An Information-Theoretic Approach to Hierarchical Clustering of Uncertain Data

Francesco Gullo[a], Giovanni Ponti[b], Andrea Tagarelli[c], Sergio Greco[c]

[a] *UniCredit, R&D Dept., Rome, Italy*
[b] *ENEA, Portici Research Center, Portici (NA), Italy*
[c] *DIMES, University of Calabria, Rende (CS), Italy*

Abstract

Uncertain data clustering has become central in mining data whose observed representation is naturally affected by imprecision, staling, or randomness that is implicit when storing this data from real-word sources. Most existing methods for uncertain data clustering follow a partitional or a density-based clustering approach, whereas little research has been devoted to the hierarchical clustering paradigm. In this work, we push forward research in hierarchical clustering of uncertain data by introducing a well-founded solution to the problem via an information-theoretic approach, following the initial idea described in our earlier work [26]. We propose a prototype-based agglomerative hierarchical clustering method, dubbed *U-AHC*, which employs a new uncertain linkage criterion for cluster merging. This criterion enables the comparison of (sets of) uncertain objects based on information-theoretic as well as expected-distance measures. To assess our proposal, we have conducted a comparative evaluation with state-of-the-art algorithms for clustering uncertain objects, on both benchmark and real datasets. We also compare with two basic definitions of agglomerative hierarchical clustering that are treated as baseline methods in terms of accuracy and efficiency of the clustering results, respectively. Main experimental findings reveal that U-AHC generally outperforms competing methods in accuracy and, from an efficiency viewpoint, is comparable to the fastest baseline version of agglomerative hierarchical clustering.

Keywords: Clustering, Hierarchical clustering, Uncertain data, Information

Email addresses: gullof@acm.org (Francesco Gullo), giovanni.ponti@enea.it (Giovanni Ponti), tagarelli@dimes.unical.it (Andrea Tagarelli), greco@dimes.unical.it (Sergio Greco)

theory, Probability distributions, Mixture models

1. Introduction

Uncertainty in data arises from a variety of real-world phenomena, such as implicit randomness in data generation/acquisition, imprecision in physical measurements, and data staling [2]. It is usually related to incomplete/missing information [31, 1], or to the probability of occurrence of a given information [40, 13, 21]. For instance, sensor measurements may be imprecise due to the presence of various noisy factors (e.g., signal noise, instrumental errors, wireless transmission) [15]. Moving objects continuously change their location so that the exact positional information at a given time instant may be unavailable [50]. In data integration, uncertainty can arise from approximation assumptions on the semantic mappings between the data sources and the mediated schema or poor knowledge about the exact mapping [4]. The biomedical research domain also abounds of data affected by uncertainty; as an example, in the context of gene expression microarray data, handling the so-called probe-level uncertainty represents a key aspect that enables more expressive data representation and more accurate processing [42].

Uncertainty can be considered at different granularities and various modeling approaches have been developed in data management [47]. In general, uncertainty can be considered at *table*, *tuple* or *attribute* level [49]: this work focuses on data containing attribute-level uncertainty modeled according to *probability distributions*, which has attracted major attention in data mining research in recent years [38, 39, 10, 41, 23]. In this work, we will hence refer to an *uncertain object* as a data object represented in terms of a multidimensional region and a probability distribution that describes the likelihood that exact object representations correspond to any specific point in that region.

Mining uncertain objects is inherently difficult as the uncertainty in data representation needs to be carefully handled in order to produce meaningful knowledge patterns. Consider for instance the scenario depicted in Fig. 1—uncertain objects are represented in terms only of their domain regions for the sake of simplicity (probability distribution assumed to be uniform for all the objects). The "true" representation of each uncertain object (black circles in Fig. 1(a)) corresponds to a point within its domain region and can be in general far away from its "observed" representation (black circles in Fig. 1(b)). Thus, considering only the observed representations may lead to discover groups of similar objects (i.e.,

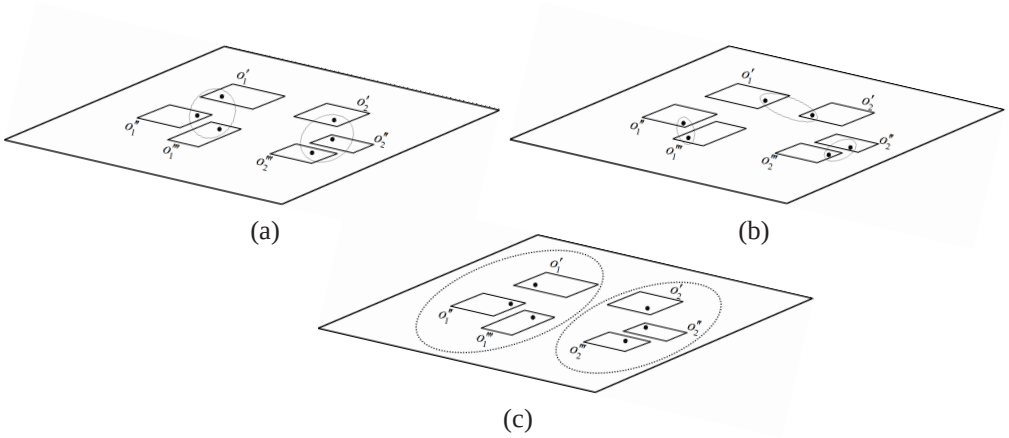


Figure 1: Clustering in an uncertain dataset: (a) true representations of objects and their desired grouping, (b) observed representations which may lead to unexpected groupings, (c) desired grouping identified by considering the object uncertainty (domain regions).

$\{o'_1, o'_2\}$, $\{o''_1, o'''_1\}$, $\{o''_2, o'''_2\}$ in Fig. 1(b)) that are substantially different from the ideal ones which would be identified by considering the true representations (i.e., $\{o'_1, o''_1, o'''_1\}$, $\{o'_2, o''_2, o'''_2\}$ in Fig. 1(a)). Instead, taking into account uncertainty, i.e., considering the whole domain regions (and pdfs) of the uncertain objects, may help to recognize the correct clustering (Fig. 1(c)).

Clustering uncertain objects has emerged in the last decade in data mining research, originally with the general mission of reviewing and extending the traditional (deterministic data) clustering methods to a particular probabilistic context of data representation. Like traditional (deterministic) clustering, a crucial step lies in the definition of a proper proximity measure. Two major approaches to comparing uncertain objects have been so far defined: one approach is to compute the difference between some aggregated values, e.g., *expected values*, from the distributions of the uncertain objects; the other approach instead exploits the whole information available from the distributions by involving the computation of the so-called *expected distance* (ED) [23]. Although relatively efficient, the expected-value-based approach may be inaccurate, since the whole information available from the distributions is collapsed into a single numerical value; by contrast, the ED-based approach is more accurate but also less efficient, as its computation typically requires slow numerical estimation of the integrals involved.

Several studies in clustering uncertain objects have led to uncertain versions of the classic K-means and other partitional algorithms [10, 41, 34, 23, 24], as well as of density-based algorithms such as DBSCAN and OPTICS [39, 38]. Surpris-

ingly, the hierarchical clustering paradigm has been nearly out of focus, despite its features (i.e., hierarchical presentation of the clusters, relative independence on input parameters and versatility in the shape of the clusters detected) make it particularly appealing to handle uncertainty in a large variety of application domains. In the following, we briefly discuss some of the application scenarios that might benefit from a hierarchical clustering approach in an uncertain data setting.

Applications. It has been widely recognized that the task of *document clustering* takes large benefit from a representation of the documents via a statistical topic model, such as Latent Dirichlet Allocation or derived models. In fact, in recent years, a number of approaches to modeling document contents have been developed based on the idea that any document can be represented as a mixture of probability distributions over its terms, and each component of the mixture refers to a *topic*. Organizing topic-model-based text data in conceptual hierarchies becomes essential in scenarios where documents need to be assigned (clustered) to multiple topics, for which an explanation in terms of hierarchically related subtopics is required.

As another example, *gene-expression data* typically exhibit the so-called *probe-level uncertainty*, i.e., uncertainty due to human/instrumental errors that affect the data-acquisition process therein. This gives rise to gene-expression data naturally represented as uncertain objects. Hierarchical clustering of this probabilistically-represented gene-expression data finds application in the task of predicting the protein functions. In this context, in fact, protein functions typically need to be organized in a hierarchy, since each protein can have more than one function, which in turn can have more than one sub-function. Hierarchical clustering here can also support hierarchical multi-label classification which is a common task in protein function prediction, where the training instances are organized according to a hierarchy predefined in a functional genomics setting.

Yet, in the *sensor data domain*, besides the motion uncertainty that leads to probabilistic representations of the data outputted by the sensors, a critical challenge is also represented by a partial observability of the motion system, i.e., *environmental-sensing uncertainty*. This type of uncertainty might be treated by using different sensors (cameras) able to detect the position of the objects from different perspectives. A critical task in this context is to exploit such sensor data in order to reconstruct a hierarchical representation of the environment therein where each level of the hierarchy corresponds to a representation of the environment at a specific level of granularity. A hierarchical clustering of the objects according to the uncertain representations of their position (coming from differ-

ent sensors) can be directly exploited to define such a hierarchical representation of the environment.

Challenges. Hierarchical algorithms for clustering uncertain objects could in principle be defined by involving one of the aforementioned distance computation approaches for uncertain objects (i.e., the faster distance between aggregate values or the more accurate expected distance) into a standard criterion of cluster linkage (e.g., single-, complete-, or group-average linkage) used in the classic agglomerative hierarchical clustering (AHC) scheme. Unfortunately, the previously discussed effectiveness and/or efficiency issues that would arise from the existing notions of uncertain object distance make this solution inappropriate.

A major challenge in hierarchical clustering of uncertain objects is thus the definition of a linkage criterion that takes the advantages of existing notions of distance between uncertain objects, and as such it should be efficient yet accurate. Particularly, to fulfil the accuracy requirement, it should exploit the whole information available from the distributions of the uncertain objects, like ED. *Information Theory* (IT) has represented a fruitful research area to devise measures for comparing probability distributions. IT measures indeed compute the distance between two distributions accurately and, in most cases, in linear time with respect to the dimensionality m of the distributions to be compared. However, the prominent existing IT measures, such as the popular ones falling into the Ali-Silvey class [5], cannot be directly used to define distances for uncertain objects, mainly because of the assumption that the distributions need to share a common event space, which does not necessarily hold for distributions associated to uncertain objects. Resorting to IT measures hence represents a novel and challenging approach to defining proper linkage criteria in the context of hierarchical clustering of uncertain objects.

Contributions. In this paper, we propose an information-theoretic approach to hierarchical clustering of uncertain objects. While our earlier work [26] initially brought the hierarchical clustering paradigm to the context of uncertain objects, here we revise the key notions of cluster merging criterion and distance between uncertain cluster prototypes, and use them to originally pose the theoretical foundations of hierarchical clustering of uncertain objects.

