Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Novel Density-Based Clustering Algorithms for Uncertain Data

Xianchao Zhang and **Han Liu** and **Xiaotong Zhang** and **Xinyue Liu**
School of Software Technology
Dalian University of Technology
Dalian 116620, China
xczhang@dlut.edu.cn, liu.han.dut@gmail.com, zxt.dut@hotmail.com, xyliu@dlut.edu.cn

Abstract

Density-based techniques seem promising for handling data uncertainty in uncertain data clustering. Nevertheless, some issues have not been addressed well in existing algorithms. In this paper, we firstly propose a novel density-based uncertain data clustering algorithm, which improves upon existing algorithms from the following two aspects: (1) it employs an exact method to compute the probability that the distance between two uncertain objects is less than or equal to a boundary value, instead of the sampling-based method in previous work; (2) it introduces new definitions of core object probability and direct reachability probability, thus reducing the complexity and avoiding sampling. We then further improve the algorithm by using a novel assignment strategy to ensure that every object will be assigned to the most appropriate cluster. Experimental results show the superiority of our proposed algorithms over existing ones.

Introduction

Clustering plays an important role in many fields such as pattern recognition and data mining. Traditional clustering algorithms deal with certain data. However, in many real applications only uncertain data is available, such as biomedical measurement (Liu et al. 2005), sensor networking (Deshpande et al. 2005), motion tracking (Trajcevski et al. 2004), financial and market data analysis, meteorological forecasting and so on (Aggarwal 2009). Uncertain data has posed a huge challenge to traditional clustering algorithms.

Several algorithms for uncertain data clustering have been proposed. Partition-based algorithms, e.g., UK-means (Chau et al. 2006), UK-medoids (Gullo, Ponti, and Tagarelli 2008), extend traditional clustering algorithms k-means and k-medoids by use of expected distance or uncertain distance. However, these partition-based approaches could not handle the uncertain information well (Gullo and Tagarelli 2012). Density-based algorithms, e.g., FDBSCAN (Kriegel and Pfeifle 2005a), do not suffer from the issues of the partition-based algorithms. Nevertheless, there still exist several problems that have not been addressed well in FDBSCAN, which is the foundation of other density-based algorithms.

In this paper, we firstly propose a novel density-based uncertain data clustering algorithm which improves upon existing algorithms from the following two aspects: (1) it employs an exact method to compute the probability that the distance between two uncertain objects is less than or equal to a boundary value, instead of the sampling-based method in previous work; (2) it introduces new definitions of core object probability and direct reachability probability, thus reducing the complexity and avoiding sampling. We then further improve the algorithm by introducing maximal direct reachability probability instead of the fixed threshold used in previous work to guarantee that every object will be assigned to the most appropriate cluster. Experimental results show the superiority of our proposed algorithms over existing ones.

Related Work

We briefly review the main algorithms for uncertain data clustering. A comprehensive survey of uncertain data mining could be found in (Aggarwal and Yu 2009).

Partition-based Algorithms

One of the earliest attempts to solve the problem of uncertain data clustering is UK-means (Chau et al. 2006). It is an adaptation of k-means by use of expected distance instead of accurate distance. (Ngai et al. 2006), (Kao et al. 2008), (Kao et al. 2010), (Ngai et al. 2011) and (Lukic, Köhler, and Slavek 2012) improve the efficiency of UK-means with some pruning techniques to avoid the computation of redundant expected distances. CK-means (Lee, Kao, and Cheng 2007) is a variant of UK-means which resorts to the moment of inertia of rigid bodies in order to reduce the time for computing expected distances. UK-medoids (Gullo, Ponti, and Tagarelli 2008) employs uncertain distance for uncertain data and exploits a k-medoids scheme. (Cormode and McGregor 2008) proposes guaranteed approximation algorithms for clustering uncertain data by using k-means, k-median and k-center. MMvar (Gullo, Ponti, and Tagarelli 2010) takes a criterion based on the minimization of the variance of cluster mixture models. However, partition-based approaches could not handle the uncertain information well (Gullo and Tagarelli 2012).

Density-based Algorithms

The fuzzy version of the DBSCAN (Ester et al. 1996) algorithm, FDBSCAN (Kriegel and Pfeifle 2005a), is the

foundation of other density-based algorithms for uncertain data clustering. FOPTICS (Kriegel and Pfeifle 2005b) is extended from the hierarchical density-based clustering algorithm OPTICS (Ankerst et al. 1999) to deal with uncertain data. (Günnemann, Kremer, and Seidl 2010) extends the density-based algorithm to subspace clustering for high dimensional uncertain data. Density-based algorithms do not suffer from the issues of the partition-based algorithms, thus they seem more promising for uncertain data. Nevertheless, as we point out in the following section, there still exist several problems that have not been addressed well.

FDBSCAN Issues

In this section, we review the FDBSCAN algorithm and point out several issues that have not been addressed well.

FDBSCAN Basics

In general, data uncertainty can be considered at table, tuple or attribute level, and is usually specified by fuzzy models, evidence-oriented models, or probabilistic models (Sarma et al. 2009). Here we focus on the attribute-level uncertainty in a probabilistic model. In particular, an uncertain object is represented by a probability density function (pdf), which describes the probability that the object appears at any position in a multidimensional space.

