Top-k Distance-based Outlier Detection on Uncertain Dataset

Salman Ahmed SHAIKH[†] and Hiroyuki KITAGAWA[†]

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

Abstract This paper studies the problem of top-k distance-based outlier detection on uncertain data. In this work, an uncertain object is modelled by a Gaussian probability density function. Since the Naive approach is very expensive due to costly distance function between uncertain objects, a populated-cell list (PC-list) based top-k distance-based outlier detection approach is proposed in this work. Where PC-list is a sorted list of non-empty cells of a grid (grid is used to index dataset objects). Using PC-list, the top-k outlier detection algorithm needs to consider only a fraction of dataset objects and hence quickly identifies candidate objects for top-k outliers. An extensive empirical study shows that our proposed approach is effective, efficient and scalable.

Key words Top-k Distance-based Outlier Detection, Uncertain Data, Gaussian Distribution, PC-list based Approach

1. Introduction

Outlier detection is one of the most important data mining techniques with vital importance in many application domains including credit card fraud detection, network intrusion detection, environment monitoring, etc. Hawkins [1] 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.

Most of the earliest outlier detection techniques were given by statistics [2]. However, most statistical techniques are univariate, and in the majority of techniques, the parameter of distribution is difficult to determine. In order to overcome these problems several distance-based approaches for outlier detection have been proposed in data mining [3], [4], [5].

Most of the outlier detection techniques proposed in data mining are suitable only for deterministic data. However, due to the incremental usage of sensors, RFIDs and similar devices for data collection these days, data contains certain degree of inherent uncertainty [6], [7], [8]. 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 [6]. In order to get reliable results from such data, uncertainty needs to be considered in calculation. Therefore in this work we study the problem of top-k distance-based outlier detection on uncertain data. This paper assumes that uncertainty in the values obtained from a sensor follows Gaussian distribution.

In the following, uncertainty of data is modelled by the most commonly used PDF, i.e., Gaussian distribution. Since the actual distance function is very costly to compute, we introduce a populated-cell list (PC-list) based top-k outlier detection technique. PC-list is a sorted list of non-empty cells of a d-dimensional grid, where grid is used to index our data. Using PC-list, our top-k outlier detection algorithm needs to consider only a fraction of the dataset objects and hence quickly identifies candidate objects for top-k outliers.

The rest of the paper is organized as follows. Sec.2. surveys the related work. Sec.3. formally defines top-k distance-based outlier detection on uncertain datasets. Its naive approach is given in Sec.4.. The PC-list and top-k algorithm are presented in Sec.5.. Sec.6. contains an extensive experimental evaluation that demonstrates the efficiency and of proposed techniques. Sec.7. concludes our paper.

2. Related Work

Distance-based outliers detection approach was introduced by Knorr, et al. in [3]. 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. [9] formulated distance-based outliers as the top-t data points whose distance to their κ^{th} nearest neighbour is largest. Angiulli et al. in [10] gave a slightly different definition of outliers than [9] by considering the average distance to their k nearest neighbour. Besides, there are some works on the detection of distance-based outliers over stream data including [5] and [11]. Both of these works are based on the Knorr, et al. definition of distance-based outliers. Furthermore, [11] gave an approximate algorithm to reduce the memory space required by its exact counterpart. Later on [5] extended [11] work by adding the concepts of multi-query and micro-cluster based distance-based outlier detection. A geometric approach of outlier detection has also been proposed in [12]. The proposed solution is only suitable for identifying abnormal nodes from the cluster of nodes placed nearby and not valid for the problem when the measurements of a single node is classified as outliers, based on the nodes past measurements. However all these approaches were given for deterministic data and could not handle uncertain data.

Recently a lot of research has focused on managing, querying and mining of uncertain datasets [13], [14]. The problem of outlier detection on uncertain datasets was first studied by Aggarwal, et al. in [13]. They represented an uncertain object by a PDF. They defined an uncertain object o to be a density-based (δ, η) outlier, if the probability of o existing in some subspace of a region with density at least η is less than δ. However, their work focuses on detecting outliers in subspaces. In practise, an outlier in subspace is not necessarily an outlier in full space as argued in [4]. [14] also proposed a distance-based outlier detection algorithm on uncertain datasets, which was later extended in [15] for probabilistic data streams. However in their works, an object's existential uncertainty is considered rather than representing an object by a PDF as in our work.

