Distance-based and Density-based Outlier Detection on Time Series Data A Comparative Study with Stabilized Approach

Maya Nayak, Prasannajit Dash
[1]Orissa Engineering College
Bhubaneswar, Odisha, India
mayanayak3299@yahoo.com, prasannajitdash11@gmail.com

ABSTRACT: *Outlier detection always plays a major role in different industrial domains ranging from major financial fraud detection to network intrusions and also major clinical diagnosis of different diseases. Over the years of research, distance based outlier detection algorithm and density based outlier detection algorithm have emerged as a viable, scalable, parameter-free alternative way to the very long before the traditional approaches. In this paper, we evaluate both the distance and density based outlier detection approach applying on a time-series data i.e. stock market data traded for three months. We begin by surveying and examining the design landscape of the above outlier detection approaches. Also implemented the outlier detection framework for the said design and analysis to understand the various pros and cons of various optimizations proposed by us on a real-time data set i.e. the current stock market data set. The combinations of optimization techniques (factors) and strategies through distance and density based outlier approaches always dominate on various types of data sets. So plays a dominant role on this real-time stock market data set.*

Keywords: Canonical Algorithm, Simple Nested Loop (SNL), Prunning, Approximate Nearest Neighbor (ANN), kNN, dbScan, Complex, dimension, Attribute, Object, Tuple

Received: 18 October 2014, Revised 23 November 2014, Accepted 27 November 2014

1. Introduction

In a larger number of data, outlier detection is said as per layman's terms, the noted physicist Stephen Hawkin's definition as "an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". For over a century, the outlier detection is the prime focus in more research work in the fields of statistics. Here, the data is considered to follow a parametric distribution and the objects never confirmed to the parametric nature are considered as outliers. Several non-parametric approaches which could not rely on the non-parametric data distribution have been proposed. The first class of methods which confirms to the non-parametric nature is the clustering algorithms. Basically partition-based clustering (k-means, fuzzy c-means), hierarchical based clustering and density-based clustering i.e. dbscan etc. The idea comes as a label to create outliers having those points that were not a core part of any clusters. While going for discovering various tasks on knowledge based are classified broadly into categories like dependency detection, class description, class identification and exception outlier detection. In correspond to the first three categories, the task is applied to a large number of objects in the dataset. The referred tasks in data mining (e.g. association rules, data classification, data clustering etc.). The fourth category focuses on a very small percentage of data objects quite discarded as noise. For last four decades, computational geometry inspired for more on outlier detection. It is based on the depth and convex hull computations. Due to the computational power, outliers are calculated relying more on the principle of data lying in outer space of the data space. The dependency on convex hull calculation hampers the scalability of such outlier detection on high dimensional data and even on low dimensional data these methods are quite very expensive with very large datasets.

To address the scalability factor on high dimensional data as well low dimensional data, the density based outlier detection approach which determines the outliers by analyzing the object's neighborhood density and presents the opportunity to identify the outliers that are missed by other methods. Here these methods give the idea of the distance that plays an important role in determining the neighborhood and local density. While these methods give assurance in improved performance compared to methods inspired on statistical or computational geometry principles, they are not scalable. The well known nearest-neighbor principle which is distance based was first proposed by N_g and K_{norr}. It employs a well-defined distance metric to detect outliers i.e. the greater is the distance of the object to its neighbors, the more likely it is an outlier. In data mining and data analytics, the distance based approach is a non-parametric approach which supports generally the larger dataset with high dimensionality. The basic algorithm for such distance-based algorithm is the nested loop (*NL*) algorithm. This nested-loop (*NL*) algorithm calculates the distance between each pair of objects and calculates the outliers in regards to those objects which are very far from the neighbor-hood objects. The NL algorithm is the quadratic complexity in nature with respect to the number of objects, making it mostly not fit for mining large database like government audit database, clinical trial data and network data. So looking into the prospect of designing of such algorithms, we have identified several important optimizations and algorithms such as the use of compact data structures. Hence, the fundamental optimization technique that should be followed by all such algorithms is to find out an approximate nearest neighbor search such that it should find out a point that is an outlier in the data set.

1.1 Related Works

For so many years the outlier tests have been conducted depending on the (i) data distribution, (ii) the distribution parameters like standard deviation, variance, mean etc.,(iii) the number of expected outliers including upper or lower outliers. There are two serious problems while testing these outliers, the first one is that all the outliers are considered as univariate (i.e. single attribute).

Symbol	Description
N	Number of outliers in dataset to be identified in the database
K	Number of nearest neighbours to be considered
$D^k(p)$	Distance between point p and it k^{th} closest neighbour
$D^k{}_{min}$	Smallest distance between a point and its k^{th} nearest neighbour from data set
$\|P\|$	Number of objects (size) of a partition *P*
$R(P)$	The MBR diagonal value of a partition *P*
MINDIST	Minimum distance between an object and *MBR* structure-
MAXDIST	Maximum distance between an object and *I* structure

Table 1. Notations

The second one is that all of them are considered as distribution based. In some situations there is no idea that whether a particular attribute follows a normal distribution i.e. a gamma distribution such that we need to perform an extensive testing to search for a mathematical distribution that fits the attribute.So depth-based methods are not expected to be more practical for more than four dimensional large datasets.

The DB-based (distance-based) techniques for outlier detection have become popular due to non-parametric nature and the ability to scale to large datasets. There are three main definitions of outliers:

(a) Outliers are objects with lesser than *k* neighbors in the dataset, where a neighbor is an object within a distance *R*

(b) Outliers are having n objects giving the highest distance values to their respective k^{th} nearest neighbor

(c) Outliers having n objects presenting the highest average distance to their respective *k* nearest neighbors

All three definitions exemplify Hawking's definition, that is, the greater is the distance of the object to its neighbors, the more likely it is an outlier. The first definition originally proposed by Knorr and Ng relies on both the definition of a neighborhood (*R*) as well as the number of neighbors *k* in order to determine whether a point is an outlier or not. The choice of *R* is an additional parameter that we will set to determine apriori.

In fact, for a useful selection of *R*, it is shown that Knorr and Ng's definition leads to a quadratic algorithm where all distances between pairs of objects must be calculated in the worst case. Another measure that magnifies such data for each object *p* is D^k (*p*), which is the distance between the object p and its k^{th}-nearest neighbor. This measure was proposed by Ramaswamy, and is the basis of the second definition of an outlier. Among the three, however, the kNN definition by Ramaswamy is definitely more frequently used and is the focus of the paper.