Definition 1 *Distance Density Function: o and o' are two objects in database D. Let $d : D \times D \rightarrow IR_0^+$ be a distance function, and let $P(a \leqslant d(o,o') \leqslant b)$ denote the probability that $d(o,o')$ is between a and b. Then a probability density function $p_d : D \times D \rightarrow (IR_0^+ \rightarrow IR_0^+ \cup \infty)$ is called a distance density function if the following condition holds:*

$$P(a \leqslant d(o,o') \leqslant b) = \int_a^b p_d(o,o')dx \tag{1}$$

Definition 2 *Distance Distribution Function: o and o' are two objects in database D. Let $d : D \times D \rightarrow IR_0^+$ be a distance function, and let $P(d(o,o') \leqslant b)$ denote the probability that $d(o,o')$ is smaller than b. Then a probability distribution function $P_d : D \times D \rightarrow (IR_0^+ \rightarrow [0...1])$ is called a distance distribution function if the following condition holds:*

$$P_d(o,o')(b) = P(d(o,o') \leqslant b) \tag{2}$$

From definition 1, $P_d(o,o')(b) = \int_{-\infty}^b p_d(o,o')dx$.

Definition 3 *Core Object Probability: Let D be a database, and let $P_d : D \times D \rightarrow (IR_0^+ \rightarrow [0...1])$ be a distance distribution function. Then, the core object probability of an object o is defined as:*

$$P^{core}_{Eps,MinPts,d,D}(o) = \sum_{\substack{A \subseteq D \\ |A| \geqslant MinPts}} \prod_{p \in A} P_d(p,o)(Eps) \prod_{p' \in D \setminus A} (1 - P_d(p',o)(Eps)) \tag{3}$$

where Eps denotes a distance threshold, $MinPts$ denotes the minimum number of objects contained in the Eps-range of a core object. From this definition, we can get that the core object probability $P^{core}_{Eps,MinPts,d,D}(o)$ is equal to the probability value $P(|N_{Eps}(o)| \geqslant MinPts)$, which indicates the likelihood that o is a core object (Kriegel and Pfeifle 2005a). $|N_{Eps}(o)|$ denotes the number of the objects in the Eps-range of o. If $P^{core}_{Eps,MinPts,d,D}(o) \geqslant 0.5$, object o can be regarded as a core object.

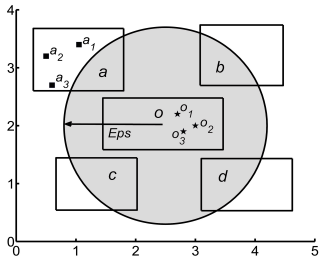


Figure 1: Issues of FDBSCAN

Definition 4 *Reachability Probability: Let D be a database, and let $P_d : D \times D \rightarrow (IR_0^+ \rightarrow [0...1])$ be a distance distribution function. p and o are two objects in D. Then, the reachability probability of p w.r.t. o is defined as follows:*

$$P^{reach}_{Eps,MinPts,d,D}(p,o) = P^{core}_{Eps,MinPts-1,d,D\setminus\{p\}}(o) \cdot P_d(p,o)(Eps) \tag{4}$$

If $P^{reach}_{Eps,MinPts,d,D}(p,o) \geqslant 0.5$, object p can be regarded as directly density-reachable to object o.

Based on the above definitions, FDBSCAN extends the traditional DBSCAN algorithm for handling uncertain data.

Unaddressed Issues

The following issues are not addressed well in FDBSCAN.

Losing uncertain information: FDBSCAN does not provide an exact function for calculating $p_d(o,o')$. Instead, it uses the sampling method to calculate the probability that the distance between two uncertain objects is less than or equal to a boundary value. However, sampling may lose some uncertain information, even cause wrong results. Take an example in Figure 1. a and o are two uncertain objects, the rectangles are their uncertain regions, Eps is a distance threshold. We need to calculate the probability that the distance between a and o is less than or equal to Eps. The sampling rate is 3. If the sampling objects of a are a_1, a_2, a_3 and the sampling objects of o are o_1, o_2, o_3, then the probability we want to calculate equals 0. But the real distance between a and o is probably less than or equal to Eps, i.e., the probability should not be 0.

High time complexity: When computing the core object probability, FDBSCAN needs to determine for each subset A of D having a cardinality higher than $MinPts$, the probability that only the objects of A are within an Eps-range of o but no objects of $D \setminus A$ (Kriegel and Pfeifle 2005a). So the number of the subsets we need to consider is $C_{|D|}^{MinPts} + C_{|D|}^{MinPts+1} + ... + C_{|D|}^{|D|} = 2^{|D|} - C_{|D|}^0 - C_{|D|}^1 - ... - C_{|D|}^{MinPts-1}$, where C is the combinatorial symbol in the binomial formula, $MinPts$ denotes the minimum number

of objects contained in the Eps-range of a core object and $|D|$ denotes the object number of the whole database. Thus we nearly need to find every subset of the whole database if $MinPts$ is very small. Take an example in Figure 1. a, b, c, d and o are five uncertain objects, the rectangles are their uncertain regions. a, b, c and d are perhaps in the Eps-range of o, $MinPts$ is 2, if we want to get the core object probability of o, we have to consider $2^5 - C_5^0 - C_5^1 = 26$ cases. Assume that there are 500 uncertain objects in the Eps-range of o and $MinPts$ is very small, then the number of the subsets we need to consider will be as large as 2^{500}, so the computation is too time consuming.

