2008 Eighth IEEE International Conference on Data Mining

Clustering Uncertain Data using Voronoi Diagrams

Ben Kao Sau Dan Lee David W. Cheung Wai-Shing Ho K. F. Chan
Department of Computer Science
The University of Hong Kong
{kao,sdlee,dcheung,wsho,kfchan}@cs.hku.hk

Abstract

We study the problem of clustering uncertain objects whose locations are described by probability density functions (pdf). We show that the UK-means algorithm, which generalises the k-means algorithm to handle uncertain objects, is very inefficient. The inefficiency comes from the fact that UK-means computes expected distances (ED) between objects and cluster representatives. For arbitrary pdf's, expected distances are computed by numerical integrations, which are costly operations. We propose pruning techniques that are based on Voronoi diagrams to reduce the number of expected distance calculation. These techniques are analytically proven to be more effective than the basic bounding-box-based technique previous known in the literature. We conduct experiments to evaluate the effectiveness of our pruning techniques and to show that our techniques significantly outperform previous methods.

1. Introduction

Clustering is a technique that has been widely studied and applied to many real-life applications. Many efficient algorithms, including the well known and widely applied k-means algorithm, have been devised to solve the clustering problem efficiently. Traditionally, clustering algorithms deal with a set of objects whose positions are accurately known. The goal is to find a way to divide objects into clusters so that the total distance of the objects to their assigned cluster centres is minimised.

Although simple, the problem model does not address situations where object locations are uncertain. Data uncertainty, however, arises naturally and often inherently in many applications. For example, physical measurements can never be 100% precise in theory (due to Heisenberg's *Uncertainty Principle*). Limitations of measuring devices thus induce uncertainty to the measured values in practice.

As another example, consider the application of clustering a set of mobile devices. By grouping mobile devices into clusters, a leader can be elected for each cluster, which can then coordinate the work within its cluster. For example, a cluster leader may collect data from its cluster's members, process the data, and send the data to a central server via an access point in batch. In this way, local communication within a cluster only requires short-ranged signals, for which a higher bandwidth is available. Long-ranged communication between the leaders and the mobile network only takes place in the form of batch communication. This results in better bandwidth utilisation and energy conservation.

We remark that device locations are uncertain in practice. A mobile device may deduce and report its location by comparing the strengths of radio signals from mobile access points. Unfortunately, such deductions are susceptible to noise. Furthermore, locations are reported periodically. Between two sampling time instances, a location value is unknown and can only be estimated by considering a last reported value and an uncertainty model. Typically, such an uncertainty model considers factors such as the speed of the moving devices and other geometrical constraints (such as road network, etc.). In this paper we consider the problem of clustering uncertain objects whose locations are specified by uncertainty regions over which arbitrary probability density functions (pdf's) are defined.

Traditional clustering methods were designed to handle point-valued data and thus cannot cope with data uncertainty. One possible way to handle data uncertainty is to first transform uncertain data into point-valued data by selecting a representative point for each object before applying a traditional clustering algorithm. For example, the centroid of an object's pdf can be used as such a representative point. However, in [4], it is shown that considering object pdf's gives better clustering results than the centroid method.

In this paper we concentrate on the problem of clustering objects with location uncertainty. Rather than a single point in space, an object is represented by a probability density function (pdf) over the space R^m being studied. We assume that each object is confined in a finite region, so that the probability density outside the region is zero. Each ob-

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.31

ject can thus be bounded by a finite bounding box. This assumption is realistic because in practice the probability density of an object is high only within a very small region of concentration. The probability density is negligible outside the region. (For example, the uncertainty region of a mobile device can be limited by the maximum speed of the device.) Given a set of such objects, our goal is to divide them into k clusters, minimising the total *expected distance* (ED) [4] from the objects to their cluster centres.

The problem of clustering uncertain objects was first described in [4], in which the UK-means algorithm was proposed. UK-means is a generalisation of the traditional k-means algorithm to handle objects with uncertain locations. The major computational cost of UK-means is the evaluation of EDs, which involves numerical integration using a large number of sample points for each pdf. To improve efficiency, [19] introduced some pruning techniques to avoid many ED computations. The pruning techniques make use of bounding boxes over objects as well as the triangle inequality to establish lower- and upper-bounds of the EDs. Using these bounds, some candidate clusters are eliminated from consideration when UK-means determines the cluster assignment of an object. The corresponding computation of expected distances from the object to the pruned clusters are thus not necessary and are avoided.

The contribution of this paper is the introduction of a new set of pruning techniques for the UK-means algorithm that are based on Voronoi diagrams [10]. These new pruning techniques take into consideration the spatial relationship among the cluster representatives. We prove that Voronoi-diagram-based technique is strictly more effective than the basic bounding-box-based technique. Another technique we investigate is the partial ED evaluation method, which can be shown to further save the computation costs of UK-means. Since our pruning techniques are orthogonal to the ones proposed in [19], we study a hybrid algorithm that integrates the various pruning techniques. Our empirical study shows that the hybrid algorithm achieves significant performance improvement.

The rest of the paper is organised as follows. We mention a few related works in Section 2. In Section 3, we formally define the problem. In Section 4, we first briefly describe the UK-means algorithm and the bounding-box-based pruning techniques. After that, we discuss our Voronoi-diagram-based pruning techniques. We present experiment results in Section 5, comparing our new pruning techniques with existing ones. Finally, Section 6 concludes the paper.

2. Related Works

Data uncertainty has been broadly classified into existential uncertainty and value uncertainty. Existential uncertainty appears when it is uncertain whether an object or a data tuple exists. For example, a data tuple in a relational database could be associated with a probability that represents the confidence of its presence [9, 2]. Value uncertainty, on the other hand, appears when a tuple is known to exist, but its values are not known precisely. A data item with value uncertainty is usually represented by a pdf over a finite and bounded region of possible values [6, 7, 5, 4]. In this paper, we study the problem of clustering objects with value (e.g., location) uncertainty.

One well-studied topic on value uncertainty is "imprecise query processing." An answer to such a query is associated with a probabilistic guarantee on its correctness. Some example studies on imprecise data include indexing structures for range query processing [6], nearest neighbour query processing [7], and imprecise location-dependent query processing [5].

