OPTICS-OF: Identifying Local Outliers

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng[1], Jörg Sander
Institute for Computer Science, University of Munich
Oettingenstr. 67, D-80538 Munich, Germany
phone: +49-89-2178-2225
fax: +49-89-2178-2192
{breunig, kriegel, ng, sander}@dbs.informatik.uni-muenchen.de

Abstract: For many KDD applications finding the outliers, i.e. the rare events, is more interesting and useful than finding the common cases, e.g. detecting criminal activities in E-commerce. Being an outlier, however, is not just a binary property. Instead, it is a property that applies to a certain *degree* to each object in a data set, depending on how 'isolated' this object is, with respect to the surrounding clustering structure. In this paper, we formally introduce a new notion of outliers which bases outlier detection on the same theoretical foundation as density-based cluster analysis. Our notion of an outlier is 'local' in the sense that the outlier-degree of an object is determined by taking into account the clustering structure in a bounded neighborhood of the object. We demonstrate that this notion of an outlier is more appropriate for detecting different types of outliers than previous approaches, and we also present an algorithm for finding them. Furthermore, we show that by combining the outlier detection with a density-based method to analyze the clustering structure, we can get the outliers almost for free if we already want to perform a cluster analysis on a data set.

1 Introduction

Larger and larger amounts of data are collected and stored in databases, increasing the need for efficient and effective analysis methods to make use of the information contained implicitly in the data. *Knowledge discovery in databases* (KDD) has been defined as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [9]. Corresponding to the kind of patterns to be discovered, several KDD tasks can be distinguished.

Most research in KDD and data mining is concerned with identifying patterns that apply to a large percentage of objects in a data set. For example, the goal of clustering is to identify a set of categories or clusters that describes the structure of the whole data set. The goal of classification is to find a function that maps each data object into one of several given classes. On the other hand, there is another important KDD task applying only to very few objects deviating from the majority of the objects in a data set. Finding exceptions and outliers has not yet received much attention in the KDD area (cf. section 2). However, for applications such as detecting criminal activities of various kinds (e.g. in electronic commerce), finding rare events, deviations from the majority, or exceptional cases may be more interesting and useful than the common cases.

1. On sabbatical from: Dept. of CS, University of British Columbia, Vancouver, Canada, rng@cs.ubc.ca.

Outliers and clusters in a data set are closely related: outliers are objects deviating from the major distribution of the data set; in other words: being an outlier means not being in or close to a cluster. However, being an outlier is not just a binary property. Instead, it is a property that applies to a certain *degree* to each object, depending on how 'isolated' the object is. Formalizing this intuition leads to a new notion of outliers which is 'local' in the sense that the outlier-degree of an object takes into account the clustering structure in a bounded neighborhood of the object. Thus, our notion of outliers is strongly connected to the notion of the density-based clustering structure of a data set. We show that both the cluster-analysis method OPTICS ("*O*rdering *P*oints *T*o *I*dentify the *C*lustering *S*tructure"), which has been proposed recently [1], as well as our new approach to outlier detection, called OPTICS-OF ("OPTICS with *O*utlier *F*actors"), are based on a common theoretical foundation.

The paper is organized as follows. In section 2, we will review related work. In section 3, we show that global definitions of outliers are inadequate for finding all points that we wish to consider as outliers. This observation leads to a formal and novel definition of outliers in section 4. In section 5, we give an extensive example illustrating the notion of local outliers. We propose an algorithm to mine these outliers in section 6 including a comprehensive discussion of performance issues. Conclusions and future work are given in section 7.

2 Related Work

Most of the previous studies on outlier detection were conducted in the field of statistics. These studies can be broadly classified into two categories. The first category is *distribution-based*, where a standard distribution (e.g. Normal, Poisson, etc.) is used to fit the data best. Outliers are defined based on the distribution. Over one hundred tests of this category, called discordancy tests, have been developed for different scenarios (see [4]). A key drawback of this category of tests is that most of the distributions used are univariate. There are some tests that are multivariate (e.g. multivariate normal outliers). But for many KDD applications, the underlying distribution is unknown. Fitting the data with standard distributions is costly, and may not produce satisfactory results.

