5

Density-Based Clustering and Anomaly Detection

Lian Duan
University of Iowa, USA

1. Introduction

As of 1996, when a special issue on density-based clustering was published (DBSCAN) (Ester et al., 1996), existing clustering techniques focused on two categories: partitioning methods, and hierarchical methods. Partitioning clustering attempts to break a data set into *K* clusters such that the partition optimizes a given criterion. Besides difficulty in choosing the proper parameter *K*, and incapacity of discovering clusters with arbitrary shape, partitioning clustering techniques are very sensitive to outliers. Although the k-medoids method (Kaufman & Rousseeuw, 1990) is more robust than k-means (MacQueen, 1967) in the presence of outliers, they cannot discover outliers. Hierarchical clustering algorithms produce a nested sequence of clusters, with a single all-inclusive cluster at the top and single point clusters at the bottom. CURE (Guha et al., 1998) is capable of finding clusters of arbitrary shapes and reduces the effect of outliers; however, it only considers cluster proximity yet ignores cluster interconnectivity, and an outlier is still assigned to the cluster which has the closest representative point to it.

To discover clusters with arbitrary shape and outliers, density-based clustering methods have been developed. These typically regard clusters as dense regions of objects in the data space that are separated by regions of low density (representing outliers or noises). DBSCAN grows clusters according to a density-based connectivity analysis. OPTICS (Ankerst et al., 1999) extends DBSCAN to produce a cluster ordering obtained from a wide range of parameter settings. DENCLUE (Hinneburg & Keim, 1998) clusters objects based on a set of density distribution functions. LOF (Breunig et al., 2000) uses a more meaningful way to assign to each object a degree of being an outlier than to consider being an outlier as a binary property. LDBSCAN (Duan et al., 2007) combines the concepts of DBSCAN and LOF to discover clusters and outliers. There are two potential benefits of combining clustering and outlier detection: increasing precision and facilitating data understanding. The goal of this chapter is to survey the core concepts and techniques in the density-based clustering and outlier detection (Duan et al., 2009) with its roots in data mining, statistics, machine learning and other communities.

This chapter is organized as follows. Section 2 presents the algorithm LDBSCAN. Section 3 discusses the cluster-based outlier detection. The comprehensive experiments on the algorithms we proposed are conducted on both synthetic data and practical data. Finally, we present some concluding remarks.

2. LDBSCAN: Local-Density-Based Spatial Clustering of Applications with Noise

In this section, we introduce our algorithm LDBSCAN. First, the basic notions used in LDBSCAN are discuss. Then, the algorithm is presented.

2.1 Basic notions of LDBSCAN

2.1.1 Problems of existing density-based algorithms

A common property of many practical data sets is that their intrinsic cluster structures cannot be characterized by global density parameters. As a result, very different local densities may be needed to reveal clusters in different regions of the data space. For example, in the data set depicted in Figure 1, it is impossible to detect the cluster A, B, C_1, C_2, and C_3 simultaneously by using one global density parameter. A global density-based decomposition can only detect the clusters A, B, and C, or C_1, C_2, and C_3. For the second partition, the objects from A and B are noise.

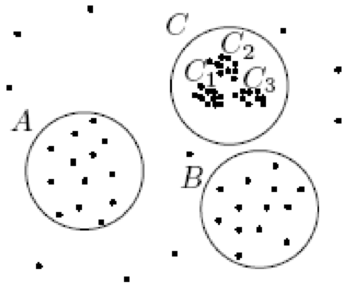


Fig. 1. Clusters with respect to different global density parameters

Optics can solve this problem; however, it only creates an augmented ordering of the database representing its density-based clustering structure instead of producing clusters of a data set explicitly. In addition, it might not be able to generate the clusters resided in other clusters appropriately and this part will be discussed in the experimental part. Therefore, an algorithm which can detect A, B, C_1, C_2, and C_3 explicitly is needed.

2.1.2 Definition of LRD and LOF

The LOF of each object represents the degree the object is being outlying and the LRD of each object represents the local-density of the object. The formal definitions for these notions of LOF and LRD are shortly introduced in the following. More details are provided in (Breunig et al., 2000).

Definition 1 (*k*-distance of an object *p*) For any positive integer *k*, the *k*-distance of object *p*, denoted as *k-distance(p)*, is defined as the distance *d(p,o)* between *p* and an object $o \in D$ such that:

1. for at least *k* objects $o' \in D \setminus \{p\}$ it holds that $d(p,o') \leq d(p,o)$
2. for at most *k-1* objects $o' \in D \setminus \{p\}$ it holds that $d(p,o') < d(p,o)$.

Definition 2 (*k*-distance neighborhood of an object *p*): Given the *k*-distance of *p*, the *k*-distance neighborhood of *p* contains every object whose distance from *p* is not greater than the *k*-distance, i.e. $N_{k\text{-}distance(p)}(p)=\{ q \in D \setminus \{p\} \mid d(p,q) \leq k\text{-}distance(p) \}$. These objects *q* are called the *k*-nearest neighbors of *p*.

As no confusion arises, the notation can be simplified to use $N_k(p)$ as a shorthand for $N_{k\text{-}distance(p)}(p)$.

Definition 3 (reachability distance of an object *p* w.r.t. object *o*): Let *k* be a natural number. The reachability distance of object *p* with respect to object *o* is defined as

$reach\text{-}dist_k(p,o)=max \{ k\text{-}distance(o), d(p,o) \}$

Definition 4 (local reachability density of an object *p*): The local reachability density of *p* is defined as

$$\mathrm{LRD_{MinPts}(p)}=1/\left(\frac{\sum_{o \in N_{MinPts}(p)} reach-dist_{MinPts}(p,o))}{|N_{MinPts}(p)|}\right)$$

Intuitively, the local reachability density of an object *p* is the inverse of the average reachability distance based on the *MinPts*-nearest neighbors of *p*.

Definition 5 (local outlier factor of an object *p*): The local outlier factor of *p* is defined as

$$\mathrm{LOF_{MinPts}(p)}=\frac{\sum_{o \in N_{MinPts}(p)} \frac{LRD_{MinPts}(o)}{LRD_{MinPts}(p)}}{|N_{MinPts}(p)|}$$

The LOF of object *p* is the average of the ratio of the LRD of *p* and those of *p*'s *MinPts*-nearest neighbors. It captures the degree to which *p* is called an outlier. It is easy to see that the higher the ratio of the LRD of *p* to those of *p*'s *MinPts*-nearest neighors is, the farther away the point *p* is from its nearest cluster, and the higher the LOF value of *p* is. Since the LOF represents the degree the object is being outlying and the LOF of most objects in a cluster is approximately equal to 1, we regard object *p* belong to a certain cluster if *LOF(p)* is lower than a threshold we set.