Depending on the application, the result of cluster analysis can be used to identify the (locally) most probable values of model parameters [11] (e.g., means of Gaussian mixtures), to identify high-density connected regions [15] (e.g., areas with high population density), or to minimise an objective function (e.g., the total within-cluster squared distance to centroids [18]). For model parameters learning, by viewing uncertain data as samples from distributions with hidden parameters, the standard Expectation-Maximisation (EM) framework [11] can be used to handle data uncertainty [13].

There has been growing interest in uncertain data mining. In [4], the well-known k-means clustering algorithm is extended to the UK-means algorithm for clustering uncertain data. In that study, it is empirically shown that clustering results are improved if data uncertainty is taken into account during the clustering process. As we have explained, data uncertainty is usually captured by pdf's, which are generally represented by sets of sample values. Mining uncertain data is therefore computationally costly due to information explosion (sets of samples vs. singular values). To improve the performance of UK-means, CK-means [17] introduced a novel method for computing the EDs efficiently. However, that method only works for a specific form of distance function. For general distance functions, [19] takes the approach of pruning, and proposed pruning techniques such as min-max-dist pruning. In this paper we take this latter approach and propose new pruning techniques that are significantly more powerful than those proposed in [19].

Apart from studies in partition-based uncertain data clustering, other directions in uncertain data mining include density-based clustering (e.g., FDBSCAN [15]), frequent itemset mining [8] and density-based classification [1].

For density-based clustering, two well-known algorithms, namely, DBSCAN and OPTICS have been extended to handle uncertain data. The corresponding algorithms are called FDBSCAN [15] and FOPTICS [16], respectively. In

DBSCAN, the concepts of core objects and reachability are defined. Clusters are then formed based on these concepts. In FDBSCAN, the concepts are re-defined to handle uncertain data. For example, under FDBSCAN, an object o is a core object if the probability that there is a "good number" of other objects that are close to o exceeds a certain probability threshold. Also, whether an object y is "reachable" from another object x depends on both the probability of y being close to x and the probability that x is a core object. FOPTICS takes a similar approach of using probabilities to modify the OPTICS algorithm to cluster uncertain data.

Clustering of uncertain data is also related to fuzzy clustering, which has long been studied in fuzzy logic [20]. In fuzzy clustering, a cluster is represented by a fuzzy subset of objects. Each object has a "degree of belongingness" with respect to each cluster. The fuzzy c-means algorithm is one of the most widely used fuzzy clustering methods [12, 3]. Different fuzzy clustering methods have been applied on normal or fuzzy data to produce fuzzy clusters [21, 23]. A major difference between the clustering problem studied in this paper and fuzzy clustering is that we focus on hard clustering, for which each object belongs to exactly one cluster. Our formulation targets for applications such as mobile device clustering, in which each device should report its location to exactly one cluster leader.

Voronoi diagram is a well-known geometric structure in computational geometry. It has also been applied to clustering. For example, Voronoi trees [10] have been proposed to answer Reverse Nearest Neighbour (RNN) queries [22]. Given a set of data points and a query point q, the RNN problem [14] is to find all the data points whose nearest neighbour is q. The TPL algorithms proposed in [24] uses more advanced pruning techniques to solve this problem efficiently.

3. Definitions

Consider a set of objects $O = \{o_1, \ldots, o_n\}$ in an m-dimensional space R^m with a distance function $d : R^m \times R^m \rightarrow R$ giving the distance $d(x, y) \geq 0$ between any points $x, y \in R^m$. Associated with each object is a pdf $f_i : R^m \rightarrow R$, which gives the probability density of o_i at each point $x \in R^m$. By the definition of pdf, we have (for all $i = 1, \ldots, n$)

$$\begin{aligned} f_i(x) &\geq 0 \quad \forall x \in R^m \\ \int_{x \in R^m} f_i(x)\, dx &= 1 \end{aligned}$$

Further, we assume that the probability density of o_i is confined in a finite region A_i, so that $f_i(x) = 0$ for all $x \in R^m \setminus A_i$.

We define the expected distance between an object o_i and any point $y \in R^m$:

$$\mathrm{ED}(o_i, y) = \int_{x \in A_i} d(x, y) f_i(x)\, dx. \qquad (1)$$

Now, given an integer constant k, the problem of clustering uncertain data is to find a set of cluster representative points $C = \{c_1, \ldots, c_k\}$ and a mapping $h : \{1, \ldots, n\} \rightarrow \{1, \ldots, k\}$ so that the total expected distance

$$\mathrm{TED} = \sum_{i=1}^{n} \mathrm{ED}(o_i, c_{h(i)})$$

is minimised.

To facilitate our discussion on bounding-box-based algorithms, we use MBR_i to denote the *minimum bounding box* of object o_i. MBR_i is the smallest box, with faces perpendicular to the principal axes of R^m, that encloses A_i. Note that Equation (1) still holds if we replace "$x \in A_i$" with "$x \in \mathrm{MBR}_i$". This fact can be exploited for optimisation when computing ED.

4. Algorithms

We first give a short description of the UK-means algorithm [4] and existing pruning techniques [19] that improve UK-means. Then, we present our new pruning techniques that are based on Voronoi diagrams.

4.1. UK-means

UK-means is an adaptation of the well known k-means algorithm to handle data objects with uncertain locations.

1: Choose k arbitrary points as c_j $(j = 1, \ldots, k)$
2: **repeat**
3: **for all** $o_i \in O$ **do** */*assign objects to clusters*/*
4: **for all** $c_j \in C$ **do**
5: Compute $\mathrm{ED}(o_i, c_j)$.
6: $h(i) \leftarrow j^*$ where j^* minimises $\mathrm{ED}(o_i, c_j)$ (among $c_j \in C$)
7: **for all** $j = 1, \ldots, k$ **do** */*readjust cluster representatives*/*
8: $c_j \leftarrow$ centroid of $\{o_i \in O \mid h(i) = j\}$
9: **until** C and h become stable

Initially, k arbitrary points $c_1, \ldots, c_k$ are chosen as the cluster representatives. Then, UK-means repeats the following steps until the result converges. First, for each object o_i, $\mathrm{ED}(o_i, c_j)$ is computed for all $c_j \in C$. Object o_i is then assigned to cluster c_{j^*} that minimises ED, i.e., $h(i) \leftarrow j^*$. Next, each cluster representative c_j is recomputed as the centroid of all o_i's that are assigned to cluster j. The two steps are repeated until the solution $C = \{c_1, \ldots, c_k\}$ and $h(\cdot)$ converge.