We develop a prototype-based agglomerative hierarchical clustering method, called U-AHC, which employs as notion of cluster prototype a mixture model that summarizes the probability distributions of the objects within a given cluster. The cluster merging criterion relies on a novel *IT distance measure*, which is designed for comparing cluster prototypes effectively (as theoretically established) and ef-

ficiently (as its complexity equals that of the most efficient existing approach to computing the distance between uncertain objects).

A major novelty of the proposed IT distance is the original combination of the two opposite ways of comparing the distributions that represent groups of uncertain objects (uncertain cluster prototypes): measuring the distance by involving the entire distributions, and computing the difference between the expected values of the distributions. The intuition behind this definition of distance lies in the fact that comparing two probability distributions by appropriate IT measures is in general effective but not always feasible, and hence resorting to the expected values when needed will compensate for the lack of IT application requirements.

We demonstrate the soundness of our compound distance measure showing that it exploits an IT-based criterion that allows for determining how well an IT measure is suited for comparing cluster prototypes and, therefore, allows for weighting the relative importance of the IT term with respect to the expected value based term. Even though the proposed distance measure is in principle suitable for being coupled with any clustering scheme, we provide theoretical justifications for using such a measure as a linkage criterion into an AHC scheme. Our main theoretical finding in this regard concerns a well-founded relationship between the proposed compound distance, the cluster prototype, and the proposed U-AHC algorihm, which shows how the "suitability" of comparing any two cluster prototypes is monotonically increasing with the various steps of the AHC scheme.

We conducted an extensive experimental evaluation on several datasets, including datasets with uncertainty generated synthetically as well as real-world data collections in which uncertainty is inherently present. Compared to state-of-the-art partitional and density-based algorithms, U-AHC achieves the highest average quality on all datasets, in terms of both external and internal cluster-validity criteria. U-AHC is also compared to the aforementioned naïve hierarchical algorithms for clustering uncertain objects, achieving considerably better accuracy (resp. efficiency) results than the fastest (resp. most accurate) naïve algorithm.

Roadmap. The rest of the paper is organized as follows. Section 2 briefly discusses related work. Section 3 describes uncertain object and uncertain prototype modeling. Section 4 discusses distance measures for probability distributions and our proposal for comparing uncertain prototypes. Section 5 presents the U-AHC algorithm. This section also provides an insight into the relations existing between U-AHC, the uncertain prototype and the prototype distance measure. Section 6 presents experimental methodology and results, while Section 7 concludes the paper.

2. Related Work

One of the earliest approaches to clustering uncertain objects is *UK-means* [10], which is an adaptation of the popular K-means algorithm. UK-means relies on the distance between uncertain objects and (deterministic) cluster centroids, at each iteration. Improvements upon the efficiency of that algorithm have been proposed in [43, 34], where some pruning techniques are introduced to avoid the calculation of redundant object-to-centroid distances, and in [41], where the *CK-means* variant is defined based on a closed-form expression similar to that employed for computing the moment of inertia of rigid bodies. CK-means essentially comprises two steps: in the first (offline) step, the distances between each object and its mass center are computed, whereas the second step performs a classic partitional relocation scheme; in this step, the distances computed in the first step are exploited to obtain a K-means-like strategy. A major issue shared by UK-means and all its optimizations is the deterministic representation of cluster centroids. In [23], the *UK-medoids* algorithm is introduced to overcome this issue. It employs distances between uncertain objects that are pre-computed offline and then employed in a classic K-medoids scheme. A kernel-based version of UK-medoids is also defined in [52].

Alternative formulations to partitional centroid-based clustering of uncertain objects are defined in [25, 27]. More specifically, in [25], we take into account the variance of the individual set members (rather than the central tendency only), and propose a criterion based on the minimization of the variance of the mixture model that summarizes a set of uncertain objects. In [27], we define an uncertain prototype of a set (cluster) of uncertain objects as an uncertain object that is defined in terms of a random variable whose realizations correspond to all possible deterministic representations deriving from the uncertain objects to be summarized. A closed-form-computable compactness criterion, coupled with the proposed notion of uncertain prototype, enables an effective yet efficient objective criterion for grouping uncertain objects. Other approaches to partitional clustering of uncertain data have been developed focusing on the uncertain K-center problem [12, 22]. Cormode and McGregor [12] propose a number of bicriteria approximation algorithms, which have however a major weakness: they are unable to preserve the number of centers; this limitation is overcome in [22].

Density-based approaches to clustering uncertain objects have been defined in [38, 39]. In [38], the $\mathcal{F}$*DBSCAN* is proposed as a fuzzy version of the popular DBSCAN, mainly based on the use of fuzzy distance functions to compute the core object and reachability probabilities. Analogously, the $\mathcal{F}$*OPTICS* algo-

rithm [39] extends OPTICS, by producing in this case an augmented ordering of the objects based on the novel notion of fuzzy object reachability-distance.

Some works have focused on the computation of the similarity/distance between uncertain data objects according to their probability distributions. Hung et al. [30] propose to approximate the distance distribution between uncertain objects using a Gaussian or Gamma distribution. In [32], Kullback-Leibler divergence is estimated in the continuous case by kernel density estimation and the Gauss transform technique is exploited to speed up the similarity computation. Moreover, in [11], uncertain cross-entropy is studied under the requirement of estimating the uncertainty distribution of an uncertain variable from the known partial information.

Volk et al. [51] propose an approach based on the possible-world scenario, where a clustering solution is derived from each possible world, and the various solutions are eventually aggregated to form a unique clustering by employing standard methods for clustering aggregation. Züfle et al. [54] focus on quality guarantees of uncertain clustering results. They utilize a sampling approach to represent an uncertain dataset as a set of sample deterministic datasets, compute over them a set of possible clusterings, and then combine these clusterings into a concise set of τ-ϕ-representative clusterings, i.e., clustering solutions that have probability at least ϕ to be distant at most τ to the actual clustering of the data if the data were certain.

Marginally related to our work is research on high dimensionality in uncertain objects by addressing the problems of subspace clustering [28] and projective clustering [3]. The work in [14] focuses on maximum-likelihood parameter estimation with application to clustering uncertain data with categorical and continuous attributes. There are also different bodies of study on rough set theory, fuzzy set theory, and granular computing [17, 46, 20, 45] models that have been used in uncertain information processing, although not directly concerning uncertain data clustering. More specifically, rough set theory deals with uncertainty that is caused by the indistinguishability of objects, fuzzy set theory deals with the uncertainty caused by the smooth boundedness of a concept, while in a granular computing framework, any information is assumed to be uncertain to some degree and can be expressed as certain information at a coarser degree. Recently, in [53], an extension to the quotient space theory is proposed to describe hierarchical structures by using weighted equivalence relations and tolerance relations.

None of the above works studies the problem of hierarchical clustering of uncertain objects. To the best of our knowledge, the only existing work dealing with such a problem is the preliminary version of this paper [26], whose key notions

of cluster merging criterion and distance between uncertain cluster prototypes are revised and made theoretically well-founded in this work.

3. Uncertain objects and uncertain prototypes

Uncertain objects are typically represented according to *multivariate* uncertainty models [23], whose formal definition is reported next.

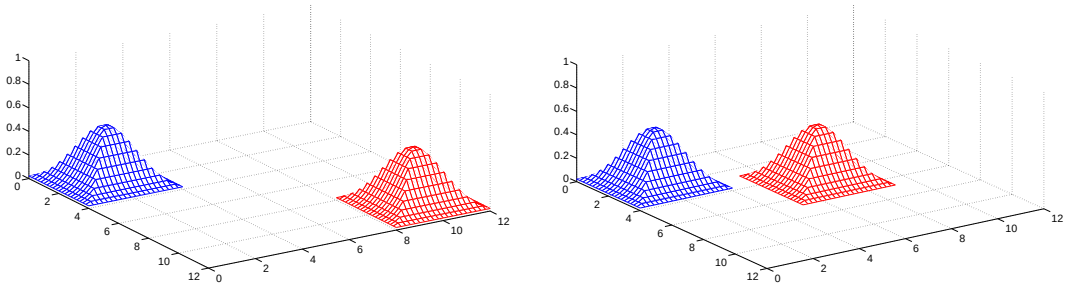
Definition 1 (multivariate uncertain object)**.** *A* multivariate uncertain object o *is a pair* $(\mathcal{R}, f)$*, where* $\mathcal{R} \subseteq \Re^m$ *is the* m*-dimensional region in which* o *is defined and* $f : \Re^m \rightarrow \Re_0^+$ *is the probability density function of* o *at each point* $\vec{x} \in \Re^m$*, such that* $f(\vec{x}) > 0, \forall \vec{x} \in \mathcal{R}$ *and* $f(\vec{x}) = 0, \forall \vec{x} \in \Re^m \setminus \mathcal{R}$*.* □

An *uncertain prototype* is an uncertain object designed to properly summarize the features of all uncertain objects in a given set. Since uncertain objects are represented by pdfs, it is reasonable to represent an uncertain prototype as a *finite mixture* obtained by the pdfs of the objects in the set to be summarized. Using finite mixtures allows for maintaining information about the uncertainty of the objects to be summarized, which makes the probabilistic representation particularly accurate. This contrasts with other definitions of prototypes (centroids) of uncertain objects that collapse the entire information about uncertainty into a single numerical value, like those employed in [10, 41]. Also, computing the mixture model of a set of random variables is fast as it can be performed in linear time w.r.t. the size of that set.

Definition 2 (uncertain prototype)**.** *Let* $C = \{o_1, \ldots, o_n\}$ *be a set of uncertain objects, where* $o_i = (\mathcal{R}_i, f_i)$*,* $\mathcal{R}_i \subseteq \Re^m$*, for each* $i \in [1..n]$*. The* uncertain prototype *of* C *is a pair* $P = (\mathcal{R}, f)$*, where* $\mathcal{R} = \bigcup_{i=1}^{n} \mathcal{R}_i$ *and* $f(\vec{x}) = (1/n) \sum_{i=1}^{n} f_i(\vec{x})$*.* □

According to the above definition, it can be straightforwardly proved that any uncertain prototype is also an uncertain object satisfying Def. 1. Also, the next proposition describes how to compute an uncertain prototype resulting from the union of two other prototypes in an efficient way, that is without iterating over all the uncertain objects that are at the basis of the prototype. Such a concept is exploited in Sect. 5 to define our U-AHC algorithm.

