Fast Top-k Distance-Based Outlier Detection on Uncertain Data

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
salman@kde.cs.tsukuba.ac.jp, kitagawa@cs.tsukuba.ac.jp

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 probability density function of a Gaussian distribution. We start with the Naive approach. We then introduce a populated-cell list (PC-list), a sorted list of non-empty cells of a grid (grid is used to index our data). Using PC-list, our 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 approximate top-k outlier detection algorithm is also presented to further increase the efficiency of our outlier detection algorithm. An extensive empirical study on synthetic and real datasets shows that our proposed approaches are efficient and scalable.

Keywords: 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 [4] 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 [6]. 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 [5], [11], [14].

Due to the increasing usage of sensors, RFIDs and similar devices for data collection these days, data contains certain degree of inherent uncertainty. The causes of uncertainty may include limitation of equipments, absence of data and delay or loss of data in transfer. In order to get reliable results from such data, uncertainty needs to be considered in calculation. In this work we study the problem of top-k distance-based outlier detection on uncertain data following the Gaussian distribution.

J. Wang et al. (Eds.): WAIM 2013, LNCS 7923, pp. 301–313, 2013.

In the following, uncertainty of data is modelled by the most commonly used PDF, i.e., the Gaussian distribution. Since the distance between uncertain data objects 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. Furthermore an approximate top-k outlier detection algorithm is also presented to increase the efficiency of our outlier detection algorithm.

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

2 Related Work

Distance-based outliers detection approach was introduced by Knorr, et al. in [5]. They defined a point p to be an outlier if at most M points are within D-distance of 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 neighbours. Besides, there are some works on the detection of distance-based outliers over stream data including [13], [14] and [15]. These works are based on the Knorr, et al. definition of distance-based outliers. Furthermore, [13] gave an approximate algorithm to reduce the memory space required by its exact counterpart. Later on [14] extended [13] 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 [2]. 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 [12], [7]. The problem of outlier detection on uncertain datasets was first studied by Aggarwal, et al. in [12]. They represented an uncertain object by a PDF. They defined an uncertain object o to be a density-based (δ, η) outlier, if the probability of o existing in some subspace of a region with density at least η is less than δ. 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 [11]. [7] also proposed a distance-based outlier detection algorithm on uncertain datasets, which was later extended in [8] 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 [1], we proposed a cell-based approach of distance-based outlier detection on uncertain data. According to [1], 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. 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 [1], we cannot obtain the outlier's ranking. Therefore in this work, we propose PC-list based approach of the top-k distance-based outlier detection, which can always obtain k strongest outliers along with their ranking, provided $k \leq N$.

3 Distance-Based Outliers in Uncertain Data

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

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

In this work, our focus is the detection of the top-k outliers on a dataset whose objects' attribute values are uncertain. This paper assumes that the uncertainty is given by the Gaussian distribution. The Gaussian distribution is chosen for representing uncertainty, because in statistics the Gaussian distribution *(or the normal distribution)* is the most important and the most commonly used.

In this paper, k-dimensional uncertain objects o_i are considered, with attribute $\overrightarrow{\mathcal{A}_i} = (x_{i,1}, ..., x_{i,k})^T$ following the Gaussian PDF with mean $\overrightarrow{\mu_i} = (\mu_{i,1}, ..., \mu_{i,k})^T$ and co-variance matrix $\Sigma_i = diag(\sigma^2_{i,1}, ..., \sigma^2_{i,k})$, respectively. Namely, the 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 object o_i. 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 $\mathcal{G}DB$.

3.1 Top-k Distance-Based Outliers in Uncertain Data

We naturally extend Definition 1 for the 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.*

The objects that lie within D-distance of an object o are called D-neighbours of o and the set of D-neighbours of o is denoted by $DN(o)$. In order to find 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 [3]. 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, $\overrightarrow{\mathcal{A}_i} - \overrightarrow{\mathcal{A}_j} = \mathcal{N}(\overrightarrow{\mu_i} - \overrightarrow{\mu_j}, \Sigma_i + \Sigma_j)$ [3]. Let $Pr(o_i, o_j, D)$ denotes the probability that $o_j \in DN(o_i)$. Then,

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

where R is a sphere with centre $(\overrightarrow{\mu_i} - \overrightarrow{\mu_j})$ and radius D. For the expression and derivation of $Pr(o_i, o_j, D)$, please refer our previous work [1]. Furthermore, we will use $Pr(\alpha, D)$ to denote $Pr(o_i, o_j, D)$ when there is no confusion, where α is an ordinary Euclidean distance between the means of $o_i \in \mathcal{GDB}$ and $o_j \in \mathcal{GDB}$. Computing this probability is usually very costly, and we have to avoid this computation as much as possible.