2.1.3 A local-density based notion of clusters

When looking at the sample set of points depicted in Figure 2, we can easily and unambiguously detect clusters of points and noise points not belonging to any of those clusters. The main reason is that within each cluster the local density of points are different from that of the outside part.

In the following, these intuitive notions of "clusters" and "noise" are formalized. Note that both notion of clusters and the algorithm LDBSCAN apply to 2D Euclidean space as to higher dimensional feature space. The key idea is that for any point *p* satisfying *LOF(p)* ≤*LOFUB*, i.e. point *p* is not an outlier and belongs to a certain cluster *C*, if point *q* is the *MinPts*-nearest neighbour of *p* and has the similar LRD with *p*, *q* belongs to the same cluster *C* of *p*. This approach works with any distance function so that an appropriate function can be chosen for a given application. In this chapter, for the purpose of proper visualization, related examples will be in 2D space using the Euclidean distance.

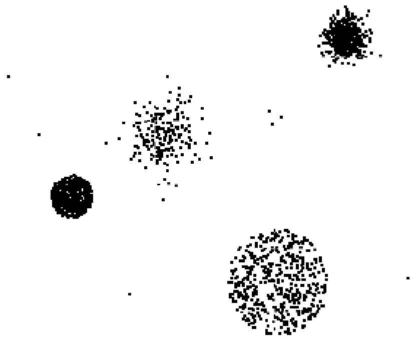


Fig. 2. Sample Data (Breunig et al., 2000)

Definition 6 (core point): A point *p* is a core point w.r.t. *LOFUB* if *LOF(p)≤LOFUB*.

If *LOF(p)* is small enough, it means that point *p* is not an outlier and must belong to some clusters. Therefore it can be regarded as a core point.

Definition 7 (directly local-density-reachable): A point *p* is directly local-density-reachable from a point *q* w.r.t. *pct* and *MinPts* if

1. $p \in N_{MinPts}(q)$ and
2. $LRD(q)/(1+pct)<LRD(p)<LRD(q)*(1+pct)$

Here, the parameter *pct* is used to control the fluctuation of local-density. However, in general, it is not symmetric if *q* is not the *MinPts*-nearest neighbour of *p*. Figure 3 shows the asymmetric case. Let *MinPts=3* and *pct=0.3*, we calculate that *LRD(p)/LRD(q)=1.27*. It shows that *p* is directly local-density-reachable from *q*, but *q* is not directly local-density-reachable from *p*.

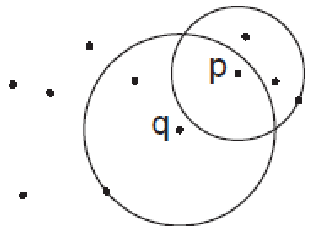


Fig. 3. Directly local-density-reachability

Definition 8 (local-density-reachable): A point *p* is local-density-reachable from the point *q* w.r.t. *pct* and *MinPts* if there is a chain of points $p_1, p_2, ..., p_n, p_1=q, p_n=p$ such that p_{i+1} is directly local-density-reachable from p_i.

Local-density-reachability is a canonical extension of direct local-density-reachability. This relation is transitive, but it is not symmetric. Figure 4 depicts the relations of some sample points and an asymmetric case. Let *MinPts=3, pct=0.3*. According to the above definitions, *LRD(p)/LRD(o)=1.27, LRD(o)/LRD(q)=0.95*. Here, *q* is local-density-reachable from *p*, but *p* is not local-density-reachable from *q*.

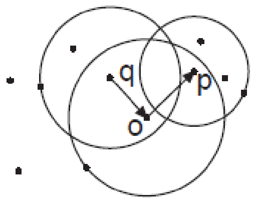


Fig. 4. Local-density-reachability

Definition 9 (local-density-connected): A point *p* is local-density-connected to a point *q* from *o* w.r.t. *pct* and *MinPts* if there is a point *o* such that both *p* and *q* are local-density-reachable from *o* w.r.t. *pct* and *MinPts*.

From the definition, local-density-connectivity is a symmetric relation show in Figure 5. Now we can make use of the above definitions to define the local-density-based cluster. Intuitively, a cluster is defined as a set of local-density-connected points which is maximal w.r.t local-density-reachability. Noised are defined relatively to a given set of clusters. Noises are simply the set of points in the dataset not belonging to any of its clusters.

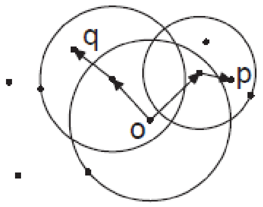


Fig. 5. Local-density-connectivity

Definition 10 (cluster): Let *D* be a database of points, and point *o* is a selected core point of *C*, i.e. $o \in C$ and $LOF(o) \leq LOFUB$. A cluster *C* w.r.t. *LOFUB*, *pct* and *MinPts* is a non-empty subset of *D* satisfying the following conditions:

1. $\forall p$: *p* is local-density-reachable from *o* w.r.t. *pct* and *MinPts*, then $p \in C$. (Maximality)
2. $\forall p,q \in C$: *p* is local-density-connected *q* by *o* w.r.t. *LOFUB*, *pct* and *MinPts*. (Connectivity)

Definition 11 (noise): Let $C_1, \ldots, C_k$ be the clusters of the database *D* w.r.t. parameters *LOFUB*, *pct* and *MinPts*. Then we define the noise as the set of points in the database *D* not belonging to any cluster C_i, i.e. noise= $\{ p \in D \mid \forall i: p \text{ not in } C_i \}$.

2.2 The algorithm

In this section, we present the algorithm LDBSCAN which is designed to discover the clusters and the noise in a spatial database according to Definition 10 and 11. First, the appropriate parameters *LOFUB*, *pct*, and *MinPts* of clusters and one core point of the respective cluster are selected. Then all points that are local-density-reachable from the given core point using the correct parameters are retrieved. Since all the parameters are relative, and not absolute as those in DBSCAN, they are easy to choose and fall in a certain range as presented in the experimental part.