In [16], we proposed a cell-based approach of distance-based outlier detection on uncertain data. According to [16], an uncertain object o is a distance-based outlier if the expected number of objects lying within its D-distance is not greater than $\theta = N(1-p)$, 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. In practise parameter p is difficult to determine and is dependent on N. An arbitrary value of p may results in a very few or a lot of outliers for different N. Moreover from [16], we cannot obtain the outlier's ranking. Therefore in this work, we propose PC-list based approach of top-k distance-based outlier detection, which can always obtain k strongest outliers along with their ranking, provided $k \leqq N$.

3. Distance-based Outliers in Uncertain Data

The very first definition of distance-based outlier detection was given by Knorr, et al. in [3]. 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.*

This definition was given for deterministic data. However, the focus of this work is the detection of top-k outliers on a dataset whose attribute values are uncertain. We assume that the uncertainty is given by Gaussian distribution. In the following, we consider d-dimensional uncertain objects o_i, with attribute $\overrightarrow{\mathcal{A}_i} = (x_{i,1}, ..., x_{i,d})^T$ following Gaussian PDF with mean $\overrightarrow{\mu_i} = (\mu_{i,1}, ..., \mu_{i,d})^T$ and co-variance matrix $\Sigma_i = diag(\sigma^2_{i,1}, ..., \sigma^2_{i,d})$, respectively. The complete database consists of a set of such objects, $\mathcal{G}DB = \{o_1, ..., o_N\}$, where $N = |\mathcal{G}DB|$ is the number of uncertain objects in $\mathcal{G}DB$. The vector $\overrightarrow{\mathcal{A}_i}$ is a random variable that follows Gaussian distribution $\overrightarrow{\mathcal{A}_i} \sim \mathcal{N}(\overrightarrow{\mu_i}, \Sigma_i)$.

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

3. 1 Top-k Distance-based Outliers in Uncertain Data

We naturally extend Definition 1 for top-k distance-based outliers on uncertain datasets as follows.

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

We call objects that lie within the D-distance of an object o as D-neighbours of o, and denote the set of D-neighbours of o as $DN(o)$. In order to find distance-based outliers in $\mathcal{G}DB$, the distance between uncertain objects needs to be calculated, which is given by another distribution known as the Gaussian difference distribution [18]. Let $\overrightarrow{\mathcal{A}_i}$ and $\overrightarrow{\mathcal{A}_j}$ be two independent d-dimensional normal random vectors with means $\overrightarrow{\mu_i} = (\mu_{i,1}, ..., \mu_{i,d})^T$ and $\overrightarrow{\mu_j} = (\mu_{j,1}, ..., \mu_{j,d})^T$ and diagonal covariance matrices $\Sigma_i = diag(\sigma^2_{i,1}, ..., \sigma^2_{i,d})$ and $\Sigma_j = diag(\sigma^2_{j,1},..., \sigma^2_{j,d})$, respectively. Then $\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j} = \mathcal{N}(\overrightarrow{\mu_i} - \overrightarrow{\mu_j}, \Sigma_i + \Sigma_j)$ [18]. 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. Here we only give the 2-dimensional expression for $Pr(o_i, o_j, D)$. However, the expressions for higher dimensional cases can be obtained using Eq. 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 $\overrightarrow{\mu_i} = (\mu_{i,1}, \mu_{i,2})^T$, $\overrightarrow{\mu_j} = (\mu_{j,1}, \mu_{j,2})^T$, $\Sigma_i = diag(\sigma^2_{i,1}, \sigma^2_{i,2})$ and $\Sigma_j = diag(\sigma^2_{j,1}, \sigma^2_{j,2})$. Assuming that $\sigma_{i,1} = \sigma_{j,1} = \sigma_{i,2} = \sigma_{j,2} = \sigma$, $Pr(o_i, o_j, D)$ is given as follows.

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

where $\alpha^2 = \alpha_1^2 + \alpha_2^2$ and $\alpha_1 = \mu_{i,1} - \mu_{j,1}$ and $\alpha_2 = \mu_{i,2} - \mu_{j,2}$. For the proof of Eq. 2, please refer to our previous work [16].

Note that $Pr(o_i, o_j, D)$ only depends on α^2 but not on coordinates of o_i and o_j. Hence we can denote $Pr(o_i, o_j, D)$ by $Pr(\alpha, D)$ when there is no confusion. Computing this probability is usually very costly, and we have to avoid this computation as much as possible.