The Naive approach of the top-k outlier detection given in Alg. 1 uses Nested-loop. In order to find whether an object $o_i \in \mathcal{GDB}$ is a top-k outlier, we need to compute its expected D-neighbours $(EN(o_i))$. Computation of $EN(o_i)$ for an object $o_i \in \mathcal{GDB}$ requires evaluation of N expensive distance functions. During the computation of $EN(o_i)$, if expected D-neighbours become greater than threshold θ, o_i is an inlier and the computation of $EN(o_i)$ is stopped. On the other hand, if $EN(o_i)$ is less than or equal to θ, o_i is added to candidate list of outliers $\mathbb{C}_{obj}$, along with its expected D-neighbours. The $\mathbb{C}_{obj}$ is kept sorted in ascending order of D-neighbours' column and the top-k objects in it are selected as outliers. In the worst case, this approach requires $O(N^2)$ evaluations of distance function, which is very expensive.

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

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

(a) If $\alpha \leq D - t\sigma', Pr(o_i, o_j, D) \approx 1$.
(b) If $\alpha \geq D + t\sigma', Pr(o_i, o_j, D) \approx 0$.

Algorithm 1. The top-k Naive Approach

```
Input: $\mathcal{G}DB$, $D$, $k$
Output: Top-$k$ Distance-based Outliers
$N \leftarrow |\mathcal{G}DB|$, $\theta \leftarrow \infty$, $\mathbb{C}_{obj} \leftarrow \phi$ (Candidate top-$k$ outliers list);
for each $o_i$ in $\mathcal{G}DB$ do
    $EN(o_i) \leftarrow 0$; (expected number of $D$-neighbours of $o$)
    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 $\mathbb{C}_{obj}$ (Keep $\mathbb{C}_{obj}$ sorted w.r.t. $EN(o)$);
    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 $\mathbb{C}_{obj}$;
```

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 $Pr\{d_M(X, \mu) \leq s\} = G_d(s^2)$ [19], 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.
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 [3]. Hence if t denotes the value of s, such that $Pr\{d_M(X, \mu) \leq t\}$ covers a large area of the Gaussian distribution (say $> 99\%$), then for $\alpha \leq D - t\sigma', Pr(o_i, o_j, D) \approx 1$ and for $\alpha \geq D + t\sigma', Pr(o_i, o_j, D) \approx 0$ ■

4.1 Structure

In order to find the top-k distance-based outliers from an uncertain dataset using the PC-list, we first quantize each object in $\mathcal{G}DB$, to a d-dimensional space that is partitioned into cells of length l (The cell length is discussed in Sec. 4.3). Let $C_{\psi_1,...,\psi_d}$ be any cell in grid $\mathcal{G}$, where positive integers $\psi_1, ..., \psi_d$ denote the cell indices. The layers $(L_1, ..., L_n)$ of $C_{\psi_1,...,\psi_d} \in \mathcal{G}$ are the neighbouring cells of $C_{\psi_1,...,\psi_d}$, as shown in Fig. 1 and are derived as follows.

$$L_1(C_{\psi_1,...,\psi_d}) = \{C_{x_1,...,x_d} | x_1 = \psi_1 \pm 1, ..., x_d = \psi_d \pm 1, C_{x_1,...,x_d} \neq C_{\psi_1,...,\psi_d}\}.$$
$$L_2(C_{\psi_1,...,\psi_d}) = \{C_{x_1,...,x_d} | x_1 = \psi_1 \pm 2, ..., x_d = \psi_d \pm 2,$$
$$C_{x_1,...,x_d} \notin L_1(C_{\psi_1,...,\psi_d}), C_{x_1,...,x_d} \neq C_{\psi_1,...,\psi_d}\}.$$

