# AN OVERVIEW OF OUTLIER DETECTION METHODS

[1]Mohd. Noor Md. Sap, [2]Ehsan Mohebi
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
[1]mohdnoor@fksm.utm.my, [2]saeh_hamo@yahoo.com

**Abstract:** Outlier detection is a necessary task in many safety critical environments such as aircraft engine rotation failure, factory production line, network intrusion, and bust analysis. Most proposed methods are based on distance and density based outlier detection which simply detects outliers by calculating distances or density between the data points. In high dimensional datasets, it's very difficult to find outliers with the measure of distance or density methods, because in such spaces the data become sparse and the imagination of data distribution is also hard. So with the *curse of dimensionality* we discuss a cluster based method that examines the behavior of the data in low dimensional projection. On the other hand most existing methods detect outliers with some parameters that needed to be defined by users in advance. In some cases it is very difficult for users to define these parameters. To solve such problems, example based method has been proposed. In this paper, wide and different modern technologies of outlier detection will be considered.

**Keywords:** Outlier, Statistic, Data Mining, SOM, Spatial Data.

## 1. INTRODUCTION

As given by Hawkins (1980), outlier can be defined as follows:

*"An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism."*

*Statistical* approaches were the earliest algorithms used for outlier detection, which are suited to quantitative real-valued data sets or at the very least quantitative ordinal data distributions.

The earliest outlier detection method was devised by Grubbs (1969), who calculates a *Z* value as the difference between the *mean value* for the attributes and the result value divided by the *standard deviation* for the attributes where the mean and standard deviation are calculated from all attribute values. For example, in normal distribution, outlier is an observation whose distance from the average is three times greater than variance (freedman et al, 1978). Barnett and Lewis (1994) provide a comprehensive treatment, listing about 100 abnormality tests for *normal, exponential, Poisson, and binomial distributions*.

The choice of appropriate abnormality tests depends on a number of factors: (i) the distribution, (ii) whether or not the distribution parameters, such as the mean and the variance, are known, (iii) the number of expected outliers, (iv) and the types of expected outliers such as upper or lower outliers in an ordered sample (Knorr et al, 1997).

K-Nearest Neighbor algorithm ($kNN$) for outlier detection with the complexity $O(mn^2)$, ($m$ is dimension and $n$ is the number of points), which calculates all the $k$ neighbors of each point. All $kNN$ methods use a distance calculation metric such as *Euclidean* or *Mehalanobis* distance:

$$\sqrt{(x-\mu)^T C^{-1}(x-\mu)}$$

In fact the later distance metric one is very expensive because it calculates the distance from a point to the mean $\mu$ defined by correlated attributes given by the Covariance matrix $C$. On the other hand this method is very expensive for high dimensional data set, because all the $k$ neighbors of each point should be calculated.

The same distance metric is also used for the *K-means* method that requires the users to specify the value of $k$ clusters. This method provides a local model of data. This algorithm represents each of $k$ clusters by a prototype vector with attribute values equivalent to the mean values across all points in the cluster. It updates cluster center to indicate the new instances.

In this paper, in section 2 we discuss the two popular distance and density based methods, then we will consider *example based* and *subspace based* methods to figure out the problem of defining some parameters in advance by users and for the curse of dimensionality. In section 3, we will compare the neural network approach in the case of SOM outlier detection methods. Spatial data and spatial outlier detection techniques will be discuss in section 4. The goal of this paper is to review the strength, weaknesses and improvements of major proposed outlier detection methods.

## 2. DATA MININING OUTLIER DETECION APPROACHES

### 2.1 Distance Based

The distance based methods (DB-outlier detection) is one of the simplest and most widely used approaches as it only calculates the distances between data points. The earliest approach introduced by Knorr and Ng (1998) is defined as:

*An object $x$ in a data set $T$ is a $DB(p, D)$-outlier if at least $p$ of objects in $T$ lie greater than distance $D$ from $x$ (see figure 1).*

They described an efficient kNN that does not calculate all neighbors of $k$, only $p < k$ neighbors will be determined so it's not sensitive to computational growth. Their definition means that if there are less than $p$ neighbors inside the distance threshold $D$ then the instance is an outlier. The problem is finding appropriate $D$ and $p$ such that outliers would be correctly detected with a small number of false detections (Hautamaki et al, 2004). Thus this process requires users to define optimal parameters ($D$ and $p$) in advance to have an acceptable performance.

