Clustering Uncertain Data via K-medoids

Francesco Gullo, Giovanni Ponti, Andrea Tagarelli

DEIS, University of Calabria, Via P.Bucci 41c, Rende (CS) I87036, Italy
e-mail: {fgullo,gponti,tagarelli}@deis.unical.it

Abstract. Uncertain data are usually represented in terms of an uncertainty region over which a probability density function (pdf) is defined. In the context of uncertain data management, there has been a growing interest in clustering uncertain data. In particular, the classic K-means clustering algorithm has been recently adapted to handle uncertain data. However, the centroid-based partitional clustering approach used in the adapted K-means presents two major weaknesses that are related to: *(i)* an accuracy issue, since cluster centroids are computed as deterministic objects using the expected values of the pdfs of the clustered objects; and, *(ii)* an efficiency issue, since the expected distance between uncertain objects and cluster centroids is computationally expensive.
In this paper, we address the problem of clustering uncertain data by proposing a K-medoids-based algorithm, called *UK-medoids*, which is designed to overcome the above issues. In particular, our UK-medoids algorithm employs distance functions properly defined for uncertain objects, and exploits a K-medoids scheme. Experiments have shown that UK-medoids outperforms existing algorithms from an accuracy viewpoint while achieving reasonably good efficiency.

1 Introduction

Handling uncertainty in data management has been requiring more and more importance in a wide range of application contexts. Indeed, data uncertainty naturally arises from, e.g., implicit randomness in a process of data generation/acquisition, imprecision in physical measurements, and data staling. Various notions of uncertainty have been defined depending on the application domain (e.g., [2–8]). In general, uncertainty can be considered at table, tuple or attribute level [9], and is usually specified by fuzzy models [10], evidence-oriented models [11, 12], or probabilistic models [13].

In this paper, we focus on data containing attribute-level uncertainty, which is modeled according to a probabilistic model. We hereinafter refer to this data as *uncertain objects*. An uncertain object is usually represented by means of *probability density functions* (pdfs), which describe the likelihood that the object appears at each position in a multidimensional space [14, 15, 1], rather than by a traditional vectorial form of deterministic values.

Attribute-level uncertainty expressed by means of probabilistic models is present in several application domains. For instance, sensor measurements may be imprecise at a certain degree due to the presence of various noisy factors (e.g.,

signal noise, instrumental errors, wireless transmission) [16, 14]. To address this issue, it is advisable to model sensor data as continuous pdfs [17, 18]. Another example is given by data representing moving objects, which continuously change their location so that exact positional information at a given time instant may be unavailable [19]. Further examples come from distributed applications, privacy preserving data mining, and forecasting or other statistical techniques used to generate data attributes [20].

Dealing with uncertain objects has raised several issues in data management and knowledge discovery. In particular, organizing uncertain objects is challenging since the intrinsic difficulty underlying the various notions of uncertainty. As a major exploratory task of data mining, *clustering* is organizing a collection of objects (whose classification is unknown) into meaningful groups (clusters), based on interesting relationships discovered in the data. Objects within a cluster will be each other highly similar, but will be very dissimilar from objects in other clusters. One of the most popular clustering approaches is represented by partitional (or partitioning) clustering [21], which iteratively assigns objects to the clusters according to a certain distance/similarity function. A major cruciality in partitional clustering is how to devise a notion of cluster prototype. In particular, a cluster prototype can be defined as a *centroid*, which is the "mean" object in the cluster, or as a *medoid*, which is an actual object that is nearest to all the other objects in the cluster. The K-means [22] and K-medoids [23] algorithms are the exemplary methods of centroid-based and medoid-based partitional clustering, respectively.

In a recent work [1], the K-means algorithm has been adapted to the uncertain data domain. However, the resulting algorithm, named UK-means, has two major weak points. First, cluster centroids are defined as deterministic objects and computed as the mean of the expected values over the pdfs of the uncertain objects in the cluster; defining centroids in this way may result in loss of accuracy, since only the expected values of the pdfs of the uncertain objects are taken into account. Second, the computation of the Expected Distance (ED) between cluster centroids and uncertain objects is computationally expensive, as it requires non-trivial numerical integral estimations; this represents an efficiency bottleneck at each iteration of the algorithm.

In this paper, we present *UK-medoids*, an algorithm for clustering uncertain objects based on the K-medoids clustering scheme. The proposed algorithm exploits a distance function for uncertain objects, which is not limited to consider only scalar values derived from the pdfs associated to the objects (e.g., pdf expected values). This allows for better estimating the real distance between two uncertain objects, leading to significant improvement of the clustering quality. Also, our algorithm does not require any expensive operation to be repeated at each iteration; indeed, the computation of the distances between uncertain objects in the dataset is performed only once, thus guaranteeing a significant improvement of the efficiency w.r.t. UK-means. Experiments have shown that our method outperforms existing algorithms from an accuracy viewpoint while achieving reasonably good efficiency.