4. Naive Approach

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

Algorithm 1 The top-k Naive Approach

```
Input: $\mathcal{G}DB$, $D$, $k$
Output: Top-$k$ Distance-based Outliers
$N \leftarrow |\mathcal{G}DB|$, $\theta \leftarrow \infty$;
$\mathbb{C}_{obj} \leftarrow \phi$ (Sorted list of candidate top-$k$ outliers);
for each $o_i$ in $\mathcal{G}DB$ do
    $EN(o_i) \leftarrow 0$; (expected number of $D$-neighbours of $o_i$)
    for each $o_j$ in $\mathcal{G}DB$ do
        $EN(o_i) += Pr(o_i, o_j, D)$;
        if $EN(o_i) > \theta$ then GOTO next $o_i$;
    end for
    Insert $o_i$ and $EN(o_i)$ into $\mathbb{C}_{obj}$ ($\mathbb{C}_{obj}$ is sorted in ascending order w.r.t. $EN(o_i)$);
    if $|\mathbb{C}_{obj}| > k$ then
        Set $\theta = EN(o')$, where $o'$ is the $k^{th}$ object in $\mathbb{C}_{obj}$;
        Remove all $o'' \in \mathbb{C}_{obj}$, such that $EN(o'') > \theta$;
    end if
end for
return $\mathbb{C}_{obj}$;
```

5. The Populated-Cells List (PC-list)

The Naive approach requires a lot of computation time to detect top-k outliers even from a small dataset due to the costly distance calculation. To overcome this problem we propose a PC-list-based approach of top-k outlier detection.

PC-list is an array of non-empty cells of a d-dimensional grid containing uncertain data objects $o \in \mathcal{G}DB$. The PC-list helps in the detection of top-k distance-based outliers by identifying the cells containing candidate outliers.

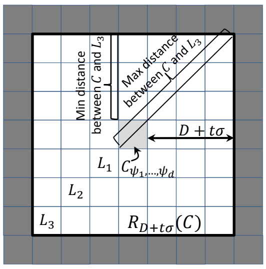

Min distance between C and L_3
Max distance between C and L_3
$D + t\sigma$
L_1
$C_{\psi_1,\ldots,\psi_d}$
L_2
L_3
$R_{D+t\sigma}(C)$

Figure 1: Cell Layers and Bounds Region

Lemma 1. *Let $o_i, o_j \in \mathcal{G}DB$ be two d-dimensional uncertain objects following Gaussian distribution and α denotes the ordinary Euclidean distance between the means of o_i and o_j. Then for $t \in \mathcal{R}$, denoting the number of standard deviations required to enclose a large probability (say $> 99\%$) of a d-dimensional Gaussian distribution, the following statements hold.*

(a) If $\alpha \leqq D - t\sigma, Pr(o_i, o_j, D) \approx 1$.

(b) If $\alpha \geqq D + t\sigma, Pr(o_i, o_j, D) \approx 0$.

Proof. The number of standard deviations s needed to enclose a given probability for a d-dimensional random variable X following Gaussian distribution can be obtained using the expression $Pr\{d_M(X, \mu) \leqq s\} = G_d(s^2)$ [**?**], where $d_M(X, \mu) = \sqrt{(X - \mu)^T \sum^{-1}(X - \mu)}$ is the Mahalanobis distance and $G_d(s^2)$ is the CDF of the chi-squared distribution with d-degrees of freedom.
Hence if t denotes the value of s, such that $Pr\{d_M(X, \mu) \leqq t\}$ covers a large area of Gaussian distribution (say $> 99\%$), then for $\alpha \leqq D - t\sigma, Pr(o_i, o_j, D) \approx 1$ and for $\alpha \geqq D + t\sigma, Pr(o_i, o_j, D) \approx 0$■

5.1 Structure

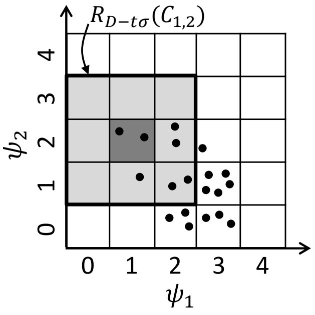
In order to find the top-k distance-based outliers from an uncertain dataset using PC-list, we first quantize each object $o \in \mathcal{G}DB$, to a d-dimensional space that is partitioned into cells of length l (The cell length is discussed in Sec. 5.3). Let $C_{\psi_1,\ldots,\psi_d}$ be any cell in grid $\mathcal{G}$, where positive integers $\psi_1, \ldots, \psi_d$ denote the cell indices. The layers $(L_1, \ldots, L_n)$ of $C_{\psi_1,\ldots,\psi_d} \in \mathcal{G}$ are the neighbouring cells of $C_{\psi_1,\ldots,\psi_d}$, as shown in Fig. 1 and are defined as follows.