2.1 Distance Based Outliers Problem Domain

The distance based outlier is taken into consideration as an object *P* in the real time dataset *R* and it is *DB* (*p*, *D*). The *DB* (*p*, *D*) is an outlier when at least the fraction *p* of the total objects in *R* lies greater than the euclidean distance *D* from *P.* Whether being standardized or without standardized in some applications, the dissimilarity (or similarity) between the objects is calculated by taking the distance between each pair of objects. The distance said above is the Euclidean distance measure which is defined as

$$d\,(i,j) = sqrt\,((\,x_{i1} - x_{j1})^2 + \,(x_{i2} - x_{j2})^2 + ... + (x_{in} - x_{jn})^2 \qquad (1)$$

where $i = (x_{i1}, x_{i2}, \ldots\ldots\ldots.., x_{in})$ and $j = (x_{j1,}\, x_{j2}, \ldots\ldots\ldots.., x_{jn})$

The euclidean distance satisfies the following mathematical requirements of a distance function :

(i) $d\,(i, j) >= 0$: Distance is a non negative number

(ii) $d\,(i, j) = 0$: Distance of any object to itself is always zero

(iii) $d\,(i, j) = d\,(j, i)$: Distance is symmetric in nature

(iv) $d\,(i, j) < = d\,(i, h) + d\,(h, j)$: Calculating the distance from object i to object *j* in space is no more than making a distance over any other object *h*

The notation i.e. *DB* (*p*, *D*) means the shorthand notation for distance based outlier using the parameters *p* and *D* which is nothing but the Hawkins definition. The definition is well defined for *k*-dimensional datasets for any value of *k*. Anyhow, the *DB*-outliers are not restricted computationally to small values of *k*. Hence the DB-outliers rely on the computations of distance values based on a metric distance function.

2.2 Properties of DB-Outliers

Definition 1: The generalization of *DB* (*p*, *D*) brings another definition Def for outliers, if there are specific values for p_0 and D_0 so that the object i.e. *O* is a *DB* (p_0, D_0) which is the outlier. Outliers are considered in a normal distribution as the observations that lie 3 or more standard deviations(σ) from the mean i.e.μ

Definition 2: Let us define Def_{normal} as '*t*' which is an outlier in a normal distribution with mean μ and standard deviation σ if $|(t - \mu) / \sigma| >= 3$. *DB* (*p*, *D*) unifies Def_{normal} with $p_0 = 0.9988$ and $D_0 = 0.13\sigma$ so that, *t* is an outlier according to Def_{normal} if t is a *DB* (0.9988, 0.13ó) as an outlier. If the value 3σ is changed to some other value, such as 4 σ, the above definition should be modified with the related p_0 and D_0 to show that *DB* (*p*, *D*) still unifies the new distribution of Def_{normal}.

Definition 3: Let us define Def_{normal} shows that t is considered as an outlier in a Poisson distribution with parameter $\mu = 3.0$ if *t*

$t >= 8$. $DB\,(p, D)$ unifies $Def_{poisson}$ with $p_0 = 0.9892$ and $D_0 = 1$.

2.3 A Simple Nested Loop Algorithm For Finding All *DB* (*p, D*) – Outliers

The algorithm *NL* i.e. the nested loop algorithm given below uses the nested loop design in blocks oriented nature. Let us assume that the total size of the buffer of *B*% of the total dataset size, the algorithm divides the entire buffer space into two equally measured arrays as the first array and the second array where the whole dataset got divided into arrays. By doing this, it computes the distance between each pair of objects or tuples. For each object *t* in the first array, a count of its D-neighbors is maintained. The process of counting almost gets done for a particular tuple whenever the number of D-neighbors exceeds *M*.

2.3.1 Pseudo-Code For Algorithm NL (Nested Loop)

```
(a) The size that holds B/2% of the dataset will be filled with a block having tuples from T.
(b) In first array, for each tuple $t_i$, steps to be followed:
1. The counter is to be initialized i.e. $count_i$ to 0
2. In the first array, for each tuple check $t_i$ whether the dist($t_i$, $t_j$) <= $D$
3. If the above distance happens to be less or equal to $D$, then increment the $t_i$ i.e. $count_i$ > M where we could mark $t_i$ as a non-outlier and further proceed to the next $t_i$
4. While the remaining blocks to be compared to the first array, then fill the second array with another block. For each unmarked tuple $t_i$ in the first array where for each tuple $t_i$ in the second array if dist($t_i$, $t_j$) < = $D$
5. If it happens, then increment the $count_i$ by 1. So if $count_i$ > $M$, mark $t_i$ as a non outlier and proceed to next $t_i$
6. For each unmarked tuple ti in the first array, report $t_i$ as an outlier
7. In the second array if it has served as the first array anytime before, then swap the names of the first and second arrays and go to step (b).
```

3. Application of Stock Market Data as Time Series Data

As the below stock market text dataset hold for the National Stock Exchange data of Tata Motors for a quarter, these dataset consists of the date of transaction, opening price of the day, high and low price for a day and the closing price for a day. Since the data is collected for three months, it is time-series data as the outliers are to be observed for multidimensional data i.e. attributes around five in the dataset. The data related to the opening and closing price is calculated in terms of euclidean distance for data in columns like open price and closed price.

The sum is taken of the euclidean distances (row-wise) between the contents in open price and close price. Hence the average euclidean distance is calculated for the total objects i.e. rows or records (tuples). Here the fraction of object is calculated by taking the value of '*p*' as 0.9. Since the total number of rows are 249 in the dataset, therefore the fraction of the object i.e. '*f*' is calculated as (249× (1–*p*)) which produces 24.9. Here the one dimensional matrix i.e. *C*1 of rank 249×1 where it contains all euclidean distances (row or tuple wise) between open and close prices. While iterating through the records from 1 to 249, if the value in C1 is less than the average euclidean distance then if the count of the object is greater than the fraction of the object, then it is not an outlier otherwise it is an outlier as shown Figure 2 and Figure 3 respectively. If the calculated distance is greater than the average euclidean distance, then also it is considered as the outlier. Then the Nested Loop (NL) is applied to calculate the non-outliers and outliers in respect to the date.