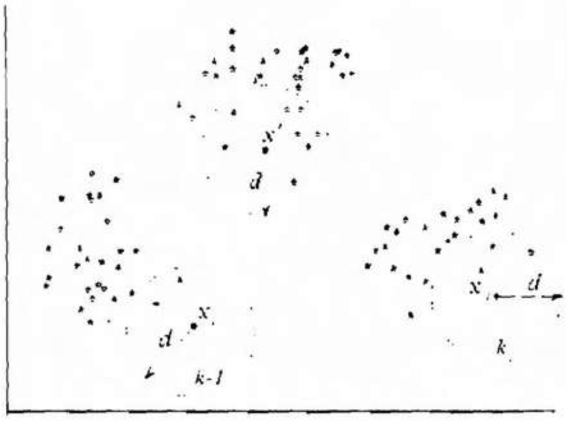

**Figure 1 Outliers that defined by Knorr and Ng**

Ramaswamy (2000) introduced an optimized algorithm, producing a list of potential outliers ($DB_n^k$). The definition of $DB_n^k$ outlier is ($k$ is user defined):

*Outliers are the top n data elements whose $D_k$ (distance to the $k^{th}$ neighbor), are greatest.*

By this definition $n$-largest k-nearest neighbor's distances in data set consider as outliers. This algorithm first partitions the input data set into disjoint subsets, and then prunes entire partitions as soon as it can be determined that they cannot contain outliers. Since people are usually interested in only a small number of outliers, our algorithm is able to determine very quickly that a significant number of the input points cannot be outliers. Furthermore, it outperforms the nested-loop and index-based algorithms by more than an order of magnitude for a wide range of parameter settings. But its complexity is not good for computational growth, because all k-nearest neighbors must be calculated. This optimized method applies kNN to partitioned data *cells*, Ramaswamay Used *Nested loop*, *Index based* and *partition* *based* algorithm to identify outliers. The result of the time complexity for these three algorithms has been compared (see figure 2).

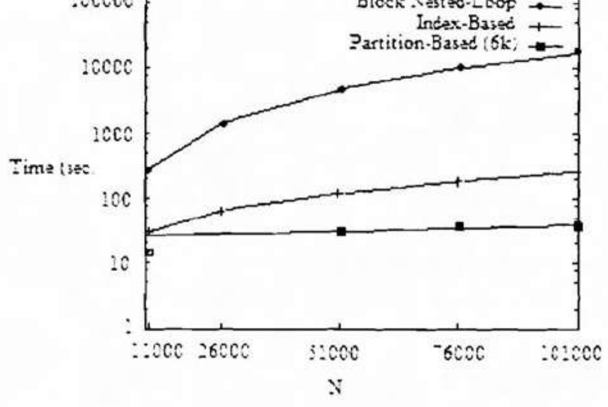

**Figure 2 Performance Results for N (Ramaswamy, 2000)**

As conclusion, a drawback of the method proposed by Ramaswamay is that user has to know in advance how many outliers there are in the data set (Hautamaki et al, 2004), because in some data sets only one outlier's distance to $k^{th}$ neighbor is so large which is clearly in sparse space and detected as outlier.

Bay and Schwabacher (2003) proposed an approach that can detect $DB_n^k$ outliers in near linear running time with the data set size. Indeed, this method is an optimized version of the nested loop algorithm by making use of the technique of *randomization* and a simple *pruning rule*. The data set is randomized and divided into small blocks, and the blocks are handled one by one. For the first block, each object is compared with every object in the whole data set in order to compute its score (which is the distance to its $k^{th}$ nearest neighbor). According to these scores, the top $n$ outliers in the first block can be decided, and the score of the $n^{th}$ outlier is used as a cut-off for the second block. As more blocks have been processed, more extreme outliers can be found and a larger cut-off can be used for the next block.

As a result, pruning becomes more efficient after each iteration. But the procedure of randomizing the whole data set is important for this method. The performance can be very poor if the data set is *sorted* or the objects clustered together in space also appear together in the data set file.

In fact the main shortcoming is that this method needs to scan the whole data set $m$ times, where $m$ is the number of blocks. When the whole data set cannot fit in the main memory, expensive disk scans could result in very poor performance. Even though the worse case complexity is still $O(dN^2)$, the experimental results show that this method can achieve near linear running time. One of the shortcomings of Knorr et al proposition is that it cannot achieve good performance with very large datasets and high dimensional datasets.

To overcome such disadvantage, D.Ren et al (2004) improved knorr's method by introducing the definition of processing *vertical structure* instead of traditional horizontal structure. The definition of neighborhood of a data point $O$ with the radius $r$ is defined as following, where $X$ is the dataset:

$$Nbr(O,r) = \{x \in X \,|\, |O - x| \leq r\},$$

And the definition of outliers is:

$$Ols(X,p,D) = \{x \in X \,|\, N(Nbr(x,D)) \leq (1-p) \times |X|\},$$

They proposed a vertical "by-neighbor" outlier detection method with local pruning (PODMP)[1], which can detect outliers efficiently and scale well in large datasets. The vertical method works as follows. First, the dataset to be mined is represented as the set of P-Trees. Secondly, one point $O$ in the dataset is selected arbitrarily; then, the $D$-neighbors are searched using the fast computation of inequality P-Tree, and the $D$-neighbors are represented with an inequality P-Tree, which is called a neighborhood P-Tree. In the neighborhood P-Tree, '1' means the point is a neighbor of the point $O$, while '0' means the point is not a neighbor. Thirdly, the number of points in $D$-neighbors is calculated efficiently by extracting values from the root node of the neighbor P-Tree (Q.Ding et al, 2002). They compared the time consuming of their method with nested loop (NL) as following (see figure 3)

