International Journal of Automation and Computing 11(2), April 2014, 128-142
DOI: 10.1007/s11633-014-0775-8

Top-k Outlier Detection from Uncertain Data

Salman Ahmed Shaikh Hiroyuki Kitagawa

Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan

Abstract: Uncertain data are common due to the increasing usage of sensors, radio frequency identification (RFID), GPS and similar devices for data collection. The causes of uncertainty include limitations of measurements, inclusion of noise, inconsistent supply voltage and delay or loss of data in transfer. In order to manage, query or mine such data, data uncertainty needs to be considered. Hence, this paper studies the problem of top-k distance-based outlier detection from uncertain data objects. In this work, an uncertain object is modelled by a probability density function of a Gaussian distribution. The naive approach of distance-based outlier detection makes use of nested loop. This approach is very costly due to the expensive distance function between two uncertain objects. Therefore, a populated-cells list (PC-list) approach of outlier detection is proposed. Using the PC-list, the proposed top-k outlier detection algorithm needs to consider only a fraction of dataset objects and hence quickly identifies candidate objects for top-k outliers. Two approximate top-k outlier detection algorithms are presented to further increase the efficiency of the top-k outlier detection algorithm. An extensive empirical study on synthetic and real datasets is also presented to prove the accuracy, efficiency and scalability of the proposed algorithms.

Keywords: Top-k distance-based outlier detection, uncertain data, Gaussian uncertainty, cell-based approach, PC-list based approach.

1 Introduction

Outlier detection is a fundamental problem in data mining. It has applications in many domains including credit card fraud detection[1], network intrusion detection[2], industrial damage detection[3], environment monitoring[4], medical sciences[5], etc. Several definitions of outlier have been given in the past, but there exists no universally agreed definition. Hawkins[6] defined an outlier as an observation that deviates so much from other observations as to arouse suspicion that it is generated by a different mechanism. Barnet and Lewis[7] 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[7, 8]. However, most statistical techniques are not useful in computer science 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 outlier detection approaches have been proposed for data mining[9−16].

Most of the outlier detection techniques proposed for data mining are suitable only for deterministic data. However, due to the increasing usage of sensors, radio frequency identifications (RFIDs), GPS and similar devices for data collection these days, data contains certain degree of inherent uncertainty[17−20]. The causes of uncertainty may include but are not limited to limitation of equipment, absence of data, inconsistent supply voltage and delay or loss of data in transmit[17]. In order to get reliable results from such data, uncertainty needs to be considered in calculation. Therefore this work presents a top-k distance-based outlier detection technique on uncertain data, where the uncertainty of data is modelled by the most commonly used probability density function, i.e., the Gaussian distribution.

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 measurements. Due to calibration errors, short-circuited connections, damaged sensors and low battery voltage, sensor reported measurements may be different from true measurements[17]. In other words, such measurements are uncertain values. Therefore, commercial sensor producers always mention accuracy (measurement error) on their products. Table 1 lists the 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.

To obtain top-k distance-based outliers from uncertain datasets using our proposed approach, distance needs to be calculated between uncertain data objects. However, the distance computation between uncertain data objects is very costly. Therefore, a populated-cells list (PC-list) based outlier detection approach is proposed in this paper to quickly identify the top-k outliers from uncertain data. The PC-list is a sorted list of non-empty cells of a d-dimensional grid, where grid is used to index data objects[21]. Using the PC-list, the top-k outlier detection algorithm needs to consider only a fraction of the dataset objects and hence quickly identifies candidate objects for the top-k outliers. Finally, exact outlier score ($\#D$-neighbors) is computed for each candidate object to find the top-k outliers and their ranking. Furthermore, two approximate top-k outlier detection algorithms are also presented in this work to increase

Manuscript received July 31, 2013; revised November 11, 2013
This work was partly supported by Grant-in-Aid for Scientific Research(A)(#24240015A).

Table 1 Uncertainty in commercial sensor measurements

Company	Sensor type	Model	Parameter	Max measurement error (%)*
Stevens[22]	Weather	WXT520	Air temperature	1
			Barometric pressure	0.2
			Relative humidity	5
			Wind speed	5
			Wind direction	3
	Pyranometer	LI200SA	Solar radiation	5
Vaisala[23]	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[24]	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 efficiency of the outlier detection algorithm. The first approximate algorithm only approximates the candidate objects' $\#D$-neighbors, while the second approximate algorithm makes use of the bounded Gaussian uncertainty to increase the efficiency of the top-k outlier detection algorithm. These approximate algorithms are denoted by top-k approx and top-k BG, respectively in the following.

The rest of the paper is organized as follows. Section 2 surveys the related work. Section 3 formally defines the top-k distance-based outlier detection on uncertain datasets. The PC-list based pruning is presented in Section 4. The top-k and the top-k approx algorithms are presented in Section 5. Section 6 presents a top-k algorithm using the bounded Gaussian uncertainty (the top-k BG). Section 7 contains an extensive experimental evaluation that demonstrates the accuracy, efficiency and scalability of the proposed algorithms. Section 8 concludes our paper.

2 Related work

The very first definition of distance-based outlier was given by Knorr et al.[10]. They defined an object o to be an outlier if at most $M = N(1-p)$ objects are within D-distance of o, where N is the number of objects in the dataset, and p is the fraction of objects that lie farther than D-distance of o. They also presented a cell-based approach to efficiently compute the distance-based outliers. Ramaswamy et al.[25] formulated distance-based outliers as the top-t data objects whose distance to their k-th nearest neighbour is largest. Angiulli and Pizzuti[26] gave a slightly different definition of outliers than Ramaswamy et al.[25] 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[17−29]. These works are based on the definition of distance-based outliers by Knorr et al.[10]. Furthermore, Angiulli and Fassetti[27] gave an approximate algorithm to reduce the memory space required by its exact counterpart. Later on Kontaki et al.[28] extended Angiulli and Fassetti′s work[27] 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[15, 30]. The problem of outlier detection on uncertain datasets was first studied by Aggarwal and Yu[30]. According to them, 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 the density of all subspaces needs to be computed and then the η-probability of each o in the dataset is computed to tell o is an outlier. Since this computation is very expensive, a sampling procedure is used to approximate the η-probability. In contrast to Aggarwal and Yu′s work[30], this paper addresses the detection of distance-based outliers in full space, where the distance between two uncertain objects is computed by the Gaussian difference distribution[31]. Therefore, the problem definition is quite different from Aggarwal and Yu[30].

Wang et al.[15] also proposed outlier detection on uncertain data. Their work focused on the uncertainty in the existence of a data object. In contrast, in this paper, the uncertainty lies in the measurements obtained from sensors. Each tuple in Wang et al.′s work[15] is associated with the confidence of appearing at a corresponding location. However in this work, each uncertain object is represented by a Gaussian probability density function (PDF), with an assumption that sensor measurements may deviate from true values due to the reasons discussed in Section 1.

We also proposed a cell-based approach of distance-based outlier detection on uncertain data[32, 33]. According to our previous works[32, 33], an uncertain object o is a distance-based outlier if the expected number of objects lying within its D-distance is not greater than $M = N(1-p)$, where N is the number of objects in the dataset and p is the fraction of objects that lies farther than D-distance of o. A prob-

lem with our current work is that parameter p is difficult to determine and is dependent on N.

An arbitrary value of p may result in a very few or a lot of outliers for a different N. Moreover, outlier′s ranking cannot be obtained.

In all the existing works including our previous work[32, 33], an object can be either classified as outlier or inlier. Since there is no universally agreed definition of outliers, different algorithms return different outliers depending upon the combination of parameter values. Some combinations return a very few while others return a lot of outliers. Moreover, no outlier ranking is available and users are unable to differentiate between strong and weak outliers. Therefore, we presented a top-k approach of distance-based outliers in one of our recent works[34]. The proposed approach[34] returns k objects with lowest outlier scores ($\#D$-neighbors), in other words, k strongest outliers along with their ranking.

This paper is an extended version of our recent work[34]. The main contributions of this paper include a bounded Gaussian approach of the top-k outlier detection presented in Section 6, complexity analysis of the proposed top-k algorithms (Section 5.3), discussion on the determination of values for parameters D and l (Section 5.4), detailed experiments comparing the accuracy of the proposed approach with the deterministic approach of outlier detection by Knorr et al.[10] and an extensive empirical study on performance using larger real and synthetic datasets. To the best of our knowledge, the top-k distance-based outlier detection on uncertain datasets has not been studied by other researchers.

3 Outliers in uncertain data

Outlier detection is a significant problem in the field of data mining. Due to its importance, several outlier detection approaches have been proposed for data mining. These include the nearest-neighbour based[25], density based[35], clustering based[36] and distance based[10, 11]. Due to the increasing usage of automatic data collection devices, i.e., sensors, RFIDs, etc., measurements contain inherent uncertainty (refer to Table 1). Therefore, outlier detection from uncertain data is gaining popularity and a lot of researchers are focusing on it[15, 30, 37]. In this work, our focus is distance-based approach because distance-based approaches are the simplest and the most commonly used. Moreover, distance-based approaches are useful in modelling other data mining techniques i.e., k-nearest neighbours, clustering, etc.

3.1 Distance-based outliers in uncertain data

The very first definition of distance-based outlier on deterministic data was given by Knorr et al.[10]. They defined an object o to be an outlier if at most M objects are within D-distance of o. This definition was given for deterministic datasets. Later we extended this definition to uncertain datasets whose attribute values are uncertain[32, 33]. In that work, we assumed that the uncertainty of the data objects follow a 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, d-dimensional uncertain objects o_i are considered, with attribute $\overrightarrow{\mathcal{A}_i} = [x_{i,1}, \cdots, x_{i,d}]^{\mathrm{T}}$ following the Gaussian PDF with mean $\overrightarrow{\mu_i} = [\mu_{i,1}, \cdots, \mu_{i,d}]^{\mathrm{T}}$ and covariance matrix $\Sigma_i = \mathrm{diag}(\sigma_{i,1}^2, \cdots, \sigma_{i,d}^2)$, respectively. Namely, vector $\overrightarrow{\mathcal{A}_i}$ is a random variable that follows the 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 an object o_i. The complete database consists of a set of such objects, $\mathcal{GDB} = \{o_1, \cdots, o_N\}$, where $N = |\mathcal{GDB}|$ is the number of uncertain objects in $\mathcal{GDB}$. Hence, the Knorr et al.[10] definition can be extended naturally to uncertain datasets as follows.

Definition 1. 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.

3.2 Top-k distance-based outliers in uncertain data

Since the focus of this work is the detection of top-k distance-based outliers and their ranking from uncertain data, Definition 1 can be modified into the top-k distance-based outliers as follows.

Definition 2. The top-k distance-based outliers are the k uncertain objects in the dataset $\mathcal{GDB}$ for which the expected number of objects $o_i \in \mathcal{GDB}$ lying within D-distance is the smallest.