The rest of the paper is organized as follows. The next section discusses some related work. Section 3 describes the uncertain data models used in the paper. Section 4 describes the notion of uncertain distance and the UK-medoids algorithm. Section 5 provides experimental evaluation of our algorithm and the competing methods. Finally, Section 6 concludes the paper.

2 Related work

In the context of uncertain data management, a lot of research has been mainly focused on data representation and modeling, indexing, query processing, and data mining (e.g., [20]). In particular, data mining applications have involved various tasks, such as classification [24], outlier detection [25], association analysis [26], and clustering [27, 15, 1, 28, 29].

As above mentioned, one of the earliest attempts to solve the problem of clustering uncertain objects is UK-means [1]. In order to improve the UK-means efficiency, [28] proposes some pruning techniques to avoid the computation of redundant EDs. Such techniques make use of lower- and upper-bounds that are ad-hoc defined for each ED to be calculated; these bounds allow for eliminating some candidate assignments of objects to cluster centroids, avoiding the corresponding ED computation. However, a major problem of this approach is that it cannot guarantee high pruning (and, hence, high efficiency), as it depends on the features of the objects in the specific dataset. In [29], the *CK-means* is proposed as a variant of UK-means that resorts to the moment of inertia of rigid bodies in order to reduce the execution time needed for computing EDs. Unfortunately, the soundness of the CK-means criterion for the ED computation is guaranteed only if the mean squared error for the definition of the EDs is used and the distance function is based on the Euclidean norm.

It should be noted that all the UK-means variants have to face the issue of computing cluster centroids, whose effectiveness depends on how well the aggregated values (e.g., the expected values) extracted from the object pdfs represent the real location of the uncertain objects. Also, computing the distance between uncertain objects is usually accomplished by calculating the Euclidean distance between the vectors of the (deterministic) expected values.

A more refined approach to the distance computation consists in defining a univariate pdf, or *fuzzy distance function*, for each pair of objects. This univariate pdf computes a probability for each distance value for two objects, and the distance between the objects is finally computed by extracting an aggregated, representative value (e.g., expected value) from the pdf of those objects. This method has been originally presented in [27] and has been proved to be more effective than the standard Euclidean distance applied to vectors of deterministic values.

Devising a fuzzy distance function is a key aspect in density-based approaches that have been proposed for clustering uncertain objects [15, 27]. In [15], a fuzzy version of the popular DBSCAN [30] algorithm, $\mathcal{F}$DBSCAN, is proposed. Fuzzy distance functions are used to compute core object and reachability probabil-

ities, which are at the basis of the density-based clustering strategy of the algorithm. A similar approach is presented in [27], where $\mathcal{F}$OPTICS is proposed as a fuzzy version of the popular hierarchical density-based clustering algorithm OPTICS [31].

It is important to note that [15, 27] focus on how to efficiently compute reachability probabilities; however, they do not provide a formal definition of fuzzy distance function that can be applied to any clustering algorithm. By contrast, we provide a definition of fuzzy distance function that does not depend on a particular clustering scheme and is well-suited to continuous as well as discrete pdfs.

3 Modeling uncertain data

Representing attribute-level uncertain objects is traditionally accomplished by using two models, namely *multivariate uncertainty* and *univariate uncertainty* models.

Using a multivariate uncertainty model, an m-dimensional uncertain object is defined in terms of an m-dimensional region and a multivariate probability density function, which stores the probability according to which the exact representation of the object coincides with any point in the region. In a univariate uncertainty model, an m-dimensional uncertain object has, for each attribute, an interval and a univariate probability density function that assigns a probability value to any point within the interval. Formally:

Definition 1 (multivariate uncertain object). *A* multivariate uncertain object o *is a pair* (R, f)*, where* $R \subseteq \Re^m$ *is the region in which* o *is defined and* $f : \Re^m \rightarrow \Re_0^+$ *is the probability density function of* o *at each point* $\boldsymbol{z} \in R$*.*

Definition 2 (univariate uncertain object). *A* univariate uncertain object o *is a tuple* $(a^{(1)}, \ldots, a^{(m)})$*. Each attribute* $a^{(h)}$ *is a pair* $(I^{(h)}, f^{(h)})$*, for each* $h \in [1..m]$*, where* $I^{(h)} = [l^{(h)}, u^{(h)}]$ *is the interval of definition of* $a^{(h)}$*, and* $f^{(h)} : \Re \rightarrow \Re_0^+$ *is the probability density function that assigns a probability value to each* $z \in I^{(h)}$*.*