Proposition 1. *Given two sets* C'*,* C'' *of uncertain objects and their corresponding prototypes* $P' = (\mathcal{R}', f')$*,* $P'' = (\mathcal{R}'', f'')$*, let* $\widehat{C}$ *be the set given by the union*



(a) (b)

Figure 2: Two cases of uncertain objects sharing no common region.

of C' and C''. The prototype of $\widehat{C}$ is defined as $\widehat{P} = (\widehat{\mathcal{R}}, \widehat{f})$ such that:

$$\widehat{\mathcal{R}} = \mathcal{R}' \cup \mathcal{R}'', \qquad \textit{and} \qquad \widehat{f} = \frac{|C'|}{|\widehat{C}|} f' + \frac{|C''|}{|\widehat{C}|} f'' - \frac{|C' \cap C''|}{|\widehat{C}|} f''',$$

where $f''' = (1/|C' \cap C''|) \sum_{o_{12}=(\mathcal{R}_{12}, f_{12}) \in C' \cap C''} f_{12}$.

Proof. Concerning the region $\widehat{\mathcal{R}}$, it holds that $\widehat{\mathcal{R}} = \bigcup_{(\mathcal{R},f)\in\widehat{C}} \mathcal{R} = \bigcup_{(\mathcal{R}_1,f_1)\in C'} \mathcal{R}_1 \cup \bigcup_{(\mathcal{R}_2,f_2)\in C''} \mathcal{R}_2 = \mathcal{R}' \cup \mathcal{R}''$.

As far as the pdf $\widehat{f}$, instead, it results that $\widehat{f} = |\widehat{C}|^{-1} \sum_{(\mathcal{R},f)\in\widehat{C}} f = |\widehat{C}|^{-1} \left(\sum_{(\mathcal{R}_1,f_1)\in C'} f_1 + \sum_{(\mathcal{R}_2,f_2)\in C''} f_2 - \sum_{(\mathcal{R}_{12},f_{12})\in C'\cap C''} f_{12} \right) = (|C'|/|\widehat{C}|) f' + (|C''|/|\widehat{C}|) f'' - (|C' \cap C''|/|\widehat{C}|) f_{\cap}$. The proposition follows. □

4. Comparing Uncertain Prototypes

Information-theoretic (IT) measures have been used for comparing pdfs in several application contexts. Comparing two pdfs by means of IT measures is efficient, since their complexity is linear in the number of statistical samples used for representing/approximating the pdfs to be compared (as such measures require just a scan of the two sets of samples), and, in most cases, it may be even linear in the dimensionality m of the pdfs (for the cases where a closed-form of the specific IT measure exists for the pdfs to be compared). Also, the IT-based comparison is generally accurate, since the whole pdf information is involved. However, as mentioned in the Introduction, the comparison makes sense only if the two pdfs share some common event space: if the two pdfs do not have any intersection in

their event spaces, any IT distance (resp. similarity) measure will evaluate equal to the maximum (resp. minimum) value. To better illustrate this concept, look at the two pairs of 2-dimensional pdfs depicted in Fig. 2: although it is clear that the pdfs in Fig. 2-(a) are more similar to each other than the pdfs in Fig. 2-(b), IT measures will not distinguish the two cases as no common region is shared between either pair of pdfs.

To overcome the above issue while retaining the (accuracy and efficiency) benefits of IT measures, we propose a generic distance measure Δ for any two uncertain prototypes P and P' which is expressed as a function φ of two different terms:

$$\Delta(P, P') = \varphi\left(\Delta_{IT}(P, P'), \Delta_{EV}(P, P')\right), \tag{1}$$

where Δ_{IT} involves a specific IT measure, and Δ_{EV} is based on the difference between the expected values of the pdfs of P and P'. The rationale of (1) is to suitably combine an IT measure Δ_{IT} (which is not always applicable) with a concise (but always available) information based on comparing the expected values of pdfs (Δ_{EV}). In the following, we show how the Δ measure can be precisely defined.

4.1. Δ_{IT} measure

The Ali-Silvey class [5] contains two of the most frequently used distance measures for probability distributions, namely *Kullback-Leibler* (KL) and *Chernoff*. From a similarity viewpoint, instead, the *Bhattacharyya coefficient* (ρ) [8, 33] provides a notion of the amount of overlap between any two pdfs f and f':

$$\rho(f, f') = \int_{\vec{x} \in \Re^m} \sqrt{f(\vec{x})\ f'(\vec{x})}\ \mathrm{d}\vec{x}. \tag{2}$$

The Bhattacharyya coefficient puts the basis for the definition of various distance functions. Among these, the following measure, known as *Hellinger distance* [37],

$$\mathcal{H}(f, f') = \sqrt{1 - \rho(f, f')}, \tag{3}$$

has a number of advantages with respect to both other distances based on the Bhattacharyya coefficient, such as the common formulation $-\log \rho$, and other IT measures, including Kullback-Leibler and Chernoff. In fact, unlike all other mentioned measures, $\mathcal{H}$ ranges within $[0, 1]$, which makes it directly combinable with measures that capture other aspects when comparing two pdfs. Also, $\mathcal{H}$ satisfies

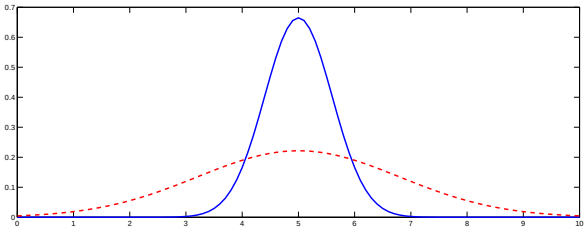


Figure 3: Two pdfs having the same expected value but different shape.

the additive property even though the random variables are not identically distributed (unlike Chernoff), is symmetric (unlike Kullback-Leibler), and obeys the triangle inequality (unlike $-\log\rho$, Kullback-Leibler, and Chernoff).

Due to the above reasons, we choose $\mathcal{H}$ to define Δ_{IT}. Moreover, as we describe later, the Bhattacharyya coefficient ρ (on which $\mathcal{H}$ is based) is proven to be a well-founded criterion for combining Δ_{IT} and Δ_{EV}.

4.2. Δ_{EV} measure

In our formulation, Δ_{EV} should reflect the distance between the expected values of the pdfs of the uncertain prototypes $P = (\mathcal{R}, f)$, $P' = (\mathcal{R}', f')$ to be compared. Therefore, Δ_{EV} should depend on $\delta(\vec{\mu}, \vec{\mu}')$, where $\vec{\mu}$ (resp. $\vec{\mu}'$) is the expected value of the pdf f (resp. f') and $\delta : \Re^m \rightarrow \Re_0^+$ is a function that measures the distance between m-dimensional points (e.g., Euclidean norm $\|\vec{\mu} - \vec{\mu}'\|_2$). At the same time, Δ_{EV} should preferably range within $[0, 1]$, in order to be directly comparable to Δ_{IT} (which ranges within $[0, 1]$ as well).

The above requirements could in principle be fulfilled by defining Δ_{EV} as done in the earlier version of this work [26], i.e., as $\Delta_{EV} = \delta(\vec{\mu}, \vec{\mu}')/E_{max}(\mathcal{D})$, where $E_{max}(\mathcal{D})$ is the maximum over the pairwise distances between the expected values of the input uncertain objects. Unfortunately, this definition of Δ_{EV} suffers from two main drawbacks. The first one is the normalization of Δ_{EV} by $E_{max}(\mathcal{D})$: a large value $E_{max}(\mathcal{D})$ may lead to Δ_{EV} values that are all close to zero, which would make it difficult discriminating among different Δ_{EV} values. The second weak point concerns the exclusive use of the expected values in the definition of Δ_{EV}: two pdfs can be very different while having close expected values, as shown in Fig. 3.

We provide here a more accurate definition of Δ_{EV} that solves both the above issues. The normalization issue is addressed by resorting to an exponential function. The second issue is overcome by taking into account standard deviations

$\vec{\sigma}$ and $\vec{\sigma}'$ of the pdfs of the prototypes P and P' to be compared, along with their expected values $\vec{\mu}$ and $\vec{\mu}'$. We accomplish this by approximating the pdfs f and f' of P and P' as *Uniform* pdfs $\widetilde{f}$ and $\widetilde{f}'$ defined over the regions (hyper-rectangles) $[\vec{\mu}-\vec{\sigma}, \vec{\mu}+\vec{\sigma}]$ and $[\vec{\mu}'-\vec{\sigma}', \vec{\mu}'+\vec{\sigma}']$, respectively, and defining Δ_{EV} as the squared expected distance ED_2 [23] between such approximated Uniform pdfs. Note that, this way, the time complexity of computing Δ_{EV} remain low, as the ED_2 distance between such Uniform pdfs has a closed-form expression that can be efficiently computed in $\mathcal{O}(m)$ as formally stated in the following theorem.

Theorem 1. *Let $\widetilde{f}$ and $\widetilde{f}'$ ($\widetilde{f} \neq \widetilde{f}'$) be two m-dimensional Uniform pdfs defined over the m-dimensional regions (hyper-rectangles) $R = [a_1, b_1] \times \cdots \times [a_m, b_m]$ and $R' = [a'_1, b'_1] \times \cdots \times [a'_m, b'_m]$, respectively. The squared expected distance $ED_2\Big(\widetilde{f}, \widetilde{f}'\Big) = \int_{\vec{x}\in R}\int_{\vec{y}\in R'} \|\vec{x}-\vec{y}\|_2^2 \ \widetilde{f}(\vec{x}) \ \widetilde{f}'(\vec{y}) \ \mathrm{d}\vec{x} \ \mathrm{d}\vec{y}$ between $\widetilde{f}$ and $\widetilde{f}'$ is equal to $ED\Big(\widetilde{f}, \widetilde{f}'\Big) = \frac{1}{6}\sum_{j=1}^{m}\big[2\big(a_j^2+a_jb_j+b_j^2\big)+2\big({a'_j}^2+a'_jb'_j+{b'_j}^2\big)-3\,(b_j+a_j)\,\big(b'_j+a'_j\big)\big]$.*