The second category of outlier studies in statistics is *depth-based*. Each data object is represented as a point in a *k*-d space, and is assigned a depth. With respect to outlier detection, outliers are more likely to be data objects with smaller depths. There are many definitions of depth that have been proposed (e.g. [13], [15]). In theory, depth-based approaches could work for large values of k. However, in practice, while there exist efficient algorithms for $k = 2$ or 3 ([13], [11]), depth-based approaches become inefficient for large data sets for $k \geq 4$. This is because depth-based approaches rely on the computation of *k*-d convex hulls which has a lower bound complexity of $\Omega(n^{k/2})$.

Recently, Knorr and Ng proposed the notion of *distance-based* outliers [12]. Their notion generalizes many notions from the distribution-based approaches, and enjoys better computational complexity than the depth-based approaches for larger values of *k*. Later in section 3, we will discuss in detail how their notion is different from the notion of local outliers proposed in this paper.

Given the importance of the area, fraud detection has received more attention than the general area of outlier detection. Depending on the specifics of the application do-

mains, elaborate fraud models and fraud detection algorithms have been developed (e.g. [8], [6]). In contrast to fraud detection, the kinds of outlier detection work discussed so far are more exploratory in nature. Outlier detection may indeed lead to the construction of fraud models.

3 Problems of Current (non-local) Approaches

As we have seen in section 2, most of the existing work in outlier detection lies in the field of statistics. Intuitively, outliers can be defined as given by Hawkins [10].

Definition 1: (Hawkins-Outlier)

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

This notion is formalized by Knorr and Ng [12] in the following definition of outliers.

Definition 2: (DB(p,d)-Outlier)

An object *o* in a data set *D* is a *DB*(*p*,*d*)-outlier if at least fraction *p* of the objects in *D* lies greater than distance *d* from *o*.

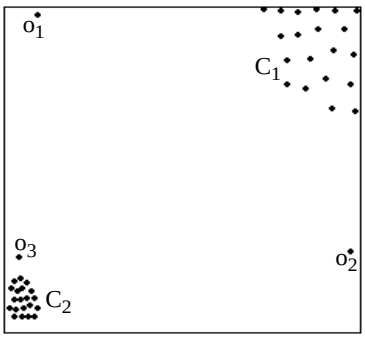

Fig. 1. 2-*d* dataset DS1

Below, we will show that definition 2 captures only certain kinds of outliers. Its shortcoming is that it takes a global view of the data set. The fact that many interesting real-world data sets exhibit a more complex structure, in which objects are only outliers relative to their local, surrounding object distribution, is ignored. We give an examples of a data set containing objects that are outliers according to Hawkins' definition for which no values for *p* and *d* exist such that they are *DB*(*p*,*d*)-outliers.

Figure 1 shows a 2-*d* dataset containing 43 objects. It consists of 2 clusters C_1 and C_2, each consisting of 20 objects, and there are 3 additional objects o_1, o_2 and o_3. Intuitively, and according to definition 1, o_1, o_2 and o_3 are outliers, and the points belonging to the clusters C_1 and C_2 are not. For an object *o* and a set of objects *S*, let $d(o,S) = min\{ d(o,s) \mid s \in S \}$.

Let us consider the notion of outliers according to definition 2:

- o_1: For every $d \le d(o_1,C_1)$ and $p \le 42/43$, o_1 is a *DB*(*p*,*d*) outlier. For smaller values of *p*, *d* can be even larger.
- o_2: For every $d \le d(o_2, C_1)$ and $p \le 42/43$, o_2 is a *DB*(*p*,*d*) outlier. Again, for smaller values of *p*, *d* can be even larger.
- o_3: Assume that for every point *q* in C_1, the distance from *q* to its nearest neighbor is larger than $d(o_3, C_2)$. In this case, no combination of *p* and *d* exists such that o_3 is an *DB*(*p*,*d*) outlier and the points in C_1 are not:
  - For every $d \le d(o_3, C_2)$, p=42/43 percent of all points are further away from o_3 than *d*. However, this condition also holds for every point $q \in C_1$. Thus, o_3 and all $q \in C_1$ are *DB*(*p*,*d*)-outliers.
  - For every $d > d(o_3, C_2)$, the fraction of points further away from o_3 is always smaller than for any $q \in C_1$, so either o_3 and all $q \in C_1$ will be considered outliers or (even worse) o_3 *is not* an outlier and all $q \in C_1$ *are* outliers.