In the same way, the high price and low price are calculated in terms of euclidean distance row wise as it checks the euclidean distance measurement for columns like high price and low price. The sum is calculated for euclidean distances (row-wise) between the data in high price and low price. Hence the average euclidean distance is calculated for the total objects means rows or records (tuples). Here the fraction of object is calculated by taking the value of '*p*' as 0.9. Since the total number of rows are 249 in the dataset, therefore the fraction of the object i.e. '*f*' is calculated as (249× (1-*p*)) which produces 24.9. So that the one dimensional matrix of order or rank 249×1 i.e. C2 where it contains all euclidean distances(row wise or tuple wise) between high and low prices. While traversing through the records from 1 to 249, if the value in C2 is less than the average euclidean distance then if the count of the object is greater than the fraction of the object, then it is not an outlier otherwise it is an outlier as shown in Figure 4 and Figure 5 respectively. If the distance is greater than the average euclidean distance, then also it is counted as outlier. Then the Nested Loop (NL) is applied to calculate the non-outliers and outliers in respect to the date.

In Figure 2 and 4, the color i.e. red indicates the non-outliers as per the date against the euclidean distance difference between Open and Close Price column data. The x-axis is the date and the y-axis is the distance differences. So in figure 3 and 5, the blue color indicates the outliers.

4. Clustering-Based Techniques for Outlier Detection

Clustering finds groups of strongly related objects. Outlier detection finds objects that are not strongly related to other objects. Thus, an object is a cluster-based outlier if the object does not belong strongly to any cluster. In detecting outliers, small clusters that are far from other clusters are considered to contain outliers. This approach is sensitive to the number of clusters selected. It requires thresholds for the minimum cluster size and the distance between a small cluster (with outliers) and other clusters. If a cluster is smaller than the minimum size, it is regarded as a cluster of outliers. This paper presents a density-based clustering technique known as Density Based Spatial Clustering of Applications with Noise (DBSCAN). The detection of time series outliers using the DBSCAN technique is focused in the paper.

4.1 Density Based Spatial Clustering of Applications with Noise (DBSCAN)

It presents a density-based clustering technique which estimates similarities between points from a data set with respect to distance and partitions them into subsets known as clusters, so that the points in each cluster share some common trait. We use the clustering technique used in data mining known as Density Based Spatial Clustering of Applications with Noise (DBSCAN).

DBSCAN is designed to discover the clusters and the noise from a given set of points by classifying a point

(i) Inside of a cluster (core point)

(ii) In the edge of a cluster (border point), or

(iii) As neither a core point nor a border point (noise) .

DBSCAN requires two important parameters; Eps, which is a specified radius around a point to other points, and MinPts, which is the minimum number of points required to form a cluster.

4.2 Key Concepts

The following definitions are the key concepts in understanding the DBSCAN algorithm:

Definition 1 The Eps - neighborhood of a point p, denoted by NEps (p), is defined by

$$N_{EPS}(p) = \{q \in P \mid dist(p,q) <= E_{ps}$$

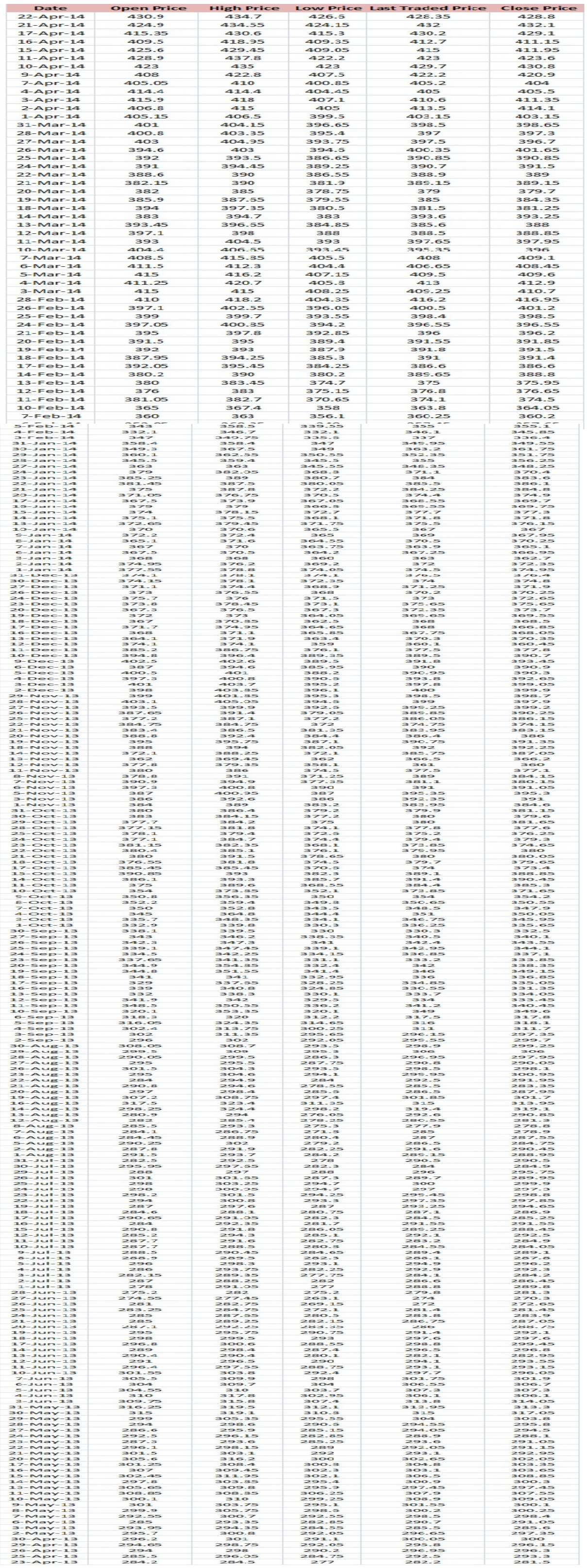
A point is a core point if it has more than a specified minimum number of points required to form a cluster (*MinPts*) within an *Eps* - neighborhood. These are points that are in the interior of a cluster. A border point has fewer than MinPts within an *Eps*, but it is in the *Eps* - neighborhood of a core point. A noise point is any point that is neither a core point nor a border point.

Definition 2: The density-based approach is an approach that regards clusters as regions in the data space in which the objects are dense and separated by regions of low object density (outliers).