The objects that lie within the D-distance of o_i are called its D-neighbors, and the set of the D-neighbors of o_i and the number of D-neighbours are denoted by $DN(o_i)$ and $\#D\text{-neighbors}(o_i)$ or $\#D(o_i)$, respectively. In order to find the top-k distance-based outliers in $\mathcal{GDB}$, the distance between uncertain objects needs to be calculated, which is given by another distribution known as the Gaussian difference distribution[31]. Let $\overrightarrow{\mathcal{A}_i}$ and $\overrightarrow{\mathcal{A}_j}$ be two independent d-dimensional normal random vectors with means $\overrightarrow{\mu_i} = [\mu_{i,1}, \cdots, \mu_{i,d}]^{\mathrm{T}}$ and $\overrightarrow{\mu_j} = [\mu_{j,1}, \cdots, \mu_{j,d}]^{\mathrm{T}}$ and diagonal covariance matrices $\Sigma_i = \mathrm{diag}(\sigma_{i,1}^2, \cdots, \sigma_{i,d}^2)$ and $\Sigma_j = \mathrm{diag}(\sigma_{j,1}^2, \cdots, \sigma_{j,d}^2)$, respectively. Then, $\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j} = \mathcal{N}(\overrightarrow{\mu_i} - \overrightarrow{\mu_j}, \Sigma_i + \Sigma_j)$[31]. Let $Pr(o_i, o_j, D)$ denotes the probability that $o_j \in DN(o_i)$. Then,

$$Pr(o_i, o_j, D) = \int_R \mathcal{N}(\overrightarrow{\mu_i} - \overrightarrow{\mu_j}, \Sigma_i + \Sigma_j) \mathrm{d}\overrightarrow{\mathcal{A}} \qquad (1)$$

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, the $Pr(o_i, o_j, D)$ expressions for higher dimensions can be derived using (1).

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 $\overrightarrow{\mu_i} = [\mu_{i,1}, \mu_{i,2}]^{\mathrm{T}}$, $\overrightarrow{\mu_j} = [\mu_{j,1}, \mu_{j,2}]^{\mathrm{T}}$, $\Sigma_i = \mathrm{diag}(\sigma_{i,1}^2, \sigma_{i,2}^2)$ and $\Sigma_j = \mathrm{diag}(\sigma_{j,1}^2, \sigma_{j,2}^2)$. The $Pr(o_i, o_j, D)$

is given as follows.

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

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 (2) is simplified as

$$Pr(o_i, o_j, D) = \frac{1}{4\pi\sigma^2} \times \int_0^D \int_0^{2\pi} \mathrm{e}^{\left\{\frac{-1}{4\sigma^2}(r^2 - 2\alpha r\cos\theta + \alpha^2)\right\}} r\mathrm{d}\theta\,\mathrm{d}r. \quad (3)$$

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, where α is an ordinary Euclidean distance between the means of $o_i \in \mathcal{G}DB$ and $o_j \in \mathcal{G}DB$. Computing this probability is usually very costly, and needs to be avoided as much as possible during the computation of outliers. In the following, we assume that the standard deviation is uniform in all the dimensions to keep the discussion simple.

The naive approach of the top-k outlier detection given in Algorithm 1 uses nested-loop. In order to find whether an object $o_i \in \mathcal{G}DB$ is a top-k outlier, its $\#D$-neighbours ($\#D(o_i)$) are computed. Computation of $\#D(o_i)$ for an object $o_i \in \mathcal{G}DB$ requires evaluation of N expensive distance functions. During the computation of $\#D(o_i)$, if it becomes greater than threshold θ, o_i is an inlier and the computation of $\#D(o_i)$ is stopped. On the other hand, if $\#D(o_i)$ is less than or equal to θ, o_i is added to the candidate list of outliers $\mathcal{C}_{obj}$, along with its $\#D$-neighbours. The $\mathcal{C}_{obj}$ is kept sorted in the ascending order of $\#D$-neighbours and the k objects in it with lowest $\#D$-neighbours are selected as the outliers. In the worst case, this approach requires $\mathrm{O}(N^2)$ evaluations of costly distance function, which is computationally very expensive.

Algorithm 1. The top-k naive approach.
Input: $\mathcal{G}DB$, D, k.
Output: Top-k distance-based outliers.
1. $N \leftarrow |\mathcal{G}DB|$, $\theta \leftarrow \infty$, $\mathcal{C}_{obj} \leftarrow \phi$ (Candidate top-k outliers list);
2. **for each** o_i in $\mathcal{G}DB$
3. $\#D(o_i) \leftarrow 0$; ($\#D$-neighbours of o_i)
4. **for each** o_j in $\mathcal{G}DB$ **do**
5. $\#D(o_i) += Pr(o_i, o_j, D)$;
6. **if** $\#D(o_i) > \theta$ **then** GOTO next o_i;
7. **end for**
8. Insert o_i and its $\#D(o_i)$ into $\mathcal{C}_{obj}$ (Keep $\mathcal{C}_{obj}$ sorted of $\#D(o_i)$);
9. if $|\mathcal{C}_{obj}| > k$ **then**
10. Set $\theta = \#D(o')$, where o' is the k-th object in $\mathcal{C}_{obj}$;
11. Remove all $o'' \in \mathcal{C}_{obj}$, such that $\#D(o'') > \theta$;
12. **end if**
13. **end for**
14. **return** $\mathcal{C}_{obj}$.

4 PC-list based outlier detection

The naive approach requires a lot of computation time to detect the top-k outliers even from a small dataset due to the costly distance calculation. To overcome this problem a populated-cells list (PC-list) based approach of the top-k distance-based outlier detection is proposed. The PC-list is an array of non-empty cells of a d-dimensional grid. The grid is used to index dataset objects in $\mathcal{G}DB$. The PC-list helps in the detection of top-k distance-based outliers by identifying the grid cells containing candidate outliers.

Lemma 2. Let $o_i, o_j \in \mathcal{G}DB$ be two d-dimensional uncertain data objects following the Gaussian distribution and α denotes an ordinary Euclidean distance between the means of o_i and o_j. Then for $t \in \mathbf{R}$, by denoting the number of standard deviations required to enclose a large probability (say $> 99\%$) of a d-dimensional Gaussian difference distribution, the following statements hold.

$$(a)\ \text{If } \alpha \leqslant D - t\sigma', \quad Pr(o_i, o_j, D) \approx 1$$
$$(b)\ \text{If } \alpha \geqslant D + t\sigma', \quad Pr(o_i, o_j, D) \approx 0$$

where σ' is the standard deviation of the Gaussian difference distribution in any one dimension (assuming that the standard deviation is uniform in all the dimensions).

Proof. The number of standard deviations s needed to enclose a given probability for a d-dimensional random variable X following the Gaussian distribution can be obtained using the expression $Pr\{d_M(X, \mu) \leqslant s\} = G_d(s^2)$[38], where $d_M(X, \mu) = \sqrt{(X-\mu)^{\mathrm{T}} \sum^{-1} (X-\mu)}$ is the Mahalanobis distance and $G_d(s^2)$ is the cumulative distribution function (CDF) of the chi-squared distribution with d-degrees of freedom.

Here, we are interested in computing the distance between two uncertain objects o_i and o_j following the Gaussian distribution. This distance is given by another Gaussian distribution known as the Gaussian difference distribution[31]. Hence if t denotes the value of s, such that $Pr\{d_M(X, \mu) \leqslant t\}$ covers a large area of the Gaussian distribution (say $> 99\%$), then for $\alpha \leqslant D - t\sigma'$, $Pr(o_i, o_j, D) \approx 1$ and for $\alpha \geqslant D + t\sigma'$, $Pr(o_i, o_j, D) \approx 0$. □

4.1 Grid $\mathcal{G}$ structure

In order to find the top-k distance-based outliers from uncertain dataset using the PC-list, each object in $\mathcal{G}DB$ is quantized to a d-dimensional grid $\mathcal{G}$ that is partitioned into cells of length l (The cell length is discussed in Section 5.4). Let $C_{\psi_1,\cdots,\psi_d}$ be any cell in $\mathcal{G}$, where positive integers $\psi_1, \cdots, \psi_d$ denote the cell indices. The layers $(L_1, \cdots, L_n)$ of $C_{\psi_1,\cdots,\psi_d} \in \mathcal{G}$ are the neighbouring cells of $C_{\psi_1,\cdots,\psi_d}$, as shown in Fig. 1 and are derived as follows:

$$L_1(C_{\psi_1,\cdots,\psi_d}) = \{C_{x_1,\cdots,x_d} | x_1 = \psi_1 \pm 1, \cdots, x_d = \psi_d \pm 1, C_{x_1,\cdots,x_d} \neq C_{\psi_1,\cdots,\psi_d}\} \quad (4)$$

$$L_2(C_{\psi_1,\cdots,\psi_d}) = \{C_{x_1,\cdots,x_d} | x_1 = \psi_1 \pm 2, \cdots, x_d = \psi_d \pm 2, C_{x_1,\cdots,x_d} \notin L_1(C_{\psi_1,\cdots,\psi_d}), C_{x_1,\cdots,x_d} \neq C_{\psi_1,\cdots,\psi_d}\}. \quad (5)$$

$L_3(C_{\psi_1,\cdots,\psi_d}), \cdots, L_n(C_{\psi_1,\cdots,\psi_d})$ are derived in a similar way. We will use C to denote $C_{\psi_1,\cdots,\psi_d}$ when there is no confusion.

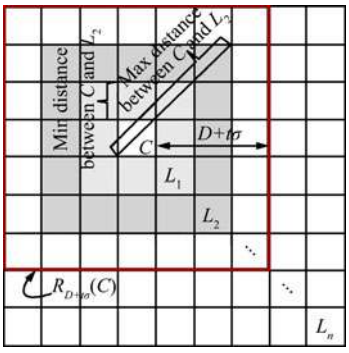
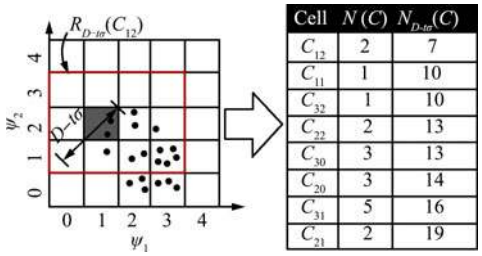


Fig. 1 Cell layers and bounds

Let $R_{D-t\sigma}(C)$ denote a region formed by $\left\lfloor \frac{D-t\sigma}{l\sqrt{d}} - 1 \right\rfloor$ neighbouring layers of $C \in \mathcal{G}$ as shown in Fig. 2. The region $R_{D-t\sigma}(C)$ is chosen in such a way that for each $o_i \in C$ and $o_j \in R_{D-t\sigma}(C)$, $Pr(o_i, o_j, D) \approx 1$. Similarly, $R_{D+t\sigma}(C)$ denotes a region formed by $\lceil \frac{D+t\sigma}{l} \rceil$ neighbouring layers of cell $C \in \mathcal{G}$ as shown in Fig. 1. Region $R_{D+t\sigma}(C)$ is chosen in such a way that for each $o_i \in C$ and $o_j \notin R_{D+t\sigma}(C)$, $Pr(o_i, o_j, D)$ approaches zero.