From this example, we infer that definition 2 is only adequate under certain, limited conditions, but not for the general case that clusters of different densities exist. In these cases definition 2 will fail to find the *local* outliers, i.e. outliers that are outliers relative to their local surrounding data space.

4 Formal Definition of Local Outliers

In this section, we develop a formal definition of outliers that more truly corresponds to the intuitive notion of definition 1, avoiding the shortcomings presented in section 3. Our definition will correctly identify local outliers, such as o_3 in figure 1. To achieve this, we do not explicitly label the objects as "outlier" or "not outlier"; instead we compute the level of outlier-ness for every object by assigning an *outlier factor*.

Definition 3: (ε-neighborhood and *k*-distance of an object *p*)

Let *p* be an object from a database *D*, let ε be a distance value, let *k* be a natural number and let *d* be a distance metric on *D*. Then:

- the ε-neighborhood $N_\varepsilon(p)$ are the objects *x* with $d(p,x) \leq \varepsilon$: $N_\varepsilon(p) = \{ x \in D \mid d(p,x) \leq \varepsilon \}$,
- the *k*-distance of *p*, *k-distance*(*p*), is the distance $d(p,o)$ between *p* and an object $o \in D$ such that at least for *k* objects $o' \in D$ it holds that $d(p,o') \leq d(p,o)$, and for at most *k*-1 objects $o' \in D$ it holds that $d(p,o') < d(p,o)$. Note that *k*-distance(*p*) is unique, although the object *o* which is called 'the' *k*-nearest neighbor of *p* may not be unique. When it is clear from the context, we write $N_k(p)$ as a shorthand for $N_{k\text{-}distance(p)}(p)$, i.e. $N_k(p) = \{ x \in D \mid d(p,x) \leq k\text{-}distance(p) \}$.

The objects in the set $N_k(p)$ are called the "*k*-nearest-neighbors of *p*" (although there may be more than *k* objects in $N_k(p)$ if the *k*-nearest neighbor of *p* is not unique).

Before we can formally introduce our notion of outliers, we have to introduce some basic notions related to the density-based cluster structure of the data set. In [7] a formal notion of clusters based on point density is introduced. The point density is measured by the number of objects within a given area. The basic idea of the clustering algorithm DBSCAN is that for each object of a cluster the neighborhood of a given radius (ε) has to contain at least a minimum number of objects (*MinPts*). An object *p* whose ε-neighborhood contains at least *MinPts* objects is said to be a *core object*. Clusters are formally defined as maximal sets of density-connected objects. An object *p* is density-connected to an object *q* if there exists an object *o* such that both *p* and *q* are density-reachable from *o* (directly or transitively). An object *p* is said to be directly density-reachable from *o* if *p* lies in the neighborhood of *o* and *o* is a core object [7].

A 'flat' partitioning of a data set into a set of clusters is useful for many applications. However, an important property of many real-world data sets is that their intrinsic cluster structure cannot be characterized by *global* density parameters. Very different local densities may be needed to reveal and describe clusters in different regions of the data space. Therefore, in [1] the density-based clustering approach is extended and generalized to compute not a single flat density-based clustering of a data set, but to create an augmented *ordering* of the database representing its density-based clustering structure. This cluster-ordering contains information which is equivalent to the density-based clusterings corresponding to a broad range of parameter settings. This cluster-ordering of a data set is based on the notions of *core-distance* and *reachability-distance*.