Nonadaptive threshold: FDBSCAN applies a fixed threshold (f_value) to judge whether an object is a core object and whether an object is directly density-reachable. The fixed threshold may cause error. For example, there are two objects p, q and only one cluster T, we set $f\_value = 0.5$, the direct reachability probability between p and any core object in T is 0.51, the direct reachability probability between q and any core object in T is 0.49. Because $f\_value = 0.5$, p could be assigned to T, q could not be assigned to T, though the direct reachability probability gap between p and q is very small. Obviously it is not reasonable.

The Proposed Algorithms

In this section, we describe our new clustering algorithms for uncertain data. Firstly we give some necessary definitions. Secondly we present the basic algorithm. Thirdly we propose an improved version of the algorithm by introducing a new assignment strategy. Finally we analyze the time complexity of the proposed algorithms.

Definitions

Consider a set of uncertain objects $D = \{o_1, o_2, ..., o_n\}$ in m-dimensional independent space R^m with a distance function $d : R^m \times R^m \to R$ defining the distance $d(o_i, o_j) \geqslant 0$ between any objects $o_i, o_j \in D$. Each uncertain object o_i is associated with a probability density function (pdf) $f_i : R^m \to R$, which gives the probability density of o_i in the uncertain region. The proposed algorithms are based on the following definitions.

Definition 5 *Given two uncertain objects o_i, o_j, whose associated pdfs are f_i, f_j respectively, x is the uncertain dimensionalities of o_i, y is the uncertain dimensionalities of o_j, objects and dimensionalities are independent of each other respectively, Eps is a distance threshold. Then the probability of $d(o_i, o_j) \leqslant Eps$, denoted by $P_{d(o_i,o_j)\leqslant Eps}$, is defined as:*

$$\begin{cases} P_{d(o_i,o_j)\leqslant Eps} = \int\limits_{x\in R^m}\int\limits_{y\in R^m} f_i(x)\cdot f_j(y)dxdy \\ \forall d(o_i,o_j) \leqslant Eps \end{cases} \quad (5)$$

Here $f_i(x) \cdot f_j(y)$ can be regarded as the joint probability density function. $d(o_i, o_j) \leqslant Eps$ is the additional limiting condition for determining the integral interval.

Take an example in Figure 2, suppose there are two uncertain objects o_v and o_w, either of them is in a linear uncertain region with uniform distribution. The object o_v moves

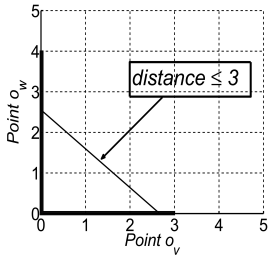


Figure 2: Computation of Equation 5

on x axis in $[0, 3]$ and o_w moves on y axis in $[0, 4]$. We need to compute the probability that the distance between o_v and o_w is less than or equal to 3. The pdfs of o_v and o_w are $f_v = 1/3$ and $f_w = 1/4$ respectively, the constraint condition is $d(o_v, o_w) \leqslant 3$, so the probability can be computed as $P_{d(o_v,o_w)\leqslant 3} = \iint f_v(x) \cdot f_w(y)dxdy = 1/12 \cdot \int_0^3 dx \int_0^{\sqrt{9-x^2}} dy = \frac{9\pi}{48}$.

Definition 6 *o_p is an uncertain object in database D, $\forall o_i \in D$, then the probability Eps-neighborhood of o_p, denoted by $PNeighborhood(o_p)$, is defined as:*

$$PNeighborhood(o_p) \leftarrow \left\{o_i | P_{d(o_i,o_p)\leqslant Eps} > 0\right\} \quad (6)$$

Definition 7 *o_p is an uncertain object in database D, $\forall o_i \in D$, then $PN_{Eps}(o_p)$ could be defined as:*

$$PN_{Eps}(o_p) = \sum_{o_i \in PNeighborhood(o_p)} P_{d(o_i,o_p)\leqslant Eps} \quad (7)$$

Set a parameter $MinPts$ which denotes the minimum number of objects contained in the Eps-range of a core object, if $PN_{Eps}(o_p) \geqslant MinPts$, o_p can be treated as a core object.

Definition 8 *o_p is an uncertain object in database D, $\forall o_i \in D$, then the core object probability of o_p, denoted by $P^{core}_{Eps,MinPts,D}(o_p)$, is defined as:*

$$P^{core}_{Eps,MinPts,D}(o_p) = PN_{Eps}(o_p) / \left|N_{Eps}(o_p)\right| \quad (8)$$

where $|N_{Eps}(o_p)|$ is the number of the objects in $PNeighborhood(o_p)$.