To find a cluster, LDBSCAN starts with an arbitrary point *p* and retrieves all points local-density-reachable from *p* w.r.t. *LOFUB*, *pct*, and *MinPts*. If *p* is a core point, this procedure yields a cluster w.r.t *LOFUB*, *pct*, and *MinPts*. If *p* is not a core point, LDBSCAN will check the next point of the dataset. In the following, we present a basic version of LDBSCAN without details of data types and generation of additional information about clusters:

```
LDBSCAN (SetOfPoints, LOFUB, pct, MinPts)
  // SetOfPoints is UNCLASSIFIED
  InitSet (SetOfPoints); // calculate LRD and LOF of each point
  ClusterID := 0;
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF LOF(Point) ≤ LOFUB THEN // core point
        ClusterID := ClusterID + 1;
        ExpandCluster(SetOfPoints, Point, ClusterID, pct, MinPts);
      ELSE // no core point
        SetOfPoint.changeClId(Point,NOISE);
      END IF
    END IF
  END FOR
END; //LDBSCAN
```

SetOfPoints is the set of the whole database. *LOFUB*, *pct* and *MinPts* are the carefully chosen parameters. The function SetOfPoints.get(i) returns the *i*-th element of SetOfPoints. Points which have been marked to be NOISE may be changed later if they are local-density-reachable from some core points of the database. The most important function used by LDBSCAN is ExpandCluster which is presented in the following:

```
ExpandCluster(SetOfPoints, Point, ClusterID, pct, MinPts)
  SetOfPoint.changeClId(Point,ClusterID);
  FOR i FROM 1 TO MinPts DO
    currentP := Point.Neighbor(i);
    IF currentP.ClId IN {UNCLASSIFIED,NOISE} and DirectReachability(currentP,Point) THEN
      TempVector.add(currentP);
      SetOfPoint.changeClId(currentP,ClusterID);
    END IF
  END FOR
  WHILE TempVector <> Empty DO
    Point := TempVector.firstElement();
    TempVector.remove(Point);
    FOR i FROM 1 TO MinPts DO
      currentP := Point.Neighbor(i);
      IF currentP.ClId IN {UNCLASSIFIED,NOISE} and DirectReachability(currentP,Point) THEN
        TempVector.add(currentP);
        SetOfPoint.changeClId(currentP,ClusterID);
      END IF
    END FOR
  END WHILE
END; //ExpandCluster
```

The function DirectReachability(currentP,Point) is presented in the following:

```
DirectReachability(currentP,Point) : Boolean
  IF LRD(currentP)>LRD(Point)/(1+pct) and LRD(currentP)<LRD(Point)*(1+pct) THEN
    RETURN True;
  ELSE
    RETURN False;
END; //DirectReachability
```

The LDBSCAN algorithm randomly selects one core point which has not been clustered, and then retrieves all points that are local-density-reachable from the chosen core point to form a cluster. It won't stop until there is no unclustered core point.

3. Cluster-Based Outliers

In this section, we give the definition of cluster-based outliers and conduct a detailed analysis on the properties of cluster-based outliers. The goal is to show how to discover cluster-based outliers and how the definition of the cluster-based outlier factor (CBOF) captures the spirit of cluster-based outliers. The higher the CBOF is, the more abnormal the cluster-based outliers are.

3.1 Definition of Cluster-Based Outliers

Intuitively, most data points in the data set should not be outliers; therefore, only the clusters that hold a small portion of data points are candidates for cluster-based outliers. Considering the different and complicated situations, it is impossible to provide a definite number as the upper bound of the number of the objects contained in a cluster-based outlier (UBCBO). Here, only a guideline is provided to find the reasonable upper bound.

Definition 12 (Upper Bound of the Cluster-Based Outlier): Let $C_1, \ldots, C_k$ be the clusters of the database D discovered by LDBSCAN in the sequence that $|C_1| \geq |C_2| \geq \ldots \geq |C_k|$. Given parameters a, the number of the objects in the cluster C_i is the *UBCBO* if $(|C_1|+|C_2|+\ldots+|C_{i-1}|) \geq |D|*a$ and $(|C_1|+|C_2|+\ldots+|C_{i-2}|) < |D|*a$.

Definition 12 gives quantitative measure to *UBCBO*. Consider that most data points in the dataset are not outliers; therefore, clusters that hold a large portion of data points should not be considered as outliers. For example, if a is set to 90%, we intend to regard clusters which contain 90% of data points as normal clusters.

Definition 13 (Cluster-based outlier): Let $C_1, \ldots, C_k$ be the clusters of the database D discovered by LDBSCAN. Cluster-based outliers are the clusters in which the number of the objects is no more than *UBCBO*.

Note that this guideline is not always appropriate. For example, in some cases the abnormal cluster deviated from a large cluster might contain more points than a certain small normal cluster. In fact, due to spatial and temporal locality, it would be more proper to choose the clusters which have small spatial or temporal span as cluster-based outliers than the clusters which contain few objects. The notion of cluster-based outliers depends on situations.

3.2 The lower bound of the number of objects contained in a cluster

Comparing with single point outliers, cluster-based outliers are more interesting. Many single point outliers are related to occasional trivial events, while cluster-based outliers concern some important lasting abnormal events. Generally speaking, it is reckless to form a cluster with only 2 or 3 objects, so the lower bound of the number of the objects contained in a cluster generated by LDBSCAN will be discussed in the following.

Definition 14 (distance between two clusters): Let C_1, C_2 be the clusters of the database D. The distance between C_1 and C_2 is defined as

$dist(C_1, C_2)=min\{ dist(p,q) \mid p\in C_1, q\in C_2\}$

Theorem 1: Let C_1 be the smallest cluster discovered by LDBSCAN w.r.t. appropriate parameters *LOFUB*, *pct* and *MinPts*, and C_2 is large enough be the closest normal cluster to C_1. Let $LRD(C_1)$ denote the minimum LRD of all the objects in C_1, i.e., $LRD(C_1)=min\{LRD(p) \mid p\in C_1\}$. Similarly, let $LRD(C_2)$ denote the minimum LRD of all the objects in C_2. Then for *LBC*, the lower bound of the number of the objects contained in a cluster, such that:

$$\text{LBC} = [\frac{(MinPts+1)LRD(q)-(LOFUB*MinPts+1)LRD(p)}{LRD(q)-LRD(p)}]+1$$

Proof (Sketch): Let p_i denote the *i*-th object in C_1 and $q_{i,j}$ be the *j*-th close object to p_i in C_2. And let *k* be the number of the objects in C_1. To simplify our proof, we only consider the situation that each point only has *k* *k*-nearest neighbors and the density within a cluster fluctuates slightly.

If $k\geq MinPts+1$, according to the definition of LOF, the LOF of any object in C_1 is approximately equal to 1. That is, $LOF(p_i)<LOFUB$ and each object in C_1 is a core point. In addition, each object in C_1 has the similar *LRD* to its neighbors which belong to the same cluster with it. According to the definition of the cluster, the cluster C_1 would be discovered by LDBSCAN. Thus, *LBC* is no more than $MinPts+1$.