Definition 4: (core-distance of an object *p*)

Let *p* be an object from a database *D*, let ε be a distance value and let *MinPts* be a natural number. Then, the *core-distance* of *p* is defined as

$$\textit{core-distance}_{\varepsilon,\textit{MinPts}}(p) = \begin{cases} \text{UNDEFINED, if } |N_\varepsilon(p)| < \textit{MinPts} \\ \textit{MinPts-distance}(p), \text{ otherwise} \end{cases}$$

The core-distance of object *p* is the smallest distance ε' ≤ ε such that *p* is a core object with respect to ε' and *MinPts* if such an ε' exists, i.e. if there are at least *MinPts* objects within the ε-neighborhood of *p*. Otherwise, the core-distance is UNDEFINED.

Definition 5: (reachability-distance of an object *p* w.r.t. object *o*)

Let *p* and *o* be objects from a database *D*, $p \in N_\varepsilon(o)$, let ε be a distance value and let *MinPts* be a natural number. Then, the *reachability-distance* of *p* with respect to *o* is defined as $\textit{reachability-distance}_{\varepsilon,\textit{MinPts}}(p, o) =$

$$\begin{cases} \text{UNDEFINED, if } |N_\varepsilon(o)| < \textit{MinPts} \\ \max(\textit{core-distance}_{\varepsilon,\,\textit{MinPts}}(o), d(o, p)) \quad \text{otherwise} \end{cases}$$

The reachability-distance of an object *p* with respect to object *o* is the smallest distance such that *p* is directly density-reachable from *o* if *o* is a core object within *p*'s ε-neighborhood. To capture this idea, the reachability-distance of *p* with respect to *o* cannot be smaller than the core-distance of *o* since for smaller distances no object is directly density-reachable from *o*. Otherwise, if *o* is not a core object, the reachability-distance is UNDEFINED. Figure 2 illustrates the core-distance and the reachability-distance.

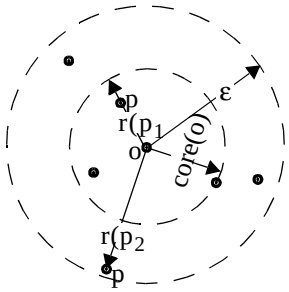

Fig. 2. Core-distance(*o*), reachability-distances r(p_1,*o*), r(p_2,*o*) for *MinPts*=4

The core-distance and reachability-distance were originally introduced for the OPTICS-algorithm [1]. The OPTICS-algorithm computes a "walk" through the data set, and calculates for each object *o* the core-distance and the smallest reachability-distance with respect to an object considered *before o* in the walk. Such a walk through the data satisfies the following condition: Whenever a set of objects *C* is a density-based cluster with respect to *MinPts* and a value ε' smaller than the value ε used in the OPTICS algorithm, then a permutation of *C* (possibly without a few border objects) is a subsequence in the walk. Therefore, the *reachability-plot* (i.e. the reachability values of all objects plotted in the OPTICS ordering) yields an easy to understand visualization of the clustering structure of the data set. Roughly speaking, a low reachability-distance indicates an object within a cluster, and a high reachability-distance indicates a noise object or a jump from one cluster to another cluster. The reachability-plot for our dataset DS1 is depicted in figure 3 (top). The global structure revealed shows that there are the two clusters, one of which is more dense than the other, and a few objects outside the clusters. Another example of a reachability-plot for the more complex data set DS2 (figure 4) containing hierarchical clusters is depicted in figure 5.

Definition 6: (local reachability density of an object *p*)

Let *p* be an object from a database *D* and let *MinPts* be a natural number. Then, the *local reachability density* of *p* is defined as

$$lrd_{MinPts}(p) = 1 / \frac{\sum_{o \in N_{MinPts}(p)} reachability\text{-}distance_{\infty, MinPts}(p, o)}{|N_{MinPts}(p)|}$$

The local reachability density of an object *p* is the inverse of the average reachability-distance from the *MinPts*-nearest-neighbors of *p*. The reachability-distances occurring in this definition are all defined, because ε=∞. The *lrd* is ∞ if all reachability-distances are 0. This may occur for an object *p* if there are at least *MinPts* objects, different from *p*, but sharing the same spatial coordinates, i.e. if there are at least *MinPts* duplicates of *p* in the data set. For simplicity, we will not handle this case explicitly but simply assume that there are no duplicates. (To deal with duplicates, we can base our notion of neighborhood on a *k-distinct-distance*, defined analogously to *k*-distance in definition 3 with the additional requirement that there be at least *k different* objects.)