Definition 3: Considering the points $p1$ and $p2$, if $p1$ is directly density reachable from $p2$ if points are close enough to each other such that dist($p1, p2$) < *Eps*, as measured using Euclidean distance or using any other distance measure. There are at least MinPts points in its neighborhood. For example, if *MinPts* = 6, then p1 must have at least 6 points as its neighbors. This concept of direct density-reachability shows that p1 is density reachable from $p2$ because the *dist* ($p1$, $p2$) < *Eps*, and $p1$ has enough points as its neighbors.

Definition 4: A point $p1$ is density reachable from a point $p2$ with respect to *Ep*s and *MinPts* if there is a chain of points $p1$, ...

Sources:http://nseindia.com/live_market/dynaContent/live_watch/get_qoute/GetCoutejsp?symbol=TATAMOTORS



Date	Open Price	High Price	Low Price	Last Traded Price	Close Price
22-Apr-14	430.9	434.7	426.5	428.35	428.8
21-Apr-14	424.9	434.55	424.15	432	432.1
17-Apr-14	415.35	430.6	415.3	430.2	429.1
16-Apr-14	409.5	418.95	409.35	412.7	411.15
15-Apr-14	425.6	429.45	409.05	415	411.95
11-Apr-14	428.9	437.8	422.2	423	423.6
10-Apr-14	423	435	423	429.7	430.8
9-Apr-14	408	422.8	407.5	422.2	420.9
7-Apr-14	405.05	410	400.85	405.2	404
4-Apr-14	414.4	414.4	404.45	405	405.5
3-Apr-14	415.9	418	407.1	410.6	411.35
2-Apr-14	406.8	415	405	413.5	414.1
1-Apr-14	405.15	406.5	399.5	403.15	403.15
31-Mar-14	401	404.15	396.65	398.5	398.65
28-Mar-14	400.8	403.35	395.4	397	397.3
27-Mar-14	403	404.95	393.75	397.5	396.7
26-Mar-14	394.6	403	394.5	400.35	401.65
25-Mar-14	392	393.5	386.65	390.85	390.85
24-Mar-14	391	394.45	389.25	390.7	391.5
22-Mar-14	388.6	390	386.55	388.9	389
21-Mar-14	382.15	390	381.9	389.15	389.15
20-Mar-14	382	385	378.75	379	379.7
19-Mar-14	385.9	387.55	379.55	385	384.35
18-Mar-14	394	397.35	380.5	381.5	381.25
14-Mar-14	383	394.7	383	393.6	393.25
13-Mar-14	393.45	396.55	384.85	385.6	388
12-Mar-14	397.1	398	388	388.5	388.85
11-Mar-14	393	404.5	393	397.65	397.95
10-Mar-14	404.4	406.55	393.45	395.35	396
7-Mar-14	408.5	415.35	405.5	408	409.1
6-Mar-14	411.5	412.3	404.4	406.65	408.45
5-Mar-14	415	416.2	407.15	409.5	409.6
4-Mar-14	411.25	420.7	405.8	413	412.9
3-Mar-14	415	415	408.25	409.25	410.7
28-Feb-14	410	418.2	404.35	416.2	416.95
26-Feb-14	397.1	402.55	396.05	400.5	401.2
25-Feb-14	399	399.7	393.55	398.4	398.5
24-Feb-14	397.05	400.35	394.2	396.55	396.55
21-Feb-14	395	397.8	392.85	396	396.2
20-Feb-14	391.5	395	389.4	391.55	391.85
19-Feb-14	392	393	387.9	391.8	391.5
18-Feb-14	387.95	394.25	385.3	391	391.4
17-Feb-14	392.05	395.45	384.25	386.6	386.6
14-Feb-14	380.2	390	380.2	389.65	388.8
13-Feb-14	380	383.45	374.7	375	375.95
12-Feb-14	376	383	375.15	376.8	376.65
11-Feb-14	381.05	382.7	370.65	374.1	374.5
10-Feb-14	365	367.4	358	363.8	364.05
7-Feb-14	360	363	356.1	360.25	360.2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
5-Feb-14	343	358.5	339.55	355	355.1
4-Feb-14	332.1	346.7	332.1	346.1	345.85
3-Feb-14	347	349.75	335.5	337	336.4
31-Jan-14	358.4	358.4	347	349.95	349.55
30-Jan-14	349.3	367.5	349	363.2	361.75
29-Jan-14	360.1	362.55	350.55	352.35	351.75
28-Jan-14	345.5	359.4	345.5	355	356.25
27-Jan-14	363	363	345.55	348.35	348.25
24-Jan-14	379	382.05	368.8	371.1	370.4
23-Jan-14	385.25	389	380.7	384	383.6
22-Jan-14	381.45	387.5	380.05	385.5	386.1
21-Jan-14	375	387.8	372.2	384.25	384.8
20-Jan-14	371.05	376.75	370.5	374.4	374.9
17-Jan-14	367.5	373.9	367.05	368.55	369.7
16-Jan-14	379	379	366.8	369.55	369.75
15-Jan-14	374	378.15	372.7	377.7	377.3
14-Jan-14	375.1	375.5	368.1	371.8	371.8
13-Jan-14	372.65	379.45	371.75	375.5	376.15