Proof. Firstly, we note that, as $\widetilde{f}$ and $\widetilde{f}'$ are Uniform pdfs, it holds that $\widetilde{f}(\vec{x}) = \Big(\prod_{j=1}^{m}(b_j-a_j)\Big)^{-1}$, $\forall \vec{x} \in \Re^m$, and $\widetilde{f}'(\vec{y}) = \Big(\prod_{j=1}^{m}(b'_j-a'_j)\Big)^{-1}$, $\forall \vec{y} \in \Re^m$. Thus, denoting with $A_{RR'}$ the product $\prod_{j=1}^{m}(b_j-a_j)\prod_{j=1}^{m}(b'_j-a'_j)$ between the areas of the regions R and R', it results that $\widetilde{f}(\vec{x})\ \widetilde{f}'(\vec{y}) = (A_{RR'})^{-1}$, $\forall \vec{x}, \vec{y} \in \Re^m$. Hence, $ED_2(\widetilde{f}, \widetilde{f}')$ can be expressed as follows:

$$
\begin{aligned}
ED_2\Big(\widetilde{f}, \widetilde{f}'\Big) &= \int\limits_{\vec{x}\in R}\int\limits_{\vec{y}\in R'} \|\vec{x}-\vec{y}\|_2^2 \ \widetilde{f}(\vec{x}) \ \widetilde{f}'(\vec{y}) \ \mathrm{d}\vec{x} \ \mathrm{d}\vec{y} = \\
&= \frac{1}{A_{RR'}} \int\limits_{\vec{x}\in R}\int\limits_{\vec{y}\in R'} \sum_{j=1}^{m}(x_j-y_j)^2 \mathrm{d}\vec{x} \ \mathrm{d}\vec{y} = \\
&= \frac{1}{A_{RR'}} \sum_{j=1}^{m} \int\limits_{\vec{x}\in R}\int\limits_{\vec{y}\in R'} \big(x_j^2-2x_jy_j+y_j^2\big) \mathrm{d}\vec{x} \ \mathrm{d}\vec{y} = \frac{1}{A_{RR'}} \sum_{j=1}^{m} \Big(\mathcal{I}_j^{(1)} - 2\,\mathcal{I}_j^{(2)} + \mathcal{I}_j^{(3)}\Big),
\end{aligned}
$$

where $\mathcal{I}_j^{(1)} = \int\limits_{\vec{x}\in R}\int\limits_{\vec{y}\in R'} x_j^2 \ \mathrm{d}\vec{x} \ \mathrm{d}\vec{y}$, $\mathcal{I}_j^{(2)} = \int\limits_{\vec{x}\in R}\int\limits_{\vec{y}\in R'} x_j \ y_j \ \mathrm{d}\vec{x} \ \mathrm{d}\vec{y}$, and $\mathcal{I}_j^{(3)} = \int\limits_{\vec{x}\in R}\int\limits_{\vec{y}\in R'} y_j^2 \ \mathrm{d}\vec{x} \ \mathrm{d}\vec{y}$.

As far as $\mathcal{I}_j^{(1)}$, it holds that:

$$\begin{aligned}
\mathcal{I}_j^{(1)} &= \int_{x_1}\mathrm{d}x_1\cdots\int_{x_{j-1}}\mathrm{d}x_{j-1}\int_{x_{j+1}}\mathrm{d}x_{j+1}\cdots\int_{x_m}\mathrm{d}x_m\int_{y_1}\mathrm{d}y_1\cdots\int_{y_m}\mathrm{d}y_m\int_{x_j}x_j^2\mathrm{d}x_j = \\
&= \frac{b_j^3-a_j^3}{3}\int_{x_1}\mathrm{d}x_1\cdots\int_{x_{j-1}}\mathrm{d}x_{j-1}\int_{x_{j+1}}\mathrm{d}x_{j+1}\cdots\int_{x_m}\mathrm{d}x_m\int_{y_1}\mathrm{d}y_1\cdots\int_{y_m}\mathrm{d}y_m = \\
&= \frac{(b_j-a_j)(a_j^2+a_jb_j+b_j^2)}{3}\prod_{\substack{k\in[1..m],\\ k\neq j}}(b_k-a_k)\prod_{k\in[1..m]}(b'_k-a'_k) = \\
&= \frac{a_j^2+a_jb_j+b_j^2}{3}\prod_{k\in[1..m]}(b_k-a_k)\prod_{k\in[1..m]}(b'_k-a'_k) \;=\; \frac{a_j^2+a_jb_j+b_j^2}{3}A_{RR'}.
\end{aligned}$$

Analogously, it results that $\mathcal{I}_j^{(3)} = \frac{1}{3}\left({a'_j}^2+a'_jb'_j+{b'_j}^2\right)A_{RR'}$. Concerning $\mathcal{I}_j^{(2)}$, we have:

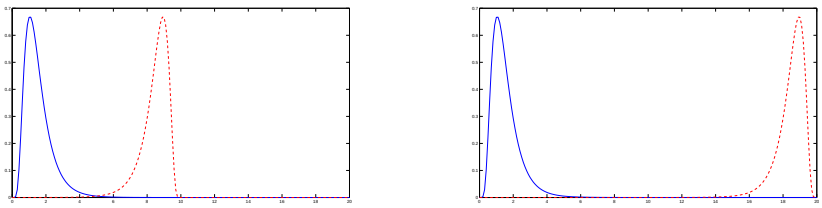
$$\begin{aligned}
\mathcal{I}_j^{(2)} &= \int_{x_1}\mathrm{d}x_1\cdots\int_{x_{j-1}}\mathrm{d}x_{j-1}\int_{x_{j+1}}\mathrm{d}x_{j+1}\cdots\int_{x_m}\mathrm{d}x_m\int_{y_1}\mathrm{d}y_1\cdots\int_{y_{j-1}}\mathrm{d}y_{j-1}\int_{y_{j+1}}\mathrm{d}y_{j+1}\cdots\int_{y_m}\mathrm{d}y_m\int_{x_j}x_j\mathrm{d}x_j\int_{y_j}y_j\mathrm{d}y_j = \\
&= \frac{(b_j^2-a_j^2)({b'_j}^2-{a'_j}^2)}{4}\prod_{\substack{k\in[1..m],\\ k\neq j}}(b_k-a_k)\prod_{\substack{k\in[1..m],\\ k\neq j}}(b'_k-a'_k) = \\
&= \frac{(b_j+a_j)(b'_j+a'_j)}{4}\prod_{k\in[1..m]}(b_k-a_k)\prod_{k\in[1..m]}(b'_k-a'_k) \;=\; \frac{(b_j+a_j)(b'_j+a'_j)}{4}A_{RR'}.
\end{aligned}$$

In conclusion, we can state that $ED_2\left(\widetilde{f},\widetilde{f'}\right) = A_{RR'}^{-1}\sum_{j=1}^m\left(\mathcal{I}_j^{(1)}-2\,\mathcal{I}_j^{(2)}+\mathcal{I}_j^{(3)}\right) = \frac{1}{6}\sum_{j=1}^m\left[2\left(a_j^2+a_jb_j+b_j^2\right)+2\left({a'_j}^2+a'_jb'_j+{b'_j}^2\right)+3\left(b_j+a_j\right)\left(b'_j+a'_j\right)\right]$, which proves the Theorem. □

In summary, the proposed Δ_{EV} distance between prototypes $P=(\mathcal{R},f)$ and $P'=(\mathcal{R}',f')$ is defined as:

$$\Delta_{EV}(P,P') = 1-e^{-ED_2\left(\widetilde{f},\widetilde{f'}\right)}. \tag{4}$$

It is easy to see that $\Delta_{EV}\in[0,1]$, and the lower the distance ED_2 between the Uniform approximations $\widetilde{f}$ and $\widetilde{f'}$, the lower the value of Δ_{EV}, and vice versa, that is $\lim_{ED_2(\widetilde{f},\widetilde{f'})\to 0}\Delta_{EV}(P,P')=0$, and $\lim_{ED_2(\widetilde{f},\widetilde{f'})\to+\infty}\Delta_{EV}(P,P')=1$.



(a) (b)

Figure 4: Uncertain objects sharing a wide common region but dissimilar from each other: two cases

4.3. Combining Δ_{IT} and Δ_{EV}

The combination of Δ_{IT} and Δ_{EV} needs to satisfy the following requirements. Δ_{IT} should prevail on Δ_{EV} as long as discriminating among different cases by means of IT-measures is possible; more precisely, it should hold that, if the IT-based comparison is meaningful, then $\Delta = \Delta_{IT}$. Conversely, if the comparison by means of IT-measures does not guarantee sufficient discrimination (like in the example in Fig. 2), then Δ should consider only Δ_{EV}, i.e., $\Delta = \Delta_{EV}$.

The above requirements are motivated as follows: if the comparison based on IT-measures is meaningful, then Δ_{IT} is sufficient for effectively computing the distance between prototypes, hence there is no need to exploit the additional term Δ_{EV} as this is implicitly taken into account by Δ_{IT}. Conversely, if the comparison in terms of Δ_{IT} is likely to be scarcely reliable, therefore Δ_{EV} should have greater relevance.

In the earlier version of this work [26], we attempted to satisfy the above requirements by resorting to a linear combination of Δ_{IT} and Δ_{EV}, where the importance of the term Δ_{IT} was determined by a factor $\gamma \in [0, 1]$ in a way directly proportional to the amount of overlap between the domain regions of the objects to be compared. This solution may incur some issues, as illustrated in Fig. 4. Let us consider two pairs of 1-dimensional uncertain objects, whose corresponding pdfs are very dissimilar; hence, for both pairs it happens that the value Δ_{IT} is close to its maximum value (i.e., Δ_{IT} close to 1). This is also the value of the overall Δ, as both the pairs of objects have large overlap, thus resulting in γ close to 1. Therefore, the objects in either pair are recognized as very dissimilar, although the objects on the left are clearly closer to each other than the objects on the right.

To overcome the above issue, we combine the terms Δ_{IT} and Δ_{EV} in a different way, that is exploiting the Bhattacharyya coefficient ρ reported in (2). By definition, ρ is directly proportional to how much the pdfs to be compared overlap

in their event space. In other words, ρ is directly proportional to the "suitability" of comparing any two pdfs by means of any IT-measure, and hence fully complies with the previously discussed requirements to satisfy when combining Δ_{IT} and Δ_{EV}. For this purpose, we incorporate ρ into our definition of Δ so that the greater ρ, the larger (resp. the smaller) the contribution given by Δ_{IT} (resp. Δ_{EV}) and vice versa. Indeed, if ρ is small, there is no way for Δ_{IT} to discriminate among the various distances, and hence, in this case, the term Δ_{EV} should prevail (both the pairs of objects in Fig. 4 have ρ close to zero); on the other hand, if ρ is high, then the comparison performed by Δ_{IT} is highly reliable, hence there is no need to exploit the term Δ_{EV} too.

Based on the above intuition, we define Δ as a linear combination of Δ_{IT} and Δ_{EV}. Denoting by $UB(\Delta_{IT})$ an upper bound to the Δ_{IT} term, it is easy to see that a reasonable form for such a combination would be $\Delta = \Delta_{IT} + (1 - UB(\Delta_{IT}))\Delta_{EV}$. Similarly, denoting by $UB(1-\Delta_{IT})$ an upper bound to the similarity counterpart $1-\Delta_{IT}$, the linear combination would become $\Delta = 1 - [(1-\Delta_{IT}) + (1 - UB(1-\Delta_{IT}))(1-\Delta_{EV})]$. Within this view, the objective now is to derive an upper bound $UB(1-\Delta_{IT})$ that complies with the reasoning explained above, that is it should rely on ρ in such a way that the higher ρ, the higher the weight given to the Δ_{IT} term in the overall combination Δ. The expression of such an upper bound is given by the following proposition.

Proposition 2. *For any two pdfs f, f' it holds that* $1 - \Delta_{IT} \leq UB(1-\Delta_{IT}) = \sqrt{\rho\,(f, f')}$.