Cell	$N(C)$	$N_{D-t\sigma}(C)$
C_{12}	2	7
C_{11}	1	10
C_{32}	1	10
C_{22}	2	13
C_{30}	3	13
C_{20}	3	14
C_{31}	5	16
C_{21}	2	19

Fig. 2 PC-list building

4.2 PC-list structure

Populated-cells list (PC-list) is an array of non-empty cells of a d-dimensional grid. Let $N(C)$ be the number of objects in C, and $N_{D-t\sigma}(C)$ be the number of objects within cells in region $R_{D-t\sigma}(C)$ (including C itself). Then the PC-list (PC) is a sorted list containing $N(C)$ and $N_{D-t\sigma}(C)$ for each non-empty cell $C \in \mathcal{G}$ as shown in Fig. 2. The tuples in the PC-list are sorted in an ascending order of $N_{D-t\sigma}(C)$ column. The idea behind sorting is that outliers tend to exist in sparse regions. Sorting tuples in the PC-list, lets us identify cells with few number of neighbouring objects or cells in sparse regions.

The PC-list constructed in such a way that the majority of cells at the top of the PC-list contain candidate outlier objects. To prune the cells in the PC-list which cannot contain top-k outliers, cell bounds are computed. In practice, only small percentage of cells at the top of the PC-list require bounds computation. The rest of the cells are pruned as inlier cells, i.e., the cells containing only inlier objects.

4.3 Cell bounds

In order to identify cells $C \in PC$, containing only inliers or candidate top-k outliers, their bounds on the $\#D$-neighbours are used. A cell C can be pruned as an inlier cell if the minimum $\#D$-neighbours for any object in C is greater than threshold θ (θ is discussed shortly). Similarly, a cell can be identified as containing top-k outliers (candidate outlier cell) if the maximum $\#D$-neighbours for any object in C is less than θ. Since the Gaussian distribution is unbounded, $Pr(o_i, o_j, D)$ is always greater than zero for $o_i, o_j \in \mathcal{G}DB$. Therefore all the cells in the PC-list need to be considered for the computation of bounds of $C \in PC$. To compute cell bounds, the minimum and the maximum ordinary Euclidean distances between cells are required. Beside this, object count of each $C \in PC$ and $Pr(\alpha, D)$ values for α ranging from the minimum to the maximum ordinary Euclidean distances between cells in $\mathcal{G}$ are also required. The $Pr(\alpha, D)$ values are precomputed and stored in a look-up table to be used by the top-k outlier detection algorithm.

4.3.1 Distance between cells

Let C_p and C_q be two cells in PC with indices $\psi_{p1}, \cdots, \psi_{pd}$ and $\psi_{q1}, \cdots, \psi_{qd}$, respectively. Let $\Delta_{\min}(C_p, C_q)$ and $\Delta_{\max}(C_p, C_q)$ denote the minimum and the maximum ordinary Euclidean distances between C_p and C_q, respectively. Distance between C_p and C_q depends on their positions in grid $\mathcal{G}$ and can be derived as

$$\Delta_{\min}(C_p, C_q) = l(\sum_{s=1}^{d} \delta_{\min,s}^2)^{\frac{1}{2}} \tag{6}$$

where $\delta_{\min,s} = \begin{cases} \psi_{ps} - (\psi_{qs} + 1), & \psi_{ps} > \psi_{qs} \\ (\psi_{ps} + 1) - \psi_{qs}, & \psi_{ps} < \psi_{qs} \\ \psi_{ps} - \psi_{qs}, & \psi_{ps} = \psi_{qs} \end{cases}$

$$\Delta_{\max}(C_p, C_q) = l(\sum_{s=1}^{d} \delta_{\max,s}^2)^{\frac{1}{2}} \tag{7}$$

where $\delta_{\max,s} = \begin{cases} (\psi_{ps} + 1) - \psi_{qs}, & \psi_{ps} \geqslant \psi_{qs} \\ \psi_{ps} - (\psi_{qs} + 1), & \psi_{ps} < \psi_{qs}. \end{cases}$

Now the bounds for the PC-list cells can be obtained using pre-computed $Pr(\alpha, D)$ values and the information available in the PC-list. Let $LB(Pr(C_p, C_q))$ and $UB(Pr(C_p, C_q))$ denote $Pr(\alpha, D)$ values at minimum $\alpha \geqslant \Delta_{\max}(C_p, C_q)$ and maximum $\alpha \leqslant \Delta_{\min}(C_p, C_q)$, respectively. Then for a $C \in PC$, its lower bound $LB(C)$ and upper bound $UB(C)$ are defined as follows:

$$LB(C) = \sum_{C' \in PC} LB(Pr(C, C')) \times N(C') \tag{8}$$

$$UB(C) = \sum_{C' \in PC} UB(Pr(C, C')) \times N(C'). \tag{9}$$

Since the major contribution in the bounds for $C \in PC$ is done by the cells in region $R_{D+t\sigma}(C)$, the bounds for $C \in PC$ can be redefined to reduce the number of precomputations and bounds computation time as follows.

$$LB(C) = \sum_{C' \in \{PC \cap R_{D+t\sigma}(C)\}} LB(Pr(C, C')) \times N(C') \tag{10}$$

$$UB(C) = \sum_{C' \in \{PC \cap R_{D+t\sigma}(C)\}} UB(Pr(C, C')) \times N(C') + Pr(l\sqrt{d}(\lceil \frac{D + t\sigma}{l} \rceil + 1), D) \times (N - \sum_{C' \in \{PC \cap R_{D+t\sigma}(C)\}} \times N(C')). \tag{11}$$

4.3.2 Number of $Pr(\alpha, D)$ pre-computations

Since the bounds of $C \in PC$ are computed using the cells in region $R_{D+t\sigma}(C)$, $Pr(\alpha, D)$ values need to be computed only for the neighbouring layers within $D + t\sigma$ distance of a cell. For $\lceil \frac{D+t\sigma}{l} \rceil$ neighbouring layers, we require $2\lceil \frac{D+t\sigma}{l} \rceil$ pre-computations. Two more pre-computations are required for the cell C itself and the objects that lie greater than $D + t\sigma$ distance of a cell. Hence the total number of pre-computations of $Pr(\alpha, D)$ required are only $2\lceil \frac{D+t\sigma}{l} \rceil + 2$.

4.4 Candidate outlier cells detection

Using the bounds discussed in Section 4.3, a cell can be pruned as an inlier cell, i.e., a cell containing only inlier objects or can be identified as containing the top-k outlier candidates. Let $\mathcal{C}_{\text{cell}}$ be a list for holding candidate outlier cells from PC-list, sorted in the ascending order of $UB(C)$. Let $C^k \in \mathcal{C}_{\text{cell}}$ be a cell with the minimum upper bound containing the k-th object. A $C \in PC$ is a candidate outlier cell whenever $\sum_{C' \in \mathcal{C}_{\text{cell}}} N(C') < k$ or $LB(C) \leqslant \theta$, where $\theta = UB(C^k)$ denotes the threshold. For a $C \in PC$, if $LB(C) > \theta$, then C cannot contain any of the top-k outliers and can be pruned. On the other hand, if $LB(C) \leqslant \theta$, C may contain the top-k outliers. C is added to $\mathcal{C}_{\text{cell}}$, such that $\mathcal{C}_{\text{cell}}$ remains sorted of its $UB(C)$ attribute. Set $\theta = UB(C^k)$ and remove C' from $\mathcal{C}_{\text{cell}}$, such that $LB(C') > \theta$, as they cannot contain the top-k outliers.

Stopping condition: The PC-list is scanned from top to bottom for candidate outlier cells. During the scanning, if a $C' \in PC$ is found such that $Pr(D-t\sigma, D) \times N_{D-t\sigma}(C') > \theta$, which is a lower bound on $\#D$-neighbours of C', C' cannot contain the top-k outliers and can be pruned. Since the PC-list is sorted of $N_{D-t\sigma}(C)$, any cell after C' must have $N_{D-t\sigma}(C) \geqslant N_{D-t\sigma}(C')$. Hence the lower bound of $C \in PC$ after C' must be greater than or equal to the lower bound of $C' \in PC$ and cannot contain the top-k outliers. Hence, the PC-list scanning can be stopped at this point.

5 Top-k and top-k approx algorithms

In this section, we present two algorithms to detect top-k distance-based outliers from uncertain datasets. The first algorithm (top-k) computes accurate $\#D$-neighbours for all the un-pruned objects, however the second algorithm (top-k approx) approximates the $\#D$-neighbours to reduce the algorithm computation cost. In Section 6, we will present another approximate top-k algorithm using the bounded Gaussian uncertainty (top-k BG).

5.1 Top-k algorithm

The Algorithm 2 first maps dataset objects to appropriate grid cells and creates the PC-list in lines 4 and 5, respectively. Since the PC-list is sorted in the ascending order of its $N_{D-t\sigma}(C)$ column, it guarantees that cells in the sparse regions of grid $\mathcal{G}$ are at the top of the PC-list. Hence, the candidate outlier cells are expected to be at the top of the list. We scan the PC-list and add the candidate outlier cells in $\mathcal{C}_{\text{cell}}$ until the stopping condition on line 8 becomes true. The number of objects in $\mathcal{C}_{\text{cell}}$ may be greater than k, hence their $\#D$-neighbours are computed to find the top-k outliers and their ranking. The object is then added to the $\mathcal{C}_{\text{obj}}$ (set of candidate outlier objects) along with its $\#D(o)$. The objects in $\mathcal{C}_{\text{obj}}$ are sorted in the ascending order of $\#D(o)$ column. As the k-th object's $\#D(o)$ is found, threshold θ is set (refer to line 10 of Algorithm 1). During the calculation of $\#D(o)$, if for some o', $\#D(o')$ becomes greater than θ, then o' can not be among the top-k outliers and is removed from further consideration.

Algorithm 2. Top-k.
Input: $\mathcal{G}DB$, D, l, k.
Output: Top-k distance-based outliers.
1. $N \leftarrow |\mathcal{G}DB|$, $\theta \leftarrow \infty$;
2. $\mathcal{C}_{\text{cell}} \leftarrow \phi$, $\mathcal{C}_{\text{obj}} \leftarrow \phi$; (Candidate outlier cells list and top-k candidate outlier objects list, respectively);
3. Map each $o \in \mathcal{G}DB$ to an appropriate cell C of grid $\mathcal{G}$;
4. Create PC-list PC, using non-empty cells of $\mathcal{G}$;
5. Sort PC w.r.t. $N_{D-t\sigma}(C)$ column;
/*Searching candidate outlier cells*/
6. **for each** C in $|PC|$ **do**
7. /*Stopping condition*/
8. **if** $N_{D-t\sigma}(C) \times Pr(D - t\sigma, D) > \theta$ **then** exit for loop.
9. Compute $LB(C)$ and $UB(C)$;
10. **if** $LB(C) \leqslant \theta$ **then**
11. Add C to $\mathcal{C}_{\text{cell}}$ (keep $\mathcal{C}_{\text{cell}}$ sorted of $UB(C)$ attribute);
12. if $\mathcal{C}_{\text{cell}}$ contains $\geqslant k$ objects **then**
13. Set $\theta = UB(C^k)$, such that C^k contain the k-th object;
14. Remove all C from $\mathcal{C}_{\text{cell}}$, such that $LB(C) > \theta$;
15. **end if**
16. **end if**
17. **end if**
/*Calculating $\#D(o)$ of candidate top-k outliers*/
18. The computation of $\#D(o)$ is similar to that of the naive approach. The only difference is that in this algorithm $\#D(o)$ are computed for the candidate objects in $\mathcal{C}_{\text{cell}}$ only.