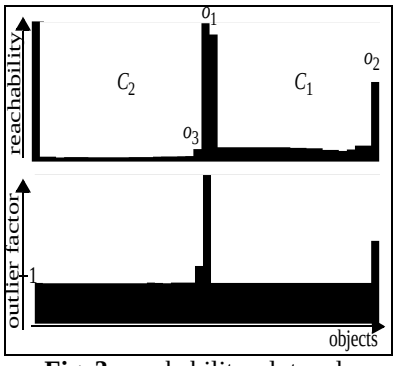

Fig. 3. reachability-plot and outlier factors for DS1

The reason for using the reachability-distance instead of simply the distance between *p* and its neighbors *o* is that it will significantly weaken statistical fluctuations of the inter-object distances: *lrd*s for objects which are close to each other in the data space (whether in clusters or noise) will in general be equaled by using the reachability-distance because it is at least as large as the core-distance of the respective object *o*. The strength of the effect can be controlled by the parameter *MinPts*. The higher the value for *MinPts* the more similar the reachability-distances for objects within the same area of the space. Note that there is a similar 'smoothing' effect for the reachability-plot produced by the OPTICS algorithm, but in this case of clustering we also weaken the so-called 'single-link effect' [14].

Definition 7: (outlier factor of an object *p*)

Let *p* be an object from a database *D* and let *MinPts* be a natural number. Then, the *outlier factor* of *p* is defined as

$$OF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|}$$

The outlier factor of the object *p* captures the degree to which we call *p* an outlier. It is the average of the ratios of the *lrd*s of the *MinPts*-nearest-neighbors and *p*. If these are identical, which we expect for objects in clusters of uniform density, the outlier factor is 1. If the *lrd* of *p* is only half of the *lrd*s of *p*'s *MinPts*-nearest-neighbors, the outlier factor of *p* is 2. Thus, the lower *p*'s *lrd* is and the higher the *lrd*s of *p*'s *MinPts*-nearest-neighbors are, the higher is *p*'s outlier factor.

Figure 3, (top) shows the reachability-plot for DS1 generated by OPTICS [1]. Two clusters are visible: first the dense cluster C_2, then points o_3 and o_1 (larger reachability values) and - after the large reachability indicating a jump - all of cluster C_1 and finally o_2. Depicted below the reachability-plot are the corresponding outlier factors (the objects are in the same order as in the reachability-plot). Object o_1 has the largest outlier factor (3.6), followed by o_2 (2.0) and o_3 (1.4). All other objects are assigned outlier factors between 0.993 and 1.003. Thus, our technique successfully highlights not only the *global* outliers o_1 and o_2 (which are also *DB*(*p*,*d*)-outliers), but also the *local* outlier o_3 (which is not a reasonable *DB*(*p*,*d*)-outlier).

5 An Extensive Example

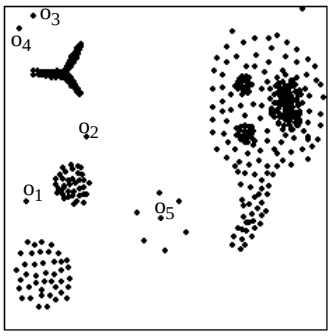

Fig. 4. Example dataset DS2

In this section, we demonstrate the effectiveness of the given definition using a complex 2-*d* example data set (DS2, figure 4, 473 points), containing most characteristics of real-world data sets, i.e. hierarchical/overlapping clusters and clusters of widely differing densities and arbitrary shapes. We give a small 2-*d* example to make it easier to understand the concepts. Our approach, however, works equally well in higher dimensional spaces. DS2 consists of 3 clusters of uniform (but different) densities and one hierarchical cluster of a low density containing 2 small and 1 bigger subcluster. The data set also contains 12 outliers.

Figure 5 (top) shows the reachability-plot generated by OPTICS. We see 3 clusters with different, but uniform densities in areas 1, 2 and 3, a large, hierarchical cluster in area 4 and its subclusters in areas 4.1, 4.2 and 4.3. The noise points (outliers) have to be located in areas N1, N2, N3 and N4.