[1] P-Tree-based outlier detection method using pruning

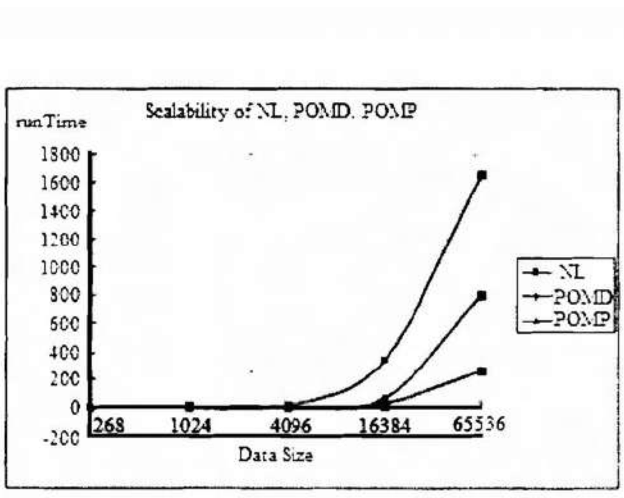

**Figure 3 Comparison of Scalability of NL, PODM, and PODMP (Bay and Schwabacher , 2003)**

In fact, as conclusion both the definitions of $DB(p, D)$ and $DB_n^k$ can only capture global outliers, because these definitions take a global view of the data set. For a data set with simple structure, for example, one that contains one or more clusters with similar density, these two definitions work well. However, for many real world data sets which have complex structure, the methods based on these two definitions might not be able to find interesting outliers.

### 2.2 Density Based

To achieve better result for finding interesting outliers and overcome some of the shortcomings of distance based method (capture global outliers, etc...), density based method has been proposed. M. Breunig et al (2000) introduced the concept of local density outliers and a measure LOF (Local Outlier Factor), which captures the degree of outlier-ness of every object in the data set, to pick up local outliers.

Aggrawal and Yu (2001), use a lower dimensional projection of data set, they focus on key attributes. They used an evolutionary search algorithm and The *Brute-force* algorithm which examines all $k$-dim projections and retain the $m$ projection which have the most negative sparsity coefficient, Then uses the searching algorithm to find the outliers. In this proposed method all points within the same cell are regarded as normal objects or outliers. Therefore, this method has a drawback that sometimes normal objects may be detected as outliers, and vice versa.

G. Kollios et al (2003) proposed a density based biased sampling method to detect $DB(k, D)$ outliers. A *kernel density estimator* is built using randomly sampled points to approximately represent the density of the data set. The estimator can be used to estimate the probability that each data point belongs to the data set. For each object, the function $N_T(x, D)$ is defined to be the number of objects whose distance is at most $D$ from the object x in the data set $T$. the definition of outliers is as following:

*An object $x \in T$ is a $DB(k, D)$-outlier only if $N_T(x, D) \leq k$.*

The proposed algorithm takes one pass over the data set to compute the density estimator function, and the complexity of this step is $O(dN)$. Since each object $x$ in the data set has to be read once in order to compute the value of $N_T(x, D)$, one full data set scan is needed. The complexity of this step is $O(\beta dN)$, where $\beta$ is the number of samples for constructing the density estimator.

One drawback of this method is that a large number of $\beta$ will improve the accuracy but increase the running time complexity. In fact how good a kernel density estimator can work in high-dimensional space has not been fully explored but it seems to be less accurate.

Brito et al (1997) proposed a *Mutual k-Nearest Neighbor* (MkNN) graph based approach. MkNN graph is a graph where an edge exists between vectors $v_i$ and $v_j$ if they both belong to each other's $k$-neighborhood. MkNN graph is undirected and is a special case of $k$-Nearest Neighbor (kNN) graph, in which every node has pointers to its $k$-nearest neighbors. Each connected component is considered as a cluster if, it contains more than one vector and an outlier when connected component contains only one vector. Potential problem with Berito's definition is that, an outlier that is too close to an inlier could be misclassified (Hautamaki et al, 2004).

To have a good performance and improve Berito's method, Hautamaki et al (2004) proposed an outlier detection method using *In-degree Number* (ODIN) algorithm that utilizes $k$-nearest neighbor graph. In this method the definition of outlier is:

*Given kNN graph $G$ for data set $S$, outlier is a vertex, whose in-degree is less than equal to threshold $T$.*

Where $T$ is a different variant of Ramaswamay's definition, i.e. it measured from *maximum the kNN distances* ($L_i$ , $i = i^{th}$ *vector*) as following:

$T = \max(L_i - L_{i-1}) \times t \qquad 0 < t < 1,$

Experimental results show that ODIN gives a good performance and produces less error rate in synthetic data sets to comparison with Berito and Ramaswamay's methodology.

### 2.3 Example Based

Most existing methods detect outliers with parameters needed to be decided by users in advance. Such parameters always contain some hidden user view of outliers. Actually, it is difficult for users to determine these parameters beforehand. Users are often experts in their problem domains, so they know what kind of objects they want to find. Therefore, users could provide some outlier examples in hand to find more objects that have similar "outlier-ness" characteristics to these examples (Yuan Li et al, 2007).