If $k\leq MinPts$, the *MinPts*-distance neighbors of p_i contain the *k-1* rest objects in C_1 and the other $MinPts$-$k+1$ neighbors in C_2 shown in Figure 6. Obviously, the *MinPts*-distance of each fixed object p_j in C_1 is greater than the distance between any object p_i in C_1 and p_j, so $reach\text{-}dist(p_i,p_j)= MinPts\text{-}distance(p_j)$. Furthermore, the $MinPts\text{-}distance(q_{i,j})<<dist(C_1,C_2)\leq d(p_i,q_{i,j})$.

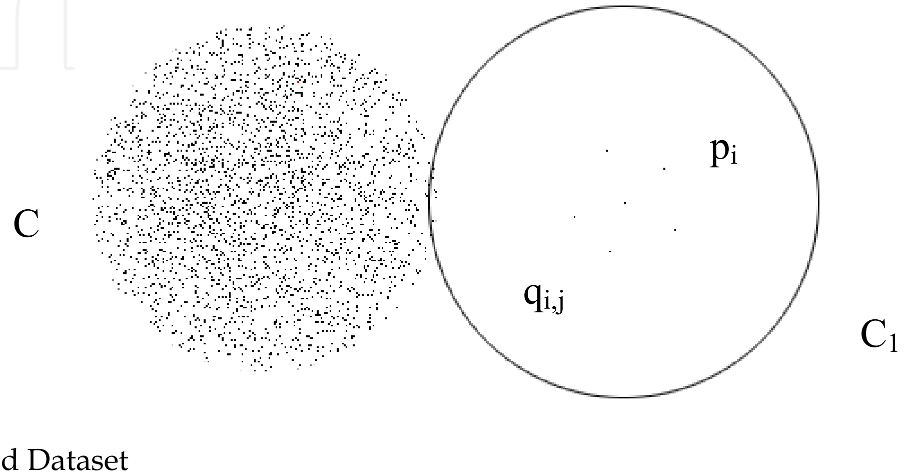


Fig. 6. 2-d Dataset

$$\Rightarrow LRD_{MinPts}(p_i) =$$
$$= MinPts / \left(\sum_{a=1}^{k} MinPts - dist(p_a) - MinPts - dist(p_i) + \sum_{a=1}^{MinPts-k+1} d(p_i, q_{i,a}) \right) \qquad (2)$$

and

$$LRD_{MinPts}(q_i) = MinPts / \sum_{o \in N_{MinPts}(q_i)} reach - dist_{MinPts}(q_i, o) \qquad (3)$$

$\forall p_i \in C_1$: Let MinPts-dist(p)=min{MinPts-dist(p_i) | p_i∈ C_1}, and then MinPts-dist(p_i) = MinPts-dist(p)+ε_i. Similarly, let d(p,q)=min{d(p_i,$q_{i,j}$) | p_i∈C_1, $q_{i,j}$∈C_2 and $q_{i,j}$ is the MinPts-neighbor of p_i} and d(p_i,$q_{i,j}$)=d(p,q)+ $\varepsilon_{i,j}$. Because we assume that the density within a cluster fluctuates slightly, MinPts-dist(p)>> ε_i and d(p,q)>> $\varepsilon_{i,j}$.

Compare the *LRD* of object p_i with that of its neighbor p_j in C_1.

$$\frac{LRD_{MinPts}(p_i)}{LRD_{MinPts}(p_j)} = \frac{\sum_{a=1}^{k} MinPts - dist(p_a) - MinPts - dist(p_j) + \sum_{a=1}^{MinPts-k+1} d(p_j, q_{j,a})}{\sum_{a=1}^{k} MinPts - dist(p_a) - MinPts - dist(p_i) + \sum_{a=1}^{MinPts-k+1} d(p_i, q_{i,a})}$$

$$= \frac{\sum_{a=1}^{k} MinPts - dist(p_a) - MinPts - dist(p) - \varepsilon_j + \sum_{a=1}^{MinPts-k+1} (d(p,q) + \varepsilon_{j,a})}{\sum_{a=1}^{k} MinPts - dist(p_a) - MinPts - dist(p) - \varepsilon_i + \sum_{a=1}^{MinPts-k+1} (d(p,q) + \varepsilon_{i,a})}$$

$$= \frac{\sum_{a=1}^{k} MinPts - dist(p_a) - MinPts - dist(p) + (MinPts - k + 1) * d(p,q) + \sum_{a=1}^{MinPts-k+1} \varepsilon_{j,a} - \varepsilon_j}{\sum_{a=1}^{k} MinPts - dist(p_a) - MinPts - dist(p) + (MinPts - k + 1) * d(p,q) + \sum_{a=1}^{MinPts-k+1} \varepsilon_{i,a} - \varepsilon_i}$$

$$\approx 1$$

Thus, the objects in C_1 have the similar LRD.

Now consider the ratio of the LRD of the object p_i to that of its neighbor q_j in C_2. Let reach-dist-max be the maximum reachability distance of the object q_j which is the object in C_2.

$\because MinPts - dist(p_i) > dist(C_1, C_2)$ and $d(p_i, q_{i,j}) > dist(C_1, C_2)$

$$\therefore \frac{LRD_{MinPts}(q_j)}{LRD_{MinPts}(p_i)} = \frac{\sum_{a=1}^{k} MinPts - dist(p_a) - MinPts - dist(p_i) + \sum_{a=1}^{MinPts-k+1} d(p_i, q_{i,a})}{\sum_{o \in N_{MinPts}(q_i)} reach - dist_{MinPts}(q_i, o)}$$

$$> \frac{MinPts * dist(C_1, C_2)}{MinPts * reach - dist - \max} = \frac{dist(C_1, C_2)}{reach - dist - \max}$$

$\because dist(C_1, C_2) >> reach - dist - \max$ and the appropriate pct<1

$\therefore \dfrac{LRD(q_j)}{LRD(p_i)} >> 2 > 1 + pct$. That is, objects in C2 will not be assigned to cluster C1.