5.2 Top-k approx algorithm

In the top-k algorithm, the minimum number of distance function computations required for the evaluation of k $\#D(o)$ is kN. However, the candidate outlier objects which require the evaluation of $\#D(o)$ may be greater that k. When the distance function is expensive to compute (as in our case), computation of even k $\#D(o)$ is very expensive. According to our distance function, the major contribution to the evaluation of $\#D(o)$ is done by the nearer objects. Hence, $\#D(o)$ for each unpruned o can be approximated with a high accuracy by considering objects only within $D + t\sigma$ distance of o according to Lemma 2, rather than considering all the objects in the dataset. It saves a lot of computation time. The rest of the algorithm is the same as that of the accurate top-k algorithm.

Maximum approximation error: For any $o \in \mathcal{G}DB$, the maximum approximation error ($\varepsilon_{\max}$) happens if all the $o' \in \mathcal{G}DB \setminus o$ are at a distance slightly greater than $D + t\sigma$

from o. Hence, $\varepsilon_{\max} = (N-1) \times Pr(D + t\sigma + \beta, D)$, where $\beta \in \mathbf{R}$ is a very small real value to make the distance greater than $D + t\sigma$.

For example, for $t = 9$, $d = 2$ and $N = 10^5$ objects, $\varepsilon_{\max} \approx 10^{-5}$. $\varepsilon_{\max}$ depends mainly on t. In practice $t \geqslant 3$ gives sufficiently accurate $\#D(o)$ for $d = 2$ and 3. For higher d values, we need to increase t value according to Lemma 2.

5.3 Complexity analysis

We will first analyse the complexity of the top-k algorithm for the 2D case. Lines 1 and 2 contain only the initializations of variables. Since there are N objects in the dataset $\mathcal{G}DB$, line 3 takes $\mathrm{O}(N)$ time. Line 4 takes $\mathrm{O}(m)$ time, where $m \ll N$ is the total number of populated cells in the cell-grid. Sorting m cells in the PC-list in line 5 takes $\mathrm{O}(m\log(m))$ time. The main loop of the algorithm in lines 6−17 is executed for all the cells in the PC-list in the worst case. The loop computes the lower and upper cell bounds, each of which takes $\mathrm{O}(m)$ time because the cell bounds computation requires the contributions of all the cells in the PC-list. Keeping the $\mathcal{C}_{\text{cell}}$ sorted in Line 11 takes $\mathrm{O}(m)$ time in the worst case. Lines 13 and 14 within the loop takes at-most $\mathrm{O}(m)$ each. Hence, the overall loop takes $\mathrm{O}(m^2)$ time. Finally, computation of the $\#D$-neighbours in Line 18 takes $\mathrm{O}(nN)$ time, where $n \ll N$ is the number of candidate objects for the top-k outliers. Thus, the worst case time complexity of the top-k outlier detection algorithm in 2D is $\mathrm{O}(nN + m^2)$.

However, in the top-k algorithm, the major cost lies in the evaluation of $\#D$-neighbours of the candidate outlier objects (Line 18). This cost is so high that it hides the cost of the rest of the algorithm. This is due to the expensive distance calculation between uncertain objects. Therefore, we give the time complexity of the proposed algorithms in terms of the number of distance function evaluations. Hence, the complexity of the top-k algorithm in 2D is $\mathrm{O}(nN)$. Although the number of distance function evaluations required for the processing of candidate outlier objects in the top-k approx algorithm is far lower than its exact counter part, its worst case complexity is still $\mathrm{O}(nN)$ in a 2D case.

The complexity of the proposed algorithms does not change with the increase in dimensions d, as long as only the number of distance function evaluations are considered for the computation of algorithms′ complexity. Although with the increase in d, the number of grid cells increases exponentially, yet the cost of the evaluation of $\#D$-neighbours for the candidate outlier objects remains dominant and hence the complexity remains the same, i.e., $\mathrm{O}(nN)$ for a higher dimensional case.

From the above analysis, it is evident that the computational complexity of the proposed algorithms is lower than the naive algorithm, which is $\mathrm{O}(N^2)$ in terms of the distance function. Hence, the execution times of the proposed algorithms are far lower than the naive algorithm. However, with the increase in d, the distance computation between uncertain objects becomes very expensive and it becomes impractical to detect outliers from very high dimensional data.

5.4 Discussion: determination of values for D and l

Let us begin by stating that there is no universally correct value for parameters D or l. Parameter D has an affect on the $\#D$-neighbors of an object, as our distance function $Pr(o_i, o_j, D)$ is dependent on D. A larger D value results in large $Pr(o_i, o_j, D)$ values and therefore large $\#D$-neighbors and vice versa. However, a very small or very large D value is not recommended as it may result in very small or very large $\#D$-neighbors, respectively for all the dataset objects and may hide the difference between strong and weak outliers. Therefore, we recommend a moderate value of D, not too large to cover the entire dataset objects and not too small to cover only the object itself. Hence, an appropriate D value may be decided by considering the dataset distribution by the end user.

Parameter l (cell length) has an affect on the performance of the algorithms rather than the accuracy. Smaller l values are good for cell pruning as they result in tighter bounds. However, very small l may increase the number of cells in the grid exponentially and the time required to construct the PC-list and the bounds computation. This phenomenon becomes severe with the increase in dimension d. On the other hand, larger l values result in looser bounds and hence reduce the pruning capability of the algorithms. Therefore, small l values are recommended for lower dimensions and relatively larger values are recommended for higher dimensions.

6 Outlier detection using the bounded Gaussian uncertainty

Approximating the Gaussian uncertainty by the bounded Gaussian uncertainty enables an approximate but more efficient outlier detection. According to this paper′s assumption, attributes of uncertain objects follow the Gaussian distribution. Therefore, according to the 3-sigma rule, there are a 95.45% chance that uncertain objects′ attribute values lie within 2 standard deviations of the observed values and a 99.73% chance that the values lie within 3 standard deviations of the observed values[39]. Hence, the conventional Gaussian distribution can be normalized within certain boundaries to increase the efficiency of the top-k outlier detection at a small cost of accuracy.

Given a two dimensional conventional Gaussian function $g_{\overrightarrow{\mathcal{A}}}(x_1, x_2)$ with mean $\overrightarrow{\mu} = (\mu_1, \mu_2)$ and co-variance matrix $\Sigma = \mathrm{diag}(\sigma^2, \sigma^2)$, the bounded Gaussian distribution $f_{\overrightarrow{\mathcal{A}}}(x_1, x_2)$ can be defined following the practice of Tao et al.[40] as follows:

$$f_{\overrightarrow{\mathcal{A}}}(x_1, x_2) = \begin{cases} \dfrac{g_{\overrightarrow{\mathcal{A}}}(x_1, x_2)}{\int_{(x_1, x_2) \in o.ur} g_{\overrightarrow{\mathcal{A}}}(x_1, x_2) \mathrm{d}x_1 \mathrm{d}x_2}, (x_1, x_2) \in o.ur \\ 0, \quad \text{otherwise} \end{cases} \tag{12}$$

where $o.ur$ denotes the uncertainty region of the bounded Gaussian distribution. This paper assumes that the uncertainty region is a sphere with centre (μ_1, μ_2) and radius $r = t\sigma$ (t is discussed in Lemma 2).

By bounding the Gaussian uncertainty, a cell can be pruned by simply counting the number of objects in its

neighbouring cells. Moreover, the major cost of outlier detection, that is, the processing of un-pruned objects also reduces significantly. This is because, with the bounded Gaussian uncertainty, the outlier detection algorithm needs to consider limited number of objects for the computation of an object's $\#D$-neighbors rather than all the objects in the dataset. Interested readers may refer to our previous work[33], for the details of the bounded Gaussian uncertainty.

6.1 Grid ($\mathcal{G}$) and PC-list structure for the bounded Gaussian

In order to identify distance-based outliers using the PC-list, each object in $\mathcal{G}DB$ is mapped into a d-dimensional space that is partitioned into cells of length l (l is discussed in Section 5.4). Let $C_{\psi_1,\cdots,\psi_d}$ be a cell in Grid $\mathcal{G}$; then cells in region $R_{D-2r}(C_{\psi_1,\cdots,\psi_d})$ are those which completely lie within $D-2r$ distance of $C_{\psi_1,\cdots,\psi_d}$, including the $C_{\psi_1,\cdots,\psi_d}$ itself. Let $n_{D-2r} = \lfloor \frac{D-2r}{l} \rfloor - 1$; then the region $R_{D-2r}(C_{\psi_1,\cdots,\psi_d})$ is derived as

$$\begin{aligned} &R_{D-2r}(C_{\psi_1,\cdots,\psi_d}) = \{C_{x_1,\cdots,x_d} | x_1 = \psi_1 \pm n_{D-2r}, \cdots \\ &x_d = \psi_d \pm n_{D-2r}, \quad \sqrt{\sum_{i=1}^{d}((x_i+1)l)^2} < D - 2r \\ &C_{x_1,\cdots,x_d} \neq C_{\psi_1,\cdots,\psi_d}. \end{aligned} \tag{13}$$

The number of cells in the region of $R_{D-2r}(C_{\psi_1,\ldots,\psi_d})$ vary depending upon n_{D-2r}. Note that the $R_{D-2r}(C_{\psi_1,\ldots,\psi_d})$ satisfies the following property.

Property 1. If $C_{x_1,\cdots,x_d} \in R_{D-2r}(C_{\psi_1,\cdots,\psi_d})$, then the objects $o_i \in C_{\psi_1,\cdots,\psi_d}$ and $o_j \in C_{x_1,\cdots,x_d}$ are at most $D-2r$ distance apart.

From Property 1, the $o_i \in C_{\psi_1,\cdots,\psi_d}$ and the $o_j \in R_{D-2r}(C_{\psi_1,\cdots,\psi_d})$ are guaranteed to be D-neighbours mutually. Hence, the $Pr(o_i, o_j, D)$ is always equal to 1. Cells in region $R_{D+2r}(C_{\psi_1,\cdots,\psi_d})$ are those which fall within $D+2r$ distance of the $C_{\psi_1,\cdots,\psi_d}$. Let $n_{D+2r} = \lceil \frac{D+2r}{l} \rceil$; then the region $R_{D+2r}(C_{\psi_1,\cdots,\psi_d})$ is derived as

$$\begin{aligned} &R_{D+2r}(C_{\psi_1,\cdots,\psi_d}) = \{C_{x_1,\cdots,x_d} | x_1 = \psi_1 \pm n_{D+2r}, \cdots \\ &x_d = \psi_d \pm n_{D+2r}, \quad \sqrt{\sum_{i=1}^{d}((x_i-1)l)^2} < D + 2r \\ &C_{x_1,\cdots,x_d} \notin R_{D-2r}(C_{\psi_1,\cdots,\psi_d}),\ C_{x_1,\cdots,x_d} \neq C_{\psi_1,\cdots,\psi_d}. \end{aligned} \tag{14}$$

Note that $R_{D-2r}(C_{\psi_1,\cdots,\psi_d})$ and $R_{D+2r}(C_{\psi_1,\cdots,\psi_d})$ satisfy the following property.