The UK-means algorithm is inefficient. This is because UK-means computes ED for each object-cluster pair in each iteration. So, given n objects and k clusters, UK-means computes nk EDs in each iteration. The computation of an ED involves numerically integrating a function that involves an object's pdf. In practice, a pdf is represented by a probability distribution matrix, with each element of the matrix representing a sample point in an MBR. To accurately represent a pdf, a large number of sample points are needed. The computation cost of an integration is thus high.

To improve the performance of UK-means, we need to reduce the time spent on ED calculations. To incorporate pruning into UK-means, we replace lines 4–6 in UK-means with the following:

1: $Q_i \leftarrow C$/**candidate clusters**/
2: Apply a pruning technique
3: **if** only one candidate remains in Q_i **then**
4: $h(i) \leftarrow j$ where $c_j \in Q_i$.
5: **else**
6: **for all** $c_j \in Q_i$ **do** /**remaining candidates**/
7: Compute $\text{ED}(o_i, c_j)$.
8: $h(i) \leftarrow j^*$ where j^* minimises $\text{ED}(o_i, c_j)$ (among $c_j \in Q_i$).

For a given object o_i, the set Q_i stores the set of candidate cluster representatives that are potentially the closest to o_i. Initially, $Q_i = C$, the set of all cluster representatives. In line 2, a pruning algorithm is applied to prune candidate representatives from Q_i that are guaranteed to be not the closest to object o_i. If all but one candidate cluster remains in Q_i, object o_i is assigned the that cluster. Otherwise, we compute the expected distances between o_i and each cluster in Q_i. Object o_i is then assigned to the cluster that gives the smallest expected distance. We describe a few pruning methods in the following sections.

4.2. MinMax Pruning

Several pruning techniques that are based on bounds on ED have been proposed in [19]. In the MinMax approach, for an object o_i and a cluster representative c_j, certain points in MBR_i are geometrically determined. The distances from those points to c_j are computed to establish bounds on ED. Formally, we define

$$\begin{aligned} \text{MinD}(o_i, c_j) &= \min_{x \in \text{MBR}_i} d(x, c_j) \\ \text{MaxD}(o_i, c_j) &= \max_{x \in \text{MBR}_i} d(x, c_j) \\ \text{MinMaxD}(o_i) &= \min_{c_j \in C} \{\text{MaxD}(o_i, c_j)\} \end{aligned}$$

It should be obvious that $\text{MinD}(o_i, c_j) \leq \text{ED}(o_i, c_j) \leq \text{MaxD}(o_i, c_j)$. Then, if $\text{MinD}(o_i, c_p) > \text{MaxD}(o_i, c_q)$ for some cluster representatives c_p and c_q, we can deduce that $\text{ED}(o_i, c_p) > \text{ED}(o_i, c_q)$ without computing the exact values of the EDs. So, object o_i will not be assigned to cluster p (since there is another cluster q that gives a smaller expected distance from object o_i). We can thus prune away cluster p without having to compute $\text{ED}(o_i, c_p)$. As an optimisation, we can prune away cluster p if $\text{MinD}(o_i, c_p) > \text{MinMaxD}(o_i)$. This gives rise to the following BB (bounding box) pruning algorithm.

1: **for all** $c_j \in C$ **do** /**for a fixed object o_i**/
2: Compute $\text{MinD}(o_i, c_j)$ and $\text{MaxD}(o_i, c_j)$.
3: Compute $\text{MinMaxD}(o_i)$.
4: **for all** $c_j \in C$ **do**
5: **if** $\text{MinD}(o_i, c_j) > \text{MinMaxD}(o_i)$ **then**
6: Remove c_j from Q_i

We call this pruning algorithm MinMax-BB. Depending on data distribution, the pruning condition $\text{MinD}(o_i, c_j) > \text{MinMaxD}(o_i)$ potentially removes many clusters from consideration in line 6. This avoids many ED computations at the expense of computing MinD and MaxD. We remark that computing MinD and MaxD requires us to consider only a few points on the perimeter of an MBR, instead of all points in an object's pdf. Thus, computing MinD and MaxD is much simpler than computing ED and it does not involve evaluating an integral. They can be computed much faster than ED.

Another pruning technique proposed in [19] makes use of the inequalities:

$$\text{ED}(o_i, c_j) \leq \text{ED}(o_i, y) + d(y, c_j) \quad (2)$$

$$\text{ED}(o_i, c_j) \geq |\text{ED}(o_i, y) - d(y, c_j)| \quad (3)$$

for any point $y \in R^m$. (Equation (2) is indeed the triangle inequality.) These inequalities give bounds on $\text{ED}(o_i, c_j)$ based on $\text{ED}(o_i, y)$. If we can compute the latter efficiently, then we can find the bounds efficiently. One possibility is to choose (for each object) certain fixed points as y, and precompute $\text{ED}(o_i, y)$. Then, evaluating the bounds using the inequalities involves only an addition, a subtraction, and an evaluation of distance $d(y, c_j)$, which are relatively cheap. Note that y is fixed for each object while c_j, a cluster representative, changes across different iterations in UK-means. So, for an object o_i, by computing one expected distance $\text{ED}(o_i, y)$, we are able to obtain bounds for many EDs that involve o_i and any cluster representative c_j.

Another pruning method proposed in [19] is called "cluster-shift" (CS). Consider a cluster j whose representatives in two consecutive iterations are c'_j and c_j in that order. If $\text{ED}(o_i, c'_j)$ has been calculated, then we can use c'_j as y in Equations (2) and (3) to bound $\text{ED}(o_i, c_j)$. An appealing aspect of CS is that in the later iterations, as the solution converges, $d(c'_j, c_j)$ decreases rapidly, making the bounds very tight. It is shown in [19] that the cluster-shift method is very effective in pruning. Also, it does not re-

quire any pre-determined fixed points y and hence no pre-computation of $\text{ED}(o_i, y)$ is needed. In our following discussion, we consider the cluster-shift method instead of the fixed-point method.

Now, the MinMax-BB algorithm can be augmented with the cluster-shift technique to improve pruning power. The bounds computed using Equations (2) and (3) allow us to refine the bounds on ED to tighter values rather than using MinD and MaxD alone. With tighter bounds, we are able to prune more candidates at little additional cost. We call this algorithm MinMax-SHIFT.