Then, if objects in C_1 form a cluster which can be discovered by LDBSCAN, the inequality, $Min(LOF_{MinPts}(p_i)) \leq LOFUB$, must be satisfied.

$$\Rightarrow Min(LOF_{MinPts}(p_i)) = Min(\frac{\sum_{a=1}^{k} LRD_{MinPts}(p_a) - LRD_{MinPts}(p_i) + \sum_{a=1}^{MinPts-k+1} LRD(q_{i,a})}{MinPts * LRD_{MinPts}(p_i)})$$

$$\geq \frac{(k-1)LRD(p) + (MinPts - k + 1)LRD(q)}{MinPts * (LRD(p) + \varepsilon_i)}$$

$$\Rightarrow \frac{(k-1)LRD(p) + (MinPts - k + 1)LRD(q)}{MinPts * (LRD(p) + \varepsilon_i)} \leq LOFUB$$

$$\Rightarrow (MinPts + 1)LRD(q) - LRD(p) \leq LOFUB * MinPts * (LRD(p) + \varepsilon_i) + k * (LRD(q) - LRD(p))$$

$$\Rightarrow k \geq \frac{(MinPts + 1)LRD(q) - (LOFUB * MinPts + 1)LRD(p) - LOFUB * MinPts * \varepsilon_i}{LRD(q) - LRD(p)}$$

$$\therefore \text{LBC} = [\frac{(MinPts + 1)LRD(q) - (LOFUB * MinPts + 1)LRD(p)}{LRD(q) - LRD(p)}] + 1$$

Since the LOF of objects deep in a cluster is approximately equal to 1, the LOFUB must be greater than 1. Then

$$\text{LBC} = [\frac{(MinPts + 1)LRD(q) - (LOFUB * MinPts + 1)LRD(p)}{LRD(q) - LRD(p)}] + 1$$
$$< [\frac{(MinPts + 1)LRD(q) - (MinPts + 1)LRD(p)}{LRD(q) - LRD(p)}] + 1 = MinPts + 2$$

In other words, LBC satisfies the inequality, *LBC≤MinPts+1*, discussed in part (a). Let's consider another extreme situation. The LOFUB is so big that (LOFUB*MinPts+1)*LRD(p) is bigger than (MinPts+1)*LRD(q), and in this case LBC is less than 1. As a matter of fact, it is impossible for LBC to be less than 1. When LOFUB is big enough, the object p which is a single point outlier still satisfies the core point condition, *LOF(p)≤LOFUB*; therefore, the object *p* is deemed as a core point that should belong to a certain cluster. In this case, it forms a cluster which contains only one object by itself.

3.3 The Cluster-Based Outlier Factor

Since outliers are far more than a binary property (Breunig et al., 2000), a cluster-based outlier also needs a value to demonstrate its degree of being an outlier. In the following we give the definition of the cluster-based outlier factor.

Definition 15 (Cluster-based outlier factor): Let C_1 be a cluster-based outlier and C_2 be the nearest non-outlier cluster of C_1. The cluster-based outlier factor of C_1 is defined as

$$CBOF(C_1) = |C_1| * dist(C_1, C_2) * \sum_{p_i \in C_2} lrd(p_i) / |C_2|$$

The cluster-based outlier factor of the cluster C_1 is the result of multiplying the number of the objects in C_1 by the product of the distance between C_1 and its nearest normal cluster C_2 and the average local reachability density of C_2. The outlier factor of cluster C_1 captures the degree to which we call C_1 an outlier. Assume that C_1 as a cluster-based outlier is deviated from its nearest normal cluster C_2. It is easy to see that the more objects C_1 contains, and the farther away C_1 is from C_2, and the more dense C_2 is, the higher the *CBOF* of C_1 is and the more abnormal C_1 is.

4. Experiments

A comprehensive performance study has been conducted to evaluate our algorithm. In this section, we describe those experiments and their results. The algorithm was run on both real-life datasets obtained from the UCI Machine Learning Repository and synthetic datasets.

4.1 LDBSCAN

In this section, we will demonstrate how the proposed LDBSCAN can successfully generate clusters which appear to be meaningful that is unable to be generated by other methods.

4.1.1 A synthetic dataset with clusters resided in other clusters

In order to test the effectiveness of the algorithm, both LDBSCAN and OPTICS are applied to a data set with 473 points as shown in Figure 7. Both LDBSCAN and OPTICS can generate the magenta cluster D, the cyan cluster E, and the green cluster F. But OPTICS can only generate the cluster G which contains all the magenta, cyan, green, and pink points. And it is more reasonable to generate a cluster which only contains the pink points because of their similarity in local-density. Therefore LDBSCAN produces the similar local-density clusters instead of the clusters produced by OPTICS with local-density exceeds certain thresholds.

The result of LDBSCAN can be influenced by the choice of the parameters. There are two totally different parameters of *MinPts*. One is for the calculation of LOF and the other is for the clustering algorithm. For most of the datasets, it seems work well when *MinPts* for LOF is between 10 and 20, and more details can be found in (Breunig et al., 2000). For convenience of presentation, $MinPts_{LOF}$ is used as a shorthand of *MinPts* for LOF and $MinPts_{LDBSCAN}$ as a shorthand of *MinPts* for the clustering algorithm.

For objects deep inside a cluster, the LOFs are approximately equal to 1. The greater the LOF is, the higher possibility for the object to be an outlier. If the value that is selected for *LOFUB* is too small, some core points may be mistakenly considered as outliers; and if the value is too large, some outliers may be mistakenly considered as core points. For most of the datasets that have been experimented with, picking 1.5 to 2.5 appears to work well.