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

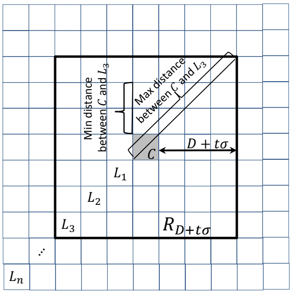
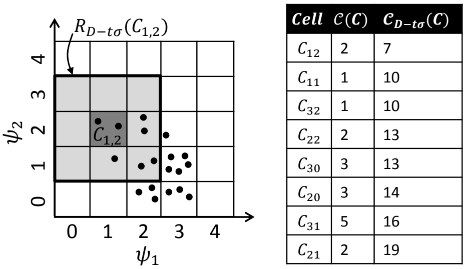


Fig. 1. Cell Layers and Bounds



Cell	$\mathcal{C}(C)$	$\mathcal{C}_{D-t\sigma}(C)$
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

Let $R_{D-t\sigma}(C)$ denotes a region formed by $\left\lfloor \frac{D-t\sigma}{l\sqrt{d}} - 1 \right\rfloor$ neighbouring layers of $C \in \mathcal{G}$. The region $R_{D-t\sigma}(C)$ is chosen in such a way that for each $o_i \in C$ and $o_j \in R_{D-t\sigma}(C)$, $Pr(o_i, o_j, D) \approx 1$. let $\mathcal{C}(C)$ is the count of objects in C, and $\mathcal{C}_{D-t\sigma}(C)$ is the count of objects within cells in region $R_{D-t\sigma}(C)$ (including C itself). Then the PC-list (PC) is a sorted list containing $\mathcal{C}(C)$ and $\mathcal{C}_{D-t\sigma}(C)$ 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 $\mathcal{C}_{D-t\sigma}(C)$ column. The idea behind sorting is that outliers tend to exist in sparse regions. Sorting tuples in the PC-list, lets us identify cells in sparse regions.

4.2 Cell Bounds

In order to identify cells $C \in PC$, containing only inliers or candidate top-k outliers, their bounds on the expected D-neighbours are used. A cell C can be pruned as an inlier cell if the minimum expected 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 if the maximum expected 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 $o_i, o_j \in \mathcal{G}$. 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 distance between cells, object count of each $C \in PC$ and precomputed $Pr(\alpha, D)$ values for α ranging from the minimum to the maximum ordinary Euclidean distances between cells in $\mathcal{G}$ are also required for the computation of $C \in PC$ bounds. The precomputed values are stored in a look-up table to be used by the top-k outlier detection algorithm.

Distance between Cells: Let C_p and C_q are two cells in PC with indices $\psi_{p1}, ..., \psi_{pd}$ and $\psi_{q1}, ..., \psi_{qd}$ 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 the grid $\mathcal{G}$ and can be derived as follows.

$$\Delta_{min}(C_p, C_q) = l.(\sum_{s=1}^{d} \delta_{min,s}^2)^{1/2} \text{ where } \delta_{min,s} = \begin{cases} \psi_{ps} - (\psi_{qs} + 1) & \psi_{ps} > \psi_{qs} \\ (\psi_{ps} + 1) - \psi_{qs} & \psi_{ps} < \psi_{qs} \\ \psi_{ps} - \psi_{qs} & \psi_{ps} = \psi_{qs} \end{cases}$$

$$\Delta_{max}(C_p, C_q) = l.(\sum_{s=1}^{d} \delta_{max,s}^2)^{1/2} \text{ where } \delta_{max,s} = \begin{cases} (\psi_{ps} + 1) - \psi_{qs} & \psi_{ps} \geq \psi_{qs} \\ \psi_{ps} - (\psi_{qs} + 1) & \psi_{ps} < \psi_{qs} \end{cases}$$

Now we can obtain bounds for cells in the PC-list 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 $\alpha \geq \Delta_{max}(C_p, C_q)$ and maximum $\alpha \leq \Delta_{min}(C_p, C_q)$ respectively. Then for a $C \in PC$, $LB(C) = (\sum_{C' \in PC} LB(Pr(C, C')) * \mathcal{C}(C')$ and $UB(C) = (\sum_{C' \in PC} UB(Pr(C, C')) * \mathcal{C}(C'))$.