10-Jan-14	370	370.6	365.5	367	367
9-Jan-14	372.2	372.4	365	369	367.95
8-Jan-14	365.1	371.6	364.55	370.5	370.25
7-Jan-14	367	370	363.75	363.9	365.1
6-Jan-14	367.5	370.5	364.2	367.25	366.95
3-Jan-14	368	368	360	363	362.7
2-Jan-14	374.95	376.2	369.2	372	372.35
1-Jan-14	377.55	378.8	374.05	374.5	374.95
31-Dec-13	374.1	378.1	374.1	376.5	376.4
30-Dec-13	374.15	378.1	372.35	374	374.8
27-Dec-13	371.1	374.4	368.9	371.25	371.9
26-Dec-13	373	376.55	368	370.2	370.25
24-Dec-13	375.7	376	371.5	373	372.65
23-Dec-13	373.8	378.45	373.1	375.65	375.65
20-Dec-13	367.3	376.5	367.3	372.35	373.7
19-Dec-13	372	373	364.05	369.65	369.55
18-Dec-13	367	370.85	362.5	368	368.5
17-Dec-13	371.7	374.95	364.65	368	366.85
16-Dec-13	368	371.1	365.85	367.75	368.05
13-Dec-13	364.1	371.9	363.4	370.3	370.35
12-Dec-13	374.1	374.1	359	360.1	360.45
11-Dec-13	385.2	386.75	376.1	377.5	377.8
10-Dec-13	394.8	396.4	389.35	389.5	390.7
9-Dec-13	402.5	402.6	389.5	391.8	393.45
6-Dec-13	387	394.6	385.95	390	390.9
5-Dec-13	400.5	401	388.2	390.95	390.3
4-Dec-13	397.3	400.8	390.5	393.8	392.65
3-Dec-13	401	403.7	395.2	397.8	399.05
2-Dec-13	398	403.85	396.1	400	399.9
29-Nov-13	399	401.85	395.3	398.5	398.7
28-Nov-13	403.1	405.05	394.5	399	397.9
27-Nov-13	393.5	399.9	392.5	399.25	399.2
26-Nov-13	387.65	391.4	379.05	389.85	390.25
25-Nov-13	377.2	387.1	377.2	386.05	386.15
22-Nov-13	384.75	384.75	372	374.75	374.15
21-Nov-13	383.4	386.5	381.35	383.95	383.15
20-Nov-13	388.8	392.4	384.4	386.4	386
19-Nov-13	395	395.75	387.1	390.75	391.35
18-Nov-13	388	394	382.05	392	392.25
14-Nov-13	372.1	388.25	372.1	385.75	387.05
13-Nov-13	362	369.45	362	366.5	366.2
12-Nov-13	377.8	379.35	358.1	361	360
11-Nov-13	380	386	374.5	377.5	377.1
8-Nov-13	378.8	391	371.25	389	384.15
7-Nov-13	390.9	394.9	377.35	381.1	380.15
6-Nov-13	397.3	400.8	390	391	391.05
5-Nov-13	387	400.95	387	395.35	395.3
3-Nov-13	386	392.6	386	392.35	391
1-Nov-13	384	389	383.2	383.95	384.6
31-Oct-13	380	386.4	379.2	379.9	381.15
30-Oct-13	383	384.15	377.2	380	379.6
29-Oct-13	377.7	384.2	375	380	381.65
28-Oct-13	377.15	381.8	374.1	377.8	377.6
25-Oct-13	378.1	379.4	372.5	375.2	376.25
24-Oct-13	377.1	384.7	374.4	379.4	379.3
23-Oct-13	381.15	382.35	368.1	373.85	374.65
22-Oct-13	380.4	385.1	376.1	379.95	380
21-Oct-13	380	391.5	378.65	380	380.05
18-Oct-13	376.55	381.8	374.5	379.7	379.65
17-Oct-13	385.45	385.45	370.5	374	373.4
15-Oct-13	390.85	393	382.3	389.1	388.85
14-Oct-13	386.1	393.3	385.7	391.4	390.45
11-Oct-13	375	389.6	368.55	384.4	385.3
10-Oct-13	354	373.85	352.1	373.85	371.65
9-Oct-13	350.8	356.35	350	354	354.2
8-Oct-13	352.2	359.4	349.8	350.65	350.55
7-Oct-13	350	352.8	343.5	348.5	347.9
4-Oct-13	345	364.8	344.4	351	350.05
3-Oct-13	335.7	348.35	334.1	346.75	345.95
1-Oct-13	332.9	339.8	330.3	336.25	335.65
30-Sep-13	338.1	339.5	330	330.3	332.5
27-Sep-13	343	346.2	338.35	340.5	340.1
26-Sep-13	342.3	347.3	341	342.4	343.55
25-Sep-13	339.1	347.45	339.1	342.95	344.1
24-Sep-13	334.5	342.25	334.15	336.85	337.1
23-Sep-13	337.65	341.35	331.1	333.2	333.85
20-Sep-13	344.9	354.85	332.4	342	338.35
19-Sep-13	344.8	351.55	341.4	346	349.15
18-Sep-13	341	341	332.95	336	336.85
17-Sep-13	329	337.55	328.25	334.85	335.05
16-Sep-13	339	340.8	324.85	330.55	331.35
13-Sep-13	332	338.3	330.1	333.7	334.05
12-Sep-13	341.9	342	329.5	334	333.45
11-Sep-13	348.5	350.55	336.2	341.2	340.45
10-Sep-13	320.1	353.35	320.1	349	349.6
6-Sep-13	318.3	320	312.2	317.5	317.8
5-Sep-13	316.05	324.35	314.65	316	318.1
4-Sep-13	302.4	313.75	300.25	311	311.7
3-Sep-13	302	311.35	295.65	296.15	297.35
2-Sep-13	296	302	292.05	299.55	299.7
30-Aug-13	308.05	308.7	293.5	298.9	299.25