$$L_1(C_{\psi_1,\ldots,\psi_d}) = \{C_{x_1,\ldots,x_d} | x_1 = \psi_1 \pm 1, \ldots, x_d = \psi_d \pm 1, \\ C_{x_1,\ldots,x_d} \neq C_{\psi_1,\ldots,\psi_d}\}.$$

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

$L_3(C_{\psi_1,\ldots,\psi_d}), \ldots, L_n(C_{\psi_1,\ldots,\psi_d})$ are defined in a similar way. We will use C to denote $C_{\psi_1,\ldots,\psi_d}$ when there is no confusion. Let $R_{D-t\sigma}(C)$ denotes the region formed by $\left\lfloor \frac{D-t\sigma}{l\sqrt{d}} - 1 \right\rfloor$ neighbouring layers of $C \in$

$\mathcal{G}$. Then PC-list (PC) is a table containing a tuple $(\psi_1, ..., \psi_d, \mathcal{C}(C), \mathcal{C}_{D-t\sigma}(C))$, denoted by $t_i (0 < i \leq |\text{PC}|)$, for each non-empty cell $C \in \mathcal{G}$ as shown in Fig.2. Here, $\psi_1, ..., \psi_d$ are the indices of C, $\mathcal{C}(C)$ is the object count of C, and $\mathcal{C}_{D-t\sigma}(C)$ is the object count within cells in $R_{D-t\sigma}(C)$ (including C itself). The region $R_{D-t\sigma}(C)$ is chosen in such a way that for each $o_i \in C$ and $o_j \in R_{D-t\sigma}(C)$, $Pr(o_i, o_j, D) \approx 1$. The tuples in the PC-list are sorted in the ascending order of $\mathcal{C}_{D-t\sigma}(C)$ column. The idea behind sorting is that outliers tend to exist in sparse region rather than dense regions. Sorting tuples in the PC-list, lets us identify and process cells in sparse regions before dense regions and makes pruning effective. Since each $t_i \in PC$ corresponds to a cell $C \in \mathcal{G}$, we will use C_{ti} to denote the cell referred by tuple t_i.



ψ_1	ψ_2	$\mathcal{C}(C)$	$\mathcal{C}_{D-t\sigma}(C)$
1	2	2	7
1	1	1	10
3	2	1	10
2	2	2	13
3	0	3	13
2	0	3	14
3	1	5	16
2	1	2	19

Figure 2: PC-list building

5.2 Cell Bounds

In order to identify cells $C_{ti} \in PC$, containing only inliers or candidate top-k outliers, their bounds on the expected number of D-neighbours are useful. A cell C_{ti} can be pruned as an inlier cell if the minimum expected number of D-neighbours for any object in cell C_{ti} is greater than threshold θ (θ is discussed later in this section). Similarly a cell can be identified as containing top-k outliers if the maximum expected number of D-neighbours for any object in C_{ti} is less than θ. Since the Gaussian distribution is unbounded, $Pr(o_i, o_j, D)$ is always greater than zero for $o_i, o_j \in \mathcal{G}$. Therefore all the cells in PC-list need to be considered for the computation of bounds of $C_{ti} \in PC$. Since our distance function depends only on α, that is, the distance between the means of two objects rather than their coordinates, we need to compute distance between cells in PC-list in order to compute their bounds. Beside distance between cells, pre-computed $Pr(\alpha, D)$ values and object count of each $C_{ti} \in PC$ are also required for the computation of $C_{ti} \in PC$ bounds.