Proof. $1 - \Delta_{IT} = 1 - \mathcal{H} \leq \sqrt{\rho} \Leftarrow 1 - \sqrt{1-\rho} \leq \sqrt{\rho} \Leftarrow \left(1 - \sqrt{\rho}\right)^2 \leq 1 - \rho \Leftarrow 2\rho - 2\sqrt{\rho} \leq 0 \Leftarrow \rho \leq \sqrt{\rho}$, which holds as $\rho \leq 1$ according to (2). □

Doing the math, we therefore obtain the following expression for Δ:

$$\begin{aligned} \Delta &= 1 - [(1-\Delta_{IT}) + (1 - UB(1-\Delta_{IT}))(1-\Delta_{EV})] = \\ &= 1 - [(1-\Delta_{IT}) + (1 - \sqrt{\rho})(1-\Delta_{EV})] = \mathcal{H} - (1 - \sqrt{\rho})e^{-ED_2}, \end{aligned}$$

which leads to the next formal definition.

Definition 3 (uncertain distance)**.** *The* uncertain distance *between two uncertain prototypes $P = (\mathcal{R}, f)$ and $P' = (\mathcal{R}', f')$ is defined as* $\Delta(P, P') = \mathcal{H}(f, f') - \left(1 - \sqrt{\rho\,(f, f')}\right) e^{-ED_2\left(\widetilde{f}, \widetilde{f'}\right)}$. □

4.4. Remarks on the proposed distance function

We now provide an insight into the behavior of the proposed distance function Δ. First, it can be noted that the requirements about ρ are satisfied, as it can be straightforwardly proved from Def. 3 that $\rho = 1 \Rightarrow \Delta = \Delta_{IT}$, and $\rho = 0 \Rightarrow \mathcal{H} = \Delta_{IT} = 1 \Rightarrow \Delta = \Delta_{EV}$. Moreover, the definition of Δ is well-founded and the overall combination is correctly related to ρ. To demonstrate this, let us consider the behavior of Δ when the two terms Δ_{IT} and Δ_{EV} are close to their extreme values.

Case $\Delta_{IT} = 1$: It holds that $\Delta_{IT} = \mathcal{H} = 1 \Rightarrow \rho = 0 \Rightarrow \Delta = \Delta_{EV}$. As required, if Δ_{IT} is high, the only way to effectively discriminate among the various cases is to use Δ_{EV}.

Case $\Delta_{IT} = 0$: Since $\Delta_{IT} = \mathcal{H} = 0$ if and only if the two pdfs to be compared are the same, and $\mathcal{H} = 0$ implies that $\rho = 1$, it holds that $\Delta_{IT} = 0 \Rightarrow \Delta = 0$. The distance between any two uncertain prototypes is correctly recognized as equal to zero if they are represented by the same pdf.

Case $\Delta_{EV} = 1$: It holds that $\Delta_{EV} = 1 \Rightarrow \Delta = \mathcal{H} = \Delta_{IT}$. The prototypes to be compared can still be similar to a certain degree (in a way inversely proportional to Δ_{IT}) even if the distance measured by Δ_{EV} is maximum.

Case $\Delta_{EV} = 0$: It holds that $\Delta_{EV} = 0 \Rightarrow \Delta = \Delta_{IT} + \left(\sqrt{\rho} - 1\right) = \sqrt{1-\rho} + \left(\sqrt{\rho} - 1\right)$. Hence, in this case, Δ is a function of ρ; in particular, it is correctly equal to 0 when ρ is equal to either its extreme value (i.e., 0 and 1). Indeed, the condition $\rho = 0 \Rightarrow \Delta = 0$ is sound because, if $\rho = 0$, then only the contribution given by Δ_{EV} should be taken into account, and Δ_{EV} is zero in this case; also, the condition $\rho = 1 \Rightarrow \Delta = 0$ is sound too, as $\rho = 1$ implies maximum similarity, and hence maximum suitability of measuring the distance according to IT-measures (and minimum Δ distance). As concerns middle values of ρ, it holds that $\Delta \leq 2\sqrt{0.5} - 1 \approx 0.42$ (particularly, the maximum is reached for $\rho = 0.5$). As desirable, in this case, the two prototypes may be recognized as somehow distant from each other, though $\Delta_{EV} = 0$ would suggest that such prototypes are identical; indeed, we recall that $\Delta_{EV} = 0$ implies only that both the expected values and the standard deviations of the prototypes are equal to each other, but this does not necessarily mean that the two pdfs do not have dissimilar forms.

As a further insight into the proposed distance measure, we remark that our function Δ is a *semimetric*, as it satisfies the axioms of non-negativity, identity of indiscernibles, and symmetry. Instead, it does not generally obey the triangle inequality. We however point out that this is not a weak point of our measure,

as the triangle-inequality property is not a strict requirement in the context of clustering [36]. A classical example in this regard is *information-theoretic clustering* [16], which uses the well-known Kullback-Leibler divergence as a (non-metric) distance measure and whose strengths have been widely attested in several contexts (e.g., document clustering [7]).

4.5. Computing Δ

The most critical operation for computing the proposed uncertain distance measure Δ is the calculation of the Bhattacharyya coefficient ρ reported in (2). Hershey and Olsen [29] show how to compute ρ for mixture models knowing in advance the ρ pairwise values between the components of the mixtures. Furthermore, Nielsen et al. [44] show that the Bhattacharyya coefficient between any two pdfs belonging to the same exponential family has a closed-form expression that can be efficiently computed in $\mathcal{O}(m)$, where m is the number of dimensions (attributes) of the uncertain objects to be compared. Although the exponential families include many of the most common probability distributions (such as Normal, Bernoulli, Beta, Binomial, Chi-square, Dirichlet, Exponential, Gamma, Multinomial, Poisson, and Weibull), it is however desirable to provide a method for computing ρ efficiently even when no closed-form can be exploited. For this purpose, we resort to a commonly used approach in the context of clustering uncertain objects: approximate uncertain objects with sets of statistical samples [38, 10, 23].

Given an input dataset $\mathcal{D}$ of uncertain objects, all samples $\vec{w}$ used for computing approximated representations are common to all uncertain objects within $\mathcal{D}$; such samples form a set $\mathcal{S}$, called *domain sample set*, which is a discrete set of m-dimensional points defined over $\bigcup_o \mathcal{R}$, with $o = (\mathcal{R}, f) \in \mathcal{D}$. Given a domain sample set $\mathcal{S}$, the Bhattacharyya coefficient ρ between any two uncertain objects $o = (\mathcal{R}, f)$ and $o' = (\mathcal{R}', f')$ can be approximated as follows:

$$\widetilde{\rho}(f, f') = \left(\sum_{\vec{w} \in \mathcal{S}} f(\vec{w}) \times \sum_{\vec{w} \in \mathcal{S}} f'(\vec{w}) \right)^{-\frac{1}{2}} \sum_{\vec{w} \in \mathcal{S}} \sqrt{f(\vec{w})\ f'(\vec{w})}. \tag{5}$$

Computing $\widetilde{\rho}$ takes $\mathcal{O}(|\mathcal{S}|\ m)$ time, which is also the overall time complexity of Δ. We recall that, in general, the expected distance ED requires the approximated representations of the objects to be compared, with overall time complexity of $\mathcal{O}(|\mathcal{S}|^2\ m)$. Thus, even if no closed-form expression is used for ρ, the proposed uncertain distance Δ remains more efficient than the standard ED.

5. Clustering Uncertain Objects

5.1. The U-AHC algorithm

We present here the proposed prototype-based AHC algorithm for clustering uncertain objects, called *Uncertain AHC* (U-AHC), whose outline is given in Alg. 1. We focus on the most general case where a domain sample set $\mathcal{S}$ is needed by U-AHC for computing the Bhattacharyya coefficient ρ.

The input of U-AHC is a dataset $\mathcal{D}$ of n uncertain objects and a number S of pdf samples used for computing the domain sample set; the output is a hierarchy $\mathbf{T}$ of clusters (a dendrogram). The algorithm follows the classic AHC scheme. A priority queue ($\mathbf{Q}$) is exploited to efficiently store the inter-cluster distances—the lower the distance between a pair of clusters, the higher the priority in $\mathbf{Q}$.

The initialization steps (Lines 1-6) are in charge of computing the domain sample set $\mathcal{S}$, the approximated representations of each object within $\mathcal{D}$, and the initial set $\mathcal{C}$ of clusters. Particularly, $\mathcal{C}$ contains n pairs, each one composed by a singleton cluster and the associated prototype, which corresponds to the only object belonging to that cluster. The initialization phase ends with the computation of the initial pair-wise distances by means of the *prototype_distance* procedure, which exploits the approximated representations of the prototypes to be compared and the uncertain distance function defined in Def. 3.

The main loop of the algorithm (Lines 7-16) is repeated until the whole hierarchy has been built. At each iteration, the two pairs $\langle C', P' \rangle$, $\langle C'', P'' \rangle$ having the minimum distance are extracted from the priority queue (Line 8) and exploited by the *compute_prototype* procedure for computing the new pair $\langle \widehat{C}, \widehat{P} \rangle$ (Line 9). The procedure *compute_prototype* merges clusters $\mathcal{C}'$ and $\mathcal{C}''$ into a single cluster $\widehat{\mathcal{C}}$, and computes the corresponding prototype $\widehat{P}$ from P' and P'' by applying (1). Afterwards, the priority queue is updated (Lines 10-14).

The computational complexity of U-AHC is stated in the following proposition. Again, we focus on the most general (worst) case which arises when the Bhattacharyya coefficient is computed according to (5), i.e., exploiting no closed-form expressions.

Proposition 3. *Given a dataset $\mathcal{D}$ of n m-dimensional uncertain objects and a domain sample set composed of S samples, the U-AHC algorithm takes $\mathcal{O}(n^2(S\,m + \log n))$ time.*

Proof. The costs of the various steps of U-AHC are summarized next. We assume that the operations of insertion/deletion/extraction of any object into/from the priority queue $\mathbf{Q}$ may be performed in $\mathcal{O}(\log |\mathbf{Q}|)$.

Algorithm 1 U-AHC

Input: a set $\mathcal{D} = \{o_1, \ldots, o_n\}$ of uncertain objects, an integer S denoting the size of the domain sample set over $\mathcal{D}$.
Output: a set of partitions $\mathbf{T}$ (i.e., a dendrogram).