4.3. Pruning with Voronoi Diagram

MinMax-based pruning techniques improve the performance of UK-means significantly by making use of efficiently evaluable bounds on ED to avoid many ED computations. However, these techniques do not consider the geometric structure of R^m or the spatial relationships among the cluster representatives. The major innovation in this paper is the introduction of Voronoi diagrams [10] as a method to exploit the spatial relationships among the cluster representatives to achieve a very effective pruning. We will show in this section that our Voronoi-diagram-based pruning technique is theoretically strictly stronger than MinMax-BB. Also, we will discuss how our Voronoi-diagram-based method can be combined with the cluster-shift method to achieve the most efficient pruning algorithm.

We start with a definition of Voronoi diagram and a brief discussion of its properties. Given a set of points $C = \{c_1, \ldots, c_k\}$, the Voronoi diagram divides the space R^m into k cells $V(c_j)$ with the following property:

$$d(x, c_p) \quad < \quad d(x, c_q) \quad \forall x \in V(c_p), c_q \neq c_p \qquad (4)$$

The boundary of a cell $V(c_p)$ and its adjacent cell $V(c_q)$ consists of points on the *perpendicular bisector*, denoted $c_p|c_q$ between the points c_p and c_q. The bisector is the hyperplane that is perpendicular to the line segment joining c_p and c_q that passes through the mid-point of the line segment. This hyperplane divides the space R^m into two halves. We denote the half containing c_p (but excluding the hyperplane itself) as $H_{p/q}$. Thus, $H_{p/q}$, $H_{q/p}$ and $c_p|c_q$ form a partition of the space R^m. Further, we have the following properties: $\forall$ distinct $c_p, c_q \in C$,

$$\begin{aligned} d(x, c_p) \quad &< \quad d(x, c_q) \quad \forall x \in H_{p/q}, \qquad (5) \\ d(x, c_p) \quad &= \quad d(x, c_q) \quad \forall x \in c_p|c_q. \end{aligned}$$

Here is how we use Voronoi diagram for pruning in UK-means: In each iteration, we first construct the Voronoi diagram from the k cluster representative points, $C = \{c_1, \ldots, c_k\}$. The Voronoi diagram leads to two pruning methods: The first one is Voronoi-cell pruning. For each

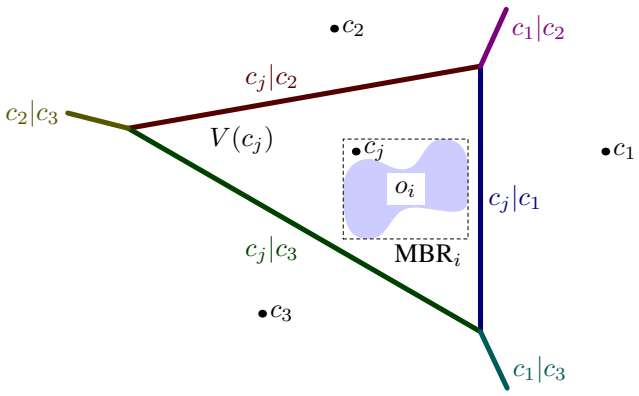

Figure 1. Voronoi-cell pruning

object o_i, we check if MBR$_i$ lies completely inside any Voronoi cell $V(c_j)$. If so, then object o_i is assigned to cluster c_j. This is because it follows from Equations (1) and (4) that:

$$\text{ED}(o_i, c_j) < \text{ED}(o_i, c_q) \; \forall c_q \in C \setminus \{c_j\}.$$

Note that in this case, no ED is computed. All clusters except c_j are pruned. An example is illustrated in Figure 1, in which $V(c_j)$ is adjacent to $V(c_1)$, $V(c_2)$ and $V(c_3)$. Since MBR$_i$ lies completely in $V(c_j)$, all points belonging to o_i lie closer to c_j than any other c_q. It follows that $\text{ED}(o_i, c_j)$ is strictly smaller than $\text{ED}(o_i, c_q)$ for all $c_q \neq c_j$. The Voronoi-cell pruning method can be summarised by the following pseudo code:

```
1: Compute the Voronoi diagram for C = {c_1, ..., c_k}.
2: for all c_j ∈ C do
3:     if MBR_i ⊆ V(c_j) then
4:         Q_k ← {c_j}.
```

The other pruning method is bisector pruning. Bisectors are the side-products of Voronoi diagram construction, and thus they are available at little extra cost. Given an object o_i, we consider every pair of distinct cluster representatives c_p, c_q from C. We then check if MBR$_i$ lies completely in $H_{p/q}$. If it does, then by Equation (5), we can deduce that $\text{ED}(o_i, c_p) < \text{ED}(o_i, c_q)$, and c_q is pruned from Q_i. The expected distance $\text{ED}(o_i, c_q)$ is not computed. The bisector-pruning method is summarised below:

```
1: for all distinct c_p, c_q ∈ C do
2:     if MBR_i ⊆ H_{p/q} then
3:         remove c_q from Q_i
```

In the following theorem, we show that bisector pruning is strictly stronger than MinMax-BB in terms of pruning effectiveness.

Theorem 1 *For any object $o_i \in O$ and cluster j ($j = 1, \ldots, k$), if bisector pruning does not prune away candidate cluster j, then neither does MinMax-BB.*