Distance between Cells: Let C_{tp} and C_{tq} are two cells in PC with indices $\psi_{p1}, ..., \psi_{pd}$ and $\psi_{q1}, ..., \psi_{qd}$ respectively. Let $\Delta_{min}(C_{tp}, C_{tq})$ and $\Delta_{max}(C_{tp}, C_{tq})$ denote the minimum and maximum ordinary Euclidean distances between cells C_{tp} and C_{tq} respectively. Distance between cells depend on the position of cells C_{tp} and C_{tq} in the grid $\mathcal{G}$ and can be defined as follows.

$$\Delta_{min}(C_{tp}, C_{tq}) = l * (\sum_{s=1}^{d} \delta_{min,s}^2)^{1/2}$$

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

$$\Delta_{max}(C_{tp}, C_{tq}) = l * (\sum_{s=1}^{d} \delta_{max,s}^2)^{1/2}$$

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

Now we can obtain bounds for cells in PC-list using pre-computed $Pr(\alpha, D)$ values and the information available in PC-list. Let $LB(Pr(C_{tp}, C_{tq}))$ and $UB(Pr(C_{tp}, C_{tq}))$ denote the $Pr(\alpha, D)$ values at minimum $\alpha \geqq \Delta_{max}(C_{tp}, C_{tq})$ and maximum $\alpha \leqq \Delta_{min}(C_{tp}, C_{tq})$ respectively. Then for $C_{ti} \in PC$, $LB(C_{ti}) = (\sum_{C'_{ti} \in PC} LB(Pr(C_{ti}, C'_{ti})) * \mathcal{C}(C'_{ti}))$ and $UB(C_{ti}) = (\sum_{C'_{ti} \in PC} UB(Pr(C_{ti}, C'_{ti})) * \mathcal{C}(C'_{ti}))$.

Let $R_{D+t\sigma}(C)$ 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. Since the major contribution in the bounds for $C \in \mathcal{G}$ is done by the cells in region $R_{D+t\sigma}(C)$, we redefine the bounds for $C_{ti} \in PC$, to reduce the number of pre-computations and bounds computation time, as follows.

$$LB(C_{ti}) = \sum_{C'_{ti} \in \{PC \cap R_{D+t\sigma}(C_{ti})\}} LB(Pr(C_{ti}, C'_{ti})) * \mathcal{C}(C'_{ti}).$$

$$UB(C_{ti}) = \sum_{C'_{ti} \in \{PC \cap R_{D+t\sigma}(C_{ti})\}} UB(Pr(C_{ti}, C'_{ti})) * \mathcal{C}(C'_{ti})$$
$$+ Pr(\alpha', D) * (N - \sum_{C'_{ti} \in \{PC \cap R_{D+t\sigma}(C_{ti})\}} \mathcal{C}(C'_{ti}).$$

where $\alpha' = D + t\sigma$, for cell objects that lie greater than $D + t\sigma$ distance from the target cell C_{ti}.

Number of Pre-computations: Since we compute bounds 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 any 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 any cell. Hence the total number of pre-computations required are only $2\lceil \frac{D+t\sigma}{l} \rceil + 2$.

5.3 Candidate Outlier Cells

The bounds are computed for each $C_{ti} \in PC$, in order to prune inlier cells or identify outlier cells. A threshold is required to decide whether a $C_{ti} \in PC$ is inlier or outlier depending upon their bounds. The definition of threshold requires an attribute of candidate outlier cells table, hence we define candidate outlier cells table first.

Let $\mathbb{C}_{cell}$ denotes a table containing tuples of the form $(\psi_1, ..., \psi_d, \mathcal{C}(C_{tj}), LB(C_{tj}), UB(C_{tj}))$, denoted by $t_j(0 < j \leqq |\mathbb{C}_{cell}|)$, for each candidate outlier cell in PC-list. Let $C^k_{tj} \in \mathbb{C}_{cell}$ be a cell containing the k^{th} object. The tuples in $\mathbb{C}_{cell}$ are sorted in ascending order of the $UB(C_{tj})$ attribute. A $C_{ti} \in PC$ is a candidate outlier cell whenever $\sum_{C_{tj} \in \mathbb{C}_{cell}} \mathcal{C}(C_{tj}) < k$ or $LB(C_{ti}) \leqq \theta$, where $\theta = UB(C^k_{tj})$ denotes the threshold.

Cell Pruning and θ Updation: For $C_{ti} \in PC$, if $LB(C_{ti}) > \theta$, C_{ti} cannot contain any of the top-k outliers and can be pruned. On the other hand, if $LB(C_{ti}) \leqq \theta$, C_{ti} may contain top-k outlier. We add C_{ti} to $\mathbb{C}_{cell}$, such that $\mathbb{C}_{cell}$ remain sorted of $UB(C_{tj})$ attribute. Moreover if $UB(C_{ti}) < \theta$, add C_{ti} to $\mathbb{C}_{cell}$, set $\theta = UB(C^k_{tj})$ and remove C'_{tj} from $\mathbb{C}_{cell}$, such that $LB(C'_{tj}) > \theta$, 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 $C'_{ti} \in PC$ is found such that $Pr(\alpha, D) * \mathcal{C}_{D-t\sigma}(C'_{ti}) > \theta$, where $\alpha = D - t\sigma$, neither C'_{ti} nor any cell after it in PC-list can contain outliers. Hence the scanning can be stopped at this position.