For each multivariate uncertain object, the probability density function involved in its representation can be either *continuous* or *discrete*. A continuous multivariate m-dimensional probability density function defined over a region $R \subseteq \Re^m$ is a function $f : \Re^m \rightarrow \Re_0^+$ such that:

$$\int_{\boldsymbol{z} \in R} f(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} = 1 \qquad \text{and} \qquad \int_{\boldsymbol{z} \in \Re^m \setminus R} f(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} = 0$$

A discrete multivariate m-dimensional probability density function defined over a set of points $S = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_v\}$ ($\boldsymbol{z}_u \in \Re^m$, for each $u \in [1..v]$) is a function

$f : \Re^m \rightarrow \Re_0^+$ such that:

$$\sum_{\boldsymbol{z} \in S} f(\boldsymbol{z}) = 1 \qquad \text{and} \qquad \int_{\boldsymbol{z} \in \Re^m \setminus S} f(\boldsymbol{z}) \mathrm{d}\ \boldsymbol{z} = 0$$

For the univariate model, a continuous (resp. discrete) univariate probability density function can be trivially defined in terms of a continuous (resp. discrete) multivariate probability density function, in which the region (resp. set) of definition is a subset of $\Re$ (i.e., $m = 1$).

We hereinafter refer to uncertainty models involving continuous probability functions. Note that this assumption does not result in loss of generality, since the corresponding "discrete" version can be obtained by simply replacing integrals with sums in the equations.

4 Clustering uncertain data

4.1 Computing uncertain distance

To measure the distance between uncertain objects, we need to devise a suitable notion of *uncertain distance*, which is involved in the proposed clustering algorithm. Uncertain distance is defined in terms of an *uncertain distance function*. In order to make the uncertain distance independent from the chosen uncertainty model, we provide definitions of uncertain distance function for both multivariate and univariate uncertainty models.

Definition 3 (uncertain distance function). *Given a set of uncertain objects $D = \{o_1, \ldots, o_n\}$, the* uncertain distance function *defined over D is a function $\Delta : D \times D \times \Re \rightarrow \Re_0^+$, for which the following conditions hold:*

$$\int_{z \in \Re} \Delta(o_i, o_j, z)\ \mathrm{d}z = 1, \quad \forall o_i, o_j \in D,$$

$$\Delta(o_i, o_j, z) = \begin{cases} 1, & \textit{if}\ \ i = j, z = 0 \\ 0, & \textit{if}\ \ i = j, z \neq 0 \end{cases}$$

For any pair of uncertain objects $o_i, o_j, i \neq j$, Δ can be derived from the pdfs associated to the uncertain objects. The definition of Δ depends on the uncertainty model used for representing o_i and o_j (Sect. 3).

Uncertain distance function for multivariate objects. If $o_i = (R_i, f_i)$, $o_j = (R_j, f_j)$ are multivariate uncertain objects, Δ is defined as:

$$\Delta(o_i, o_j, z) = \int_{\boldsymbol{x} \in R_i} \int_{\boldsymbol{y} \in R_j} I[dist(\boldsymbol{x}, \boldsymbol{y}) = z]\ f_i(\boldsymbol{x})\ f_j(\boldsymbol{y})\ \mathrm{d}\boldsymbol{x}\ \mathrm{d}\boldsymbol{y} \tag{1}$$

where $dist(\boldsymbol{x}, \boldsymbol{y})$ is a distance measure between any pair $\boldsymbol{x}, \boldsymbol{y} \in \Re^m$ (e.g., Euclidean distance), and $I[A]$ is the *indicator function*, which is equal to 1 when the event A occurs, 0 otherwise.

Uncertain distance function for univariate objects. If $o_i = ((I_i^{(1)}, f_i^{(1)}), \ldots, (I_i^{(m)}, f_i^{(m)}))$, $o_j = ((I_j^{(1)}, f_j^{(1)}), \ldots, (I_j^{(m)}, f_j^{(m)}))$ are univariate uncertain objects, Δ is defined as:

$$\Delta(o_i, o_j, z) = \int_{x_1 \in \Re} \cdots \int_{x_m \in \Re} I[f_{dist}(x_1, \ldots, x_m) = z] \prod_{h=1}^{m} \Psi^{(h)}(o_i, o_j, x_h) \, \mathrm{d}x_1 \cdots \mathrm{d}x_m \tag{2}$$

where

- $\Psi^{(h)} : D \times D \times \Re \rightarrow \Re$,
- $\Psi^{(h)}(o_i, o_j, x_h) = \int_{u \in I_i^{(h)}} \int_{v \in I_j^{(h)}} I[|u - v| = x_h] \; f_i^{(h)}(u) \; f_j^{(h)}(v) \; \mathrm{d}u \; \mathrm{d}v, \quad h \in [1..m]$,
- $f_{dist} : \Re^m \rightarrow \Re$ is a function that computes a scalar value from the components of a vector $(x_1, \ldots, x_m)$. In this work, this function is defined as $f_{dist} = \sqrt{(1/m) \sum_{h=1}^{m} {x_h}^2}$.