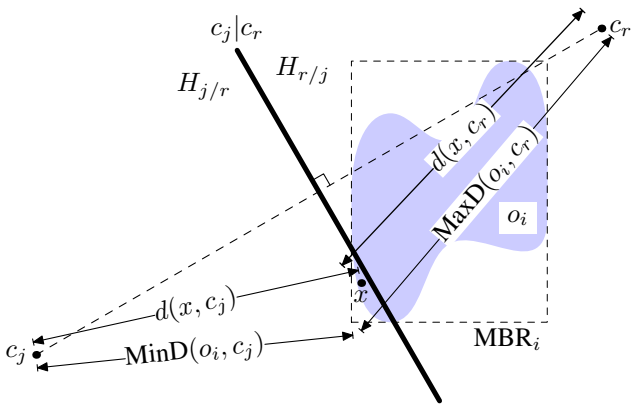

Figure 2. Illustration of the proof of Theorem 1

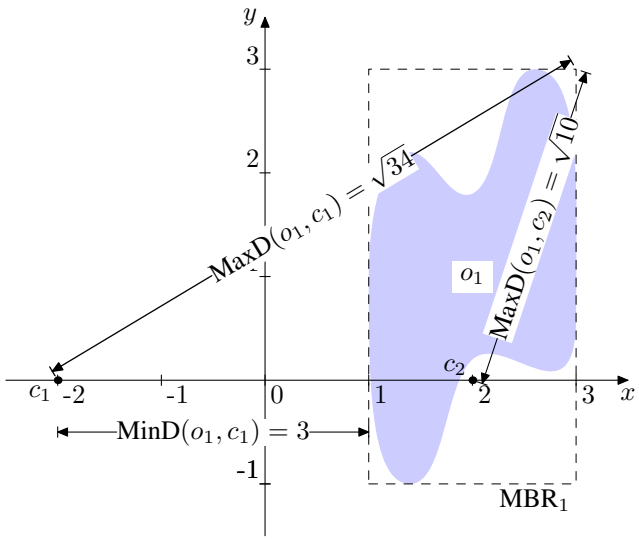

Figure 3. A counter-example

Proof: Let c_r be the cluster representative that gives the smallest MaxD with object o_i, i.e., $\text{MaxD}(o_i, c_r) = \text{MinMaxD}(o_i)$. We consider two cases:

Case 1: $r = j$. Then,

$$\begin{array}{lcll} \text{MinD}(o_i, c_j) & \leq & \text{MaxD}(o_i, c_j) & \text{by definition} \\ & = & \text{MaxD}(o_i, c_r) & \text{since } r = j \\ & = & \text{MinMaxD}(o_i) & \text{by definition of } c_r \end{array}$$

Since MinMax-BB prunes cluster j only when $\text{MinD}(o_i, c_j) > \text{MinMaxD}(o_i)$, we conclude that MinMax-BB does not prune away cluster j in this case. The theorem thus holds in this case.

Case 2: $r \neq j$. The bisector $c_j|c_r$ is well defined and the space R^m can be partitioned into $\{H_{r/j}, c_j|c_r, H_{j/r}\}$. We consider 2 subcases:

Case 2a: MBR_i lies completely in $H_{r/j}$. In this case, cluster j will be pruned by bisector pruning. So, the theorem holds for this case because the antecedent is not satisfied.

Case 2b: MBR_i overlaps with $H_{j/r} \cup (c_j|c_r)$. Now, consider a point x in $\text{MBR}_i \cap (H_{j/r} \cup (c_j|c_r))$, as illustrated in Figure 2. We have:

$$\begin{array}{ll} \text{MinD}(o_i, c_j) \leq d(x, c_j) & \text{since } x \in \text{MBR}_i \\ \leq d(x, c_r) & \text{since } x \in H_{j/r} \cup (c_j|c_r) \\ \leq \text{MaxD}(o_i, c_r) & \text{by definition of MaxD} \\ = \text{MinMaxD}(o_i) & \text{by definition of } c_r \end{array}$$

Again, the pruning criterion of MinMax-BB is not satisfied and MinMax-BB cannot prune away cluster j. The theorem thus holds.

Hence, we conclude that if bisector pruning does not prune away cluster j, neither does MinMax-BB. Q. E. D.

The converse of the theorem, however, does not hold. That is, there are cases in which MinMax-BB fails to prune a cluster while bisector pruning can. Figure 3 shows such an example in R^2 with 2 clusters. Suppose $c_1 = (-2, 0)$ and $c_2 = (2, 0)$. Then, $c_1|c_2$ is the line $x = 0$, i.e., the y-axis. Now, consider an object o_1 with MBR_1 bounded by the lines $x = 1$, $x = 3$, $y = -1$, $y = 3$. Since MBR_1 lies completely in $H_{2/1}$, bisector pruning can prune away cluster 1. How about MinMax-BB? Note that $\text{MinD}(o_1, c_1) = 3$; $\text{MaxD}(o_1, c_1) = \sqrt{34}$ and $\text{MaxD}(o_1, c_2) = \sqrt{10}$. So, we have $\text{MinMaxD}(o_1) = \sqrt{10} > \text{MinD}(o_1, c_1)$ and hence the pruning condition of MinMax-BB is not satisfied. So, MinMax-BB cannot prune away cluster 1.

We have thus shown that bisector pruning is strictly stronger than MinMax-BB in terms of pruning effectiveness. Note that in implementation, bisectors are a side-product of Voronoi diagram computation. It is therefore advantageous to perform both Voronoi-cell pruning and bisector pruning together. As the Voronoi diagram and bisectors depend only on the cluster representatives c_j ($j = 1, \ldots, k$), we can move the computation of the Voronoi diagram to the outermost loop in the UK-means algorithm as a further optimisation. We call the resulting algorithm VDBi (for Voronoi Diagram with Bisector pruning).

4.4. Partial ED Computation

Given two cluster representatives c_p and c_q and an object o_i, bisector pruning prunes cluster q if $\text{MBR}_i \subseteq H_{p/q}$. If no bisector that involves cluster q can be found to prune cluster q, the expected distance $\text{ED}(o_i, q)$ may have to be computed. Interestingly, it is not necessary that we compute the complete integral of $\text{ED}(o_i, q)$. Our next pruning technique attempts to prune a cluster by computing ED partially.

Again, consider two clusters p and q and an object o_i. If

MBR_i intersects the bisector $c_p|c_q$, we partition MBR_i into two parts X and Y ($X \cup Y = \text{MBR}_i$ and $X \cap Y = \emptyset$) such that $X \subseteq V(c_p)$. The expected distance $\text{ED}(o_i, c_p)$ can then be computed by two "smaller" integrals:

$$\begin{aligned} & \text{ED}(o_i, c_p) \\ = & \int_{x \in \text{MBR}_i} d(x, c_p) f_i(x)\, dx \\ = & \int_{x \in X} d(x, c_p) f_i(x)\, dx + \int_{x \in Y} d(x, c_p) f_i(x)\, dx \\ = & \text{ED}_X(o_i, c_p) + \text{ED}_Y(o_i, c_p). \end{aligned}$$

Similarly, we have $\text{ED}(o_i, c_q) = \text{ED}_X(o_i, c_q) + \text{ED}_Y(o_i, c_q)$.