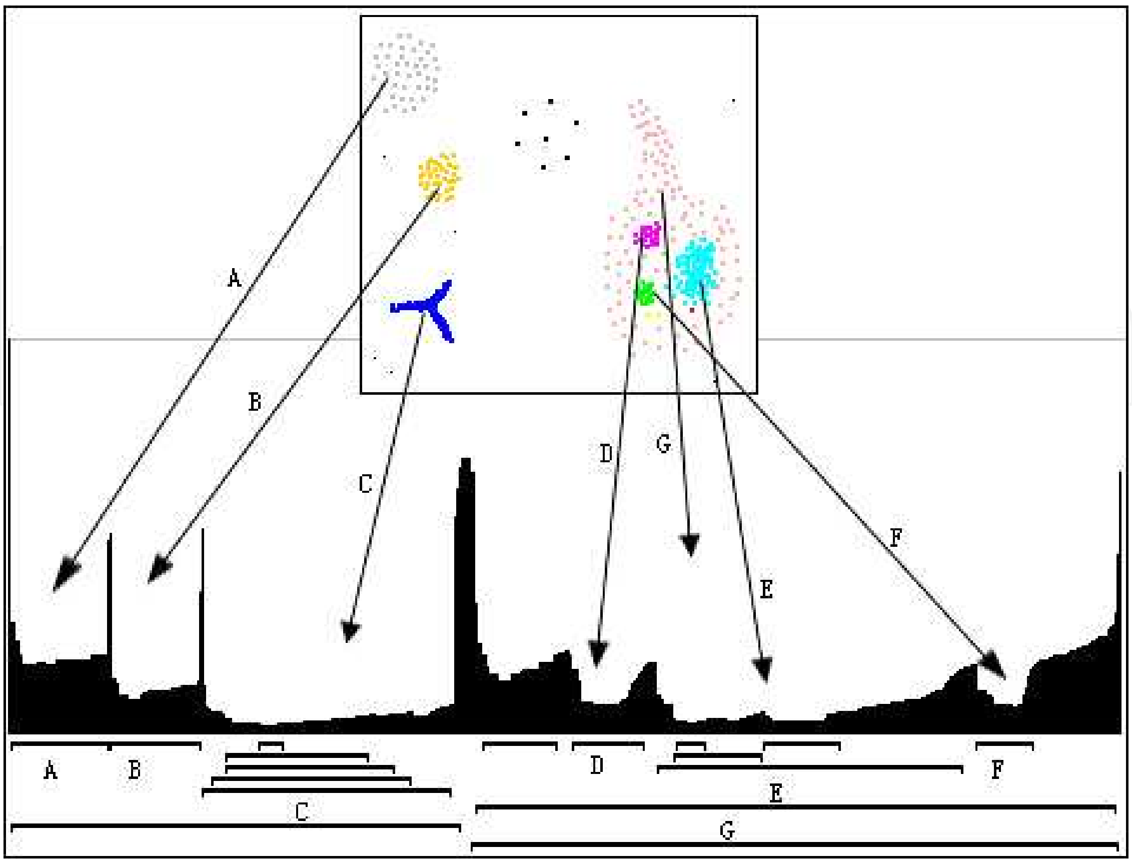


Fig. 7. Reachability-plot for a data set with hierarchical clusters of different sizes, densities and shapes

However, it also depends. For example, we identified multiple clusters, e.g., a cluster of pictures from a tennis match and the reasonable *LOFUB* is up to 7. In Figure 8, the red points are those whose LOF exceeds the *LOFUB* when $MinPts_{LOF}$*=15*.

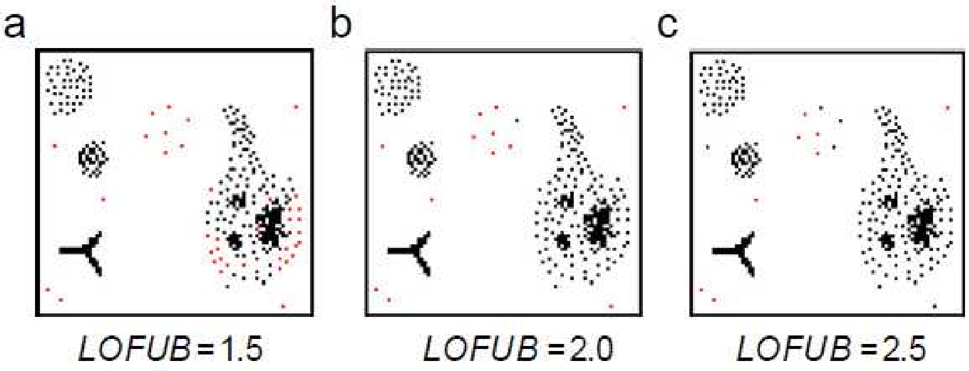


Fig. 8. Core points and outliers

Parameter *pct* controls the local-density fluctuation as it is accepted. The value of *pct* depends on the fluctuation of the cluster. Generally speaking, it is between 0.2 and 0.5. Of course in some particular situations, other values out of this range can be chosen. Let $MinPts_{LOF}$*=15,* $MinPts_{LDBSCAN}$*=10*, and *LOFUB=2.0*. Figure 9 shows the different clustering results with different values of *pct*.

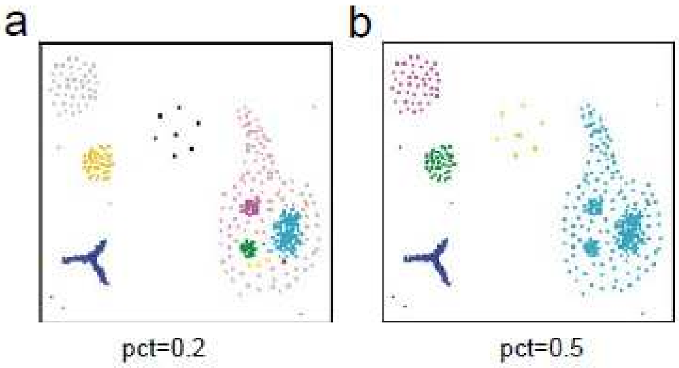


Fig. 9. Clustering results with different values of *pct.*

Parameter $MinPts_{LDBSCAN}$ determines the stand-by objects belonging to the same cluster of the core point. Clearly $MinPts_{LDBSCAN}$ can be as small as 1. However, if $MinPts_{LDBSCAN}$ is too small, some reasonable objects may be missed. Thus we suggest that $MinPts_{LDBSCAN}$ is at least 5 in order to take enough reasonable objects into account. The upper bound of $MinPts_{LDBSCAN}$ is based on a more subtle observation. Let $p \in C_1$, $q \in C_2$, C_1 has the similar density with C_2. p and q are the nearest objects between C_1 and C_2. Consider the simple situation that *distance*(C_1, C_2) is small enough shown in Figure 10, obviously that as $MinPts_{LDBSCAN}$ values increase, there will be a corresponding monotonic sequence of changes to *MinPts-distance(p)*. As the $MinPts_{LDBSCAN}$ values increase, once *MinPts-distance(p)* is greater than *distance*(C_1, C_2), C_1 and C_2 will be generated into one cluster. In Figure 10, clustering with any core point in C_1 is started. When $MinPts_{LDBSCAN}$ reaches 10, C_1 and C_2 will be generated into one cluster *C*. Therefore, the value for $MinPts_{LDBSCAN}$ should not be too large. When $MinPts_{LDBSCAN}$ reaches 15, enough candidates will be considered. The value ranges from 5 to 15 can be chosen for $MinPts_{LDBSCAN}$.

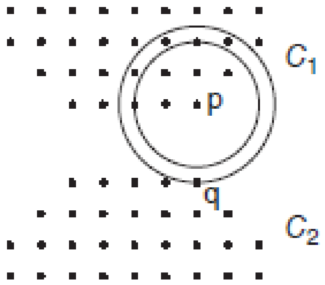


Fig. 10. Different values for $MinPts_{LDBSCAN}$.