Definition 9 *Given two uncertain objects o_p and o_q in database D, o_p is a core object, then the direct reachability probability of o_q w.r.t. o_p, denoted by $P^{dir-reach}_{Eps,MinPts,D}(o_q, o_p)$, is defined as:*

$$\begin{aligned} P^{dir-reach}_{Eps,MinPts,D}(o_q,o_p) &= P^{core}_{Eps,MinPts-1,D\setminus\{o_q\}}(o_p) \\ &\quad \cdot P_{d(o_q,o_p)\leqslant Eps} \\ &= (PN_{Eps}(o_p) - P_{d(o_q.o_p)\leqslant Eps})/ \\ &\quad (|N_{Eps}(o_p)| - 1)\cdot P_{d(o_q,o_p)\leqslant Eps} \end{aligned} \quad (9)$$

Explanation: $P^{core}_{Eps,MinPts-1,D\setminus\{o_q\}}(o_p)$ means the probability that at least $MinPts-1$ objects from $D\setminus o_q$ are located in the Eps-range of o_p; $P_{d(o_q,o_p)\leqslant Eps}$ means the probability that the distance between o_p and o_q is less than or equal to Eps. As these two events are independent of each other, their product means the probability that at least

$MinPts$ objects from D are located in the Eps-range of o_p and o_q is one of them.

Through these definitions, we can use the new methods to compute core object probability and direct reachability probability, without finding most subsets of the whole dataset and using any sampling method, thus reducing the complexity and avoiding the loss of uncertain information.

PDBSCAN Algorithm

Our proposed algorithm is a probabilistic density-based uncertain data clustering algorithm, called PDBSCAN. It is based on the principle that a cluster is a set of objects which are directly density-reachable from an arbitrary core object in the cluster.

Algorithm 1 and Algorithm 2 show the details of PDBSCAN algorithm and the expand_cluster procedure. $clu\_num = k$ means the current cluster number is k, k is a positive integer. $class(i) = 0$, -1 or $1...k$ respectively means that the object o_i currently does not belong to any cluster, has been determined to belong to noise or belong to cluster 1, ..., cluster k. $type(i) = 0$, -1 or 1 respectively means the object o_i is a border object, a noise object or a core object. $visited(i) = 1$ or 0 respectively means the object o_i has been processed or not.

After a preliminary phase (Lines 1-2), PDBSCAN starts from an unvisited object o_p, calculates the corresponding $PNeighborhood$ and PN_{Eps} (Lines 3-5). If PN_{Eps} equals 1, it indicates that the checked object is the only object in its Eps-range, obviously it is a noise object (Lines 6-7). If PN_{Eps} is between 1 and $MinPts$, the information is not enough to straightly judge the type of the checked object, PDBSCAN does not need to do anything here, so this if-condition is left out in the algorithm. If PN_{Eps} is greater than or equal to $MinPts$, it means that the checked object is a core object, then PDBSCAN adds the objects whose direct reachability probability from the checked object are greater than or equal to the given threshold (f_value) as the cluster members (Lines 8-16), and calls the expand_cluster procedure which further expands the cluster by adding the objects whose direct reachability probability from the core objects in the cluster members are greater than or equal to the given threshold (f_value) (Line 17). After the expansion procedure, if no objects can be added to the current cluster, the algorithm finds another core object, adds cluster members and expands the cluster. The procedure is repeated until no objects can be added to any cluster, and the remaining objects are treated as outliers (Lines 21-23).

Improved PDBSCAN Algorithm

PDBSCAN has addressed the first two issues in FDBSCAN. We continue to solve the third issue by using the maximal direct reachability probability instead of the fixed direct reachability probability threshold. The definition of the maximal direct reachability probability is as below.

Definition 10 *o_i is an uncertain object in database D, for an arbitrary core object $o_p \in D$, the maximal direct reachability probability of o_i, denoted by $P^{dir-reach-\max}_{Eps,MinPts,D}(o_i)$, is defined as:*

$$P^{dir-reach-\max}_{Eps,MinPts,D}(o_i) \leftarrow \max\left\{P^{dir-reach}_{Eps,MinPts,D}(o_i,o_p)\right\}$$

Algorithm 1 PDBSCAN

Input: Uncertain dataset $D = \{o_1, o_2, ..., o_n\}$, Eps, $MinPts$, f_value
Output: A set of clusters, types of all objects in D