Now, since $X \subseteq V(c_p)$, by Equation (4), we know that $\text{ED}_X(o_i, c_p) < \text{ED}_X(o_i, c_q)$. We compute the integrals $\text{ED}_Y(o_i, c_p)$ and $\text{ED}_Y(o_i, c_q)$ and if $\text{ED}_Y(o_i, c_p) < \text{ED}_Y(o_i, c_q)$, we can conclude that $\text{ED}(o_i, c_p) < \text{ED}(o_i, c_q)$. Cluster q can thus be pruned. Otherwise, if q cannot be pruned, we have to compute $\text{ED}(o_i, c_q)$ later, but we need not do so from scratch. We only need to compute $\text{ED}_X(o_i, c_q)$ and then add it to the already computed value of $\text{ED}_Y(o_i, c_q)$ to get $\text{ED}(o_i, c_q)$. Therefore, the effort spent on computing $\text{ED}_Y(o_i, c_q)$ can be reused for computing complete ED later if necessary. Thus, the partial computation of $\text{ED}(o_i, c_q)$ involves little overhead. We incorporate the above idea of partial ED computation into VDBi to improve the pruning power of the algorithm. We call the resulting algorithm VDBiP.

4.5. Hybrid Pruning

Our Voronoi-diagram-based pruning methods are based on a different principle than the MinMax-based methods. They are thus orthogonal and can be combined to achieve a better performance. For example, we can combine VDBi with the cluster shift technique, i.e., we attempt to prune candidate cluster representatives using bisector pruning and if that fails, we apply cluster-shift pruning. Similarly, we can combine VDBiP with the cluster shift method. We call these hybrid pruning algorithms VDBi-SHIFT and VDBiP-SHIFT. Note that there is no need to combine MinMax-BB with VDBi, as we have shown in Theorem 1 that VDBi prunes a superset of what MinMax-BB prunes.

5. Experiments

We have performed a series of experiments to compare the performance of our Voronoi-diagram-based pruning methods with MinMax-based pruning methods [19]. We compare the algorithms VDBi, VDBiP, MinMax-BB, VDBi-SHIFT, VDBiP-SHIFT and MinMax-SHIFT. All the algorithms are implemented in Visual C++. Experiments are carried out on a PC with a Pentium-4 2GHz CPU and 768MB of main memory.

Table 1. Parameters for the Experiments

Parameter	Description	Baseline Value
n	no. of uncertain objects	20000
k	number of clusters	49
d	max. side length of MBR	10
s	no. of samples per object	196

5.1. Data Sets

Following [19], we generated many sets of data for experiments. For each dataset, a set of n MBRs are generated in the 2D space $[0, 100] \times [0, 100]$. Each MBR's side length is generated randomly, but bounded above by d. The MBR is then divided into a $\sqrt{s} \times \sqrt{s}$ grid. There are thus a total of s grid cells, each corresponding to a sample point. Each sample point is associated with a randomly generated probability value, normalised so that the sum of probabilities of the MBR is equal to 1. These probability values give a discretised representation of the pdf f_i of the corresponding object.

In each set of experiments, we generate such a data set as well as k random points to serve as the initial cluster centres. The data set and initial cluster centres are then fed to the six algorithms. The clustering results from all algorithms are compared to ensure that they are the same. For each set of parameters, 10 sets of experiments are run and the average values are taken and reported.

The parameters used for the experiments are summarised in Table 1. The rightmost column of the table shows the baseline values of the various parameters.

5.2. Results of Baseline Experiment

We carried out the first set of experiments using the parameters shown in Table 1. The results are shown in Table 2. The execution times taken by the algorithms are given in the second column. For algorithms that employ Voronoi diagrams, the column labelled t_V shows the amount of time spent on the computation of Voronoi diagrams. The column N_{ED} shows the number of ED calculations per object per iteration.

Note that if we had performed the same experiment with UK-means, the number N_{ED} for UK-means would be k [19]. This is because in each iteration, UK-means computes for each object all k expected distances from the object to the k cluster representatives. In our baseline setting, $k = 49$ and therefore N_{ED} for UK-means would be 49. From Table 2, we see that the pruning algorithms are very effec-

Table 2. Results for Baseline Experiment

Algorithm	time (sec.)	t_V	N_{ED}
MinMax-BB	1648.4	n. a.	1.586
VDBi	818.6	10.49	1.282
VDBiP	806.4	10.80	1.001
MinMax-SHIFT	1286.7	n. a.	0.680
VDBi-SHIFT	638.5	10.50	0.568
VDBiP-SHIFT	628.5	10.46	0.439

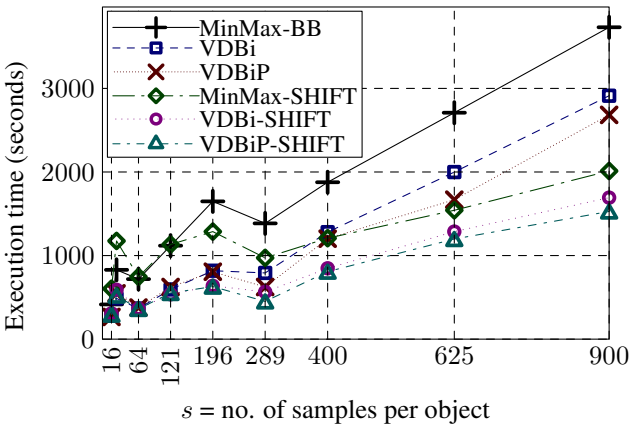

Figure 4. Effects of s on exec. time

tive. (The smaller the value of N_{ED}, the more effective the pruning is.) All of the pruning algorithms reduce N_{ED} from $k = 49$ (UK-means) to below 1.6. That is a reduction of more than 96.7%. We can see that VDBi is more effective than MinMax-BB, confirming Theorem 1. The optimisation further introduced by partial ED computation (Section 4.4) is also significant. Finally, the pruning effects are even better when we consider the hybrid algorithms that combine any of MinMax-BB, VDBi, VDBiP with the cluster-shift technique (Section 4.5).

It should be noted that the computation of Voronoi diagrams took less than 1.7% of the execution time of the relevant algorithms. This effort is certainly paid off by the amount of ED calculation saved. As bisector pruning is strictly stronger than MinMax-BB pruning as proved in Theorem 1, we can conclude that Voronoi-diagram-based pruning is a more practical and effective pruning technique than MinMax-BB.

5.3. Effects of the Number of Samples

Next, we varied s, the number of samples used to represent an object's pdf. In this experiment, s is varied from 16 to 900. The execution times taken by the algorithms are plotted in Figure 4.