1: $\mathcal{S} \leftarrow \text{domain\_sample\_set}(S)$, $\mathcal{C} \leftarrow \{\langle\{o_1\}, o_1\rangle, \ldots, \langle\{o_n\}, o_n\rangle\}$
2: $\mathbf{T} \leftarrow \{\mathcal{C}\}$, $\mathbf{Q} \leftarrow \emptyset$
3: **for all** $\langle C', P'\rangle, \langle C'', P''\rangle \in \mathcal{C}, C' \neq C''$ **do**
4: $\quad\Delta \leftarrow \text{prototype\_distance}(P', P'')$
5: $\quad\mathbf{Q}.insert(\langle C', P'\rangle, \langle C'', P''\rangle, \Delta)$
6: **end for**
7: **repeat**
8: $\quad(\langle C', P'\rangle, \langle C'', P''\rangle) \leftarrow \mathbf{Q}.removeMin()$
9: $\quad\langle \widehat{C}, \widehat{P}\rangle \leftarrow \text{compute\_prototype}(\langle C', P'\rangle, \langle C'', P''\rangle)$
10: $\quad$**for all** $\langle C, P\rangle \in \mathcal{C}, C \neq C', C \neq C''$ **do**
11: $\quad\quad\mathbf{Q}.remove(\langle C, P\rangle, \langle C', P'\rangle)$, $\mathbf{Q}.remove(\langle C, P\rangle, \langle C'', P''\rangle)$
12: $\quad\quad\Delta \leftarrow \text{prototype\_distance}(P, \widehat{P})$
13: $\quad\quad\mathbf{Q}.insert(\langle C, P\rangle, \langle \widehat{C}, \widehat{P}\rangle, \Delta)$
14: $\quad$**end for**
15: $\quad\mathcal{C} \leftarrow \mathcal{C} \setminus \{\langle C', P'\rangle, \langle C'', P''\rangle\} \cup \{\langle \widehat{C}, \widehat{P}\rangle\}$, $\mathbf{T} \leftarrow \mathbf{T} \cup \{\mathcal{C}\}$
16: **until** $|\mathcal{C}| = 1$

- computing the domain sample set, the approximated representation of each object within $\mathcal{D}$, and the initial set $\mathcal{C}$ of clusters (Line 1) take $\mathcal{O}(S\ n\ m)$, $\mathcal{O}(S\ n\ m)$, and $\mathcal{O}(n)$ time, respectively; also, the initialization of the priority queue (Lines 3-6) is performed in $\mathcal{O}(n^2\ (S\ m + \log n))$ time, as n^2 pairs have to be inserted into $\mathbf{Q}$ and the *prototype_distance* procedure computes the uncertain distance Δ in $\mathcal{O}(S\ m)$;

- the main loop (Lines 7-16) is repeated n-1 times; therefore, each step of this loop has the following *global* time complexity:

  – extracting from $\mathbf{Q}$ the pair having the minimum distance (Line 8) is $\mathcal{O}(n\ \log n)$;

  – computing the new pair $\langle \widehat{C}, \widehat{P}\rangle$ by means of the procedure *compute_prototype* (Line 9) comprises three steps, i.e., *(i)* merging the clusters C', C'', *(ii)* computing the new prototype $\widehat{P}$ from P' and P'' according to (1), and *(iii)* computing the approximated representation of $\widehat{P}$ according to (1). The first two steps take $\mathcal{O}(m \sum_{i=1}^{n-1} \max_{C \in \mathcal{C}^{(r)}} |C|) = \mathcal{O}(m \sum_{i=1}^{n-1} i) = \mathcal{O}(n^2\ m)$, where $\mathcal{C}^{(r)}$ is the set of clusters computed at the r-th iteration. The approxi-

mated representation of $\widehat{P}$ is computed according to (1) (whose cost is $\mathcal{O}(1)$) for each m-dimensional sample within $\mathcal{S}$; thus, it globally takes $\mathcal{O}(S\ n\ m)$;

– in the internal loop (Lines 10-14), inserting/deleting into/from the priority queue (Lines 11 and 13) takes $\mathcal{O}(n^2\ \log n)$ (because inserting/deleting into/from $\mathbf{Q}$ is $\mathcal{O}(\log|\mathbf{Q}|)$ with $|\mathbf{Q}| = \mathcal{O}(n^2)$, and the internal loop is repeated $\mathcal{O}(n\sum_{i=1}^{n-1}(n-i)) = \mathcal{O}(n^2)$ times), whereas the prototype distance (Line 12) takes $\mathcal{O}(S\ n^2\ m)$;

• updating $\mathcal{C}$ and $\mathbf{T}$ (Line 15) can be performed in $\mathcal{O}(n)$.

In conclusion, summing up all above costs, it holds that U-AHC works in $\mathcal{O}(n^2(S\ m + \log n))$ time. □

5.2. Impact of Δ on the U-AHC algorithm

As any uncertain prototype is an uncertain object satisfying Def. 1 (cf. Sect. 3), the proposed function Δ defined in Def. 3 may in principle be used as a distance measure between uncertain objects, and it can be thus involved into any clustering scheme. But, as discussed in Sect. 4, the significance of using Δ to compare uncertain objects mainly depends on the Bhattacharyya coefficient ρ between the two objects; particularly, we are aware that the contribution of the IT term Δ_{IT} to the overall Δ is minimal for low ρ. Nevertheless, we point out that our objective is not to define a general distance measure for uncertain objects, but rather a prototype-based criterion suitable for hierarchical clustering of uncertain objects. And in the context of hierarchical clustering of uncertain objects we are interested in, we theoretically show that the above aspect becomes irrelevant as involving uncertain prototypes defined as mixture models into a prototype-based AHC algorithm and comparing such prototypes by means of our Δ is makes the significance of comparing any two uncertain prototypes monotonically increasing with the iterations of the AHC scheme.

In other words, our main goal here is to show how the proposed distance function finds theoretical justifications when used as a linkage criterion into an AHC scheme, while this is not generally true when other clustering schemes are employed. This makes the proposed distance well-suited in the context of hierarchical clustering of uncertain objects we consider in this work.

We state the main theoretical finding in the next theorem.

Theorem 2. *Consider a generic iteration of the U-AHC algorithm where $\mathcal{C}$ denotes the current set of clusters, $C', C'' \in \mathcal{C}$ the two clusters being merged,*

$\widehat{C} = C' \cup C''$ the new cluster formed, and C any cluster belonging to $\mathcal{C}$ such that $C \neq C'$ and $C \neq C''$. Let $P' = (\mathcal{R}', f')$, $P'' = (\mathcal{R}'', f'')$, $\widehat{P} = (\widehat{\mathcal{R}}, \widehat{f})$, and $P = (\mathcal{R}, f)$ be the prototypes of C', C'', $\widehat{C}$, and C, respectively. Then $\rho(f, \widehat{f}) \geq \left(|C'|/|\widehat{C}|\right) \rho(f, f') + \left(|C''|/|\widehat{C}|\right) \rho(f, f'')$.

Proof. As according to (2) it holds that $\rho(f_1, f_2) = \int_{\Re^m} \sqrt{f_1\, f_2}\, \mathrm{d}\vec{x}$, to prove the theorem we have to demonstrate that:

$$\int_{\Re^m} \sqrt{f\ \widehat{f}}\, \mathrm{d}\vec{x} \ \geq\ \frac{|C'|}{|\widehat{C}|} \int_{\Re^m} \sqrt{f\ f'}\, \mathrm{d}\vec{x} + \frac{|C''|}{|\widehat{C}|} \int_{\Re^m} \sqrt{f\ f''}\, \mathrm{d}\vec{x}. \tag{6}$$

According to Proposition 1, we have that $\widehat{f} = \left(|C'|/|\widehat{C}|\right) f' + \left(|C''|/|\widehat{C}|\right) f'' - \left(|C' \cap C''|/|\widehat{C}|\right) f_{\cap}$. Since the two clusters to be merged are disjoint, (6) becomes $\widehat{f} = \left(|C'|/|\widehat{C}|\right)\ f' + \left(|C''|/|\widehat{C}|\right)\ f''$, which can be rewritten as:

$$\int_{\Re^m} \sqrt{\frac{|C'|}{|\widehat{C}|} f\, f' + \frac{|C''|}{|\widehat{C}|} f\, f''}\, \mathrm{d}\vec{x} \geq \int_{\Re^m} \left(\frac{|C'|}{|\widehat{C}|} \sqrt{f\, f'} + \frac{|C''|}{|\widehat{C}|} \sqrt{f\, f''} \right) \mathrm{d}\vec{x}. \tag{7}$$

Denoting with $g_1(\vec{x})$ (resp. $g_2(\vec{x})$) the function within the integral at the left (resp. right) hand side of (7), it can be noted that to prove (7) it is sufficient to demonstrate that $g_1(\vec{x}) \geq g_2(\vec{x})$, $\forall \vec{x} \in \Re^m$. To this end, let $A = \sqrt{f\, f'}$, $B = \sqrt{f\, f''}$, and $a = |C'|/|\widehat{C}|$, $b = |C''|/|\widehat{C}|$ ($a + b = 1$); it results that:

$$g_1(\vec{x}) = \sqrt{\frac{|C'|}{|\widehat{C}|} f\, f' + \frac{|C''|}{|\widehat{C}|} f\ f''} = \sqrt{a\ A^2 + b\ B^2},$$

$$g_2(\vec{x}) = \frac{|C'|}{|\widehat{C}|} \sqrt{f\ f'} + \frac{|C''|}{|\widehat{C}|} \sqrt{f\, f''} = a\ A + b\ B.$$

Thus, g_1 (resp. g_2) is defined as the weighted quadratic (resp. arithmetic) mean of the terms A and B, where the weights are given by a and b. As the (weighted) quadratic mean is never lower than the (weighted) arithmetic mean, it holds that $g_1(\vec{x}) \geq g_2(\vec{x})$, $\forall \vec{x} \in \Re^m$. The theorem follows. □

Corollary 1. *It holds that*

$$\begin{cases} \rho(f, \widehat{f}) = \rho(f, f') = \rho(f, f''), & \text{if } \rho(f, f') = \rho(f, f'') \\ \rho(f, \widehat{f}) > \dfrac{|C'|}{|\widehat{C}|} \rho(f, f') + \dfrac{|C''|}{|\widehat{C}|} \rho(f, f''), & \text{otherwise.} \end{cases}$$

The above theorem states that, at each iteration of the proposed U-AHC algorithm, the value of $\rho(f, \widehat{f})$ between the prototypes of cluster C and the cluster $\widehat{C}$ formed by merging the closest clusters C' and C'', is never lower than the (weighted) arithmetic mean of $\rho(f, f')$, $\rho(f, f'')$ between C and C', C''. Moreover, Corollary 1 shows that the bound derived from Theorem 2 is strict if $\rho(f, f') \neq \rho(f, f'')$.

Since ρ is considered as a measure of the "suitability" of comparing any two prototypes by means of an IT proximity measure, the above results may be interpreted as follows: the suitability of comparing any cluster C to the new formed one $\widehat{C}$ acts as a monotonic property w.r.t. the (weighted) arithmetic mean of the respective suitabilities of the merging clusters. These results confirm that the overall accuracy of comparing any pair of clusters in the proposed U-AHC is not decreasing (and, in many cases, strictly increasing) at each iteration of U-AHC.