Let $R_{D+t\sigma}(C)$ denotes the 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 $C \in PC$, to reduce the number of pre-computations and bounds computation time, as follows.

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

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

Number of Pre-computations: Since the bounds are pre-computed for the cells in region $R_{D+t\sigma}(C)$, $Pr(\alpha, D)$ values are 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 required are only $2\lceil \frac{D+t\sigma}{l} \rceil + 2$.

4.3 Candidate Outlier Cells

Let $\mathbb{C}_{cell}$ is a list containing candidate outlier cells from PC, sorted in ascending order of $UB(C)$. Let $C^k \in \mathbb{C}_{cell}$ is a cell with the minimum upper bound containing the k^{th} object. A $C \in PC$ is a candidate outlier cell whenever $\sum_{C' \in \mathbb{C}_{cell}} \mathcal{C}(C') < k$ or $LB(C) \leq \theta$, where $\theta = UB(C^k)$ denotes the threshold.

Cell Pruning and θ Updation: For a $C \in PC$, if $LB(C) > \theta$, C cannot contain any of the top-k outliers and can be pruned. On the other hand, if $LB(C) \leq \theta$, C may contain the top-k outlier. C is added to $\mathbb{C}_{cell}$, such that $\mathbb{C}_{cell}$ remain sorted of its $UB(C)$ attribute. Set $\theta = UB(C^k)$ and remove C' from $\mathbb{C}_{cell}$, 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 $Pr(D-t\sigma, D)* \mathcal{C}_{D-t\sigma}(C') > \theta$, neither C' nor any cell after it in PC-list can contain outliers. Hence the PC-list scanning can be stopped at this point.

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., $l = \sigma$.

Algorithm 2. The Top-k Distance-based Outliers

```
Input: $\mathcal{G}DB$, $D$, $l$, $k$
Output: Top-$k$ Distance-based Outliers
$N \leftarrow |\mathcal{G}DB|$, $\theta \leftarrow \infty$;
$\mathbb{C}_{cell} \leftarrow \phi$, $\mathbb{C}_{obj} \leftarrow \phi$; (Candidate outlier cells and top-$k$ outliers list respectively)
Create cell grid $\mathcal{G}$ depending upon dataset values and cell length $l$;
Map each $o \in \mathcal{G}DB$ to an appropriate cell $C \in \mathcal{G}$;
Create PC-list $PC$, using non-empty cells of $\mathcal{G}$;
Sort $PC$ w.r.t. $\mathcal{C}_{D-t\sigma}(C)$ column;
/*Searching candidate outlier cells*/
for each $C$ in $|PC|$ do
    if $\mathcal{C}_{D-t\sigma}(C) * Pr(D - t\sigma, D)$ then Exit for loop. /*Stopping condition*/
    Compute $LB(C)$ and $UB(C)$;
    if $LB(C) \leq \theta$ then
        Add $C$ to $\mathbb{C}_{cell}$ (keep $\mathbb{C}_{cell}$ sorted of $UB(C)$ attribute);
        if $\mathbb{C}_{cell}$ contains $\geq k$ objects then
            Set $\theta = UB(C^k)$, such that $C^k$ contain the $k^{th}$ object;
            Remove all $C$ from $\mathbb{C}_{cell}$, such that $LB(C) > \theta$;
        end if
    end if
end for
/*Calculating $EN(o)$ of candidate top-$k$ outliers*/
The computation of $EN(o)$ is similar to that of the Naive approach. The only
difference is that in this algorithm we compute $EN(o)$ for the candidate objects in
$\mathbb{C}_{cell}$ only.
```

4.4 Outlier Detection Algorithms

In this section, we present two algorithms to detect top-k distance-based outliers from uncertain datasets. The first algorithm computes accurate expected

D-neighbours for all the un-pruned objects, however the second algorithm approximates the expected D-neighbours to reduce the algorithm computation cost.