1: Compute the probability $P_{d(o_i,o_j)\leqslant Eps}, \forall o_i, o_j \in D$ by Equation 5
2: Initialization: $\forall o_i \in D, class(i) = 0, type(i) = 0, visited(i) = 0, clu\_num = 1$
3: **for** each unvisited object o_p in dataset D **do**
4: Compute $PNeighborhood(o_p)$ by Equation 6
5: Compute $PN_{Eps}(o_p)$ by Equation 7
6: **if** $PN_{Eps}(o_p) = 1$ **then**
7: $class(p) \leftarrow -1$,$type(p) \leftarrow -1$,$visited(p) \leftarrow 1$
8: **else if** $PN_{Eps}(o_p) \geqslant MinPts$ **then**
9: $class(p) \leftarrow clu\_num$,$type(p) \leftarrow 1$,$visited(p) \leftarrow 1$
10: Compute $P^{dir-reach}_{Eps,MinPts,D}(o_i, o_p), \forall o_i \in PNeighborhood(o_p)$
11: **for** $o_i \in PNeighborhood(o_p)$ **do**
12: **if** $P^{dir-reach}_{Eps,MinPts,D}(o_i, o_p) \geqslant f\_value$ **then**
13: $class(i) \leftarrow clu\_num$
14: $PNeighborhood(o_p)' \leftarrow \{o_i | P^{dir-reach}_{Eps,MinPts,D}(o_i, o_p) \geqslant f\_value\}$
15: **end if**
16: **end for**
17: **Expand_cluster**$(PNeighborhood(o_p)', clu\_num, f\_value, MinPts)$
18: $clu\_num \leftarrow clu\_num + 1$
19: **end if**
20: **end for**
21: **for** each o_i which satisfies $class(i) = 0$ **do**
22: $class(i) \leftarrow -1$, $type(i) \leftarrow -1$,$visited(i) \leftarrow 1$
23: **end for**

Algorithm 2 $Expand\_cluster(PNeighborhood(o_p)', clu\_num, f\_value, MinPts)$

1: **for** each object o_q in $PNeighborhood(o_p)'$ **do**
2: **if** o_q is unvisited **then**
3: $visited(q) \leftarrow 1$
4: Compute $PNeighborhood(o_q)$ by Equation 6
5: Compute $PN_{Eps}(o_q)$ by Equation 7
6: **if** $PN_{Eps}(o_q) \geqslant MinPts$ **then**
7: $class(q) \leftarrow clu\_num, type(q) \leftarrow 1$
8: Compute $P^{dir-reach}_{Eps,MinPts,D}(o_i, o_q), \forall o_i \in PNeighborhood(o_q)$
9: **for** $o_i \in PNeighborhood(o_q)$ **do**
10: **if** $P^{dir-reach}_{Eps,MinPts,D}(o_i, o_q) \geqslant f\_value$ **then**
11: $class(i) \leftarrow clu\_num$
12: $PNeighborhood(o_q)' \leftarrow \{o_i | P^{dir-reach}_{Eps,MinPts,D}(o_i, o_q) \geqslant f\_value\}$
13: **end if**
14: **end for**
15: **end if**
16: $PNeighborhood(o_p)' \leftarrow PNeighborhood(o_p)' \cup PNeighborhood(o_q)'$
17: **else if** o_q is visited **then**
18: Remove o_q from $PNeighborhood(o_p)'$
19: **end if**
20: **end for**

Algorithm 3 PDBSCANi

Input: Uncertain dataset $D = \{o_1, o_2, ..., o_n\}$, Eps, $MinPts$
Output: A set of clusters, types of all objects in D

1-10: The same with lines 1-10 in PDBSCAN, the only change is that PDBSCANi adds $P^{dir-reach-\max}_{Eps,MinPts,D}(o_i)$ and its initialization value is 0
11: **for** $o_i \in PNeighborhood(o_p)$ **do**
12: **if** $P^{dir-reach}_{Eps,MinPts,D}(o_i, o_p) > P^{dir-reach-\max}_{Eps,MinPts,D}(o_i)$ **then**
13: $class(i) \leftarrow clu\_num$
14: $PNeighborhood(o_p)' \leftarrow \{o_i | P^{dir-reach}_{Eps,MinPts,D}(o_i, o_p) > P^{dir-reach-\max}_{Eps,MinPts,D}(o_i)\}$
15: $P^{dir-reach-\max}_{Eps,MinPts,D}(o_i) \leftarrow P^{dir-reach}_{Eps,MinPts,D}(o_i, o_p)$
16: **end if**
17: **end for**
18: **Expand_cluster_max**$(PNeighborhood(o_p)', clu\_num, P^{dir-reach-\max}_{Eps,MinPts,D}(o_i), MinPts)$
19-24: The same with lines 18-23 in PDBSCAN

Algorithm 4 $Expand\_cluster\_max(PNeighborhood(o_p)', clu\_num, P^{dir-reach-\max}_{Eps,MinPts,D}(o_i), MinPts)$

1-9: The same with lines 1-9 in $Expand\_cluster$
10: **if** $P^{dir-reach}_{Eps,MinPts,D}(o_i, o_q) > P^{dir-reach-\max}_{Eps,MinPts,D}(o_i)$ **then**
11: $class(i) \leftarrow clu\_num$
12: $PNeighborhood(o_q)' \leftarrow \{o_i | P^{dir-reach}_{Eps,MinPts,D}(o_i, o_q) > P^{dir-reach-\max}_{Eps,MinPts,D}(o_i)\}$
13: $P^{dir-reach-\max}_{Eps,MinPts,D}(o_i) \leftarrow P^{dir-reach}_{Eps,MinPts,D}(o_i, o_q)$
14: **end if**
15-21: The same with lines 14-20 in $Expand\_cluster$

If o_i is a core object, then $P^{dir-reach-\max}_{Eps,MinPts,D}(o_i) = 1$; If o_i is a noise object, then $P^{dir-reach-\max}_{Eps,MinPts,D}(o_i) = 0$.