Cell Length l: Due to the complexity of our distance function, it is not possible to derive a single cell length l suitable for all the combinations of D and variances. Very small cell length increases the number of cells in the Grid exponentially and the time required to construct the PC-list. A good starting point of the cell length that we found through experiments is the standard deviation, i.e., $l = \sigma$.

5.4 Outlier Detection Algorithms

In this section, we present a top-k distance-based outlier detection algorithm on uncertain dataset. The algorithm 2 first maps dataset objects to appropriate grid cells and create PC-list in lines 4 and 5 respectively. Since the PC-list is sorted in ascending order of its $\mathcal{C}_{D-t\sigma}(C_{ti})$ column, it guarantees that the cells in sparse region of $\mathcal{G}$ are at the top of the PC-list. Hence candidate outlier cells are expected to be at the top of the list. We scan the PC-list and add candidate outlier cells in $\mathbb{C}_{cell}$ until the stopping condition on line 8 becomes true. The number of objects in $\mathbb{C}_{Cell}$ may be greater than k, hence additional objects need to be removed.

Algorithm 2 The Top-k Distance-based Outliers

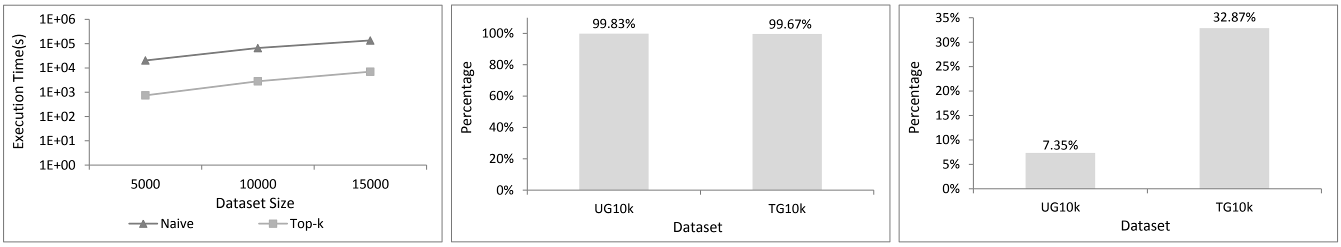
```
Input: $\mathcal{G}DB$, $D$, $l$, $k$
Output: Top-$k$ Distance-based Outliers
 1: $N \leftarrow |\mathcal{G}DB|$, $\theta \leftarrow \infty$;
 2: $\mathbb{C}_{cell} \leftarrow \phi$; (Candidate outlier cells list)
 3: $\mathbb{C}_{obj} \leftarrow \phi$; (Candidate outlier objects list)
 4: Create cell grid $\mathcal{G}$ depending upon dataset values and cell
    length $l$;
 5: Map each $o \in \mathcal{G}DB$ to an appropriate cell $C \in \mathcal{G}$;
 6: Create PC-list $PC$, using non-empty cells of $\mathcal{G}$;
 7: Sort $PC$ w.r.t. $\mathcal{C}_{D-t\sigma}(C)$ column;
    /*Searching candidate outlier cells*/
 8: for each $C_{ti}$ in $|PC|$ do
 9:   if $\mathcal{C}_{D-t\sigma}(C_{ti}) * Pr(D - t\alpha, D) > \theta$ then Exit for loop.
10:   Compute $LB(C_{ti})$ and $UB(C_{ti})$;
11:   if $LB(C_{ti}) \leqq \theta$ then
12:     Add $C_{ti}$ to $\mathbb{C}_{cell}$; (keep $\mathbb{C}_{cell}$ sorted of $UB(C_{tj})$)
13:     if $\mathbb{C}_{cell}$ contains $\geqq k$ objects then
14:       Set $\theta = UB(C^k_{tj})$, such that $C^k_{tj}$ contain the $k^{th}$ object;
15:       Remove all $C_{tj}$ from $\mathbb{C}_{cell}$, such that $LB(C_{tj}) > \theta$;
16:     end if
17:   end if
18: end for
    /*Calculating $EN(o)$ of candidate top-$k$ outliers*/
    The computation of $EN(o)$ is similar to that of Naive approach
    in Algorithm 1. The only difference is that in this algorithm
    we compute $EN(o)$ for the candidate objects in $\mathbb{C}_{cell}$ only.
```