It can be proved that the condition $\int_{z \in \Re} \Delta(o_i, o_j, z) \; \mathrm{d}z = 1$ holds for both the definitions of Δ, for all o_i, o_j in the dataset.

Given an uncertain distance function Δ, we now provide a definition of uncertain distance by extracting a single, well-representative numerical value from Δ.

Definition 4 (uncertain distance). *Given a set of uncertain objects $D = \{o_1, \ldots, o_n\}$, let Δ be the uncertain distance function defined over D. The* uncertain distance *is a function $\delta : D \times D \rightarrow \Re_0^+$, which is defined as:*

$$\delta(o_i, o_j) = \int_{z \in \Re} z \Delta(o_i, o_j, z) \; \mathrm{d}z \tag{3}$$

According to Eq. (3), $\delta(o_i, o_j)$ is the expected value of the uncertain distance function Δ between o_i and o_j. Note that, if o_i, o_j are multivariate uncertain objects, $\delta(o_i, o_j)$ can be directly computed as:

$$\delta(o_i, o_j) = \int_{\boldsymbol{x} \in R_i} \int_{\boldsymbol{y} \in R_j} dist(\boldsymbol{x}, \boldsymbol{y}) \; f_i(\boldsymbol{x}) \; f_j(\boldsymbol{y}) \; \mathrm{d}\boldsymbol{x} \; \mathrm{d}\boldsymbol{y} \tag{4}$$

whereas, if o_i, o_j are univariate uncertain objects, $\delta(o_i, o_j)$ can be calculated as:

$$\delta(o_i, o_j) = f_{dist}(\psi^{(1)}(o_i, o_j), \ldots, \psi^{(m)}(o_i, o_j)) \tag{5}$$

where

$$\psi^{(h)}(o_i, o_j) = \int_{x \in I_i^{(h)}} \int_{y \in I_j^{(h)}} |x - y| \; f_i^{(h)}(x) \; f_j^{(h)}(y) \; \mathrm{d}x \; \mathrm{d}y, \quad h \in [1..m].$$

4.2 The UK-medoids algorithm

In this section we present our K-medoids-based algorithm for clustering uncertain objects, named *UK-medoids*. The outline of UK-medoids is given in Algorithm 1.

Algorithm 1 UK-medoids

Input: a set of uncertain objects $D = \{o_1, \ldots, o_n\}$; the number of output clusters k
Output: a set of clusters $\mathcal{C}$

1: compute distances $\delta(o_i, o_j), \forall o_i, o_j \in D$
2: compute the set $S = \{m_1, \ldots, m_k\}$ of initial medoids
3: **repeat**
4: $S' \leftarrow S$
5: $S \leftarrow \emptyset$
6: $\mathcal{C} = \{C_1, \ldots, C_k\} \leftarrow \{\emptyset, \ldots, \emptyset\}$
7: **for all** $o \in D$ **do**
8: {*assign each object to the closest cluster, based on its uncertain distance to cluster medoids*}
9: $m_j \leftarrow \arg\min_{o' \in S'} \delta(o, o')$
10: $C_j \leftarrow C_j \cup \{o\}$
11: **end for**
12: **for all** $C \in \mathcal{C}$ **do**
13: {*recompute the medoid of each cluster*}
14: $m \leftarrow \arg\min_{o \in C} \sum_{o' \in C} \delta(o, o')$
15: $S \leftarrow S \cup \{m\}$
16: **end for**
17: **until** $S \neq S'$
18: **return** $\mathcal{C}$

The input for the UK-medoids algorithm is a dataset D of n uncertain objects and the number k of clusters to be discovered, and the output is a set $\mathcal{C}$ of k clusters. Initially, all the uncertain distances between any pair of objects $o_i, o_j \in D$ are computed (Line 1). The distances are calculated only once and are used at each iteration of the algorithm. Then, the set of k initial medoids is computed (Line 2). The initial medoids can be selected by means of either random chance or a suitable procedure aimed to choose well-separated medoids (e.g., that proposed for the *Partitioning Around Medoids* (PAM) algorithm [32]).

After the initialization steps, the algorithm performs the main loop (starting from Line 3) which is comprised of two phases. In the first phase (Lines $7 - 11$), each object o in D is assigned to the cluster represented by the medoid m closest to o. In the second phase, the medoids in the set S are recomputed according to the objects assigned to each cluster (Lines $12 - 16$). Such phases are iteratively repeated until a local optimum has not been reached, i.e., there has been some change in the current S w.r.t. the previous iteration (Line 17).