4.1.2 Comet-like clusters

In order to demonstrate the accuracy of the clustering results of LDBSCAN, both LDBSCAN and OPTICS are applied to a 2-dimension dataset shown in the following Figure 11. LDBSCAN discovers the cluster C_1 consisting of small rectangle points, the cluster C_2 consisting of small circle points, and the outlier P_1, P_2, P_3 denoted by hollow rectangle points. OPTICS discovers the clusters whose reachability-distance falls into the dents and assigns the point to a cluster according to its reachability-distance, regardless its

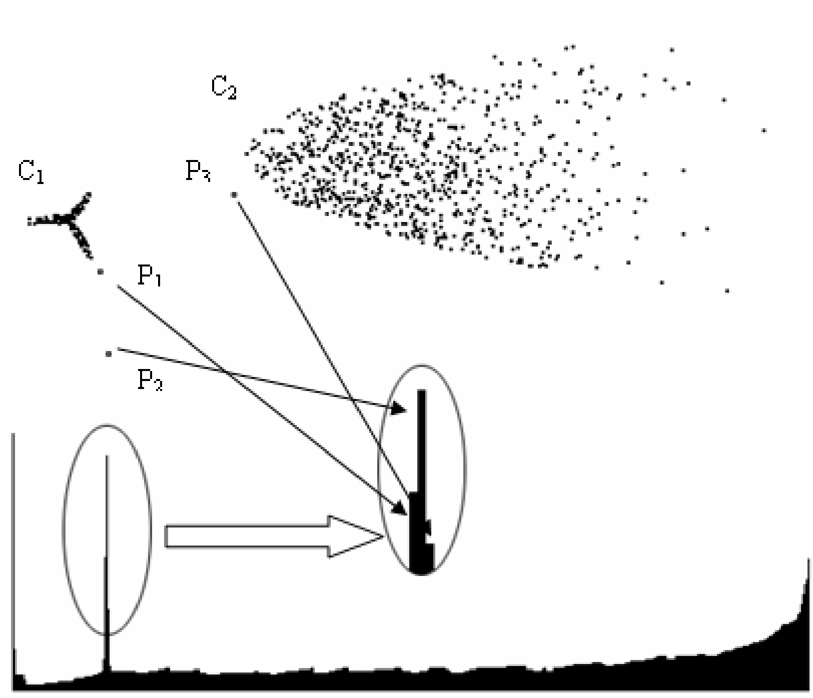


Fig. 11. Clusters with different local density borders.

neighborhood density. Because the reachability-distance of the point P_3 is similar to that of the points in the right side of the cluster C_2, the side whose density is relatively low, OPTICS would assign the point P_3 to the cluster C_2, while LDBSCAN discovers the point P_3 as an outlier due to its different local density from its neighbors. Although both OPTICS and LDBSCAN can discover the points P_1, P_2 as outliers, the clustering result of OPTICS is not accurate especially when the border density of a cluster varies, such as the comet-like cluster.

4.2 Cluster-based outliers

The performance of cluster-based outliers is tested in this section.

4.2.1 Wisconsin breast cancer data

The second used dataset is the Wisconsin breast cancer data set, which has 699 instances with nine attributes, and each record is labeled as benign (458 or 65.5%) or malignant (241 or 34.5%). In order to avoid the situation in which the local density can be ∞ if there are more than MinPts objects, different from each other, but sharing the same spatial coordinates, only 3 duplicates of certain spatial coordinates are reserved and the rest are removed. In addition, the 16 records with missing values are also removed. Therefore, the resultant dataset has 327 (57.8%) benign records and 239 (42.2%) malignant records.

The algorithm processed the dataset when *pct=0.5, LOFUB=3, MinPts=10*, and *a=0.95*. Both LOF and our algorithm find the 4 following noise records which are sing point outliers shown in Table 1. Understandably, our algorithm processes based on the result of LOF, and thus both can find the same single point outliers.

Sample code number	Value	Type	LOF
1033078	2,1,1,1,2,1,1,1,5	Benign	3.142
1177512	1,1,1,1,10,1,1,1,1	Benign	4.047
1197440	1,1,1,2,1,3,1,1,7	Benign	3.024
654546	1,1,1,1,2,1,1,1,8	Benign	4.655

Table 1. Single point outliers in Wisconsin breast cancer dataset

Besides the single point outliers, our algorithm discovers 3 clusters shown in Table 2, among which there are 2 big clusters and 1 small cluster. One big cluster A contains 296 benign records and 6 malignant records, and the other one B contains 26 benign records and 233 malignant records. The small cluster *C* contains only 1 record *p*. Among all the MinPts-nearest neighbors of the only one record in *C*, six neighbors belong to the cluster *A* and the other four belong to the cluster *B*. The record p is in the middle of cluster *A* and *B*, and *LOF(p)= 1.795*. It is closer to *A* than *B*, but has the similar local reachability density to *B* rather than *A*. Thus, it forms a cluster by itself. This kind of special record cannot be easily discovered by LOF when its *MinPts*-nearest neighborhood overlaps with more than one cluster.

Cluster Name	Number of Benign Records	Number of Malignant Records	Average Local Reachability Density
A	296	6	0.743
B	26	233	0.167
C	1	0	0.170

Table 2. Clusters in Wisconsin breast cancer dataset

4.2.2 Boston housing data

The Boston housing dataset, which is taken from the StatLib library, concerns housing values in suburbs of Boston. It contains 506 instances with 14 attributes. Before clustering, data need to be standardized in order to assign each variable an equal weight. Here the z-score process is used because using mean absolute deviation is more robust than using standard deviation (Han & Kamber, 2006). The algorithm processed the dataset when *pct=0.5, LOFUB=2, MinPts=10*, and *a=0.9*. One single point outlier, 3 normal clusters and 6 cluster-based outliers are discovered. There are few single point outliers in this dataset. The maximum LOF, the value of the 381st record, is 2.624 which indicates that there is not a significant deviation. In addition, the 381st record is assigned to the 9th cluster which is a cluster-based outlier. Its LOF exceeds *LOFUB* due to the small number of the objects contained in the 9th cluster to which it belongs. The small number, which is less than *MinPts*, would affect the accuracy of LOF. Eight of all the nine records whose LOF exceeds *LOFUB* are assigned to a certain cluster and the LOF of the only single point outlier, the 215th record, is 2.116. The 215th record has a smaller proportion of owner-occupied units built prior to 1940, the 7th attribute, than its neighbors.

