng, R., Chau, M., Yip, K.Y.: Efficient clustering of uncertain data. In: ICDM, pp. 436–445 (2006)
20. Nievergelt, J., Hinterberger, H., Sevick, K.C.: The Grid file: an adaptable, symmetric multikey file structure. ACM Trans. Database Syst. **9**(1), 38–71 (1984)
21. Orair, G.H., Teixeira, C.H.C., Meira, W.: Distance-based outlier detection: consolidation and renewed bearing. In: Proc. of the VLDB Endowment, pp. 1469–1480 (2010)
22. Pukelsheim, F.: The three sigma rule. Am. Stat. **48**(2), 88–91 (1994)
23. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: ACM SIGMOD, pp. 427–438 (2000)
24. Shaikh, S.A., Kitagawa, H.: Distance-based outlier detection on uncertain data of Gaussian distribution. In: APWeb, pp. 109–121 (2012)
25. Sharma, A.B., Golubchik, L., Govindan, R.: Sensor faults: detection methods and prevalence in real-world datasets. ACM Trans. Sens. Netw. **6**(3), 23:1–39 (2010)
26. Sloan Digital Sky Survey. http://www.sdss.org. Accessed 16 July 2012
27. Stevens Water Monitoring Systems, Inc. http://www.stevenswater.com/. Accessed 7 March 2013
28. Tao, Y., Xiao, X., Cheng, R.: Range search on multidimensional uncertain data. ACM Trans. Database Syst. **32**(3), 15:1–54 (2007)
29. Thistleton, W., Marsh, J.A., Nelson, K., Tsallis, C.: Generalized Box-Muller method for generating q-Gaussian random deviates. IEEE Trans. Inf. Theory **53**(12), 4805–4810 (2007)
30. Vaisala Corporation. http://www.vaisala.com/. Accessed 7 March 2013
31. Wang, B., Xiao, G., Yu, H., Yang, X.: Distance-based outlier detection on uncertain data. In: IEEE 9th International Conference on Computer and Information Technology, pp. 293–298 (2009)

32. Weisstein, E.W.: Normal Difference Distribution. From MathWorld—A Wolfram Web Resource. http://mathworld.wolfram.com/NormalDifferenceDistribution. Accessed 27 Jan 2012
33. Xylem Corporation. http://www.globalw.com/. Accessed 7 March 2013