6. Experiments

We evaluated U-AHC in terms of effectiveness and efficiency, and compared it with existing algorithms for clustering uncertain objects: partitional methods, i.e., UK-means (UKM) [10], CK-means (CKM) [41], and UK-medoids (UKmed) [23], density-based methods, i.e., $\mathcal{F}$DBSCAN ($\mathcal{F}$DB) [38], and $\mathcal{F}$OPTICS ($\mathcal{F}$OPT) [39], and sampling-based methods, i.e., Representative Clustering (RepClus) [54] (cf. Sect. 2).[1]

In the evaluation we also involved two *baseline* hierarchical algorithms, called F(ast)-AHC and A(ccurate)-AHC, which correspond to two naïve approaches to clustering uncertain objects that focus on either efficiency (F-AHC) or accuracy (A-AHC). Particularly, F-AHC follows a standard AHC strategy along with a *group-average* cluster merging criterion based on a distance between uncertain objects that is efficiently computed (in $\mathcal{O}(S\ m)$ time) as difference between expected values. The asymptotic time complexity of F-AHC is $\mathcal{O}(n^2(S\ m + \log n))$, and is the same as the proposed U-AHC. A-AHC follows the same AHC scheme as F-AHC, but employs the more accurate yet less efficient expected distance ED, which takes $\mathcal{O}(S^2\ m)$ time and contributes to increase the overall time complexity to $\mathcal{O}(n^2(S^2\ m + \log n))$. As a result, F-AHC is expected to be efficient but not that accurate. The opposite (i.e., high accuracy and poor efficiency) is instead expected for A-AHC. The ultimate goal of this comparison is thus to assess that

[1] We used the implementation of Representative Clustering included in the extended version of the ELKI framework [48].

Table 1: Datasets used in the experiments: benchmark datasets (left) and real datasets (right).

dataset	*# objects*	*# attributes*	*# classes*
Iris	150	4	3
Wine	178	13	3
Glass	214	10	6
Ecoli	327	7	5
Yeast	1,484	8	10
Image	2,310	19	7
Abalone	4,124	7	17
Letter	7,648	16	10
KDDcup	4,000,000	42	23

dataset	*# objects*	*# attributes*
Neuroblastoma	22,282	14
Leukaemia	22,690	21

U-AHC is able of achieving the best tradeoff between accuracy and efficiency, thus demonstrating that U-AHC is (i) more accurate than F-AHC (while remaining comparable to it in terms of efficiency), and (ii) more efficient than A-AHC.

Domain sample sets and approximated representations of the uncertain objects were computed by the Monte Carlo and Markov Chain Monte Carlo sampling methods.[2] To avoid that results were biased by random chance (due to non-deterministic operations, such as computing initial centroids/medoids/partitions), all accuracy and efficiency measurements were averaged over 50 runs. Moreover, we performed a tuning phase for parameters ϵ and μ of $\mathcal{F}$DBSCAN and $\mathcal{F}$OPTICS, and we ultimately set these parameters to the values that allowed each method to achieve the best accuracy results. As far as the clustering methods and the distance between clusterings to be used by Representative Clustering, we follow what suggested in the original paper [54] and use DBSCAN [18] and PAM [35] for producing the base clusterings and the ultimate consensus clustering, respectively, and Adjusted Random Index (ARI) as a distance measure between clusterings.

Quality of clustering solutions was evaluated by means of both external and internal criteria. External criteria exploit the availability of reference classifications in order to evaluate how well a clustering fits a predefined scheme of known classes (natural clusters). We employed the well-known *F-measure* (F), which ranges within $[0, 1]$ such that higher values correspond to better quality results. Denoting with $\widetilde{\mathcal{C}} = \{\widetilde{C}_1, \ldots, \widetilde{C}_h\}$ a reference classification

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

and with $\mathcal{C} = \{C_1, \ldots, C_k\}$ a clustering solution, F-measure is defined as $F(\mathcal{C}, \widetilde{\mathcal{C}}) = |\mathcal{D}|^{-1} \sum_{i=1}^{h} |\widetilde{C}_i| \max_{j \in [1..k]} F_{ij}$, where $F_{ij} = 2\ P_{ij}\ R_{ij} / (P_{ij} + R_{ij})$, $P_{ij} = |C_j \cap \widetilde{C}_i| / |C_j|$, and $R_{ij} = |C_j \cap \widetilde{C}_i| / |\widetilde{C}_i|$, for each $i \in [1..h]$, $j \in [1..k]$.

We also used internal criteria based on *intra-cluster* ($intra(\mathcal{C})$) and *inter-cluster* ($inter(\mathcal{C})$) distances (for a given clustering solution $\mathcal{C}$) which express cluster cohesiveness and cluster separation, respectively. Such distance values were finally combined in a single value $Q(\mathcal{C}) = inter(\mathcal{C}) - intra(\mathcal{C})$, such that the lower $intra(\mathcal{C})$ and the higher $inter(\mathcal{C})$, the better the clustering quality $Q(\mathcal{C})$. Since $intra$ and $inter$ values were normalized within $[0, 1]$, Q ranges within $[-1, 1]$.

Experiments were carried out on benchmark and real datasets, whose main characteristics are summarized in Table 1. Benchmark datasets are selected from [6], whereas real datasets correspond to two microarray datasets available from [9] which are about gene expressions in biological tissues generated by microarray analysis. Note that we synthetically generated uncertainty in benchmark datasets, as they originally contain deterministic values; conversely, this was not necessary for real microarray datasets since they inherently exhibit *probe-level* uncertainty, which can easily be modeled in the form of Normal pdfs according to the *multi-mgMOS* method [42].[3]

Uncertainty generation in benchmark datasets. Based on previous work [10], we developed the following uncertainty generation strategy. Given a (deterministic) benchmark dataset $\mathcal{D}$, we firstly generated a pdf $f_{\vec{w}}$ for each (deterministic) point $\vec{w}$ within $\mathcal{D}$. We considered the *Uniform*, *Normal* and *Exponential* pdfs, as they are commonly encountered in real uncertain data scenarios [2]. Every $f_{\vec{w}}$ was defined in such a way that its expected value corresponds exactly to $\vec{w}$ (i.e., $\vec{\mu}(f_{\vec{w}}) = \vec{w}$), whereas all other parameters (such as the width of the intervals of the Uniform pdfs and the standard deviation of the Normal pdfs) were randomly chosen. We exploited the pdfs $f_{\vec{w}}$ to simulate what actually happens in real contexts of uncertain data (cf. Fig 2). Thus, we focused on two evaluation cases: 1) the clustering task is performed by considering only the observed (i.e., non-uncertain) representations of the various data objects; 2) the clustering task is performed by involving an uncertainty model. The ultimate goal was to assess whether the results obtained in Case 2 are better than those obtained in Case 1.

In Case 1, we generated a *perturbed dataset* $\mathcal{D}'$ from $\mathcal{D}$ by adding to each point $\vec{w} \in \mathcal{D}$ random noise sampled from its assigned pdf $f_{\vec{w}}$. Thus, each point

[3] We used the Bioconductor package PUMA (*Propagating Uncertainty in Microarray Analysis*) available at http://www.bioinf.manchester.ac.uk/resources/puma/.

$\vec{w} \in \mathcal{D}$ gives rise to a *perturbed* point $\vec{w}' \in \mathcal{D}'$. As a result, $\mathcal{D}'$ still contains deterministic data. Then, each of the selected clustering methods was run on $\mathcal{D}'$ so as to output a clustering denoted by $\mathcal{C}'$. A score $F(\mathcal{C}', \widetilde{\mathcal{C}})$ was hence obtained by comparing $\mathcal{C}'$ to the reference classification of $\mathcal{D}$ (denoted by $\widetilde{\mathcal{C}}$) by means of F-measure.

In Case 2, when uncertainty is taken into account, we created an *uncertain dataset* $\mathcal{D}''$ from $\mathcal{D}'$ as follows. For each perturbed point $\vec{w}' \in \mathcal{D}'$, we derived an uncertain object $o = (\mathcal{R}, f)$ so that $f = f_{\vec{w}'}$ (i.e., a pdf whose expected value corresponds to $\vec{w}'$), while $\mathcal{R}$ was defined as the region containing most of the area (e.g., 95%) of $f_{\vec{w}'}$. Again, we run each of the selected methods on $\mathcal{D}''$ so as to get a clustering solution $\mathcal{C}''$ and a score $F(\mathcal{C}'', \widetilde{\mathcal{C}})$.

Finally, we compared the scores obtained in Case 1 and Case 2, respectively, by computing $\Theta(\mathcal{C}', \mathcal{C}'', \widetilde{\mathcal{C}}) = F(\mathcal{C}'', \widetilde{\mathcal{C}}) - F(\mathcal{C}', \widetilde{\mathcal{C}})$, $\Theta \in [-1, 1]$; the higher Θ, the better the quality of $\mathcal{C}''$ w.r.t. $\mathcal{C}'$, and, therefore, the better the performance of the clustering method when the uncertainty is considered w.r.t. the no-uncertainty case.

Results

All accuracy and efficiency results obtained by U-AHC refer to the version of the algorithm that involves the sampling method for computing the Bhattacharyya coefficient ρ described in Sect. 4; as previously discussed, in this way we were able to assess the behavior of our proposed algorithm in the most general case.

Accuracy on benchmark datasets. Tables 2–3 show accuracy results on benchmark datasets for Uniform (U), Normal (N), and Exponential (E) distributions, in terms of external (Θ) and internal (Q) cluster validity criteria, respectively. In both tables, we report for each method: *(i)* the score for each type of pdf averaged over all datasets (for short, *average pdf score*), *(ii)* the score averaged over all datasets and pdfs (for short, *overall average score*), and *(iii)* the overall average gain of our U-AHC computed as the difference between the overall average score of U-AHC and the overall average scores of the other algorithms. Note that the implementation of the RepClus method within the ELKI framework [48] does not provide support for exponential distributions. Thus, we will report RepClus results only for Uniform and Normal distributions.

Let us first focus on comparison with non-hierarchical competitors. Considering Θ results, U-AHC performed better than the other methods over most datasets and distributions (especially Normal and Exponential). In general, looking at the overall average scores, U-AHC outperformed all of non-hierarchical methods, with the following order: $\mathcal{F}$DB, RepClus, $\mathcal{F}$OPT, UKM, CKM, and UKmed.

Table 2: Accuracy results on benchmark datasets (external validity criteria).