Yuan Li et al 2007 propose a new method which makes a combination of *Subspace Based* and *Example Based* algorithms to detect DB-Outliers in high dimensional datasets.
There is a problem in searching the most suitable subspace where user examples are outstanding greatly. Given a $\vec{d}$-dimensional dataset, if we examine all subspaces whose dimensionalities vary from 1 to $\vec{d}$, it takes a lot of time. They detect the most suitable subspace where user examples are isolated more significantly than in any other subspaces with a Genetic Algorithm.This method will also figure out the shortcomings of Aggrawal and Yu's method.

### 2.4 Subspace Based

Clustering algorithms like ROCK (Guha et al, 1999), DBSCAN (Ester et al, 1996) can also handle outliers, but their main goal is to find clusters. Clustering outliers, are often regarded as *noise*, because these algorithms are optimized for producing meaningful clusters, and this prevents them from having beneficent results in clustering (Z.He et al, 2002).
Knorr and Ng (1999) have proposed algorithms to identify $DB(f,D)$ outliers in attribute subspaces instead of the full attribute space of a given data set. The outliers in a specific subspace were classified into two categories: *non-trivial* and *trivial* outliers. The definition of outlier is as following:

*Suppose that $P$ is a $DB(f,D)$ outlier in the attribute space $\mathfrak{R}$. $P$ is a non-trivial outlier if $P$ is not a $DB(f,D)$ outlier in any subspace of $\mathfrak{R}$, otherwise, $P$ is a trivial outlier.*

Two algorithms, called *Up Lattice* and *Jump Lattice* were proposed. Up Lattice works in a bottom-up, level-wise fashion. This algorithm searches for outliers in 1-$D$ spaces first, then in 2-$D$ spaces, and so on. The standard nested-loop or cell-based algorithms are used to search outliers in a specific attribute space depending on its dimensionality. It is quite normal that we cannot find any outliers in the low dimensional spaces. In this situation, the bottom-up strategy wastes a lot of time before reaching a space containing outliers.

In order to overcome this drawback, the *Jump Lattice* algorithm starts searching outliers in an attribute space $\Re$ from a specific level. If any outlier is found, a drill-down procedure is called to search outliers in the subspaces of $\Re$.

Aggarwal and Yu (2005) presented a density based definition for subspace outliers based on the concept of abnormal lower-dimensional projection. The definition of outliers introduced as follows:

*An object is considered to be an outlier if in some lower dimensional projection it presents in a local region of abnormal low density.*

The first step is to find the cubes with abnormally low density, and then drill down to the details to pick up all objects that are located in those cubes and define the density of the cubes. Second each attribute is divided into $\phi$ ranges, and each range contains a fraction $f = 1/\phi$ of the objects. The presence or absence of an object in a $\acute{d}$-dimensional cube is a Bernoulli random variable with probability $f^{\acute{d}}$ if the data is uniformly distributed.

The third issue of detecting the density based subspace outliers is how to choose the projection parameters $\acute{d}$ and $\phi$. Experimental results showed that the values of $\acute{d}$ and $\phi$ should be picked small enough so that sparsity coefficient of cube containing exactly one object is reasonably negative.

## 3. NEURAL NETWORK OUTLIER DETECTION APPROACHES

### 3.1 Introduction

Outlier detection is an important problem in pattern recognition and neural network research. We know that usually outliers are considered in statistics. Box and Whisker plots and scatter are very sensitive to outlying data. In comparison to standard density estimator, SOMs are relatively fast and in expensive when dimensionality of data is large.

Japkowicz (1995) presented novelty detection by using an auto-associative neural network with one hidden layer. This method is trained with a training set consisting of positive instances only on the concept of back propagation, after training it results in a good performance on positive instances but poorly on reconstructing subsequent negative instances. This method computes *error* (absolute) by the sum of each iterations absolute error.

Crook and Hayes (1995) introduced novelty detection which is based on calculating the *energy* of a Hopfield network. It is useful in on-line learning but this model is optimized and could classify more patterns than a Hopfield network could recognizes. Each pattern is $N$ bits long, to store $P$ patterns in the network, network weights $w_{ij}$ will be computed. The energy $(E)$ of network will be compared with a threshold in order to find outliers.

### 3.2 SOM Outlier Detection

Generally this network is known as *self-organizing map* (SOM) (Kohonen 2001), is one of the best neural networks. The SOM represents interesting low dimensional projections of high-dimensional data which is based on unsupervised learning to map nonlinear statistical relationships between high dimensional input data into two-dimensional lattice. This mapping retains the relationship between input data as faithfully as possible, thus a topology-preserving representation of input similarities in the term of distances in the output space.