29-Aug-13	299.5	309	295.3	306	306
28-Aug-13	290.05	299.5	286.3	296.95	297.95
27-Aug-13	295	295.5	287.75	290.8	290.05
26-Aug-13	301.5	304.3	293.5	298.5	298.1
23-Aug-13	295	304.6	294.1	299.95	300.95
22-Aug-13	284	294.9	284	292.5	291.95
21-Aug-13	290.8	294.6	278.55	285.5	283.35
20-Aug-13	297	298.4	285.5	286.5	287.95
19-Aug-13	307.2	308.75	297.4	301.85	301.7
16-Aug-13	317.5	323.4	311.35	315	313.95
14-Aug-13	298.25	324.4	298.2	319.4	319.1
13-Aug-13	280.9	294	276.05	292.6	290.85
12-Aug-13	282	285.4	278.25	280.55	281.3
8-Aug-13	285.5	293.3	275.3	277.9	278.8
7-Aug-13	284.1	286.75	271.8	285	278.9
6-Aug-13	284.45	288.9	280.4	287	287.55
5-Aug-13	290.25	302	279.2	286.5	284.75
2-Aug-13	287.8	291.9	282.25	291.6	290.45
1-Aug-13	291.5	293.7	284.2	289.15	288.95
31-Jul-13	282.5	292.5	278	290.5	290.5
30-Jul-13	295.95	297.55	282.3	284	284.9
29-Jul-13	288	297	288	296	295.75
26-Jul-13	301	301.55	287.3	289.7	289.95
25-Jul-13	298	303.25	294.7	300	299.9
24-Jul-13	298	300.95	294.7	297	297.3
23-Jul-13	298.2	301.5	294.25	295.45	298.8
22-Jul-13	294	300.8	293.3	297.35	297.85
19-Jul-13	287	297.6	287	293.25	294.65
18-Jul-13	284.6	288.1	280.75	287.1	286.9
17-Jul-13	290.65	291.05	282.3	284.5	285.25
16-Jul-13	284	292.35	281.7	291.55	291.55
15-Jul-13	290.8	291.8	286.05	289.25	288.45
12-Jul-13	285.2	294.3	285.1	292.1	292.5
11-Jul-13	287.7	291.6	282.75	283.2	284.9
10-Jul-13	287.7	288.5	280.8	284.55	284.05
9-Jul-13	288.5	290.45	284.65	289.4	289.1
8-Jul-13	288.9	289.5	282.3	288.1	287.8
5-Jul-13	296	298.3	293.1	294.9	296.2
4-Jul-13	286	293.75	282.25	292.9	292.3
3-Jul-13	282.15	289.35	277.75	284.1	284.2
2-Jul-13	287	288.25	282	286.6	286.45
1-Jul-13	278	291.05	277	288.8	289.8
28-Jun-13	275.2	282	275.2	279.8	281.3
27-Jun-13	274.55	277.45	263.1	274	270.3
26-Jun-13	281	282.75	269.15	272	272.65
25-Jun-13	283.25	284.75	272.1	281.4	281.45
24-Jun-13	285	287.55	280.5	283.8	283.9
21-Jun-13	285	289.25	282.15	286.75	287.05
20-Jun-13	287.1	292.25	283.35	286	288.75
19-Jun-13	295	295.75	290.75	291.4	292.1
18-Jun-13	298	299.5	293	297.6	297.6
17-Jun-13	296.8	300.6	288.55	298.8	299.45
14-Jun-13	289	298.4	287.4	296.5	296.8
13-Jun-13	290.4	290.4	280.1	282.1	282.95
12-Jun-13	291	296.5	290	294.1	293.55
11-Jun-13	296.4	297.55	288.75	293.1	293.15
10-Jun-13	301.55	303.8	292.4	297.7	296.05
7-Jun-13	305.5	309.9	298	301.75	301.9
6-Jun-13	304	309.7	304	306.55	306.7
5-Jun-13	304.55	310	303.7	307.3	307.3
4-Jun-13	310	317.8	302.95	306.1	306.1
3-Jun-13	309.75	315.8	307.4	313.8	314.05
31-May-13	316.25	319.5	312.1	313.95	313.3
30-May-13	315	319.1	310.8	315	317.05
29-May-13	299	305.35	295.55	304	303.8
28-May-13	294	298.6	290.5	294.55	295.8
27-May-13	286.6	295.9	285.15	294.05	294.5
24-May-13	292.5	296.15	282.85	288.9	288.1
23-May-13	287.3	293.6	285.25	293.6	291.05
22-May-13	296.1	298.15	289	292.05	291.15
21-May-13	301.5	303.1	292	293.1	292.95
20-May-13	305.6	316.2	300	302.65	302.05
17-May-13	301.25	308.4	300.8	304.8	303.35
16-May-13	307	309.45	302.3	303.1	303.65
15-May-13	302.45	311.95	302.1	306.5	308.85
14-May-13	297.8	303.85	295.4	300.9	300.3
13-May-13	305.65	309.8	295.9	297.45	297.45
11-May-13	308.85	308.85	306.25	307.9	307.55
10-May-13	300.1	310	299.25	308.9	309.05
9-May-13	301	303.75	295.1	301.55	300.1
8-May-13	299.9	305.75	298.5	300.2	300.25
7-May-13	292.55	300.7	292.55	298.5	298.4
6-May-13	285	293.35	282.85	290.7	291.05
3-May-13	293.95	294.35	284.55	285.5	285.6
2-May-13	295.7	300.8	292.05	296.65	297.35
30-Apr-13	296.2	301	291.1	300.05	300
29-Apr-13	294.65	298.75	292.05	295.8	296.15
26-Apr-13	294	298	290.2	296.95	296.3
25-Apr-13	285.5	296.05	284.75	292.5	293.3
23-Apr-13	284.2	284.5	277	282.2	281.5