Property 2. If $C_{x_1,\cdots,x_d}$ is neither in $R_{D-2r}(C_{\psi_1,\cdots,\psi_d})$ nor in $R_{D+2r}(C_{\psi_1,\cdots,\psi_d})$, and $C_{x_1,\cdots,x_d} \neq C_{\psi_1,\cdots,\psi_d}$, then objects $o_i \in C_{\psi_1,\cdots,\psi_d}$ and $o_j \in C_{x_1,\cdots,x_d}$ are greater than $D+2r$ distance apart.

From Property 2, it can be guaranteed that $o_i \in C_{\psi_1,\cdots,\psi_d}$ and $o_j \in C_{x_1,\cdots,x_d}$ are greater than $D+2r$ distance apart. Hence, the $Pr(o_i, o_j, D)$ is always equal to 0. In the following, C is used to denote $C_{\psi_1,\cdots,\psi_d}$ when there is no confusion.

The PC-list structure for the bounded Gaussian is similar to that of the conventional Gaussian (refer to Section 4.2) except the column $N_{D-t\sigma}(C)$. Instead of $N(C)$ and $N_{D-t\sigma}(C)$, the PC-list contains $N(C)$ and $N_{D-2r}(C)$ for each non-empty cell of $\mathcal{G}$. The tuples in the PC-list are sorted in an ascending order of $N_{D-2r}(C)$ column.

6.2 Cell bounds for the bounded Gaussian

In order to identify cells $C \in PC$ containing only inliers or candidate top-k outliers, their bounds on the $\#D$-neighbours are used. A cell C can be pruned as an inlier cell if the minimum $\#D$-neighbours for any object in C is greater than threshold θ (θ is discussed in Section 4.4). Similarly, a cell can be identified as containing top-k outliers if the maximum $\#D$-neighbours for any object in C is less than θ. In case of the bounded Gaussian distribution, $Pr(o_i, o_j, D) = 0$ if the means of o_i and o_j are greater than $D+2r$ distance. Hence, only cells within regions R_{D-2r} and R_{D+2r} of a $C \in PC$ need to be considered for the computation of its lower and upper bounds, respectively.

Thus for a $C \in PC$, its lower bound $LB(C)$ and upper bound $UB(C)$ are defined as

$$LB(C) = \sum_{C' \in \{PC \cap R_{D-2r}(C)\}} N(C') \tag{15}$$

$$UB(C) = \sum_{C' \in \{PC \cap R_{D+2r}(C)\}} N(C'). \tag{16}$$

6.3 Candidate outlier cell detection and stopping condition for the bounded Gaussian

For the bounded Gaussian case, the procedure of candidate outlier cell detection is similar to that discussed in Section 4.4. However, the stopping condition is slightly different. During the scanning of PC-list, if $C' \in PC$ is found such that $N_{D-2r}(C') > \theta$, which is a lower bound on $\#D$-neighbors of C', C' cannot contain the top-k outliers and can be pruned. Since the PC-list is sorted of $N_{D-2r}(C)$, any cell after C' must have $N_{D-2r}(C) \geqslant N_{D-2r}(C')$. Hence, the PC-list scanning can be stopped safely at this position.

6.4 Top-k BG algorithm

The major part of the top-k BG algorithm is the same as that of the top-k algorithm (Algorithm 2). The main difference lies in the construction of the PC-list and the computation of bounds as discussed in Sections 6.1 and 6.2, respectively. Moreover, evaluation of the candidate outlier objects now only requires objects within $D+2r$ distance of the target object rather than all the objects in the dataset, bringing down the overall cost of execution.

7 Experiments

Extensive experiments are conducted on synthetic and real datasets to evaluate the accuracy and efficiency of the proposed algorithms. All algorithms were implemented in C++, GNU compiler. All experiments were performed on a system with an Intel Core 2 Duo E8600 3.33 GHz CPU and 2 GB main memory running Ubuntu 12.04 OS. All programs run in main memory and no I/O cost is considered.

7.1 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[41]. This method generates pair of independent, standard, normally distributed (zero mean, unit variance) random numbers, given a source of uniformly distributed random numbers. High dimensional (3D, 4D and 5D) uni-modal Gaussian datasets are also generated for the evaluation of our proposed approaches on high dimensional data. Unless specified, 2-dimensional and high 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[42] and SDSS is obtained from Sloan Digital Sky Survey[43]. 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[43]. 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[44].

All the datasets are normalized to have a domain of [0 1000] 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: $D = 100$, $\sigma = 10$, $l = 10$, $t = 3$, $r = t\sigma$ and $k = 10$. In the following figures, the Knorr et al.[10] algorithm is denoted by Knorr and the proposed top-k, the top-k approximate and the top-k bounded Gaussian algorithms are denoted by top-k, top-k approx and top-k BG, respectively.

7.2 Accuracy

Firstly, experiments are performed to evaluate the accuracy of the proposed algorithms. Since there are no known algorithms for the top-k distance-based outlier detection on uncertain data, the deterministic algorithm for distance-based outlier detection given by Knorr et al.[10] is used as a baseline. Slight changes are made in the Knorr′s algorithm to obtain the top-k outliers from it. 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 accuracy 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 50 (with a step of 10) to generate perturbed datasets of five different levels. Experiments show that the proposed algorithms are superior to 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 is defined as the ability of the algorithm to present only true outliers. The recall is defined 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: $D = 70$, $\sigma = 10$, $\sigma_p = 30$, $l = 10$, $t = 3$, $r = t\sigma$ and $k = 50$.

Firstly, the precision-recall trade-off curves are presented for different datasets. In all the graphs in Fig. 3 both the precisions and recalls of the proposed algorithms are higher than those of the baseline approach. Moreover, the precision-recall curves of the top-k and the top-k approx are exactly the same. This is due to the fact that both the algorithms returned the same outliers. Although there was a slight difference in the $\#D$-neighbours of the outliers returned by both the proposed algorithms, but this difference was not big enough to change the top-k outlier objects or their ranking. The precision-recall of the top-k BG algorithm is also better than that of the baseline algorithm and in most datasets is equal to the top-k algorithm.

In Fig. 3 (a), the precision-recall curves are almost same for all the algorithms, however, in Figs. 3 (b), 3 (c) and 3 (d) the precision-recall curves of the proposed algorithms are comparatively higher than that of the baseline approach. Specially, the low recall in Figs. 3 (c) and 3 (d) shows the presence of a large number of false positive outliers in the outliers obtained from the baseline algorithm.

The accuracy of the proposed algorithms is also evaluated with the increasing level of uncertainty. From Fig. 4, it is clear that the precision falls with the increasing uncertainty level. Moreover, the precision of the proposed algorithms is always higher than that of the baseline algorithm in Fig. 4, which means that fewer false-positive outliers were returned by the proposed algorithms than by the baseline algorithm. Similar results are illustrated for recall in Fig. 5. In all four plots of Fig. 5, the recall is somewhat consistent with increasing uncertainty level for the proposed algorithms. This proves that the proposed algorithms are better than the baseline algorithm in retrieving only true outliers, even from the noisy data.

7.3 Efficiency

In this subsection, experiments are conducted to evaluate the efficiency of the proposed top-k outlier detection algorithms presented in Sections 5 and 6. Fig. 6 (a) compares the execution times of the naive and the proposed algorithms on UG dataset. Please note the use of logarithmic scale in all the efficiency graphs to keep the graph lines visible. The proposed algorithms are several times faster than its naive counterpart due to their strong pruning capability as can be observed from Fig. 6 (b). The stopping condition discussed in Sections 4.4 and 6.3 helps identify candidate outlier cells very quickly. Fig. 6 (c) shows the percentage of cells considered in the PC-list to identify candidate outlier cells. The percentage is comparatively higher for trimodal Gaussian dataset because the dataset is relatively sparse and hence results in larger number of candidate outlier cells.

Moreover, the top-k approx and the top-k BG algorithms are thousands of times faster than the top-k algorithm. This is due to the fact that these algorithms, in contrast to the top-k algorithm, do not consider all the dataset objects for the computation of $\#D$-neighbors of the candidate objects.

From theoretical analysis in Section 5.2 and experiments we found that the top-k approx algorithm gives an accuracy of up to several decimal digits in the evaluation of $\#D(o)$ and hence the outliers obtained from the top-k and the top-k approx are the same.

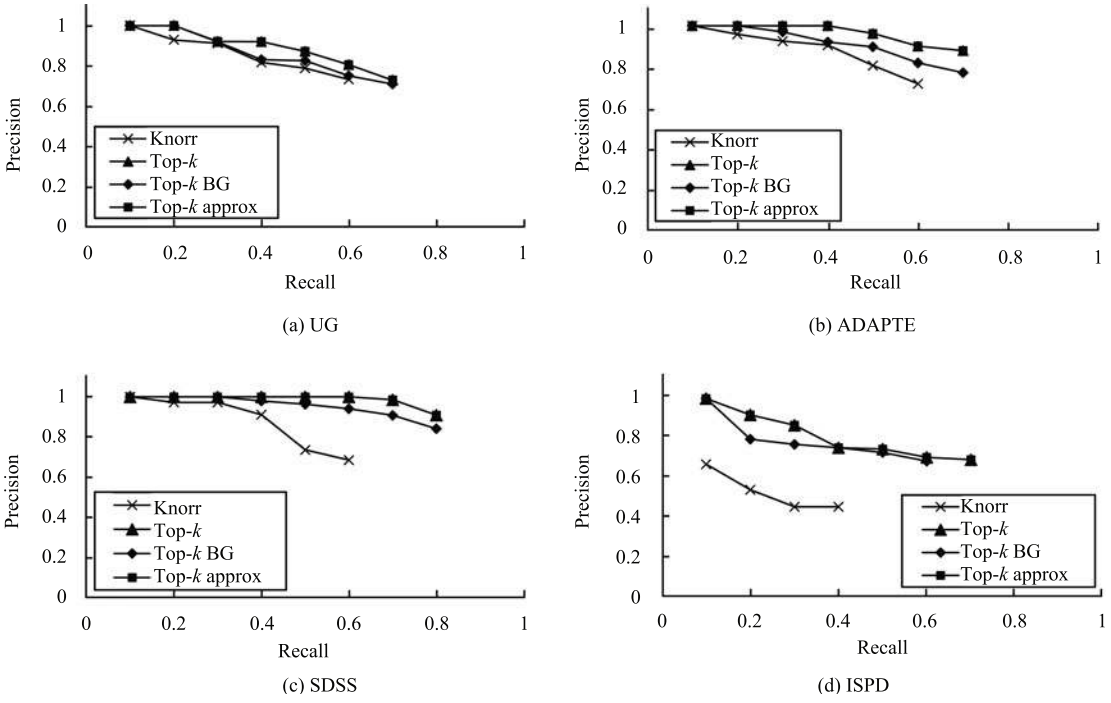


Fig. 3 Precision-recall trade-off curves ($D = 70,\ \sigma = 10,\ \sigma_p = 30,\ l = 10,\ t = 3$ and $k = 50$)