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 $\mathcal{C}_{D-t\sigma}(C)$ column, it guarantees that cells in the sparse regions of the 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 $\mathbb{C}_{cell}$ until the stopping condition on line 8 becomes true. The number of objects in $\mathbb{C}_{Cell}$ may be greater than k, hence we calculate expected D-neighbours $EN(o)$ of candidate objects to find the top-k outliers and their ranking. The 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 in ascending order of $EN(o)$ column. As the k^{th} object's $EN(o)$ is found, threshold θ is set (refer line 10 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 the top-k outliers and is removed from further consideration.

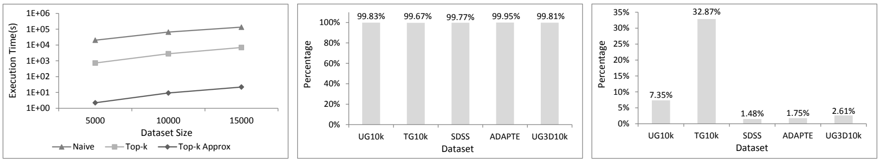
Approximate Top-k Algorithm: In the top-k algorithm, the minimum number of distance function computations required for the evaluation of k $EN(o)$ is kN, however the candidate outlier objects which require the evaluation of $EN(o)$, may be greater that k. When the distance function is expensive to compute (as in our case), computation of even k $EN(o)$ is very expensive. According to our distance function, the major contribution in the evaluation of $EN(o)$ is done by the nearer objects. Hence $EN(o)$ for each un-pruned o can be approximated with high accuracy by considering objects only within $D + t\sigma$ distance of o according to Lemma 1, rather than considering all the objects in dataset. Rest of the algorithm is same as that of Alg.2.

Maximum Approximation Error: For any $o \in \mathcal{G}DB$, maximum approximation error (ε_{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 $\varepsilon_{max} = (N-1) * Pr(D + t\sigma + \beta, D)$, where $\beta \in \mathbb{R}$ is a very small real value to make distance greater than $D + t\sigma$.

For example for $t = 9$, $d = 2$ and $N = 10^5$ objects, $\varepsilon_{max} \approx 10^{-5}$. ε_{max} depends mainly on t. In practice $t \geq 6$ gives sufficiently accurate $EN(o)$ for $d = 2$ and 3. For higher d values, we need to increase t value according to Lemma 1.

5 Experiments

We conducted extensive experiments on synthetic and real data to evaluate the effectiveness and scalability of our proposed algorithms. All algorithms were implemented in C++, GNU compiler. All experiments were performed on a system with an Intel Core 2 Duo CPU 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.