Proposition 1. *Given a dataset D of n uncertain objects, Algorithm 1 works in $\mathcal{O}(n^2\ I)$, where I is the maximum number of iterations.*

5 Experimental evaluation

We devised an experimental evaluation aimed to assess the ability of our algorithm in clustering uncertain objects, both in terms of accuracy and efficiency. We also compared our UK-medoids to K-means-based uncertain data clustering algorithms, i.e., UK-means and its variant CK-means.

5.1 Evaluation methodology

Datasets. Experimental analysis was performed on benchmark datasets from the UCI Machine Learning Repository.[1] We chose four datasets with numerical real-value attributes, namely Iris, Wine, Glass, and Ecoli.

Table 1. Datasets used in the experiments

dataset	*objects*	*attributes*	*classes*
Iris	150	4	3
Wine	178	13	3
Glass	214	10	6
Ecoli	327	7	5

Table 1 shows the main characteristics of the datasets. Iris contains measurements on different iris plants. Wine reports results of a chemical analysis of Italian wines derived from three different cultivars. In Glass, each glass instance is described by the values of its chemical components. Ecoli contains data on the Escherichia Coli bacterium, which are identified with values coming from different analysis techniques.

All the selected datasets originally contain deterministic values, hence the uncertainty was synthetically generated for each object of any dataset. In case of univariate uncertain objects, we generated the uncertain interval $I^{(h)}$ and the pdf $f^{(h)}$ defined over $I^{(h)}$, for each attribute $a^{(h)}$, with $h \in [1..m]$ of the object o. The interval $I^{(h)}$ was randomly chosen as a subinterval within $[min_{o_h}, max_{o_h}]$, where min_{o_h} (resp. max_{o_h}) is the minimum (resp. maximum) deterministic value of the attribute h, over all the objects belonging to the same ideal class of o. As concerns $f^{(h)}$, we considered two continuous density functions, namely *Uniform* and *Normal* pdfs, and *Binomial* as a discrete mass function. We set the parameters of Normal and Binomial pdfs in such a way that their mode corresponded to the deterministic value of the h-th attribute of the object o.

We performed experiments for multivariate uncertain objects as well. In this case, we generated uncertainty starting from the univariate model, assuming

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

statistical independence for the pdfs of the attributes of any object. Since univariate and multivariate models gave similar results, here we report only results on the univariate models for the sake of brevity.

Clustering validity criteria. To assess the quality of clustering solutions we exploited the availability of reference classifications for the datasets. The objective was to evaluate how well a clustering fits a predefined scheme of known classes (natural clusters). To this purpose, we resorted to the *F-measure* [33], which is one of the most commonly used external validity criteria, and is defined in terms of the Information Retrieval notions' *Precision* and *Recall*.

Given a collection D of uncertain objects, let $\Gamma = \{\Gamma_1, \ldots, \Gamma_H\}$ be the reference classification of the objects in D, and $\mathcal{C} = \{C_1, \ldots, C_K\}$ be the output partition yielded by a clustering algorithm. Precision of cluster C_j with respect to class Γ_i is the fraction of the objects in C_j that has been correctly classified:

$$P_{ij} = \frac{|C_j \cap \Gamma_i|}{|C_j|}$$

Recall of cluster C_j with respect to class Γ_i is the fraction of the objects in Γ_i that has been correctly classified:

$$R_{ij} = \frac{|C_j \cap \Gamma_i|}{|\Gamma_i|}$$

Using a macro-averaging strategy on the local values of precision and recall, the overall precision (P) and recall (R) are computed as:

$$P = \frac{1}{H} \sum_{i=1}^{H} \max_{j \in [1..K]} P_{ij}, \qquad R = \frac{1}{H} \sum_{i=1}^{H} \max_{j \in [1..K]} R_{ij},$$

Finally, in order to score the quality of $\mathcal{C}$ w.r.t. Γ by means of a single value, the overall F-measure ($F \in [0, 1]$) is computed as the harmonic mean of the overall precision and recall:

$$F = \frac{2PR}{P + R} \tag{6}$$

Settings. In K-means-based approaches, the set of initial centroids is randomly selected. Therefore, to avoid that clustering results were biased by random chance, we averaged accuracy and efficiency measurements over 100 different runs. We made a similar choice also for UK-medoids, since we noted that the use of a refined strategy for selecting initial medoids (e.g., the procedure proposed in [32]) gave no significant improvement w.r.t. random selection.