The SOM neural network is made up by two main procedures: the *training* process to the input pattern and the clustering process. In the training process, it continuously matches the input pattern with the *connection weights* $(m_i)$ and finds the competition winner which is more match with the input pattern by measuring the Euclidean distance $d(x, m_i)$, then it auto adjust the value of connection weight according to the input pattern by the following equation:

$$m_i(t+1) = m_i(t) + h_{c(x),i}\big(x(t) - m_i(t)\big) \qquad h_{c(x),i} = neighborhood\,function$$

The SOM establishes the connection weight matrix to map the input pattern and accumulates knowledge. At the end, it uses the connection weight matrix to classify the new input pattern according to learning mode (Yan and Yaoguang, 2005).

Yan and Yaoguang (2005) used a new kernel function and replace the Euclidian distance in the SOM neural network to predict the silicon content of molten iron. In fact the distance between output $i$ and input pattern $x_k$ expressed by:

$$K(x_k, M_i) = \log(1 + \|x_k - M_i\|^2/\sigma^2)$$

which achieved the prediction probability of 93%.

The result of SOM algorithm is a set of topologically ordered codebook vectors related to each input data. These codebooks distances are small in the low dimensional output lattice if the related inputs are close to each other in the input space.

In the aim of clustering a unit (neuron) is assigned to the cluster, whose reprehensive centre $x_i$ in the input space is nearest to the unit (The cluster centers are initialized in a random order). Outlying neurons can be found by looking at the median distance from each centre to its adjacent neighbors and analyzing the position on the map of neurons with the largest values of this distance.

In the comparison of SOM and $K$-means, SOM works reasonably well and could visualize multivariate outliers. The $K$-means algorithm largely depends on not only number but also the placement of initial codebooks and it's prone to get stuck in local minima.

To have a good result with the SOM, the *learning rate* and *neighborhood function* should be chosen properly according to the input data. That's one of the limitations about the SOM that has been suggested. Predetermining a good layout for SOM is also difficult. Sometimes the predetermined size of network is too big or too small, in either case the resulting map will be of poor quality. To solve such problems GSOM[2] (Zhou and Yan Fu 2005) has been proposed.

The GSOM is based on a 2-Dimensional map but the nodes are connected in a triangular way. Every node $c$ has an n-dimension model vector $pos(c)$ indicating its virtual position in $R^n$. For each input vector $x$ the nearest node is determined. The winning neuron satisfies the following relation:

$$\|pos(winner) - x\| \leq \|pos(c) - x\|$$

In this scheme of SOM network, some nodes will be added or removed from the network by the means of an *error value*. There is an error value related to each node. If the value of error

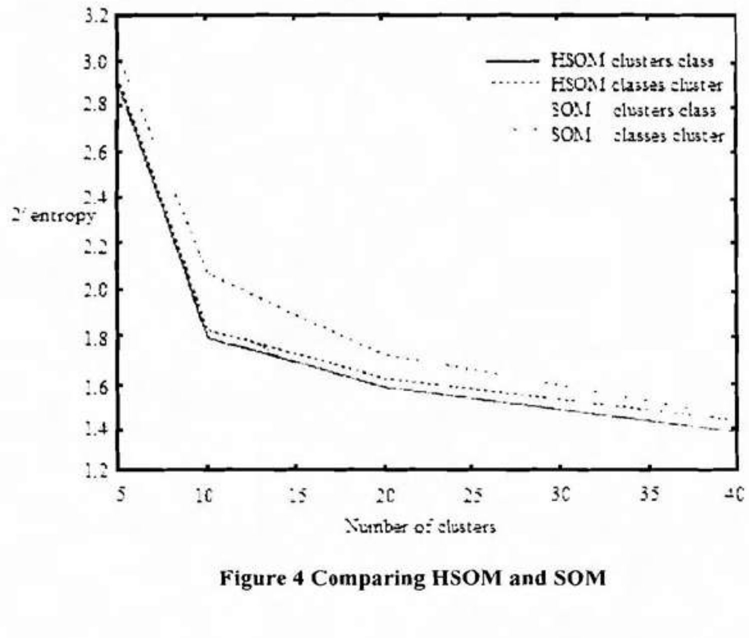

**Figure 4 Comparing HSOM and SOM**

is high then it means the density beside that node is very low, where a new node will be added to network. After $k_{remove}$ input is vector appended to the network if some nodes never fire or become winners then these nodes will be removed.

The SOM does the *vector quantization* which becomes optimized and more powerful by HSOM[3] (Lampinen and Erkki 1992). In HSOM the output of first SOM are fed into another SOM as input, causing SOM to divide into a distinct cluster representation. Luttrell (1989) showed that hierarchical maps minimize the *decoding square error* if the training neighborhoods in SOM equal the probability distribution of errors in the codes. Figure 4 shows a simple comparison between SOM and HSOM as following.

## 4. SPATIAL OUTLIER DETECTION APPROACHES

Spatial outliers are spatial objects whose non-spatial attribute values are significantly different from the value of their neighborhoods. Spatial outlier detection methods in the literature of spatial statistics can be grouped into two categories, *graphical approach* and *quantitative tests* (Chang et.al 2003).