From the figure, we see that the execution times of the algorithms generally increase as s increases. This is because the time to compute an ED grows linearly with s. We observe from the figure that when s is large (e.g., $s \geq 196$), the relative performance of the six algorithms is mostly consistent with that observed in our baseline experiment (Table 2). When s is smaller (e.g., $s \leq 121$), however, we see that MinMax-SHIFT is not faster than MinMax-BB. This is because for small values of s, the cost of computing an expected distance is small. The extra overhead that MinMax-SHIFT spends on performing pruning test cannot be paid off by the amount of time saved by reducing the number of ED computation. Similarly, we observe that the advantage of VDBi-SHIFT and VDBiP-SHIFT over VDBi and VDBiP is not very significant for small values of s. The cluster-shift technique is thus more useful when pdf's are represented by a large number of sample points. We would like to emphasise that pruning techniques that are based on Voronoi diagrams still perform better than MinMax-based pruning techniques, even for small values of s.

Note that the value of s only affects the amount of time taken to compute an ED, not the *number* of ED computations. In the following experiments, we will concentrate on measuring the pruning effectiveness of the algorithms. Therefore, in the following experiments, we will omit execution times and report N_{ED} only. Since N_{ED} is unaffected by the value of s, we have reduced the value of s down to 16 in the experiments so that the experiments could be completed faster. Again, we have already established that for large values of s, the relative efficiency of the algorithms is similar to that reported in Table 2. We have also confirmed that the Voronoi-diagram-based algorithms outperform MinMax-BB for all ranges of s studied.

5.4. Effects of the Number of Objects

In our next set of experiments, we vary the number of uncertain objects, n, from 4000 to 80000. Other parameters are given their baseline values (Table 1). The resulting values of N_{ED} are plotted against n in Figure 5. Apparently, the effectiveness of the pruning algorithms is insensitive to the number of uncertain objects.

5.5. Effects of the Number of Clusters

In another experiment, we vary the number of clusters, k, from 4 and 144. The other parameters are kept at their baseline values. Figure 6 shows the results. We see from the graph that N_{ED} increases with k. This is because with a larger number of clusters, cluster representatives are generally less spread out. It is therefore less likely that the pruning algorithms will be able to prune all but one cluster for a given object. Hence, more ED will have to be computed to determine the cluster assignment. For example, under Voronoi-cell pruning, a larger number of clusters implies

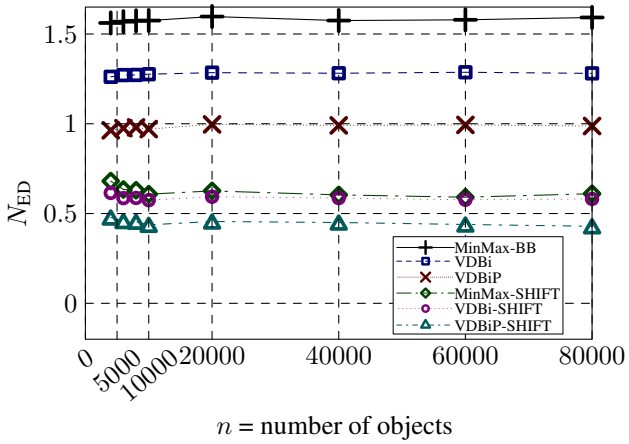

Figure 5. Effects of n on N_{ED}

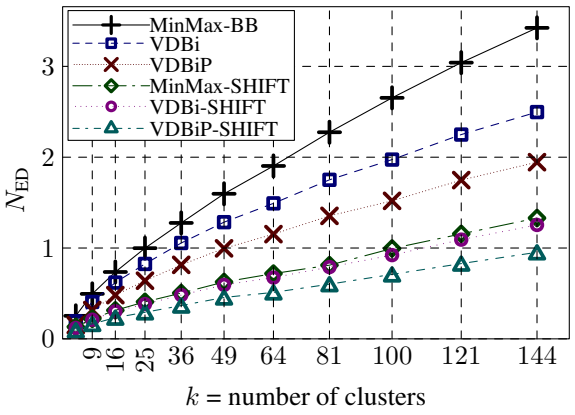

Figure 6. Effects of k on N_{ED}

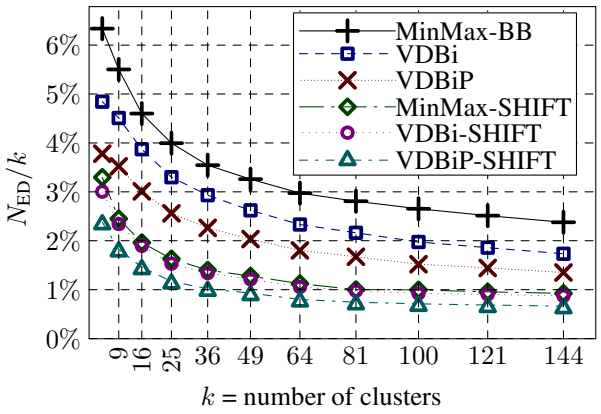

Figure 7. N_{ED}/k vs. k

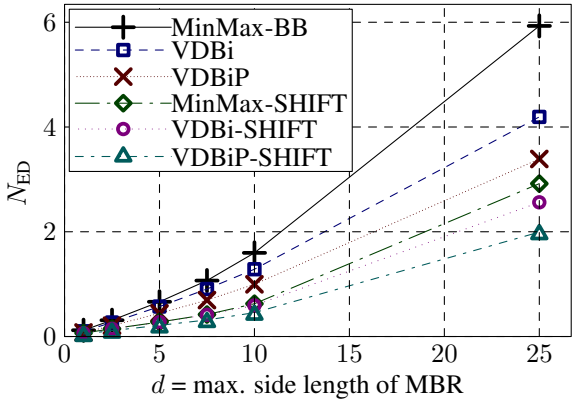

Figure 8. Effects of d on N_{ED}

smaller Voronoi cells. It is thus less likely that an object is found to be enclosed entirely within a particular Voronoi cell so that all but one cluster representative are pruned.

From Figure 6, we see that the N_{ED} curves are always significantly lower than k. Recall that UK-means performs k ED computations per object per iteration, Figure 6 thus show that all six pruning algorithms are very effective for a wide range of values of k. To better illustrate the algorithms' pruning effectiveness with respect to the basic UK-means algorithm, we plot N_{ED}/k against k in Figure 7. The figure thus shows the fraction of expected distances computed by the various algorithms compared with UK-means.