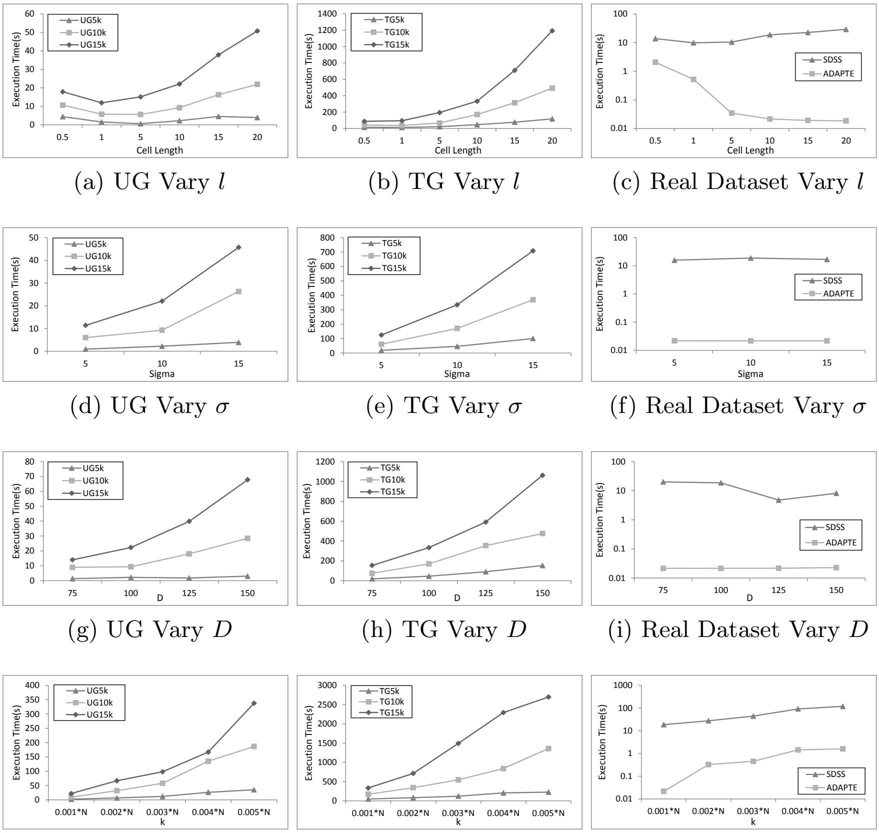


(a) Naive vs. PC-list (b) Pruning Percentage (c) Stopping Condition

Fig. 3. Effectiveness of the PC-list based approach

We use two synthetic datasets and two real datasets for our experiments. Synthetic datasets, unimodal Gaussian (UG) and trimodal Gaussian (TG) are 2-dimensional and are generated using BoxMuller method [16]. A 3-dimensional unimodal Gaussian (UG3D) dataset is also generated for the evaluation of our proposed approaches on 3-dimensional data. A shorthand notation "$DatasetName + DatasetSize''$ (e.g. UG5k to denote 5,000 tuples of unimodal Gaussian dataset) is used in figures. As for real-world data, we use two datasets: ADAPTE and SDSS. ADAPTE is obtained from CISL Research data archive [17] and SDSS is obtained from Sloan Digital Sky Survey [18]. 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 (or "RA") and Declination (or "Dec") coordinates of stars and galaxies.

All the datasets are normalized to have a domain of [0,1000] on every dimension. For each point z in a 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, following parameter values are used: $D = 100$, $\sigma = 10$, $l = 10$ and $k = 0.1\%$ of the respective dataset size. For approximate top-k algorithm, we considered objects only within $D + 6\sigma$ distance of each un-pruned object o. Pre-computation time is not included in the measurements. We first conduct experiments to evaluate the efficiency of our proposed algorithms presented in Sec.4.4. Fig. 3a compares the execution times of Naive and proposed algorithms on UG dataset. Our proposed algorithms are several times faster than its Naive counterpart due to their strong pruning capability as can be observed from Fig.3b. Stopping condition discussed in Sec.4.3 helps identify candidate outlier cells very quickly. Fig.3c shows the percentage of cells considered in the PC-list to identify candidate outlier cells. 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 approximate top-k algorithm is thousands of times faster than the ordinary top-k algorithm, due to the reason discussed in Sec.4.4. From theoretical analysis in Sec. 4.4 and experiments we found that the approximate top-k algorithm gives an accuracy of up to several decimal digits in the evaluation of $EN(o)$ and hence the outliers obtained from both the algorithms are same. Therefore in the rest of this section, we will evaluate only approximate top-k algorithm.



(a) UG Vary l (b) TG Vary l (c) Real Dataset Vary l

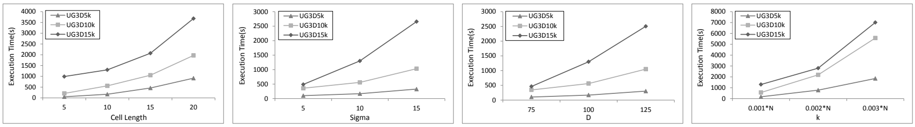
(d) UG Vary σ (e) TG Vary σ (f) Real Dataset Vary σ

(g) UG Vary D (h) TG Vary D (i) Real Dataset Vary D

(j) UG Vary k (k) TG Vary k (l) Real Dataset Vary k

Fig. 4. Varying parameters l, σ, D and k for 2D datasets

Graphs in Fig.4 show the effect of varying different parameters on the execution times. It is obvious from the graphs in Figs. 4a, 4b and 4c 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. Therefore we recommend the use of cell length equal to the standard deviation as discussed in Sec. 4.3. In Fig. 4c, k is very small due to the small size of ADAPTE dataset and therefore pruning time dominates the algorithm execution time. Consequently as the number of cells decreases due to the increase in cell length, algorithm execution time decreases. Next we perform experiments by varying the parameter σ. As σ increases, the uncertainty of the object 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 Figs. 4d, 4e and 4f. Figs. 4g, 4h and 4i show the effect 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 need to be considered for the approximation of $EN(o)$. Therefore it increases the execution time of the overall algorithm for larger values of D. From Figs. 4j, 4k and 4l, increase in k results in an increase in execution time of algorithm, which is quite obvious behaviour of our algorithm. Figure 5 shows similar effects of varying different parameters on three dimensional dataset.