In order to do so, exact expected D-neighbours $EN(o)$ of objects in candidate cells are calculated. The object o is then added to the $\mathbb{C}_{obj}$ (set of candidate outlier objects) along with its $EN(o)$. The objects in $\mathbb{C}_{obj}$ are sorted with respect to $EN(o)$. As the k^{th} object's $EN(o)$ is found, threshold θ is updated (refer line 11 of Alg.1). During the calculation of $EN(o)$, if for some o', $EN(o')$ becomes greater than θ, then o' can not be among top-k outliers and can be removed from further consideration.

6. Experiments

We conducted extensive experiments on synthetic datasets to evaluate the effectiveness and accuracy of our proposed algorithm. The algorithm is implemented in C++, GNU compiler. All experiments are performed on a system with an Intel Core 2 Duo, E8400 3.00GHz CPU and 2GB main memory running Ubuntu 12.04 OS. All programs run in main memory and no I/O cost is considered.

We use two synthetic datasets for our experiments: unimodal Gaussian (UG) and trimodal Gaussian (TG). UG and TG are 2-dimensional and are generated using Box-Muller method [17]. A shorthand notation "$DatasetName + DatasetSize$" (e.g. UG5k to denote 5,000 tuples of unimodal



(a) Naive vs. PC-list

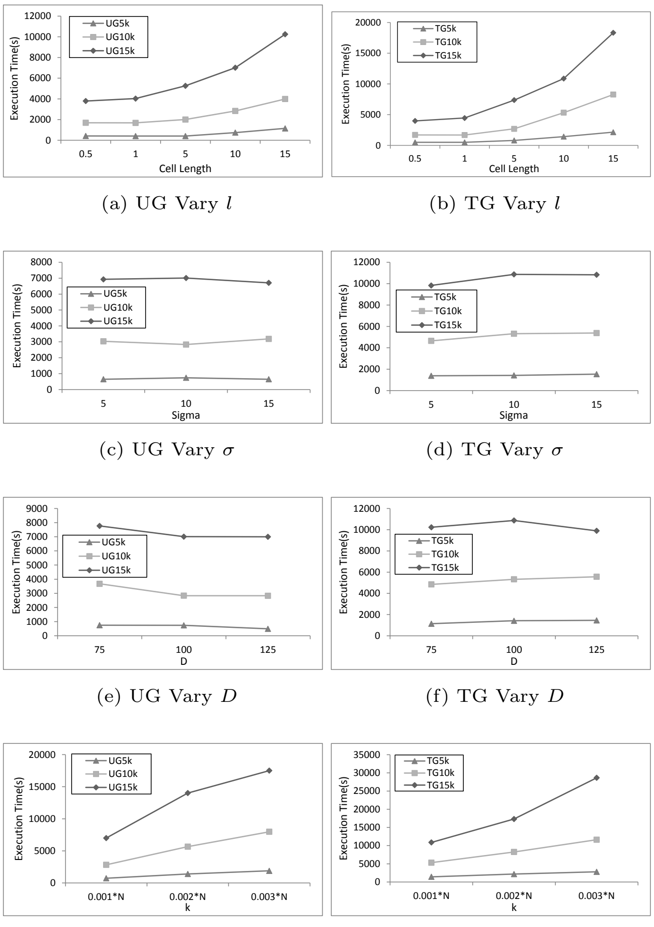
(b) Pruning Percentage

(c) Stopping Condition

Figure 3

Gaussian dataset) is used in figures.

All the datasets are normalized to have a domain of [0,1000] on every dimension. For each point z in any dataset, we create an uncertain object o, whose uncertainty is given by Gaussian distribution with mean z and standard deviation σ in all the dimensions. Unless specified, the following parameter values are used: $D = 100$, $\sigma = 10$, $l = 10$ and $k = 0.1\%$ of the respective dataset size. Pre-computation time is not included in the measurements.