Algorithm 3 and Algorithm 4 show the improved version of PDBSCAN (PDBSCANi) and the new expand_cluster procedure. For PDBSCANi, the algorithm framework is similar to PDBSCAN. Because of the limited space, the same part has been left out in the algorithm description. The important change is when finding cluster members, for each object o_i, with the update of the maximal direct reachability probability of o_i, PDBSCANi keeps recording the corresponding core object o_p from which o_i gets the maximal direct reachability probability, further assigns o_i to the corresponding cluster which o_p belongs to.

Through this way, we can address the nonadaptive threshold issue and guarantee that every object will be assigned to the most appropriate cluster, thus further improving the performance.

Time Complexity

For PDBSCAN, let n denote the number of uncertain objects, m denote the dimensionality of the uncertain data, S denote the number of independent probability density functions employed for representing probability distributions. In the preparation phase, the computation of the probability $P_{d(o_i,o_j)\leqslant Eps}$ for $\forall o_i, o_j \in D$ has a time complexity of $O(n^2 m S^2)$. During the main loop, in the worst case we need n scans, the time complexity is $O(n)$. Then the overall time complexity of the algorithm is $O(n^2 m S^2)$. PDBSCANi has the same complexity as PDBSCAN.

Experiments

Settings

We use 7 UCI[1] benchmark datasets for evaluation. The description of the datasets is shown in Table 1. These datasets are originally established as collections of data with deterministic values, we follow the method in (Gullo et al. 2008) to generate uncertainty in these datasets. We generate uncertainty with three kinds of distribution: uniform distribution, normal distribution and Laplace distribution.

We compare PDBSCAN and PDBSCANi with existing typical uncertain data clustering algorithms, UK-means, CK-means, UK-medoids, FDBSCAN and FOPTICS. For UK-means, CK-means and UK-medoids, the sets of initial centroids are randomly selected. Therefore, to avoid that the clustering results are affected by random chance, we average the results over 100 different runs. For FDBSCAN and FOPTICS, we follow the methods in (Kriegel and Pfeifle 2005a) and (Kriegel and Pfeifle 2005b) respectively and choose a sampling rate of $s = 30$. For FDBSCAN, FOPTICS, PDBSCAN and PDBSCANi, these algorithms are sensitive to parameters, so we adjust the parameters continuously until the accuracy of each method becomes the best and stable, the method of determining the parameters could refer to (Ester et al. 1996) and (Ankerst et al. 1999).

[1] http://archive.ics.uci.edu/ml/

Table 1: Datasets used in the experiment

Dataset	Objects	Attributes	Classes
Iris	150	4	3
Wine	178	13	3
Glass	214	9	6
Ecoli	327	7	5
Yeast	1484	8	10
Image	2310	19	7
Abalone	4124	7	17

Accuracy

We use purity (Manning, Raghavan, and Schütze 2008), which is one of the most commonly used criteria, to evaluate the accuracy of the clustering results. To compute purity, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned objects and dividing by the whole number. Formally:

$$purity(\Omega, C) = \frac{1}{N} \sum_k \max_j |w_k \cap c_j| \qquad (10)$$

where $\Omega = \{w_1, w_2, ..., w_k\}$ is the set of clusters and $C = \{c_1, c_2, ..., c_k\}$ is the set of classes.

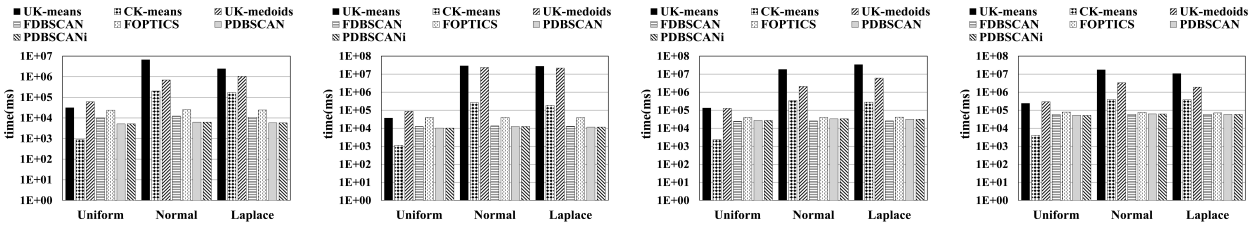
Table 2 shows the accuracy results. The last three rows of this table report, for each algorithm, (i) the score for each type of pdf averaged over all datasets (for short, Avg.); (ii) the score averaged over all datasets and pdfs (for short, all avg.score); (iii) the overall gain of PDBSCANi computed as the difference between the overall average score of PDBSCANi and the overall average scores of the other algorithms (for short, all avg.gain).

From the overall average scores, it can be seen that the score of PDBSCANi is always higher than those of the other algorithms. PDBSCAN also performs better than the competitive algorithms in most cases, whereas PDBSCANi is better than PDBSCAN with 0.0527 all avg.gain. Density-based algorithms generally perform better than partition-based algorithms on average, the reason is that the computation of expected distances or uncertain distances may cause the loss of uncertain information. UK-medoids performs the worst on average. PDBSCANi is the best with 0.7604 all avg.score.