Figure 1. Stock Market Data in National Stock Exchange, Mumbai for Tata Motors

pn, such that *pi* + 1 is directly density-reachable from *pi*.

Definition 5: A point *p*0 is density-connected to a point *pn* with respect to *Eps* and *MinPts* if there is a point *q* such that both p0 and pn are density-reachable from *q* with respect to *Eps* and *MinPts*.

4.3 The Algorithm

In general, using parameters Eps and MinPts, DBSCAN finds a cluster by starting with an arbitrary point *p* from a set of points and retrieves all points density-reachable from p with respect to Eps and *MinPts*. Suppose *p* is a core point, if *p* has neighboring points greater than or equal to the value of *MinPts*, a cluster is started. Otherwise, the point is labeled as an outlier, and a new unvisited point is retrieved and processed leading to the discovery of a further cluster of core points. A point can be in a cluster, and it can be an outlier. After all the points have been visited, any points not belonging to any clusters are considered outliers. The formal details are given in Table 2, and Algorithm 1 presents pseudocode for the DBSCAN algorithm.

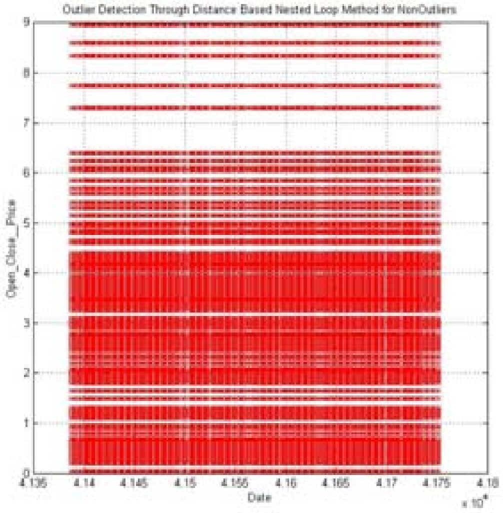


Figure 2. Open_Close Non Outliers in respect to date

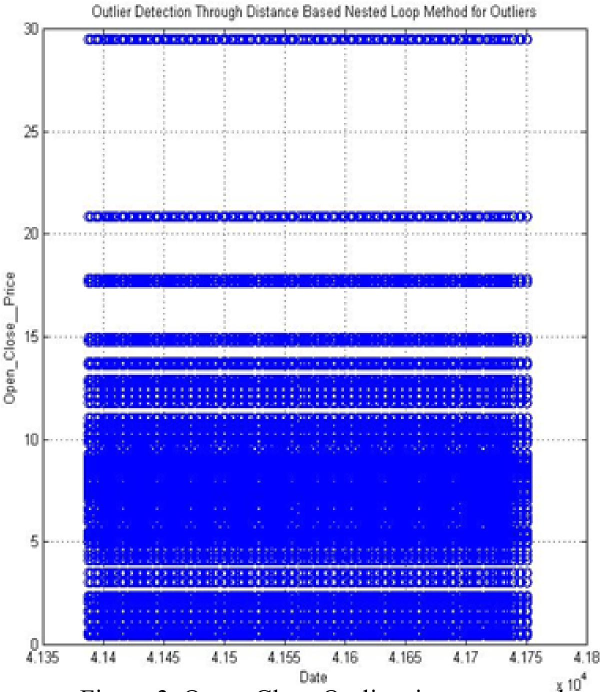


Figure 3. Open_Close Outliers in respect to date

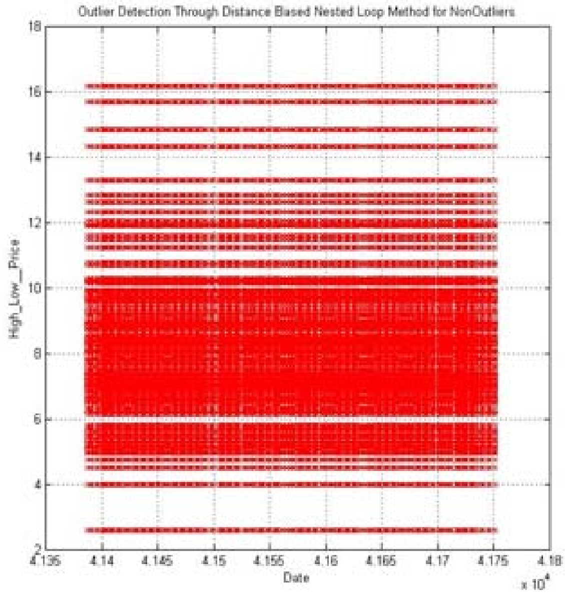


Figure 4. High_Low Non Outliers in Respect to Date

(a) Select the values of *Eps* and *MinPts* for a data set *P* to be clustered
(b) Start with an arbitrary point *p* and retrieve all points density-reachable
(c) If p is a core point that contains at most MinPts points, then a cluster is formed, otherwise, label *p* as an outlier
(d) A new unvisited point is retrieved and processed leading to the discovery of further clusters of core points
(e) Repeat step (b) until all the points have been visited
(f) Label any points not belonging to any cluster as outliers

Table 2. DBSCAN Algorithm

Algorithm

Aim: Clustering the data with Density-Based Scan Algorithm with Noise (DBSCAN)

Input: SetOfPoints (P) - data set (m, n); m-objects, n-variables

Eps : neighborhood radius

MinPts : minimal number of objects required to form a cluster

Output: A vector specifying assignment of a point to certain cluster

For example : 1st and 3rd points can be in cluster No.1 and 2nd and 4th points can be in cluster No.2

(a) As a function in MATLAB, the function DBSCAN takes three parameters i.e. SetOfPoints, MinPts and Eps where the output is the parameter i. e. IsPointAnOutlier.

(b) Call the Normalize method which takes input as SetOfPoints and set the output to 'SetOfPoints'

(c) Assign the Clusterid as zero

(d) For each and every unvisited point p in a 'SetOfPoints', mark p as visited

(e) After that the method 'getNeighbors' passes the parameters as SetOfPoints, P, Eps, the output gives the list of neighbors i.e. PListOfNeigbors.

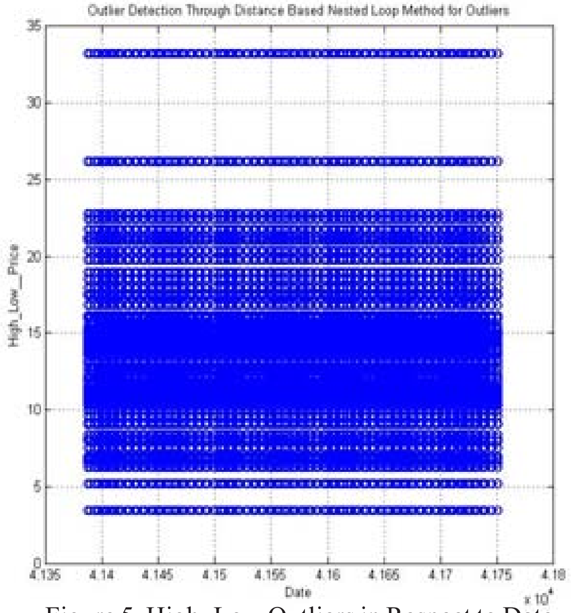


Figure 5. High_Low Outliers in Respect to Date

(f) If the size of 'PListOfNeigbors' is less than the minimum point i.e. 'MinPts', then mark 'p' as the outlier

(g) Then iterate the loop like the next cluster is assigned it to 'Clusterid'

(h) Further the method i.e. 'expandCluseter' passes the parameters as SetOfPoints, P, N, Clusterid, Eps, MinPts, so add the 'p' to the cluster having the Clusterid

```
function expandCluster(SetOfPoints, P, N, Clusterid, Eps, MinPts)
add p to cluster Clusterid
WHILE there is unvisited point p' in ListOfNeighbors
mark p' as visited
PListOfNeigbors' = getNeighbors(p', Eps)
IF PListOfNeigbors' <= MinPts
PListOfNeigbors = PListOfNeigbors joined with PListOfNeigbors
add p' to cluster Clusterid
ENDIF
ENDWHILE
RETURN
ENDEXPANDCLUSTER

function = getNeighbors (SetOfPoints, P, Eps)
RETURN Eps-Neighborhood of p in SetOfPoints as a list of points
ENDGETNEIGHBORS
```

DBSCAN has several advantages including its ability to find arbitrarily shaped clusters. It does not require the user to know the number of clusters in the data in advance. DBSCAN is very robust to outliers and requires just two parameters, Eps and MinPts. However, DBSCAN is highly affected by the distance measure used in finding the distance between two points. Its effectiveness in clustering data points depends on the distance measure used. The Euclidean distance measure is commonly used, but any