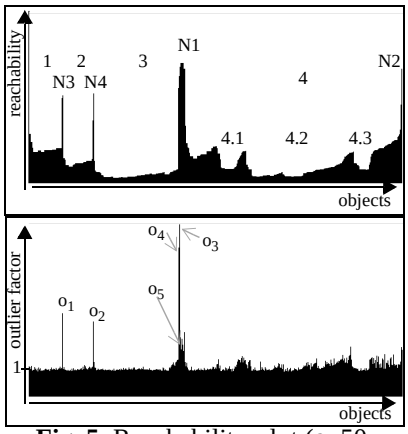

Fig. 5. Reachability-plot (ε=50, *MinPts*=10) and outlier factors OF_{10} for DS2

Figure 5 (bottom) shows the outlier factors for *MinPts*=10 (objects in the same order). Most objects are assigned outlier factors of around 1. In areas N3 and N4 there is one point each (o_1 and o_2) with outlier factors of 3.0 and 2.7 respectively, characterizing outliers with local reachability densities about half to one third of the surrounding space. The most interesting area is N1. The outlier factors are between 1.7 and 6.3. The first two points with high outlier factors 5.4 and 6.3 are o_3 and o_4. Both only have one close neighbor (the other one) and all other neighbors are far away in the cluster in area 3, which has a high density (recall that for MinPts=10 we are looking at the 10-nearest-neighbors). Thus, o_3 and o_4 are assigned large outlier factors. The other points in N1, however, are assigned much smaller

(but still significantly larger than 1) outlier factors between 1.7 and 2.4. These are the points surrounding o_5 which can either be considered a small, low density cluster or outliers, depending on ones viewpoint. Object o_5 as the center point of this low density cluster is assigned the lowest outlier factor of 1.7, because it is surrounded by points of equal local reachability density.

We also see that, from the reachability-plot, we can only infer that all points in N1 are in an area of low density, because the reachability values are high. However, no evaluation concerning their outlierness is possible.

6 Mining Local Outliers - Performance Considerations

To compute the outlier-factors $OF_{MinPts}(p)$ for all objects p in a database, we have to perform three passes over the data. In the first pass, we compute $N_{MinPts}(p)$ and $core\text{-}distance_{\infty,MinPts}(p)$. In the second pass, we calculate $reachability\text{-}distance_{\infty,MinPts}(p,o)$ of p with respect to its neighboring objects $o \in N_{MinPts}(p)$ and $lrd_{MinPts}(p)$ of p. In the third pass, we can compute the outlier factors $OF(p)$. The runtime of the whole procedure is heavily dominated by the first pass over the data since we have to perform k-nearest-neighbor queries in a multidimensional database, i.e. the runtime of the algorithm is O(n * runtime of a *MinPts*-nearest-neighborhood query).

Obviously, the total runtime depends on the runtime of k-nearest-neighbor query. Without any index support, to answer a k-nearest-neighbor query, a scan through the whole database has to be performed. In this case, the runtime of our outlier detection algorithm would be $O(n^2)$. If a tree-based spatial index can effectively be used, the runtime is reduced to $O(n \log n)$ since k-nearest-neighbor queries are supported efficiently by spatial access methods such as the R*-tree [3] or the X-tree [2] for data from a vector space or the M-tree [5] for data from a metric space. The height of such a tree-based index is O(log n) for a database of n objects in the worst case and, at least in low-dimensional spaces, a query with a reasonable value for k has to traverse only a limited number of paths.

If also the algorithm OPTICS is applied to the data set, i.e. if we also want to perform some kind of cluster analysis, we can drastically reduce the cost for the outlier detection. The algorithm OPTICS retrieves the ε-neighborhood $N_\varepsilon(p)$ for each object p in the database, where ε is an input parameter. These ε-neighborhoods can be utilized in the first pass over the data for our outlier detection algorithm: only if this neighborhood $N_\varepsilon(p)$ of p does not already contain *MinPts* objects, we have to perform a *MinPts*-nearest-neighbor query for p to determine $N_{MinPts}(p)$. In the other case, we can retrieve $N_{MinPts}(p)$ from $N_\varepsilon(p)$ since then it holds that $N_{MinPts}(p) \subseteq N_\varepsilon(p)$. Our experiments indicate that in real applications, for a reasonable value of ε and *MinPts*, this second case is much more frequent than the first case.