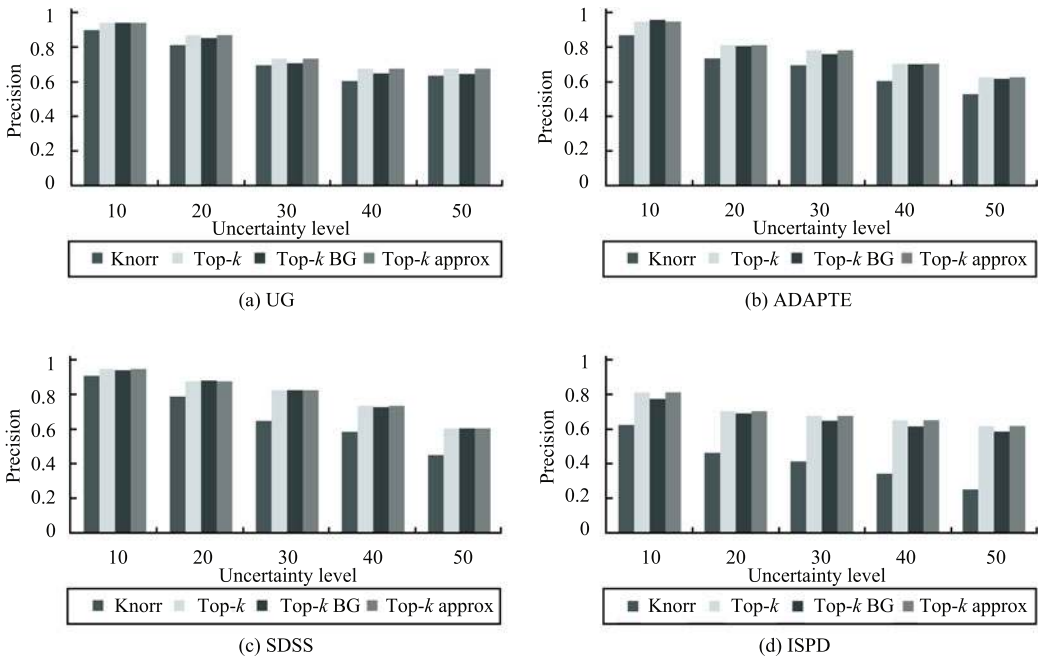


Fig. 4 Precision with increasing σ_p ($D = 70,\ \sigma = 10,\ \sigma_p = 30,\ l = 10,\ t = 3$ and $k = 50$)

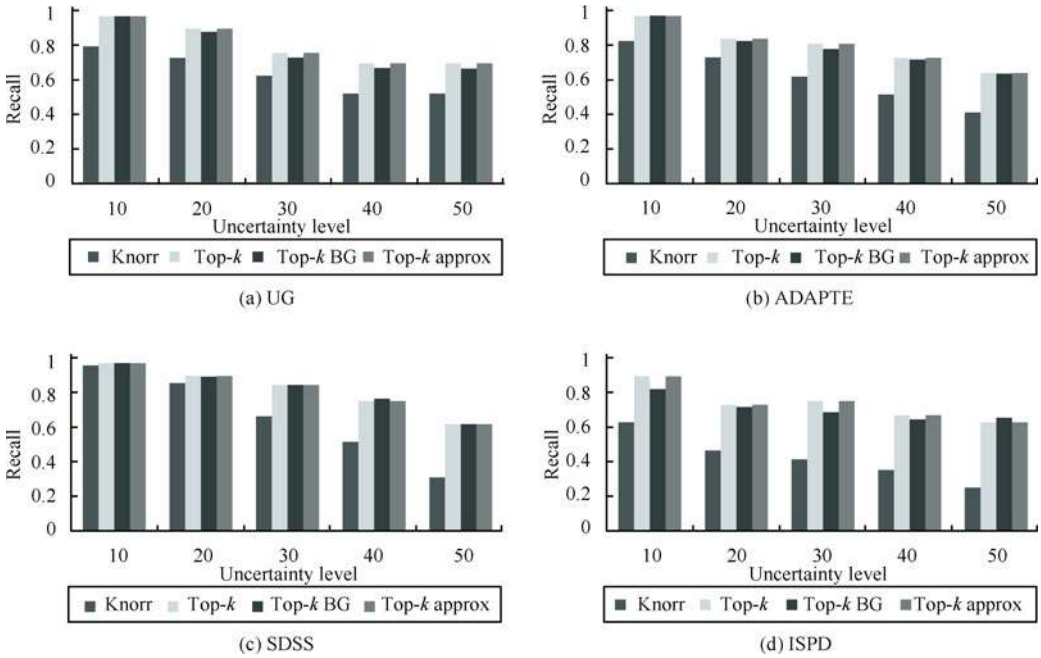


Fig. 5 Precision with increasing σ_p ($D = 70,\ \sigma = 10,\ \sigma_p = 30,\ l = 10,\ t = 3$ and $k = 50$)

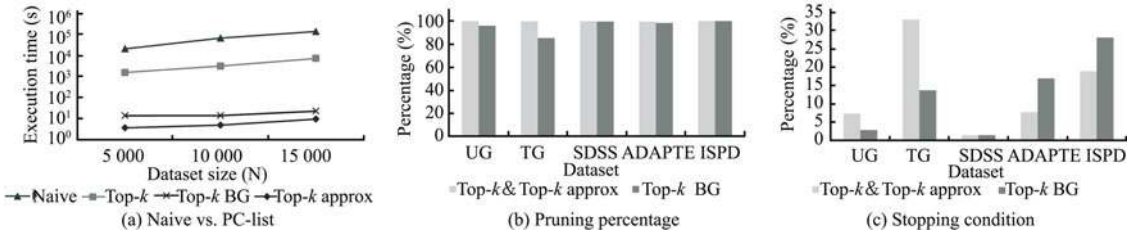


Fig. 6 Effectiveness of the PC-list based approach

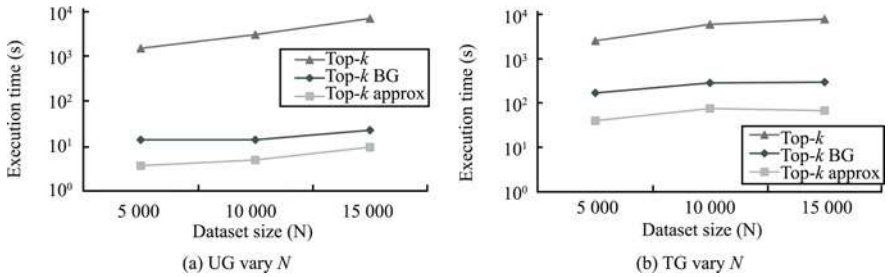


Fig. 7 Varying dataset size N ($D = 100,\ \sigma = 10,\ l = 10,\ t = 3$ and $k = 10$)

In addition, the execution time of the top-k approx algorithm in all the experiments is several times lower than its exact counterpart. On the other hand, the accuracy of the top-k BG algorithm is not as high as that of the top-k approx algorithm, however its efficiency is higher than that of the top-k approx, algorithm for the higher dimensional data (refer to Fig. 12) and it does not require any pre-computation.

Fig. 7 shows the affect of varying the number of objects in the synthetic datasets UG and TG. With the increase in the number of dataset objects, execution times of all the algorithms also increase. However, in Fig. 7 (b), the execution times do not appear to increase from $N = 10\,000$ to $N = 15\,000$. As discussed in the previous sections, the major cost of the proposed algorithms lie in the processing of the un-pruned objects (candidate outlier objects). In Fig. 7 (b), the number of un-pruned objects for $N = 15\,000$ is less than that of $N = 10\,000$, which is the cause of stability in the graph lines from $N = 10\,000$ to $N = 15\,000$.

Graphs in Figs. 8–11 show the affect of varying different parameters on the execution times. Firstly, consider the variation of parameter l in Fig. 8. The numbers above and below the graph lines show the number of candidate objects requiring exact $\#D$-neighbors computation. As cell-length varies, the number of objects requiring $\#D$-neighbors computation also varies. As discussed in previous sections, the major cost of our algorithms lie in the processing of candidate objects, because they required computation of $\#D$-

neighbors. Hence, as the cell length increases, the execution time of the algorithms also increases due to the increase in number of candidate objects. Moreover, it is obvious from the graphs in Fig. 8 that smaller cell lengths require lower execution times. However, very small cell length increases the number of cells exponentially and therefore the execution time of the algorithm. This phenomenon is discussed in Section 5.4.

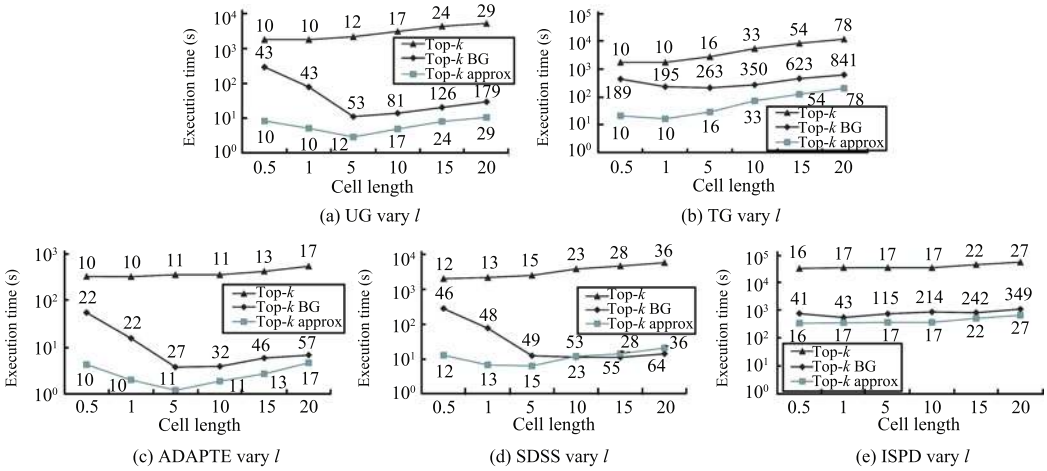


Fig. 8 Varying parameter l ($D = 100,\ \sigma = 10,\ t = 3$ and $k = 10$)