We computed the integrals involved into the distances calculation by taking into account lists of samples derived from the pdfs. To accomplish this, we employed the classic *Monte Carlo* sampling method.[2] We also performed a preliminary tuning phase to properly set the number of samples S; in particular, for

[2] We used the SSJ library, available at http://www.iro.umontreal.ca/∼simardr/ssj/

Table 2. Clustering quality results (F-measure)

dataset	*pdf*	UK-means	CK-means	**UK-medoids**
Iris	Uniform	0.45	*0.50*	**0.84**
	Normal	0.84	*0.85*	**0.88**
	Binomial	*0.62*	0.58	**0.87**
Wine	Uniform	0.46	*0.50*	**0.80**
	Normal	0.69	*0.70*	**0.70**
	Binomial	*0.63*	0.58	**0.73**
Glass	Uniform	0.26	*0.29*	**0.71**
	Normal	*0.63*	0.59	**0.68**
	Binomial	0.27	*0.29*	**0.67**
Ecoli	Uniform	0.30	*0.33*	**0.73**
	Normal	0.73	*0.74*	**0.77**
	Binomial	*0.50*	0.44	**0.72**

each method and dataset, we chose S in such a way that there was no significant improvement in accuracy for any $S' > S$. In general, the optimal S depended on the width of the uncertainty interval/region; however, according to our experiments, 50 and $400 \div 500$ samples represented a reasonably good choice, for univariate and multivariate uncertainty model, respectively.

5.2 Results

Accuracy. Table 2 summarizes the F-measure results obtained by UK-medoids and the other methods. We can observe that UK-medoids drastically outperformed UK-means and CK-means on all the datasets, with Uniform and Binomial pdfs. In particular, compared to best competing method, the accuracy improvement obtained by our UK-medoids was from 34% to 42% with Uniform pdfs and from 10% to 38% with Binomial pdfs. In case of Normal pdfs, UK-medoids performed $3 \div 5$% better than the other methods on three datasets, whereas all the methods behaved similarly in Wine. The reduction of gap between UK-medoids and K-means-based approaches on Normal pdfs can be explained in that, according to our uncertainty generation scheme, the expected value of a Normal pdf associated to any attribute of each uncertain object was set equal to the deterministic value of the attribute for that object. This allowed the centroid generation strategy of UK-means and CK-means to perform well in that case.

It should be also noted that UK-means and CK-means performed similarly for all the pdfs and datasets, as expected, since they employ a similar clustering scheme; the only differences between the two methods are due to random choices, such as selection of initial centroids and pdf sampling for the computation of the integrals.

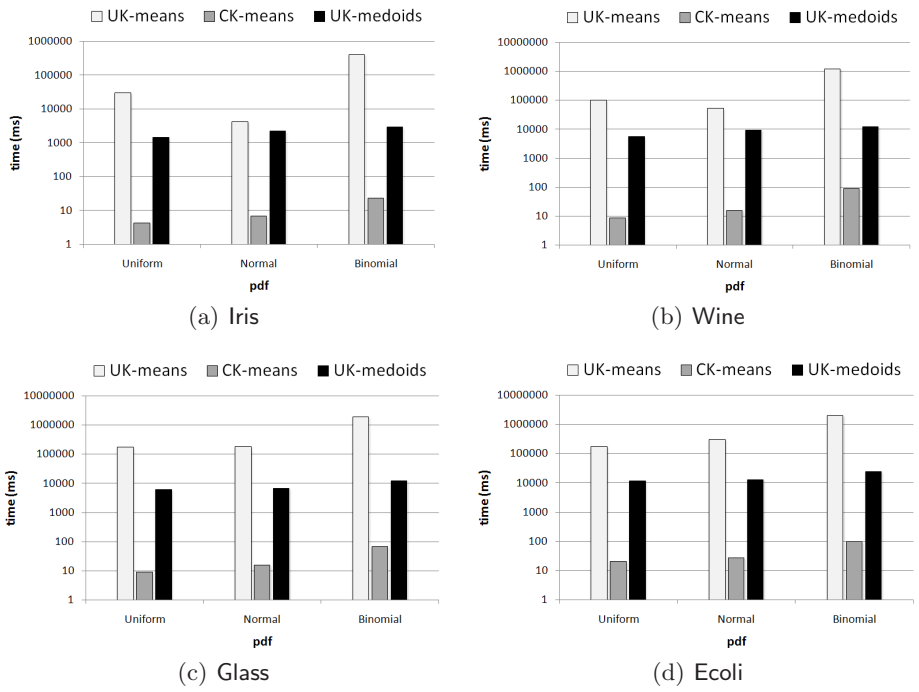

Fig. 1. Clustering time performances

Efficiency. To evaluate the efficiency of UK-medoids and the competing methods, we measured their time performances in clustering uncertain objects.[3] Figure 1 shows the total execution times (in milliseconds) obtained by the methods on the various datasets. For UK-medoids and CK-means, we calculated the sum of the times obtained for the pre-computing phase (i.e., uncertain distances computation for UK-medoids and cluster centroids computation for CK-means), together with the algorithm runtimes.