However, the 6 cluster-based outliers are more interesting than the only single point outlier. Table 3 demonstrates the information of all the 9 clusters, and the additional information of the cluster-based outliers is shown in Table 4. The 3rd cluster, which is a cluster-based outlier and has the maximum CBOF, deviates from the 1st cluster. Its 12th attribute, 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town, is much lower than that of the 1st cluster. Both the 9th cluster and the 6th cluster deviate from the 1st cluster. Although the 6th cluster contains more object than the 9th cluster, the CBOF of the 6th cluster is less than that of the 9th cluster because the 9th cluster is farther away from the 1st cluster than the 6th cluster. The records in the 9th cluster have significantly big per capita crime rate by town, comparing with those of the 1st cluster. However, it is not easy to do not differentiate the records in the 6th cluster from those of the 1st cluster. Moreover, the relationship between the 4th cluster and the 8th cluster is also impressive. There are 35 records which show that its tract bounds the Charles River, demonstrated by the 4th attribute, in the whole dataset, and 34 of them is discovered in the 4th cluster. The only exceptional record, the 284th record, has a slightly high proportion of residential land zoned for lots over 25,000 square feet, the 2nd attribute, and a relatively low proportion of non-retail business acres per town, the 3rd attribute. The area denoted by the 284th record is more like a residential area than the other areas along the Charles River.

Cluster Id	Number of Records	Average Local Reachability Density
1	82	0.556
2	345	0.528
3	26	0.477
4	34	0.266
5	1	0.303
6	9	0.228
7	1	0.228
8	1	0.155
9	6	0.127

Table 3. Clusters in Boston housing dataset

Cluster Id	CBOF	Nearest cluster	dist(C1, C2)	The nearest object pair	The contained records
3	54.094	1	3.744	436--445	412,416,417,420,424,425, 426,427,429,430,431,432, 433,434,435,436,437,438, 439,446,451,455,456,457, 458,467
9	24.514	1	7.353	415--385	381,406,411,415,419, 428
6	20.005	1	4.000	399--401	366,368,369,372,399, 405,413,414,418
7	2.452	2	4.648	103--35	103
5	2.269	1	4.084	410--461	410
8	1.468	4	5.522	284--283	284

Table 4. Cluster-based outliers in Boston housing dataset

5. Conclusion

In this chapter, we have examined various density-based techniques, DBSCAN, OPTICS, LOF, LDBSCAN and cluster-based outlier detection, and have described several applications of these techniques. Clustering is a process of grouping data based on a measure of similarity, and outlier detection is a process of discovering the data objects which are grossly different from or inconsistent with the remaining set of data. Both clustering and outlier detection is a subjective process; the same set of data often needs to be processed differently for different applications. This subjectivity makes the process of clustering and outlier detection hard. That is why a single algorithm or approach is not adequate to solve all the problems.

The most challenging step is feature extraction and pattern representation. In this chapter, the step of pattern representation is conveniently avoided by assuming the pattern representations are available as input to the clustering and outlier detection algorithm. Especially in the case of large data sets, it is difficult for the user to keep track of the importance of each feature. Comparing with partitioning and hierarchical methods, density-based methods stand out both in discovering clusters with arbitrary shape and in outlier detection. Among them, the OPTICS and LDBSCAN are most successful used due to their accuracy. They can effectively discover clusters with different local density. In summary, clustering and outlier detection is an interesting, useful and challenging problem. Density-based techniques are good at accuracy; however, the potential can only be exploited after making several designed choices carefully.

6. References

Ankerst, M.; Breunig, M. M. ; Kriegel, H.-P. & Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. In Proceedings of the 1999 ACM SIGMOD international conference on Management of data (SIGMOD '99). ACM, New York, NY, USA, 49-60.

Breunig, M. M. ; Kriegel, H.-P. ; Ng, R. T. & Sander, J. (2000), LOF: identifying density-based local outliers, Proceedings of the 2000 ACM SIGMOD international conference on Management of data, p.93-104, May 15-18, 2000, Dallas, Texas, United States.

Duan, L. ; Xu, L. ; Guo, F. ; Lee, J. & Yan, B. (2007). A local-density based spatial clustering algorithm with noise. Inf. Syst. 32, 7 (November 2007), 978-986.

Duan, L. ; Xu, L. ; Liu, Y. & Lee, J. (2009). Cluster-based Outlier Detection. Annals of Operations Research. Vol 168, No. 1, pp. 151-168.

Ester, M. ; Kriegel, H.-P. ; Sander, J. & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, E., Han, J., Fayyad, U.M. (Eds.), Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI, Menlo Park, CA. pp. 226-231.

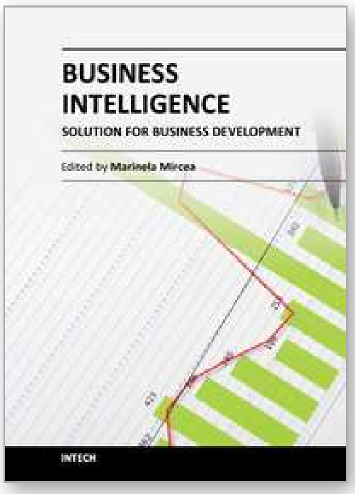
Guha, S.; Rastogi, R. & Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. In Proceedings of the 1998 ACM SIGMOD international conference on Management of data (SIGMOD '98), Ashutosh Tiwary and Michael Franklin (Eds.). ACM, New York, NY, USA, 73-84.

Han, J., and Kamber, M. (2006). Data Mining: Concepts and Techniques. Elsevier.

Hinneburg, A. & Keim, D. A. (1998). An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining, New York, NY, pp. 58-65.

Kaufman, L.; Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley & Sons.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Math. Statist, Prob., 1: 281-297.



BUSINESS INTELLIGENCE
SOLUTION FOR BUSINESS DEVELOPMENT
Edited by Marinela Mircea
INTECH

Business Intelligence - Solution for Business Development
Edited by Dr. Marinela Mircea

ISBN 978-953-51-0019-5
Hard cover, 108 pages
Publisher InTech
Published online 01, February, 2012
Published in print edition February, 2012

The work addresses to specialists in informatics, with preoccupations in development of Business Intelligence systems, and also to beneficiaries of such systems, constituting an important scientific contribution. Experts in the field contribute with new ideas and concepts regarding the development of Business Intelligence applications and their adoption in organizations. This book presents both an overview of Business Intelligence and an in-depth analysis of current applications and future directions for this technology. The book covers a large area, including methods, concepts, and case studies related to: constructing an enterprise business intelligence maturity model, developing an agile architecture framework that leverages the strengths of business intelligence, decision management and service orientation, adding semantics to Business Intelligence, towards business intelligence over unified structured and unstructured data using XML, density-based clustering and anomaly detection, data mining based on neural networks.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821