### 4.1 Graphical Approach

In graph based spatial outlier detection the main idea is based on graph connectivity (Shekhar et al. 2002). For spatial outlier detection methods, the choice of statistics is important and depends on what kind of data is considered. The statistic that Shekhar et al. used is $S(x) = [f(x) - E_{y\in N(x)}(f(y))]$, where $f(x)$ is the attribute function, $N(x)$ is the fixed set of neighbors of $x$ and $E_{y\in N(x)}(f(y))$ is the average attribute value for neighbors of $x$. In fact $S(x)$ denotes the difference of the attribute value of each node and the average of each neighbor. Detection of outliers can be considered as $|(S(x)-\mu_s)/\sigma_s| > \theta$. $\mu_s$ and $\sigma_s$ are the mean and standard deviation of all $S(x)$.

The most costly part of the algorithm is to *find neighbor nodes set*. The I/O cost of finding neighbor nodes set is determined by the connectivity residue ratio (CRR), i.e. how the nodes are grouped into disk pages. If the node and its entire neighbor nodes can be reside in the same disk page, there will be no redundant I/O operation required.

[2] Growing Self Organizing Map
[3] Hierarchical Self Organizing Map

## 4.2 Quantitative Tests

Chang et.al (2003), proposed two iterative algorithms that detect outlier by multi iterations and also employ a non-iterative algorithm which uses median as the neighborhood function namely, *iterative r algorithm*, *iterative z algorithm* and *median algorithm* respectively.

The first and second algorithm compute the $k$ nearest neighbors set $NN_k(x_i)$ for each spatial point $x_i$ and a neighborhood function $g(x_i)$ which is the average attribute values of $KNN$ of $x_i$ . Consider both algorithms, to detect the spatial outliers, the attribute value of each point $x_i$ (attribute function $f : X \longrightarrow R$) will be compared to those attribute values of its neighborhoods by a comparison function $h(x_i)$. Then a point $x_i$ is an outlier if $y_i$ is the maximum value of the set $\{y_1, y_2, ..., y_n\}$ , where $y_i = h(x_i)$.

This means that $x_i$ is an outlier if compare to threshold $\theta, |y_i|$ is large enough. Once an outlier is detected, some corrections are made immediately, such as replacing the attribute value of outliers by the average of its neighbors to avoid normal points labeled as outlier candidates.

In the third algorithm (median), instead of the average value, $g(x_i)$ is the median (in the ordered data set $\{x_1, x_2, ..., x_{2m+1}\}$ the median is $x_{m+1}$) of the attribute values in the data set: $\{f(x) : x \in NN_k(x_i)\}$

All the three proposed algorithms will detect true outliers more efficient than the $z$ algorithm (Shekhar et al. 2001), *Scatter plot* (Luc 1994) and *Moran Scatter plot* algorithm (Luc 1995).

The method introduced by Zhan et.al (2004), introduced a set of multi-attributive and multi-dimensional spatial objects ($s_{ii}$ in a matrix $j \times j$) each with $J$ attributes corresponding in a two-dimensional matrix $R$, could accurately detect spatial outliers after the attributed correlations $F'(s_i)$ was calculated by $R \times F(s_i)$, with the attribute function $F : s_i \longrightarrow R$. This method also employs an *important values* attribute set $p$ ($0 < p_i \leq 9$ for $i = 1,2,..., j$) which is the importance degree of each attribute in the data set $\{r_1, r_2, ..., r_n\}$.

Consider object $s_i$, assuming the $k$ spatial objects in neighborhood of $s_i$, In order to compute the distribution value of neighborhoods connecting with $s_i$ , an aggregate function of attribute correlations is proposed.

$$F'_{aggr}(s_i) = \left( \sum_{ii=0}^{k} R \times F(s_i) \right) / k$$

The estimation of multi attributive set $V$ :

$$V(s_i) = P \times (F'(s_i) - F'_{aggr}(s_i))$$

According to the theory of multi dimensional distribution of random function if $\mu$ and $\sigma$ are the sample mean and variance of the set $V$, then the set $Y = \{Y(s_i) = |(V(s_i) - \mu/(\sigma))|\}$ is the standard value of each $V(s_i)$. To detect the outliers we conclude that $s_i$ is an extreme value in the original data set if $V(s_i)$ has an extreme in value the standard data set, as before it should be compared to the threshold $\theta$.

To gain better result in *complexity* of computation of this algorithm an auxiliary secondary index (the dynamic index R-tree structure) is used to support the query operation. The experimental test shows that the algorithm detects true outliers more efficiently than the $z$ and the median algorithm.

Hung et al. (2005) introduced new spatial density based outlier detection method with a stochastically searching algorithm, named SODSS. This method reduced many neighborhood queries. It does not scan the data base one by one to find the neighborhood of each spatial point like DBSCAN. In fact the algorithm divides the data set into three segments or labeled data sets, cluster set, candidate set and outlier.