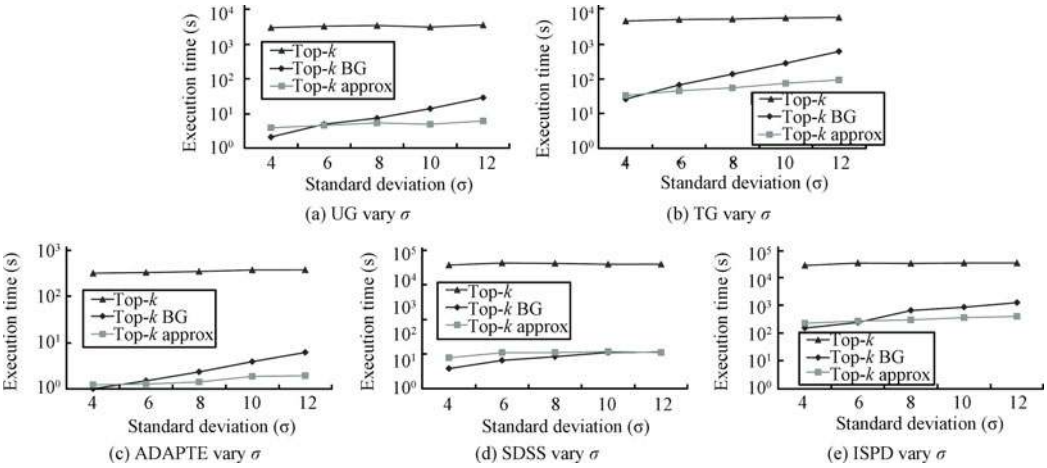


Fig. 9 Varying parameter σ ($D = 100,\ l = 10,\ t = 3$ and $k = 10$)

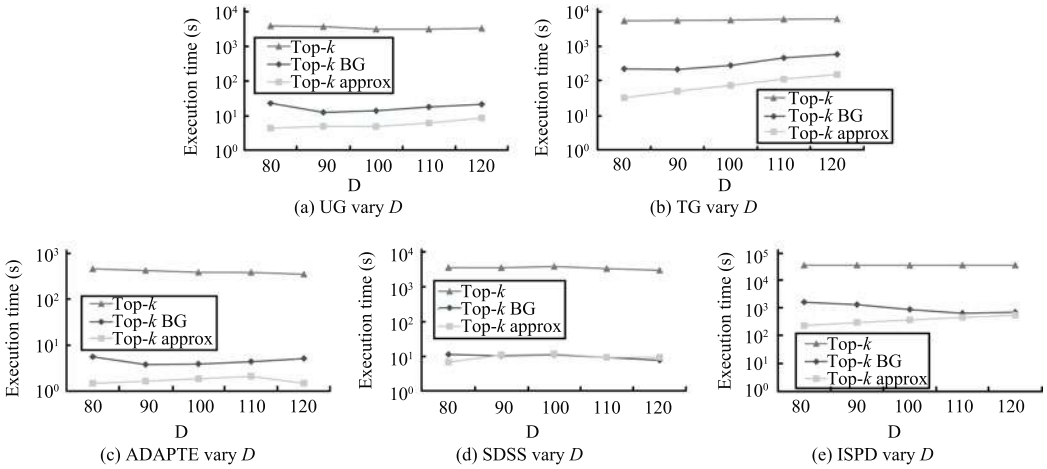


Fig. 10 Varying parameter D ($\sigma = 10,\ l = 10,\ t = 3$ and $k = 10$)

Next we perform experiments by varying the parameter σ. As σ increases, the uncertainty of 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, which results in higher execution times as can be observed from graphs in Fig. 9. Moreover, it can be observed from Fig. 9 that for smaller a σ, the computation cost is the lowest for the top-k BG algorithm. Please recall that the bounded Gaussian uncertainty is bounded by radius $r = t\sigma$. Hence, smaller σ results in a smaller r and it helps in early pruning of objects, bringing down the overall cost of the algorithm.

Graphs in Fig. 10 show the affect of varying parameter D. For each un-pruned o from the PC-list-based pruning, increase in D results in an increase in the $\#D$-neighbours, which needs to be considered for the approximation of $\#D(o)$. Therefore, it increases the execution time of the top-k approx algorithm. Similarly, the execution time of the top-k BG algorithm increases with the increase in D. Since increase in D results in an increase in region R_{D+2r} and the number of objects need to be considered for the computation of an object′s $\#D$-neighbors. However the execution times of the top-k algorithm decrease with the increase in the parameter D. This is due to the fact that for larger D, $Pr(o_i, o_j, D)$ is higher. Hence an un-pruned object is easily pruned if it is an inlier, reducing the overall cost of the algorithm.

From graphs in Fig. 11, increase in k results in an increase in execution times of the algorithms, which is quite obvious behaviour of the algorithms.

Finally, experiments are performed by varying the number of dimensions. Experiments in Fig. 12 are performed on the synthetic dataset UG with $N = 1\,000$. Computation cost of the top-k and the top-k approx algorithms increases with the increase in dimensions, however the computation cost decreases for the top-k BG algorithm with the increase in dimensions. The reason for this decrease in execution time is the sparsity of data objects in higher dimensions. As a result, the top-k BG algorithm can quickly identify the cells in sparse regions and can obtain top-k outliers. Moreover, we found that there were no un-pruned objects for the top-k BG algorithm for dimensions 3 to 5. Fig. 12 (b) can help further in understanding the graph of Fig. 12 (a). For the top-k and the top-k approx algorithms, pruning percentage decreases with the increase in d, causing the execution times to increase with d. On the other hand, pruning percentage increases with the increase in d for the top-k BG algorithm, causing the execution times to decrease dramatically. Fig. 12 (c) shows the percentage of cells considered before the execution of stopping condition. From the graph it is very obvious that it decreases with d for the top-k and the top-k approx algorithms and increases for the top-k BG algorithm.

8 Conclusion

In this work, an exact (top-k) and two approximate (top-k approx and top-k BG) algorithms on top-k distance-based outlier detection from uncertain datasets of the Gaussian distribution are proposed. All the algorithms make use of a cell grid and a PC-list (populated-cells list) to quickly identify the candidate outlier objects. The only difference between the top-k and the top-k approx algorithms is the

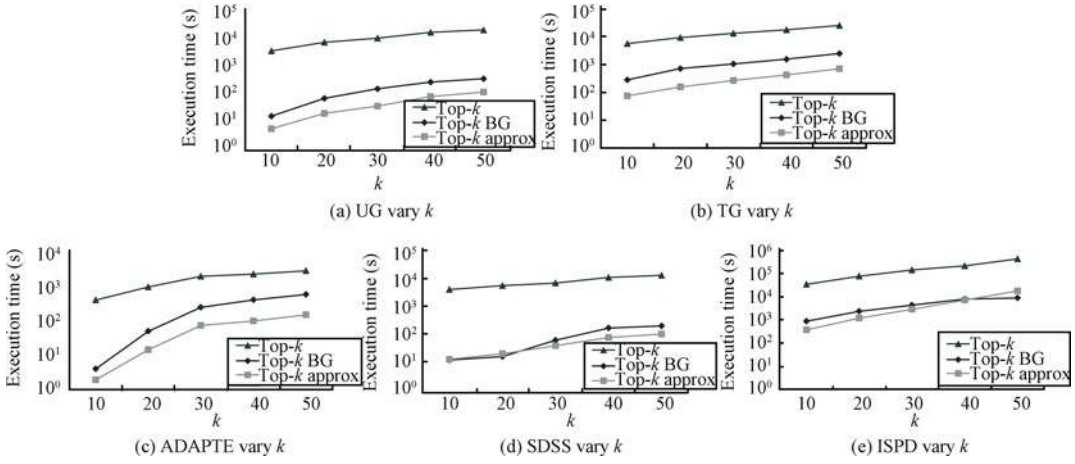


Fig. 11 Varying parameter k ($D = 100,\ \sigma = 10,\ l = 10$ and $t = 3$)

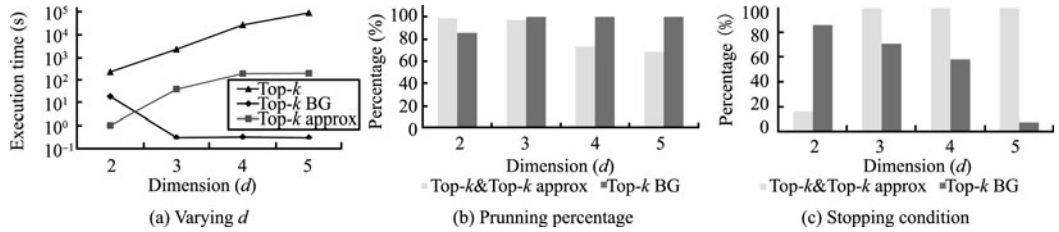


Fig. 12 Varying dimensions d ($N = 1000,\ D = 100,\ \sigma = 10,\ l = 10,\ t = 3$ and $k = 10$)

computation of $\#D$-neighbours. The exact top-k algorithm computes $\#D$-neighbours of the top-k candidate objects by considering all the objects in the dataset, however, the top-k approx algorithm considers only nearer objects for its $\#D$-neighbour computation. The top-k BG algorithm makes use of the bounded Gaussian uncertainty to reduce the computation cost of outlier detection. An extensive empirical study on real and synthetic datasets is also presented to prove the accuracy, efficiency and scalability of the proposed algorithms.

Appendix

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

$$f_{\overrightarrow{\mathcal{A}}}(x_1, \cdots, x_k) = \frac{1}{\sqrt{(2\pi)^k \det\Sigma}} \times \mathrm{e}^{\left\{ -\frac{(\overrightarrow{\mathcal{A}} - \overrightarrow{\mu})^{\mathrm{T}} \Sigma^{-1} (\overrightarrow{\mathcal{A}} - \overrightarrow{\mu})}{2} \right\}}. \quad \text{(A1)}$$

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

$$f_{\overrightarrow{\mathcal{A}}}(x_1, \cdots, x_k) = \prod_{1 \leqslant i \leqslant k} \frac{1}{\sqrt{2\pi\sigma_i^2}} \mathrm{e}^{\left\{ -\frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right\}}. \quad \text{(A2)}$$

Let o_i and o_j be two k-dimensional uncertain objects with attributes $\overrightarrow{\mathcal{A}_i} = [x_{i,1}, \cdots, x_{i,k}]^{\mathrm{T}}$ and $\overrightarrow{\mathcal{A}_j} = [x_{j,1}, \cdots, x_{j,k}]^{\mathrm{T}}$, means $\overrightarrow{\mu_i} = [\mu_{i,1}, \cdots, \mu_{i,k}]^{\mathrm{T}}$ and $\overrightarrow{\mu_j} = [\mu_{j,1}, \cdots, \mu_{j,k}]^{\mathrm{T}}$ and diagonal covariance matrices $\Sigma_i = \mathrm{diag}(\sigma_{i,1}^2, \cdots, \sigma_{i,k}^2)$ and $\Sigma_j = \mathrm{diag}(\sigma_{j,1}^2, \cdots, \sigma_{j,k}^2)$, respectively. Assuming that $\overrightarrow{\mathcal{A}_i}$ and $\overrightarrow{\mathcal{A}_j}$ are independent random vectors, then $\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j} = \mathcal{N}(\overrightarrow{\mu_i} - \overrightarrow{\mu_j}, \Sigma_i + \Sigma_j)$[31].