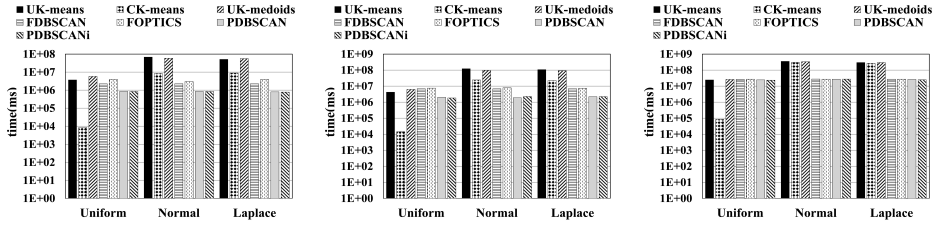
While the accuracy scores of the competitive algorithms vary largely on different datasets, the accuracy score of PDBSCANi is relatively stable with more than 0.5 in the worst case. Specifically, for yeast and abalone, all the competitive algorithms get very low accuracy scores. This is because that these two datasets both contain large numbers of clusters which are highly imbalanced in sizes and complex in distributions, thus they are hard for the competitive algorithms to detect. However, PDBSCANi still gets more than

Table 2: Accuracy results of the experiment

Dataset	pdf	UK-means	CK-means	UK-medoids	FDBSCAN	FOPTICS	PDBSCAN	PDBSCANi
Iris	Uniform	0.8760	0.8858	0.9533	0.9667	0.9733	**0.9800**	**0.9800**
	Normal	0.7467	0.8600	0.8750	0.7933	0.9600	0.9533	**0.9800**
	Laplace	0.7067	0.8733	0.8740	0.7800	0.9533	0.9200	**0.9800**
Wine	Uniform	0.7213	0.6757	0.7118	0.6742	0.6798	0.7584	**0.8483**
	Normal	0.4916	0.6348	0.4899	0.6573	0.6685	0.7303	**0.8146**
	Laplace	0.6742	0.6067	0.4893	0.6517	0.6742	0.7079	**0.8146**
Glass	Uniform	0.5706	0.5659	0.5098	0.5888	0.5093	**0.6168**	**0.6168**
	Normal	0.4593	0.4631	0.4776	0.6215	0.5234	0.6402	**0.6822**
	Laplace	0.4907	0.5248	0.4921	0.5748	0.5234	0.5935	**0.6355**
Ecoli	Uniform	0.7135	0.7119	0.6379	0.6850	0.7523	0.8135	**0.8165**
	Normal	0.5841	0.7682	0.6599	0.6820	0.6850	0.7156	**0.8991**
	Laplace	0.7645	0.6618	0.6453	0.6820	0.7462	0.7676	**0.7982**
Yeast	Uniform	0.3396	0.3394	0.3538	0.3854	0.4030	**0.6651**	**0.6651**
	Normal	0.3346	0.3382	0.3313	0.3774	0.4023	0.6058	**0.6806**
	Laplace	0.3384	0.3327	0.3323	0.3666	0.3982	0.6489	**0.6631**
Image	Uniform	0.5042	0.4920	0.4165	0.4766	0.5299	**0.8359**	**0.8359**
	Normal	0.5246	0.5330	0.4679	0.4593	0.5290	0.7996	**0.8398**
	Laplace	0.5143	0.5295	0.4862	0.4459	0.5294	0.6879	**0.8286**
Abalone	Uniform	0.2076	0.1662	0.2455	0.2000	0.2056	0.5335	**0.5336**
	Normal	0.2162	0.1659	0.2224	0.1967	0.2049	0.4867	**0.5456**
	Laplace	0.1988	0.1620	0.2067	0.1942	0.2022	0.4003	**0.5112**
Avg.	Uniform	0.5618	0.5481	0.5469	0.5681	0.5790	0.7433	**0.7566**
	Normal	0.4796	0.5376	0.5034	0.5411	0.5676	0.7045	**0.7774**
	Laplace	0.5268	0.5273	0.5037	0.5279	0.5753	0.6752	**0.7473**
all avg.score		0.5227	0.5377	0.5180	0.5457	0.5740	0.7077	**0.7604**
all avg.gain		**0.2377**	**0.2227**	**0.2424**	**0.2147**	**0.1864**	**0.0527**	—



(a) Iris (b) Wine (c) Glass (d) Ecoli



(e) Yeast (f) Image (g) Abalone

Figure 3: Efficiency results of the experiment

0.5 accuracy scores on these two datasets, since it avoids the shortcomings of both partition-based algorithms and previous density-based algorithms.

In summary, PDBSCAN performs better than the competitive algorithms, but not so good as PDBSCANi. PDBSCANi performs the best among all the tested algorithms.

Efficiency

Figure 3 shows the efficiency results (in milliseconds) on different datasets for different distributions. From the results, it can be seen that UK-means and UK-medoids are the two slowest algorithms. CK-means runs faster than our proposed algorithms PDBSCAN and PDBSCANi on uniform distribution, but slower than ours on the other two distributions. The runtime of FOPTICS is always a little higher than PDBSCAN and PDBSCANi. When the scale of the database is small, the speed of FDBSCAN is almost equal to PDBSCAN and PDBSCANi. However, it runs slower than PDBSCAN and PDBSCANi on large scale datasets. Overall PDBSCAN and PDBSCANi are competitive compared to other algorithms in terms of efficiency.