Unlike DBSCAN and GDBSCAN, once the algorithm has labeled the neighbors as a part of a cluster, it will not examine each neighborhood for each of those neighbors. Neighborhood query could be computed in $O(\log n)$ using $R^* - tree$ data structure. With the new approach the complexity of computation decreases from $O(n^2)$ to $O(k \log n)$, where $k$ is related to the threshold or maximum numbers of neighbors and it is much smaller than $n$.

## 5. DISCUSSION AND CONCLUSION

We have discussed several modern methods of outlier detection in this paper. Table 1 summarized our discussion on the weaknesses and strengths of the mentioned outlier detection methods.

We conclude from our review of existing outlier detection methods and clustering methods that they all suffer from the following disadvantages:

- Depending on pre-specified values for some parameters and they should be improved by new methods
- Overcoming the curse of high dimensionality and doing well in large data sets.

- In some mapping methodologies there is a curse of neighborhood preservation of projected input data. This means that the projected neighborhood of an input data $x_i$, may be projected far from $x_i$ in the output projection area.

**Table 1 Overview of discussed outlier detection methods**

| Technique | Strength | Weakness | Improvement |
|---|---|---|---|
| $DB(p, D)$ (knorr and Ng 1998) | efficient because it does not calculate all neighbors of $k$ | finding appropriate $D$ and $p$ such that outliers would be correctly and cannot achieve good performance with very large datasets | PODMP algorithm (D.Ren et al 2004) And Subspace Based algorithm (knorr and Ng 1999) |
| $DB_n^k$ a potential list of outliers (Ramaswamy 2000) | it outperforms the nested-loop and index-based algorithms by more than an order of magnitude for a wide range of parameter settings | user has to know in advance how many outliers there are in the data set | ODIN algorithm (Hautamaki et al 2004) And $DB_n^k$ technique of *randomization* and a simple *pruning rule* (Bay and Schwabacher 2003) |
| **Brute-force algorithm** (Aggrawal and Yu 2001) | lower dimensional projection of data set which is efficient in high dimensional data sets | sometimes normal objects may be detected as outliers, and vice versa | combination of Subspace Based and Example Based algorithms (Yuan Li et al 2007) |
| **$K$-Means** | It converges very fast and has good quality in comparison to other statistical methods | Like the hill-climbing methods it may stuck in local minima, it's result depends on not only number of initialized code book but also the placement of the initial code books | $K$-Means Hierarchical and HSOM |
| SOM (Kohonen 2001) | In multivariate data set is very fast and inexpensive And It not Model-based which is easy to interpret | The best result is gained when the layout of network is properly choose | GSOM (Zhou and Yan Fu 2005) and HSOM (Lampinen and Erkki 1992) |

- Considering the SOM neural network, the dimension or size of the network should be properly chosen according to the input data size.
- There are some parameters in the SOM neural network outlier detection method that if initialized well then the quality of the identification of outliers would be more accurate in comparison to existing methods.

As conclusion, the best application to detect multivariate outliers in large data sets would be chosen from the known neural network outlier detection methodologies, which are fast and inexpensive as compared to statistical and data mining outlier detection approaches. The introduction of a new approach which comes from the combination of statistical and neural network methods could also result in a good outlier detection methodology.

## AKNOWLEDGMNETS

This research was supported by the Research Management Center of Universiti Technologi Malaysia (UTM) and the Malaysian Ministry of science, Technology and Innovation (MOSTI) under vote number 79224.

## REFERENCES

Aggarwal, C. C. & Yu, P. S. (2001). Outlier Detection for High Dimensional Data. Proceedings of the ACM SIGMOD Conference 2001.

Aggarwal, C. C. and Yu, P. S., "An effective and efficient algorithm for high-dimensional outlier detection." VLDB J., Vol. 14, No. 2, 2005, pp. 211–221.

Barnett, V. & Lewis, T. (1994). Outliers in Statistical Data, 3rd edn. John Wiley & Sons.

Bay, S. D. and Schwabacher, M., "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," Proceedings of Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, D.C. USA, 2003, pp. 29–38.

Bradley, P. S., Fayyad, U. M. & Mangasarian, O. L. (1999). Mathematical Programming for Data Mining: Formulations and Challenges. INFORMS Journal on Computing 11(3): 217–238.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J., "Lof: Identifying density-based local outliers," Proceedings of the 2000 ACM SIGMOD International Conference on Management Data, Dallas, Texas, USA, ACM, 2000, pp. 93–104.

Brito, E. L. Chavez, A. J. Quiroz, and J. E. Yukich. Connectivity of the mutual -nearest-neighbor graph in clustering and outlier detection. Statistics & Probability Letters, 35(1):33–42, August 1997.

Bishop, C. M. (1994). Novelty detection & Neural Network validation. Proceedings of the IEE Conference on Vision, Image and Signal Processing, 217–222.

Crook, P. & Hayes, G. (1995). A Robot Implementation of a Biologically Inspired Method for Novelty Detection. Proceedings of TIMR-2001, Towards Intelligent Mobile Robots Manchester.

Chang-Lu, D.Cheng, and Y.Kou. (2003), Algorithms for Spatial Outlier Detection. Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03) pp. 597–600.