(a) UG Vary l

(b) TG Vary l

(c) UG Vary σ

(d) TG Vary σ

(e) UG Vary D

(f) TG Vary D

(g) UG Vary k

(h) TG Vary k

Figure 4: Varying l, σ, D and k

We first conduct experiments to evaluate the efficiency of our proposed top-k algorithm presented in Sec.5. 4. Fig. 3a compares the execution times of the Naive and the proposed algorithm on UG dataset. Our proposed algorithm is several times faster than its Naive counterpart due to its strong pruning capability as can be observed from Fig.3b. Stopping condition discussed in Sec.5. 3 helps identify candidate outlier cells very quickly. Fig.3c shows the percentage of cells considered in the PC-list to identify candidate outlier cells before the execution of stopping condition. 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.

Graphs in Fig.4 show the effect of varying different parameters on the execution times. It is obvious from the graphs in Figs. 4a and 4b 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. On the other hand, large cell length decreases the pruning effect and hence increases the execution time of our algorithm. Therefore we recommend the use of cell length equal to the standard deviation as discussed in Sec. 5. 3.

From Figs. 4c, 4d, 4e and 4f we can observe that our algorithm is quite consistent and its performance is not much affected by the variation in parameters σ and D.

Next we perform experiments by varying parameter k. Figs. 4g and 4h show that increase in k results in increase in execution time of algorithm, which is quite obvious behaviour of our algorithm.

7. Conclusion and Future Work

In this work, we propose a top-k distance-based outlier detection approach on uncertain datasets of Gaussian distribution based on PC-list. Sorted PC-list helps identify candidate outlier objects very quickly without considering all the objects in the dataset. An extensive empirical study demonstrate the efficiency and scalability of our proposed approach.

Acknowledgement

This work has been partly supported by Grant-in-Aid for

Scientific Research(A)(#24240015A).

Refereces

[1] Hawkins D., ”Identi?cation of Outliers”, Chapman and Hall, 1980.

[2] Barnett V., Lewis T., ”Outliers in Statistical Data”, John Wiley, 1994.

[3] Edwin M. Knorr , Raymond T. Ng , Vladimir Tucakov.: ”Distance-Based Outliers: Algorithms and Applications”, The VLDB Journal, 2000.

[4] H.Nguyen, V.Gopalkrishnan, and I.Assent.: “An unbiased distance-based outlier detection approach for high-dimensional data”, DASFAA, 2011.

[5] M.Kontaki, A.Gounaris, A.N.Papadopoulos, K.Tsichlas, and Y.Manolopoulos.: “Continuous monitoring of distance-based outliers over data streams”, ICDE, 2011.

[6] 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., June 2010.

[7] Helm I., Jalukse L., Leito I.: Measurement Uncertainty Estimation in Amperometric Sensors: A Tutorial Review. Sensors 2010.

[8] Diao, Y., Li, B., Liu, A., Peng, L., Sutton, C., Tran, T., Zink, M.: Capturing Data Uncertainty in High-Volume Stream Processing., CIDR, 2009.

[9] Sridhar Ramaswamy , Rajeev Rastogi , Kyuseok Shim.: ”Efficient Algorithms for Mining Outliers from Large Data Sets”, ACM, SIGMOD, 2000.

[10] Fabrizio Angiulli and Clara Pizzuti.: “Fast Outlier Detection in High Dimensional Spaces”, PKDD, 2002.

[11] Fabrizio Angiulli and Fabio Fassetti.: “Detecting distance-based outliers in streams of data”, CIKM, 2007.

[12] Sabbas Burdakis, Antonios Deligiannakis,: ”Detecting Outliers in Sensor Networks Using the Geometric Approach”, ICDE, 2012.

[13] Aggarwal, C.C., Yu, P.S.: ”Outlier Detection with Uncertain Data”, SDM, 2008.

[14] Bin Wang, Gang Xiao, Hao Yu and Xiaochun Yang.: ”Distance-Based Outlier Detection on Uncertain Data”, IC-CIT, 2009.

[15] B.Wang, X.Yang, G.Wang, and G.Yu.: “Outlier detection over sliding windows for probabilistic data streams”, Journal of Comp. Sc.& Tech., 25(3), 2010.

[16] Salman Ahmed Shaikh and Hiroyuki Kitagawa,: ”Distance-based Outlier Detection on Uncertain Data of Gaussian Distribution”, APWeb, 2012.

[17] W.Thistleton, J.A.Marsh, K.Nelson and C.Tsallis, Generalized Box-Muller method for generating q-Gaussian random deviates, IEEE Trans. on Info. Theory, 2007.

[18] Weisstein, Eric W.:”Normal Difference Distribution”, From MathWorld. http://mathworld.wolfram.com.