(a) UG3D Vary l (b) UG3D Vary σ (c) UG3D Vary D (d) UG3D Vary k

Fig. 5. Varying parameters l, σ, D and k for 3D dataset

6 Conclusion

In this work, the top-k distance-based outlier detection approach on uncertain datasets of the Gaussian distribution based on the PC-list is proposed. PC-list helps identify candidate outlier objects very quickly without considering all the objects in dataset. Moreover an approximate top-k outlier detection approach is presented to further reduce the algorithm computation cost. An extensive empirical study on real and synthetic datasets demonstrate the effectiveness and scalability of our proposed approaches.

Acknowledgement: This work has been partly supported by Grant-in-Aid for Scientific Research(A)(#24240015A).

References

1. Shaikh, S.A., Kitagawa, H.: Distance-Based Outlier Detection on Uncertain Data of Gaussian Distribution. In: Sheng, Q.Z., Wang, G., Jensen, C.S., Xu, G. (eds.) APWeb 2012. LNCS, vol. 7235, pp. 109–121. Springer, Heidelberg (2012)
2. Burdakis, S., Deligiannakis, A.: Detecting Outliers in Sensor Networks Using the Geometric Approach. In: ICDE (2012)
3. Weisstein, E.W.: Normal Difference Distribution, From MathWorld - A Wolfram Web Resource, `http://mathworld.wolfram.com`
4. Hawkins, D.: Identication of Outliers. Chapman and Hall (1980)
5. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-Based Outliers: Algorithms and Applications. The VLDB Journal (2000)
6. Barnett, V., Lewis, T.: Outliers in Statistical Data. John Wiley (1994)
7. Wang, B., Xiao, G., Yu, H., Yang, X.: Distance-Based Outlier Detection on Uncertain Data. In: ICCIT (2009)
8. Wang, B., Yang, X., Wang, G., Yu, G.: Outlier detection over sliding windows for probabilistic data streams. Journal of Comp. Sc. & Tech. 25(3) (2010)

9. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient Algorithms for Mining Outliers from Large Data Sets. In: ACM, SIGMOD (2000)
10. Angiulli, F., Pizzuti, C.: Fast Outlier Detection in High Dimensional Spaces. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, pp. 15–27. Springer, Heidelberg (2002)
11. Nguyen, H.V., Gopalkrishnan, V., Assent, I.: An unbiased distance-based outlier detection approach for high-dimensional data. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part I. LNCS, vol. 6587, pp. 138–152. Springer, Heidelberg (2011)
12. Aggarwal, C.C., Yu, P.S.: Outlier Detection with Uncertain Data. In: SDM (2008)
13. Angiulli, F., Fassetti, F.: Detecting distance-based outliers in streams of data. In: CIKM (2007)
14. Kontaki, M., Gounaris, A., Papadopoulos, A.N., Tsichlas, K., Manolopoulos, Y.: Continuous monitoring of distance-based outliers over data streams. In: ICDE (2011)
15. Ishida, K., Kitagawa, H.: Detecting Current Outliers: Continuous Outlier Detection over Time-Series Data Streams. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 255–268. Springer, Heidelberg (2008)
16. Thistleton, W., Marsh, J.A., Nelson, K., Tsallis, C.: Generalized Box-Muller method for generating q-Gaussian random deviates. IEEE Trans. on Info. Theory (2007)
17. CISL Research Data Archive, `http://rda.ucar.edu`
18. Sloan Digital Sky Survey, `http://www.sdss.org`
19. Bajorski, P.: Statistics for Imaging, Optics and Photonics. A John Wiley & Sons Publication (2012)