From the figure, we see that the values of N_{ED}/k are very small. The pruning algorithms are thus very effective. For example, when $k = 4$, MinMax-BB and VDBiP-SHIFT computed 6.34% and 2.38% of the EDs computed by UK-means, respectively. These translate into a pruning effectiveness of 93.66% and 97.62%, respectively. The best-performing algorithm VDBiP-SHIFT thus computes 62.5% fewer EDs than MinMax-BB. Also, we see that N_{ED}/k decreases as k increases for all six pruning algorithms. In other words, the fraction of ED pruned by the algorithms increases when there are more clusters. The pruning effectiveness of VDBi is seen to be consistently better than MinMax-BB over the whole range of k value. By achieving additional pruning using partial ED computation, VDBiP performs even better than VDBi. Furthermore, all hybrid algorithms are more effective than their non-hybrid counterparts. The results also show that, consistently, VDBiP-SHIFT is more effective than VDBi-SHIFT, which in turns, performs better than MinMax-SHIFT.

5.6. Effects of the Size of MBR

To study the effect of the extent of uncertainty on the algorithms' performance, we vary d, the maximum side length of an object's MBR, from 1.0 to 25. Other parameters are kept at their baseline values. Essentially, a larger MBR implies a larger uncertainty region and so an object's location is *more uncertain*. The results are shown in Figure 8.

We can see from the graph that N_{ED} increases as the size of the MBRs increases. This is because as the size of the MBRs increases, it is more likely that the MBRs overlap among one another or with multiple Voronoi cells. The former causes MinMax-based pruning to fail and the latter causes Voronoi-cell pruning and bisector pruning to fail. Even though the pruning effectiveness decreases when d becomes large, Figure 8 shows that the overall pruning effectiveness is still very impressive (recall that N_{ED} for the basic UK-means algorithm is 49). Comparing MinMax-BB and VDBiP-SHIFT, the latter prunes over two thirds of the EDs computed by the former. Our Voronoi-diagram-based pruning algorithms are thus seen to outperform the corresponding MinMax-based algorithms by a wide margin.

6. Conclusions

In this paper we have studied the problem of clustering uncertain objects whose locations are represented by probability density functions. We have discussed the UK-means algorithm [4], which was the first algorithm to solve the problem. We have explained that the computation of expected distances dominates the clustering process, especially when the number of samples used in representing objects' pdfs is large. We have mentioned an existing pruning technique MinMax-BB and its improved variant MinMax-SHIFT [19]. Although these techniques can improve the efficiency of UK-means, they do not consider the spatial relationship among cluster representatives.

To further improve the performance of UK-means, we have devised new pruning techniques that are based on Voronoi diagrams. The VDBi algorithm achieves effective pruning by two pruning methods: Voronoi-cell pruning and bisector pruning. We have proved theoretically that bisector pruning is strictly stronger than MinMax-BB. Furthermore, we have proposed the idea of pruning by partial ED calculations and have incorporated the method in VDBiP. We have also noticed that the different pruning techniques, employing different pruning criteria, can be combined. This leads to two hybrid algorithms VDBi-SHIFT and VDBiP-SHIFT that are highly effective.

We have conducted extensive experiments to evaluate the relative performance of the various pruning algorithms. The results show that our new pruning techniques outperform MinMax-BB consistently over a wide range of experimental parameters. The overhead of computing Voronoi diagrams for our Voronoi-diagram-based technique is paid off by the large number of ED calculations saved. The experiments also consistently demonstrated that the hybrid algorithms can prune more effectively than the other algorithms. Therefore, we conclude that our innovative pruning techniques based on Voronoi diagrams are effective and practical.

References

[1] C. C. Aggarwal. On density based transforms for uncertain data mining. In *ICDE*, pages 866–875, 2007.

[2] D. Barbará, H. Garcia-Molina, and D. Porter. The management of probabilistic data. *IEEE Trans. Knowl. Data Eng.*, 4(5):487–502, 1992.

[3] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, USA, 1981.

[4] M. Chau, R. Cheng, B. Kao, and J. Ng. Uncertain data mining: An example in clustering location data. In *PAKDD*, pages 199–204, Singapore, 9–12 Apr. 2006. Springer.

[5] J. Chen and R. Cheng. Efficient evaluation of imprecise location-dependent queries. In *ICDE*, pages 586–595, 2007.

[6] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *IEEE Trans. Knowl. Data Eng.*, 16(9):1112–1127, 2004.

[7] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In *VLDB*, pages 876–887, 2004.

[8] C. K. Chui, B. Kao, and E. Hung. Mining frequent itemsets from uncertain data. In *PAKDD*, pages 47–58, 2007.

[9] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *The VLDB Journal*, 16(4), 2007.

[10] F. K. H. A. Dehne and H. Noltemeier. Voronoi trees and clustering problems. *Inf. Syst.*, 12(2):171–175, 1987.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B39:1–38, 1977.

[12] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(32V57), 1973.

[13] H. Hamdan and G. Govaert. Mixture model clustering of uncertain data. In *Proc. Intl. Conf. on Fuzzy Systems*, 2005.

[14] F. Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In *SIGMOD*, 2000.

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

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

[17] S. D. Lee, B. Kao, and R. Cheng. Reducing UK-means to K-means. In *The 1st Workshop on Data Mining of Uncertain Data (DUNE), in conjunction with ICDM*, 2007.

[18] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Math. Stat. and Prob.*, pages 281–297, 1967.

[19] W. K. Ngai, B. Kao, C. K. Chui, R. Cheng, et al. Efficient clustering of uncertain data. In *ICDM*, pages 436–445, 2006.

[20] E. H. Ruspini. A new approach to clustering. *Information and Control*, 15(1):22–32, 1969.

[21] M. Sato, Y. Sato, and L. C. Jain. *Fuzzy Clustering Models and Applications*. Physica-Verlag, 1997.

[22] I. Stanoi, M. Riedewald, D. Agrawal, and A. E. Abbadi. Discovery of influence sets in frequently updated databases. In *VLDB*, pages 99–108, Roma, Italy, 11–14 Sept. 2001.

[23] M. Tabakov. A fuzzy clustering technique for medical image segmentation. In *2006 International Symposium on Evolving Fuzzy Systems*, pages 118–122, Sept. 2006.

[24] Y. Tao, D. Papadias, and X. Lian. Reverse kNN search in arbitrary dimensionality. In *VLDB*, pages 744–755, 2004.