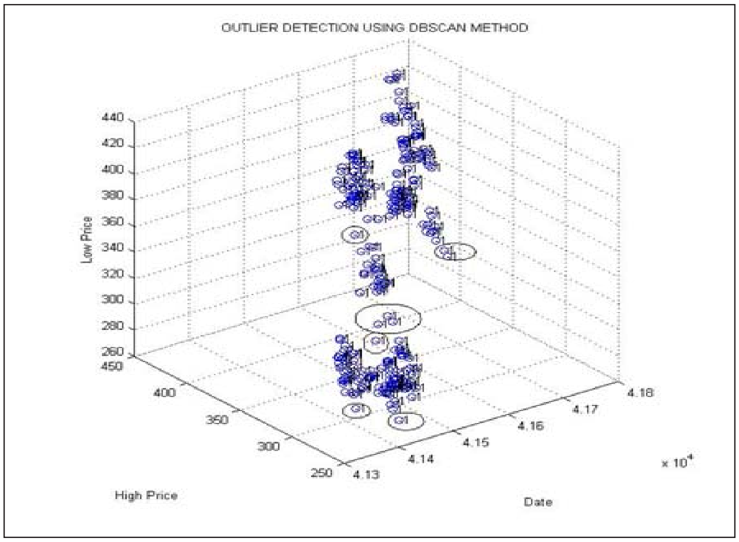


Figure 6. Outliers of high and low price in respect to date

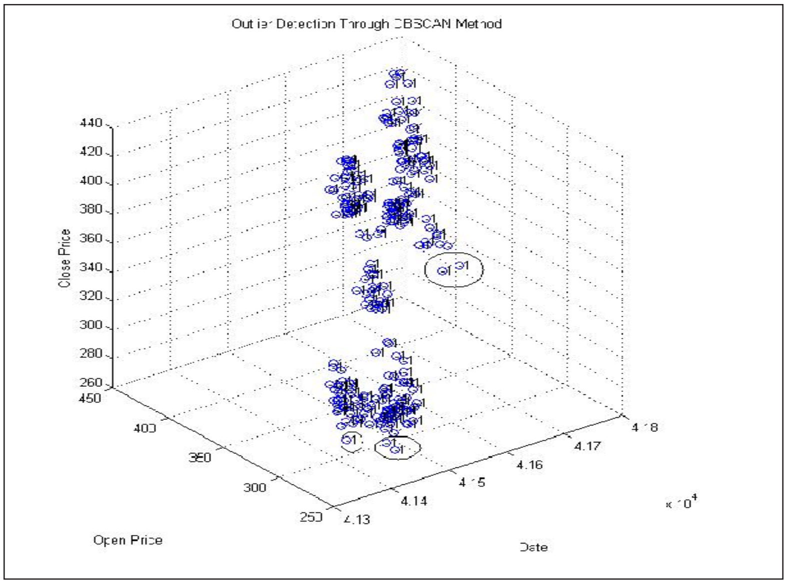


Figure 7. Outliers of open and close price in respect to date

other distance measure can be used. Also, before computing the distances between two points with different units, the data points must be normalized.

4.4 Selecting the Parameters Eps and MinPts

The DBSCAN algorithm requires two user-defined parameters *Eps* and *MinPts*. The values of these parameters have a big impact on the performance of the DBSCAN. For instance, if *Eps* is large enough, then all points form a single cluster, and no points are labeled as outliers. Likewise, if *Eps* is too small, majority of the points are labeled as outliers. There are several approaches that can be used to determine the values of *Eps* and *MinPts*.

The first approach uses the parameters specified by the experts. The parameters are provided by an expert who is very familiar with the data set to be clustered. An expert can provide the parameters and ruAn the DBSCAN algorithm, which provides graphs adjuststhe parameters, and runs the algorithm until he/she gets good results. Good results are determined by the expert showing which points from the data sets are considered to be outliers. Using visualization, an expert looks at the graphs, knowledge of the data set. An expert selects parameters that can be used as default parameters for that data set.

The k - dist approach looks at the behavior of the distance from a point to its kth nearest neighbor. If *k* is not larger than the cluster size, the value of *k* - dist is small for points that belong to the same cluster. The *k* - dist for points not in the cluster is relatively large. The idea is to pick a value of k to be the *MinPts*. The following steps are performed to find the value of *k*:

1. Compute the k - dist, (distance to its k^{th} nearest neighbor) for each of the data points.
2. Sort k - dist measures in increasing order.
3. Plot the sorted k - dist values. We expect to see a sharp change at the value of k - dist that corresponds to a suitable value of Eps

The figure above shows that the objects are encircled in the plotted graph shows the outliers of Open Price and Close Price against the corresponding dates as this has gone through the dbscan algorithm.

5. Conclusion

The nested loop (NL) algorithm avoids the explicit construction of any indexing structure and its time complexity is $O(kN^2)$.Compared to a tuple-by-tuple brute force algorithm that pays no attention to I/O's. Algorithm Nested Loop (NL) is always superior because it tries to minimize I/O's.

6. References

[1] Knorr, E. and et al. (2000). Distance-based outliers: Algorithms and applications. VLDB Journal

[2] Knorr, E., Ng, R. (1997). A unified notion of outliers Properties and computation. *In:* ACM SIGKDD.

[3] Knorr, E., Ng, R. (1999). Finding intentional knowledge of distance-based outliers. *In:* VLDB.

[4] Knorr, E., Ng, R.T. (1998). Algorithms for mining distance-based outliers in large datasets, *In:* Proc. Int'lConf. on VLDB

[5] Barnett, V., Lewis, T (1994). Outliers in Statistical Data. John Wiley.

[6] Han., Kamber. (2007). Data Mining: Concepts and Techniques Morgan Kaufmann publications

[7] Marakas, George. Data Warehousing, Data Mining and Visualisation, Pearson publications

[8] Zuriana, A. B., Rosmayati, M., Akbar, A., Mustafa, M. D.,(2006). A Comparative Study for Outlier Detection Techniques in Data Mining, CIS.

[9] Aggarwal, C. C., Yu, S. P.(2005). An effective and efficient algorithm for high-dimensional outlier detection, The VLDB Journal, .14, p. 211–221.-