Conclusion

In this paper, we have proposed a novel density-based uncertain data clustering algorithm, which improves upon existing algorithms by new definitions and computational methods. We then improve the algorithm by a more appropriate cluster assignment strategy. The proposed algorithms address remaining issues in existing density-based algorithms. Experimental results show that the proposed algorithms outperform existing algorithms in terms of accuracy. From an efficiency view, the proposed algorithms are also competitive

compared to existing algorithms. For future work, we will extend the definitions and methods to hierarchical clustering and subspace clustering for uncertain data.

Acknowledgments

This work was supported by National Science Foundation of China (No. 61272374, 61300190)

References

Aggarwal, C. C., and Yu, P. S. 2009. A survey of uncertain data algorithms and applications. *IEEE Trans. Knowl. Data Eng.* 21(5):609–623.

Aggarwal, C. C. 2009. *Managing and Mining Uncertain Data*, volume 35 of *Advances in Database Systems*. Kluwer.

Ankerst, M.; Breunig, M. M.; Kriegel, H.-P.; and Sander, J. 1999. Optics: Ordering points to identify the clustering structure. In *SIGMOD Conference*, 49–60.

Chau, M.; Cheng, R.; Kao, B.; and Ng, J. 2006. Uncertain data mining: An example in clustering location data. In *PAKDD*, 199–204.

Cormode, G., and McGregor, A. 2008. Approximation algorithms for clustering uncertain data. In *PODS*, 191–200.

Deshpande, A.; Guestrin, C.; Madden, S.; Hellerstein, J. M.; and Hong, W. 2005. Model-based approximate querying in sensor networks. *VLDB J.* 14(4):417–443.

Ester, M.; Kriegel, H.-P.; Sander, J.; and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 226–231.

Gullo, F., and Tagarelli, A. 2012. Uncertain centroid based partitional clustering of uncertain data. *PVLDB* 5(7):610–621.

Gullo, F.; Ponti, G.; Tagarelli, A.; and Greco, S. 2008. A hierarchical algorithm for clustering uncertain data via an information-theoretic approach. In *ICDM*, 821–826.

Gullo, F.; Ponti, G.; and Tagarelli, A. 2008. Clustering uncertain data via k-medoids. In *SUM*, 229–242.

Gullo, F.; Ponti, G.; and Tagarelli, A. 2010. Minimizing the variance of cluster mixture models for clustering uncertain objects. In *ICDM*, 839–844.

Günnemann, S.; Kremer, H.; and Seidl, T. 2010. Subspace clustering for uncertain data. In *SDM*, 385–396.

Kao, B.; Lee, S. D.; Cheung, D. W.; Ho, W.-S.; and Chan, K. F. 2008. Clustering uncertain data using voronoi diagrams. In *ICDM*, 333–342.

Kao, B.; Lee, S. D.; Lee, F. K. F.; Cheung, D. W.-L.; and Ho, W.-S. 2010. Clustering uncertain data using voronoi diagrams and r-tree index. *IEEE Trans. Knowl. Data Eng.* 22(9):1219–1233.

Kriegel, H.-P., and Pfeifle, M. 2005a. Density-based clustering of uncertain data. In *KDD*, 672–677.

Kriegel, H.-P., and Pfeifle, M. 2005b. Hierarchical density-based clustering of uncertain data. In *ICDM*, 689–692.

Lee, S. D.; Kao, B.; and Cheng, R. 2007. Reducing uk-means to k-means. In *ICDM Workshops*, 483–488.

Liu, X.; Milo, M.; Lawrence, N. D.; and Rattray, M. 2005. A tractable probabilistic model for affymetrix probe-level analysis across multiple chips. *Bioinformatics* 21(18):3637–3644.

Lukic, I.; Köhler, M.; and Slavek, N. 2012. Improved bisector pruning for uncertain data mining. In *ITI*, 355–360.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to information retrieval*. Cambridge University Press.

Ngai, W. K.; Kao, B.; Chui, C. K.; Cheng, R.; Chau, M.; and Yip, K. Y. 2006. Efficient clustering of uncertain data. In *ICDM*, 436–445.

Ngai, W. K.; Kao, B.; Cheng, R.; Chau, M.; Lee, S. D.; Cheung, D. W.; and Yip, K. Y. 2011. Metric and trigonometric pruning for clustering of uncertain data in 2d geometric space. *Inf. Syst.* 36(2):476–497.

Sarma, A. D.; Benjelloun, O.; Halevy, A. Y.; Nabar, S. U.; and Widom, J. 2009. Representing uncertain data: models, properties, and algorithms. *VLDB J.* 18(5):989–1019.

Trajcevski, G.; Wolfson, O.; Hinrichs, K.; and Chamberlain, S. 2004. Managing uncertainty in moving objects databases. *ACM Trans. Database Syst.* 29(3):463–507.