In the figure, it can be noted that our UK-medoids was $1 \div 2$ orders of magnitude faster than UK-means, which was the slowest method on all datasets. The slowness of UK-means is mainly due to the EDs computation needed for each object in the dataset, at each iteration of the algorithm.

As expected, CK-means outperformed UK-medoids on all datasets, which is explained by a difference between the computational complexities of the two algorithms. Indeed, both the phases of pre-computing and algorithm execution are quadratic (resp. linear) with the number of objects in the dataset for UK-medoids (resp. CK-means). However, it should be emphasized that the CK-means algorithm is less general than the other methods, as it works only if the

[3] Experiments were conducted on a platform Intel Pentium IV 3GHz with 2GB memory and running Microsoft WinXP Pro

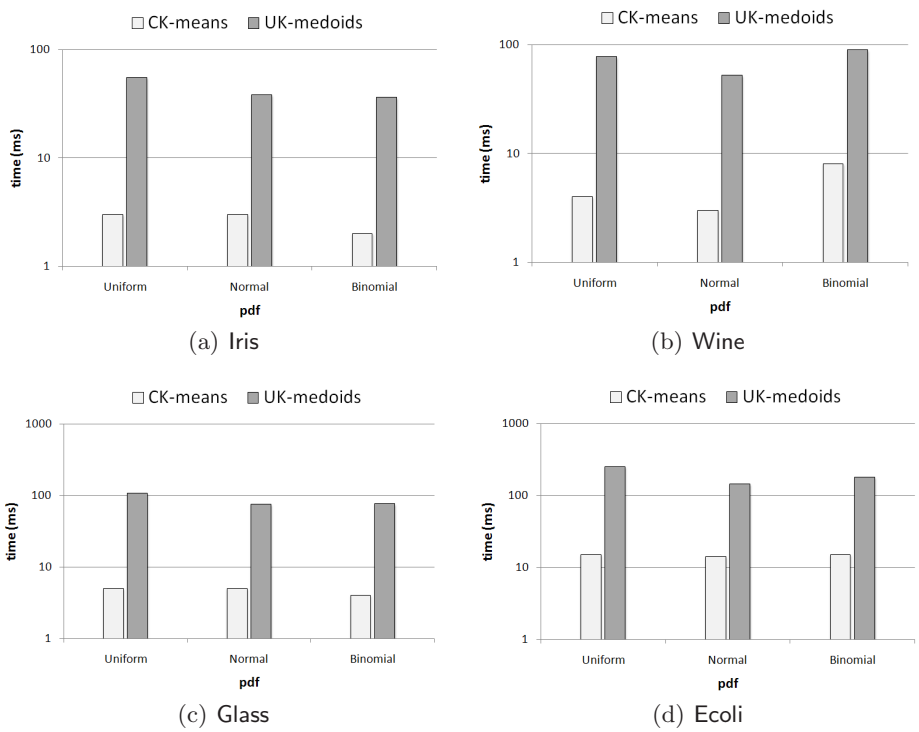

Fig. 2. Performance of the algorithm runtimes (pre-computing phases are ignored)

mean squared error for the definition of the EDs is used and the distance function is based on the Euclidean norm.

We also measured separately the times of the pre-computing phases, which involve the calculation of uncertain distances (in UK-medoids) and cluster centroids (in CK-means). Figure 2 shows that the gap between UK-medoids and CK-means was reduced w.r.t. that measured by including the total runtimes (Figure 1). This result confirms that the major difference between UK-medoids and CK-means is given by the pre-computing phase. Thus, in case of multiple runs of the two algorithms, we can state that the performance of UK-medoids and CK-means are comparable, since the pre-computing phase has to be performed once.

6 Conclusion

We addressed the problem of clustering uncertain objects based on an efficient K-medoids clustering scheme. We provided distance functions for both univariate and multivariate uncertain objects, which are well-suited to continuous as well as discrete pdfs. Moreover, these functions are designed to better estimate the real

distance between two uncertain objects since they are not limited to consider only scalar values derived from the object pdfs.

Our UK-medoids has been experimentally shown to outperform other existing methods in terms of accuracy, regardless of the choice of uncertainty density function. Also, from an efficiency viewpoint, UK-medoids performs up to two orders of magnitude faster than the baseline method UK-means.