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

$$f_{\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j}}(x_1, \cdots, x_k) = \prod_{1 \leqslant m \leqslant k} \frac{1}{\sqrt{2\pi(\sigma_{i,m}^2 + \sigma_{j,m}^2)}} \times \mathrm{e}^{\left\{ -\frac{(x_m - (\mu_{i,m} - \mu_{j,m}))^2}{2(\sigma_{i,m}^2 + \sigma_{j,m}^2)} \right\}}. \quad \text{(A3)}$$

The normal difference distribution of 2-dimensional uncertain objects o_i and o_j is given by

$$f_{\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j}}(x_1, x_2) = \frac{1}{2\pi\sqrt{(\sigma_{i,1}^2 + \sigma_{j,1}^2)(\sigma_{i,2}^2 + \sigma_{j,2}^2))}} \times \mathrm{e}^{\left\{ -\left(\frac{(x_1 - \alpha_1)^2}{2(\sigma_{i,1}^2 + \sigma_{j,1}^2)} + \frac{(x_2 - \alpha_2)^2}{2(\sigma_{i,2}^2 + \sigma_{j,2}^2)} \right) \right\}} \quad \text{(A4)}$$

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

$$Pr(o_i, o_j, D) = \frac{1}{2\pi\sqrt{(\sigma_{i,1}^2 + \sigma_{j,1}^2)(\sigma_{i,2}^2 + \sigma_{j,2}^2)}} \times \int_0^D \int_0^{2\pi} \mathrm{e}^{\left\{ -\left(\frac{(r\cos\theta - \alpha_1)^2}{2(\sigma_{i,1}^2 + \sigma_{j,1}^2)} + \frac{(r\sin\theta - \alpha_2)^2}{2(\sigma_{i,2}^2 + \sigma_{j,2}^2)} \right) \right\}} r \, \mathrm{d}\theta \, \mathrm{d}r. \quad \text{(A5)}$$

References

[1] A. Elías, A. Ochoa-Zezzatti, A. Padilla, J. Ponce. Outlier analysis for plastic card fraud detection a hybridized and multi-objective approach. *Hybrid Artificial Intelligent Systems*, Berlin, Heidelberg: Springer, pp. 1–9, 2011.

[2] M. V. Mahoney, P. K. Chan. Learning rules for anomaly detection of hostile network traffic. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, IEEE, Melbourne, FL, USA, pp. 601–604, 2003.

[3] G. Manson, G. Pierce, K. Worden. On the long-term stability of normal condition for damage detection in a composite panel. *Key Engineering Materials*, vol. 204–205, pp. 359–370, 2001.

[4] H. Garces, D. Sbarbaro. Outliers detection in environmental monitoring databases. *Engineering Applications of Artificial Intelligence*, vol. 24, no. 2, pp. 341–349, 2011.

[5] N. Alaydie, F. Fotouhi, C. K. Reddy, H. Soltanian-Zadeh. Noise and outlier filtering in heterogeneous medical data sources. In *Proceedings of Workshops on Database and Expert Systems Applications*, IEEE, Bilbao, Spain, pp. 115–119, 2010.

[6] D. M. Hawkins. *Identification of Outliers*, London: Chapman and Hall, 1980.

[7] V. Barnett, T. Lewis. *Outliers in Statistical Data*, New York: Wiley, 1994.

[8] O. Z. Maimon, L. Rokach. *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, Norwell: Kluwer Academic, 2005.

[9] C. C. Aggarwal. *Outlier Analysis*, New York: Springer-Verlag, 2013.

[10] E. M. Knorr, R. T. Ng, V. Tucakov. Distance-based outliers: Algorithms and applications. *The VLDB Journal*, vol. 8, no. 3–4, pp. 237–253, 2000.

[11] E. M. Knorr, R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of 24th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 392–403, 1998.

[12] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *Proceedings of the 19th International Conference on Data Engineering*, IEEE, Bangalore, India, pp. 315–326, 2003.

[13] V. Kumar. Parallel and distributed computing for cybersecurity. *IEEE Distributed Systems Online*, vol. 6, no. 10, pp. 1–9, 2005.

[14] G. H. Orair, C. H. C. Teixeira, W. Meira, Y. Wang, S. Parthasarathy. Distance-based outlier detection: Consolidation and renewed bearing. In *Proceedings of the VLDB Endowment*, vol. 3, no. 1–2, pp. 1469–1480, 2010.

[15] B. Wang, G. Xiao, H. Yu, X. C. Yang. Distance-based outlier detection on uncertain data. In *Proceedings of the 9th IEEE International Conference on Computer and Information Technology*, IEEE, Xiamen, China, pp. 293–298, 2009.

[16] C. Zhu, H. Kitagawa, S. Papadimitriou, C. Faloutsos. Outlier detection by example. *Journal of Intelligent Information Systems*, vol. 36, no. 2, pp. 217–247, 2011.

[17] A. B. Sharma, L. Golubchik, R. Govindan. Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks*, vol. 6, no. 3, pp. 1–39, 2010.

[18] I. Helm, L. Jalukse, I. Leito. Measurement uncertainty estimation in amperometric sensors: A tutorial review. *Sensors*, vol. 10, no. 5, pp. 4430–4455, 2010.

[19] Y. Diao, B. D. Li, A. N. Liu, L. P. Peng, C. Sutton, T. Tran, M. Zink. Capturing data uncertainty in high-volume stream processing. In *Proceedings of the 4th Biennial Conference on Innovative Data Systems Research*, Asilomar, California, USA, 2009.

[20] A. A. Omer, J. P. Thomas, L. Zhu. Mutual authentication protocols for RFID systems. *International Journal of Automation and Computing*, vol. 5, no. 4, pp. 348–365, 2008.

[21] J. Nievergelt, H. Hinterberger, K. C. Sevick. The Grid file: An adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems*, vol. 9, no. 1, 38–71, 1984.

[22] Stevens Water Monitoring Systems, Inc., [Online], Available: http://www.stevenswater.com/, March 7, 2013.

[23] Vaisala Corporation, [Online], Available: http://www.vaisala.com/, March 7, 2013.

[24] Xylem Corporation, [Online], Available: http://www.globalw.com/, March7, 2013.

[25] S. Ramaswamy, R. Rastogi, K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of 2000 ACM SIGMOD International Conference on Management of Data*, ACM, New York, NY, USA, pp. 427–438, 2000.

[26] F. Angiulli, C. Pizzuti. Fast outlier detection in high dimensional spaces. In *Proceedings of the 6th European Conference, PKDD 2002*, Springer, Helsinki, Finland, pp. 15–26, 2002.

[27] F. Angiulli, F. Fassetti. Detecting distance-based outliers in streams of data. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, ACM, New York, NY, USA, pp. 811–820, 2007.

[28] M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, Y. Manolopoulos. Continuous monitoring of distance-based outliers over data streams. In *Proceedings of the 27th IEEE International Conference on Data Engineering*, IEEE, Hannover, pp. 135–146, 2011.

[29] K. Ishida, H. Kitagawa. Detecting current outliers: Continuous outlier detection over time-series data streams. In *Proceedings of the 19th International Conference Database and Expert Systems Applications*, Springer, Berlin, Heidelberg, pp. 255–268, 2008.

[30] C. C. Aggarwal, P. S. Yu. Outlier detection with uncertain data. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 483–493, 2008.

[31] E. W. Weisstein. Normal Difference Distribution. From MathWorldA Wolfram Web Resource, [Online], Available: http://mathworld.wolfram.com/, Jan 27, 2012.

[32] S. A. Shaikh, H. Kitagawa. Distance-based outlier detection on uncertain data of Gaussian distribution. In *Proceedings of the 14th Asia-Pacific International Conference on Web Technologies and Applications*, Springer-Verlag, Berlin, Heidelberg, pp. 109–121, 2012.

[33] S. A. Shaikh, H. Kitagawa. Efficient distance-based outlier detection on uncertain datasets of Gaussian distribution. *World Wide Web*, 2013. (Online first).

[34] S. A. Shaikh, H. Kitagawa. Fast top-k distance-based outlier detection on uncertain data. In *Proceedings of the 14th International Conference on Web-age Information Management*, Springer, Berlin, Heidelberg, pp. 301–313, 2013.

[35] M. M. Breunig, H. P. Kriegel, R. T. Ng, J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ACM, New York, NY, USA, pp. 93–104, 2000.

[36] Z. Y. He, X. F. Xu, S. C. Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1641–1650, 2003.

[37] B. Jiang, J. Pei. Outlier detection on uncertain data: Objects, instances, and inferences. In *Proceedings of the 27th IEEE International Conference on Data Engineering*, IEEE, Hannover, pp. 422–433, 2011.

[38] P. Bajorski. *Statistics for Imaging, Optics, and Photonics*, New York: John Wiley & Sons Publication, 2012.

[39] F. Pukelsheim. The three sigma rule. *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[40] Y. F. Tao, X. K. Xiao, R. Cheng. Range search on multidimensional uncertain data. *ACM Transactions on Database Systems*, vol. 32, no. 3, pp. 1–54, 2007.

[41] W. J. Thistleton, J. A. Marsh, K. Nelson, C. Tsallis. Generalized Box-Müller method for generating q-Gaussian random deviates. *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4805–4810, 2007.

[42] CISL Research Data Archive, [Online], Available: http://rda.ucar.edu, July 16, 2012.

[43] Sloan Digital Sky Survey, [Online], Available: http://www.sdss.org, July 16, 2012.

[44] International Surface Pressure Databank (ISPDv2) 1768 to 2010, [Online], Available: http://rda.ucar.edu/datasets/ds132.0/index.html, July 16, 2012.

Salman Ahmed Shaikh received his B. Eng. degree in computer systems and the M. Eng. degree in communication systems and networks from the Mehran University of Engineering and Technology, Pakistan in 2005 and 2008, respectively. He is currently a Ph. D. candidate at Kitagawa Data Engineering Lab, Faculty of Engineering, Information and Systems, University of Tsukuba, Japan. He is a member of International Association of Computer Science and Information Technology (IACSIT) and Pakistan Engineering Council (PEC).

His research interests include uncertain data processing, data mining and stream processing.

E-mail: salman@kde.cs.tsukuba.ac.jp (Corresponding author)

Hiroyuki Kitagawa received his B. Sc. degree in physics and his M. Sc. and Dr. Sc. degrees in computer science, all from the University of Tokyo, in 1978, 1980 and 1987, respectively. He is currently a full professor at Faculty of Engineering, Information and Systems and at Center for Computational Sciences, University of Tsukuba, Japan. He is a member of *ACM, IEEE Computer Society, the Database Society of Japan, IEICE, IPSJ, and JSSST*. He is now vice president of the Database Society of Japan, an IEICE Fellow, an IPSJ Fellow, and an associate member of the Science Council of Japan.

His research interests include data integration, stream processing, data intensive computing, data mining, and information retrieval.

E-mail: kitagawa@cs.tsukuba.ac.jp