7 Conclusions

Finding outliers is an important task for many KDD applications. All proposals so far considered 'being an outlier' a binary property. We argue instead, that it is a property that applies to a certain *degree* to each object in a data set, depending on how 'isolated' this object is, with respect to the surrounding clustering structure. We formally defined

the notion of an *outlier factor*, which captures exactly this relative degree of isolation. The outlier factor is local by taking into account the clustering structure in a bounded neighborhood of the object. We demonstrated that this notion is more appropriate for detecting different types of outliers than previous approaches. Our definitions are based on the same theoretical foundation as density-based cluster analysis and we show how to analyze the cluster structure and the outlier factors efficiently at the same time.

In ongoing work, we are investigating the properties of our approach in a more formal framework, especially with regard to the influence of the *MinPts* value. Future work will include the development of a more efficient and an incremental version of the algorithm based on the results of this analysis.

References

1. Ankerst M., Breunig M. M., Kriegel H.-P., Sander J.: "*OPTICS: Ordering Points To Identify the Clustering Structure*", Proc. ACM SIGMOD Int. Conf. on Management of Data, Philadelphia, PA, 1999.
2. Berchthold S., Keim D., Kriegel H.-P.: "*The X-Tree: An Index Structure for High-Dimensional Data*", 22nd Conf. on Very Large Data Bases, Bombay, India, 1996, pp. 28-39.
3. Beckmann N., Kriegel H.-P., Schneider R., Seeger B.: *"The R*-tree: An Efficient and Robust Access Method for Points and Rectangles"*, Proc. ACM SIGMOD Int. Conf. on Management of Data, Atlantic City, NJ, ACM Press, New York, 1990, pp. 322-331.
4. Barnett V., Lewis T.: "*Outliers in statistical data*", John Wiley, 1994.
5. Ciaccia P., Patella M., Zezula P.: *"M-tree: An Efficient Access Method for Similarity Search in Metric Spaces"*, Proc. 23rd Int. Conf. on Very Large Data Bases, Athens, Greece, 1997, pp. 426-435.
6. DuMouchel W., Schonlau M.: "*A Fast Computer Intrusion Detection Algorithm based on Hypothesis Testing of Command Transition Probabilities*", Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining, New York, NY, AAAI Press, 1998, pp. 189-193.
7. Ester M., Kriegel H.-P., Sander J., Xu X.: *"A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise"*, Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, 1996, pp. 226-231.
8. Fawcett T., Provost F.: "*Adaptive Fraud Detection*", Data Mining and Knowledge Discovery Journal, Kluwer Academic Publishers, Vol. 1, No. 3, pp. 291-316.
9. Fayyad U., Piatetsky-Shapiro G., Smyth P.*: "Knowledge Discovery and Data Mining: Towards a Unifying Framework*", Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, 1996, pp. 82-88.
10. Hawkins, D.: "*Identification of Outliers*", Chapman and Hall, London, 1980.
11. Johnson T., Kwok I., Ng R.: "*Fast Computation of 2-Dimensional Depth Contours*", Proc. 4th Int. Conf. on KDD, New York, NY, AAAI Press, 1998, pp. 224-228.
12. Knorr E. M., Ng R. T.: "*Algorithms for Mining Distance-Based Outliers in Large Datasets*", Proc. 24th Int. Conf. on Very Large Data Bases, New York, NY, 1998, pp. 392-403.
13. Preparata F., Shamos M.: "*Computational Geometry: an Introduction*", Springer, 1988.
14. Sibson R.: *"SLINK: an optimally efficient algorithm for the single-link cluster method"*, The Computer Journal, Vol. 16, No. 1, 1973, pp. 30-34.
15. Tukey J. W.: "*Exploratory Data Analysis*", Addison-Wesley, 1977.