References

1. Chau, M., Cheng, R., Kao, B., Ng, J.: Uncertain Data Mining: An Example in Clustering Location Data. In: Proc. PAKDD Conf. (2006) 199–204
2. Imielinski, T., Lipski Jr., W.: Incomplete Information in Relational Databases. Journal of the ACM **31**(4) (1984) 761–791
3. Abiteboul, S., Kanellakis, P., Grahne, G.: On the Representation and Querying of Sets of Possible Worlds. In: Proc. SIGMOD Conf. (1987) 34–48
4. Sadri, F.: Modeling Uncertainty in Databases. In: Proc. ICDE Conf. (1991) 122–131
5. Lakshmanan, L.V.S., Leone, N., Ross, R.B., Subrahmanian, V.S.: ProbView: A Flexible Probabilistic Database System. ACM TODS **22**(3) (1997) 419–469
6. Dalvi, N.N., Suciu, D.: Efficient Query Evaluation on Probabilistic Databases. In: Proc. VLDB Conf. (2004) 864–875
7. Green, T., Tannen, V.: Models for Incomplete and Probabilistic Information. IEEE Data Engineering Bulletin **29**(1) (2006) 17–24
8. Aggarwal, C.C.: On Density Based Transforms for Uncertain Data Mining. In: Proc. ICDE Conf. (2007) 866–875
9. Tao, Y., Xiao, X., Cheng, R.: Range Search on Multidimensional Uncertain Data. TODS **32**(3) (2007) 15–62
10. Galindo, J., Urrutia, A., Piattini, M.: Fuzzy Databases: Modeling, Design, and Implementation. Idea Group Publishing (2006)
11. Lee, S.K.: An Extended Relational Database Model for Uncertain and Imprecise Information. In: Proc. VLDB Conf. (1992) 211–220
12. Lim, E.P., Srivastava, J., Shekhar, S.: An Evidential Reasoning Approach to Attribute Value Conflict Resolution in Database Integration. TKDE **8**(5) (1996) 707–723
13. Sarma, A.D., Benjelloun, O., Halevy, A., Widom, J.: Working Models for Uncertain Data. In: Proc. ICDE Conf. (2006) 7–18
14. Cheng, R., Kalashnikov, D.V., Prabhakar, S.: Evaluating probabilistic queries over imprecise data. In: Proc. SIGMOD Conf. (2003) 551–562
15. Kriegel, H.P., Pfeifle, M.: Density-Based Clustering of Uncertain Data. In: Proc. ACM SIGKDD Conf. (2005) 672–677
16. Cantoni, V., Lombardi, L., Lombardi, P.: Challenges for Data Mining in Distributed Sensor Networks. In: Proc. ICPR Conf. (2006) 1000–1007
17. Faradjian, A., Gehrke, J., Bonnet, P.: GADT: A Probability Space ADT for Representing and Querying the Physical World. In: Proc. ICDE Conf. (2002) 201–211
18. Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J.M., Hong, W.: Model-based approximate querying in sensor networks. VLDB Journal **14**(4) (2005) 417–443
19. Li, Y., Han, J., Yang, J.: Clustering Moving Objects. In: Proc. ACM SIGKDD Conf. (2004) 617–622

20. Aggarwal, C.C., Yu, P.S.: A Survey of Uncertain Data Algorithms and Applications. Technical Report RC24394, IBM Research Division, Thomas J. Watson Research Center (October 2007)
21. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall (1988)
22. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proc. Berkeley Symposium on Mathematical Statistics and Probability. (1967) 281–297
23. L. Kaufman and P. J. Rousseeuw: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley (1990)
24. Bi, J., Zhang, T.: Support Vector Classification with Input Data Uncertainty. In: Proc. NIPS Conf. (2004) 483–493
25. Aggarwal, C.C., Yu, P.S.: Outlier Detection with Uncertain Data. In: Proc. SDM Conf. (2008) 483–493
26. Chui, C.K., Kao, B., Hung, E.: Mining Frequent Itemsets from Uncertain Data. In: Proc. PAKDD Conf. (2007) 47–58
27. Kriegel, H.P., Pfeifle, M.: Hierarchical Density-Based Clustering of Uncertain Data. In: Proc. ICDM Conf. (2005) 689–692
28. Ngai, W.K., Kao, B., Chui, C.K., Cheng, R., Chau, M., Yip, K.Y.: Efficient Clustering of Uncertain Data. In: Proc. ICDM Conf. (2006) 436–445
29. S. D. Lee and B. Kao and R. Cheng: Reducing UK-means to K-means. In: Proc. ICDM Workshops. (2007) 483–488
30. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proc. ACM SIGKDD Conf. (1996) 226–231
31. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering Points To Identify the Clustering Structure. In: Proc. SIGMOD Conf. (1999) 49–60
32. Kaufmann, L., Rousseeuw, P.J.: Clustering by means of medoids. In: Proc. Statistical Data Analysis based on the L_1 Norm Conf. (1987) 405–416
33. van Rijsbergen, C.J.: Information Retrieval. Butterworths (1979)