Datta, P. & Kibler, D. (1995). Learning prototypical concept descriptions. Proceedings of the 12th International Conference on Machine Learning, 158–166, Morgan Kaufmann.

Ester, M., Kriegel, H. -P. & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, 226–231. AAAI Press.

Freedman, R. Purves, and R. Pisani. Statistics. W.W. Norton, New York, 1978.

Guha, R. Rastogi and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes". In Proceedings of the 15th International Conference on Data Engineering, page 512, March 1999.

Grubbs, F. E. (1969). Procedures for detecting outlying observations,Technometrics,11, 1–21.

Han and M. Kamber, "Data Mining: Concepts and Techniques". The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, 550 pages, August 2000.

Hautamaki, Ismo Karkkainen and Pasi Franti (2004). Outlier Detection Using k-Nearest Neighbour Graph. Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04).

Hawkins (1980). Identification of outliers. Chapman and Hall, London. 1980.

Huang, X.Qin, C.Chen, and Q.Wang.(2005), Density Based Spatial Outlier Detecting. Springer-Verlag Berlin Heidelberg, ICCS 2005, LNCS 3514, pp. 979 – 986.

Kohonen, T. Self-Organizing Maps. Springer-Verlag Berlin and Heidelberg, Germany (2001)

Knorr, Edwin M. and Raymond T. Ng (1997). A Unified Notion ofOutliers: Properties and Computation. 3rd International Conference on Knowledge Discovery and Data Mining Proceedings, 1997, pp. 219-222.

Knorr, E. M. & Ng, R. T. (1998). Algorithms for Mining Distance-Based Outliers in Large Datasets. Proceedings of the VLDB Conference, 392–403, New York, USA.

Knorr, E. M. and Ng, R. T., Finding intensional knowledge of distance-based outliers, Proceedings of 25th International Conference on Very Large Data Bases, Morgan Kaufmann, 1999, pp. 211–222.

Japkowicz, N., Myers, C. & Gluck M. A. (1995). A Novelty Detection Approach to Classification. Proceedings of the 14th International Conference on Artificial Intelligence (IJCAI-95), 518–523.

Lampinen and Erkki Oja.(1992). Clustering Properties of Hierarchical Self-Organizing Maps. Journal of Mathematical Imaging and Vision 2, 261-272 (1992).

Luc.A. Exploratory Spatial Data Analysis and Geographic Information Systems. In M. Painho, editor, New Tools for Spatial Analysis, pages 45–54, 1994.

Luc.A. Local Indicators of Spatial Association: LISA.Geographical Analysis, 27(2):93–115, 1995.

Ng, R. T. & Han, J. (1994). Efficient and Effective Clustering Methods for Spatial Data Mining. Proceedings of the 20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago, Chile, 144–155. Morgan Kaufmann Publishers.

Q. Ding, M. Khan, A. Roy, and W. Perrizo, The P-tree algebra. Proceedings of the ACM SAC, Symposium on Applied Computing, 2002.

Ramaswamy, S., Rastogi, R. & Shim, K. (2000). Efficient Algorithms for Mining Outliers from Large Data Sets. Proceedings of the ACM SIGMOD Conference on Management of Data, Dallas, TX, 427–438.

Ren, Imad Rahal, William Perrizo (2004). A Vertical Distance-based Outlier Detection Method with Local Pruning., 2004, Washington, DC, USA. Copyright 2004 ACM, CIKM'04 November 8-13.

Skalak, D. B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. Machine Learning: Proceedings of the Eleventh International Conference, 293–301.

Shekhar, Ch.T Lu, and P.Zhang. (2002). Detecting Graph-based Spatial Outliers. Intelligent Data Analysis: An International Journal, 6(5):451–468.

Shekhar, C.-T. Lu, and P. Zhang.(2001). Detecting Graph-Based Spatial Outlier: Algorithms and Applications (A Summaryof Results). In Proc. of the Seventh ACM-SIGKDD Int'l Conference on Knowledge Discovery and Data Mining, Aug 2001.

Tang, J., Chen, Z., Fu, A. & Cheung, D. (2002). A Robust Outlier Detection Scheme in Large Data Sets, 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Taipei, Taiwan, May, 2002.

Zhan-Quan Wang, Sh.Wang, T.Hong, and X.Wan, (2004). A Spatial Outlier Detection Algorithm Based Multi-Attributive Correlation. Proceedngs of the Thxd hternahond Conference on Macbme Le- and Cybernehcs, Shanghs, pp. 26-29.

Zhou and Yan Fu, (2005). Clustering High-Dimensional Data Using Growing SOM*. Springer-Verlag Berlin Heidelberg. ISNN 2005, LNCS 3497, pp. 63–68, 2005.

Yan and Yaoguang,(2005). Research and application of SOM neural network which based on kernel function.

Yuan Li and Hiroyuki Kitagawa, (2007). DB-Outlier Detection by Example in High Dimensional Datasets. Proc. Proc. 3rd IEEE International Workshop on Databases for Next-Generation Researchers (SWOD), pp.73-78, 2007.