		Theta ($\Theta \in [-1,1]$)								
dataset	*pdf*	UKM	CKM	UKmed	$\mathcal{F}$DB	$\mathcal{F}$OPT	RepClus	A-AHC	F-AHC	**U-AHC**
Iris	U	-.062	.028	.023	-.102	.005	.037	.058	-.015	.003
	N	-.010	.013	.010	-.063	.044	.051	.030	.054	.033
	E	-.249	-.380	-.045	-.383	.023	—	.024	-.088	-.147
Wine	U	-.179	.047	.175	-.179	.174	-.029	.035	.083	.179
	N	-.184	.024	-.085	-.185	.030	-.015	.010	.054	.196
	E	-.208	-.127	-.104	-.208	.006	—	.022	-.138	.022
Glass	U	.066	.079	.084	-.298	.012	-.015	.150	.167	.221
	N	-.025	.012	-.070	-.040	-.136	-,.044	.216	.243	.153
	E	-.231	-.302	.009	-.334	-.182	—	.203	.032	.214
Ecoli	U	.199	.332	.223	-.136	.023	-.061	.337	.325	.114
	N	.131	.272	.045	.061	.015	-.064	.270	.213	.227
	E	-.160	-.303	-.034	-.383	-.239	—	.114	-.122	.120
Yeast	U	.220	.279	.315	-.085	.252	-.047	.446	.219	.251
	N	.159	.145	-.035	.079	-.001	-.107	.307	.344	.365
	E	-.098	-.201	-.055	-.311	-.195	—	.144	-.007	.157
Image	U	.278	.274	.241	-.283	-.113	.112	.026	-.099	-.113
	N	.122	.132	-.061	-.251	-.081	.127	-.081	-.048	.174
	E	-.024	-.204	.087	-.307	-.137	—	.002	-.119	.064
Abal.	U	.120	.092	.379	-.092	.291	-.038	.454	.204	.324
	N	.034	-.031	.009	.095	-.039	-.101	.213	.181	.289
	E	.080	-.084	.025	-.182	.315	—	.415	.130	.390
Letter	U	.008	.113	.237	-.338	-.201	-.092	.189	.206	.386
	N	-.076	-.082	-.039	-.340	-.203	-.107	-.024	-.071	.037
	E	-.202	-.399	.033	-.431	-.294	—	.048	-.182	.045
KDDcup	U	.008	.009	.031	-.117	.127	.011	-.074	-.195	-.196
	N	.077	.047	.029	-.013	.021	.001	-.002	-.099	.265
	E	-.133	-.095	-.110	-.171	.059	—	.092	-.208	.195
avg score	U	.073	.139	.190	-.181	.063	-.014	.180	.099	.130
	N	.025	.059	-.022	-.073	-.039	-.029	.104	.097	.193
	E	-.136	-.233	-.022	-.301	-.072	—	.119	-.078	.118
overall avg. score		-.013	-.011	.049	-.185	-.016	-.021	.114	.039	.147
overall avg. gain		+.160	+.158	+.098	+.332	+.163	+.168	+.033	+.108	—

More in detail, among the competitors, $\mathcal{F}$DB had the worst performances on all types of distributions, while UKmed (resp. CKM) was the most accurate method using Uniform and Exponential (resp. Normal) distributions. Also in terms of criterion Q, U-AHC achieved higher results than the competing methods, on average. For this evaluation, the least yet significant gain by U-AHC was against RepClus, while $\mathcal{F}$OPT behaved slightly worse than U-AHC on average (however being significantly less accurate than U-AHC on Normal and Exponential distributions), and the other density-based method confirmed to be the worst performing method in general — this might be explained due to the difficulty in setting parameters ϵ and μ.

Concerning the two hierarchical competitors, U-AHC was in general much more accurate than its fast naïve counterpart (F-AHC), thus confirming one of the

Table 3: Accuracy results on benchmark datasets (internal validity criteria). Notation nz stands for values with precision over three decimal digits (i.e., values within (-5.0E-4, +5.0E-4))

		Quality ($Q \in [-1, 1]$)								
dataset	*pdf*	UKM	CKM	UKmed	$\mathcal{F}$DB	$\mathcal{F}$OPT	RepClus	A-AHC	F-AHC	**U-AHC**
Iris	U	.151	.145	.148	.197	.093	.331	.146	.144	.152
	N	.263	.194	.194	.238	.135	.244	.298	.203	.324
	E	.118	-.001	.081	-.004	.202	—	.274	.029	.050
Wine	U	-.001	-.002	.012	-.002	.128	.007	.608	-.002	.185
	N	-.020	.012	.042	.022	.009	.009	.282	.009	.031
	E	nz	nz	.001	nz	nz	—	.337	nz	nz
Glass	U	.001	.001	.060	-.013	.001	.079	.510	.006	.004
	N	.057	.062	.041	.042	.006	.105	.202	.164	.201
	E	.004	.001	.006	-.002	nz	—	.192	.024	.026
Ecoli	U	.101	.031	.187	nz	.449	.016	.642	.144	.089
	N	.141	.060	.029	.086	.284	.027	.344	.084	.141
	E	.001	nz	.003	nz	nz	—	.303	nz	-.001
Yeast	U	.041	.016	.193	nz	.029	.004	.669	.068	.063
	N	.053	.031	.005	.040	.222	.003	.185	.129	.170
	E	nz	nz	nz	nz	nz	—	.120	nz	nz
Image	U	nz	nz	nz	nz	nz	.081	.133	nz	nz
	N	.065	.074	.010	-.001	.004	.011	.341	.327	.240
	E	nz	nz	nz	nz	nz	—	.102	nz	nz
Abal.	U	.040	.025	.071	-.018	.010	.024	.273	.050	.060
	N	.103	.055	.031	.086	.054	.027	.124	.119	.043
	E	nz	nz	nz	nz	nz	—	.116	nz	nz
Letter	U	nz	nz	nz	nz	nz	.062	.210	nz	.003
	N	.352	.303	.357	-.022	.207	.107	.233	nz	.004
	E	nz	nz	nz	nz	nz	—	.210	nz	.003
KDDcup	U	.069	.066	.040	.021	.133	.064	.242	.134	.197
	N	.006	.092	.023	.061	.115	.006	.199	.086	.144
	E	.012	.088	.111	-.001	.025	—	.172	.023	.166
avg score	U	.047	.032	.084	.021	.133	.074	.364	.066	.089
	N	.113	.098	.081	.061	.115	.060	.221	.126	.145
	E	.015	.010	.022	-.001	.025	—	.195	.023	.042
overall avg. score		.058	.047	.063	.027	.091	.067	.260	.072	.092
overall avg. gain		+.034	+.046	+.030	+.065	+.001	+.025	-.168	+.020	—

major claims of this work. Indeed, U-AHC achieved higher Θ and Q results than F-AHC in most cases; more specifically, it outperformed F-AHC on 21 (resp. 16) out of 27 dataset-by-pdf configurations, with maximum gain of 0.403 (resp. 0.187) on KDDcup-Exponential (resp. Wine-Uniform) in terms of Θ (resp. Q). Compared to A-AHC, on average U-AHC behaved better in terms of Θ, especially on Normal and Exponential distributions, with overall average gain of 0.033, while a relatively large gap is observed for Q results. This is actually not surprising since the expected distance ED employed in A-AHC is the same measure as that used for defining the cluster validity criterion Q, while this does not hold for our U-AHC; thus, the assessment in terms of Q is inherently biased in favor of A-AHC. Overall, the higher performance by A-AHC might be explained considering that it employs the same hierarchical scheme as U-AHC, however equipped with the

Table 4: Accuracy results on real datasets.

		Quality ($Q \in [-1, 1]$)								
dataset	*# clusters*	UKM	CKM	UKmed	$\mathcal{F}$DB	$\mathcal{F}$OPT	RepClus	A-AHC	F-AHC	**U-AHC**
Neuroblastoma	2	.057	.059	.044	-.004	.010	.071	.452	.143	.917
	3	.061	.058	.047	-.004	.017	.071	.880	.187	.670
	5	.060	.062	.043	-.004	.009	.067	.803	.141	.847
	10	.068	.066	.048	-.004	.008	.075	.830	.093	.607
	15	.060	.062	.044	-.004	.010	.077	.667	.066	.578
	20	.061	.060	.047	-.004	.009	.077	.594	.061	.487
	25	.065	.057	.041	-.004	.009	.071	.524	.056	.465
	30	.047	.053	.043	-.004	.008	.072	.458	.049	.466
Leukaemia	2	.207	.266	.221	-.018	.068	.061	.698	.219	.445
	3	.392	.316	.256	-.018	.080	.068	.657	.238	.258
	5	.451	.372	.245	-.018	.061	.074	.829	.153	.160
	10	.455	.368	.238	-.018	.213	.081	.899	.135	.150
	15	.451	.320	.246	-.018	.192	.070	.737	.111	.145
	20	.479	.322	.213	-.018	.186	.071	.764	.091	.126
	25	.558	.296	.215	-.018	.353	.075	.707	.088	.127
	30	.448	.296	.213	-.018	.369	.059	.678	.082	.122
Neuroblastoma	*avg. score*	.060	.060	.045	-.004	.010	.072	.651	.100	.630
Leukaemia	*avg. score*	.430	.320	.231	-.018	.190	.070	.746	.140	.192
overall	*avg. score*	.245	.190	.138	-.011	.100	.071	.699	.120	.411
overall	*avg. gain*	+.166	+.221	+.273	+.422	+.311	+.340	-.288	+.291	—

more accurate expected distance between uncertain objects; on the other hand, as discussed later in this section, A-AHC is much less efficient than U-AHC.

Accuracy on real datasets. Table 4 shows accuracy results obtained on Neuroblastoma and Leukaemia, and also summarizes *(i)* the scores on each dataset by averaging over the cluster numbers, and *(ii)* the scores and gains by averaging over all cluster numbers and datasets (for short, *overall average score*). Due to the unavailability of reference classifications for such datasets, we varied the number of clusters and assessed the results based on Q only. Specifically, we varied the number of clusters from 2 to 30, since $\mathcal{F}$DB (which is able to automatically discover the number of clusters) detected a number of clusters around 15 for both datasets.

Compared to the non-hierarchical competing methods, looking at the average scores, U-AHC outperformed all of them on Neuroblastoma, with average gains above 0.620, whereas on Leukaemia U-AHC had varying competitive behavior. In general, U-AHC achieved the best overall average performance, with maximum, average and minimum gains of 0.422 (w.r.t. $\mathcal{F}$DB), 0.217, and 0.166 (w.r.t. UKM), respectively. Like for the benchmark datasets, our U-AHC was inferior to A-AHC and superior to F-AHC; more specifically, U-AHC was more accurate than F-AHC on all 16 dataset-by-number-of-clusters configurations, with average gains of 0.530 and 0.052, on Neuroblastoma and Leukaemia, respectively.

Table 5: Efficiency results (seconds).

	dataset (benchmark)								*dataset (real)*	
algorithm	Iris	Wine	Glass	Ecoli	Yeast	Image	Abalone	Letter	Neuroblast.	Leukaemia
U-AHC	0.43	0.58	0.83	1.95	46	118	416	1,459	7,054	8,284
F-AHC	0.08	0.09	0.12	0.29	12	33	133	520	4,568	5,479
A-AHC	68.09	137.07	175.31	355.29	8,030	30,773	60,281	315,559	>1.0E+6	>1.0E+6
U-AHC/F-AHC	*5.4*	*6.1*	*6.9*	*6.6*	*4.0*	*3.6*	*3.1*	*2.8*	*1.5*	*1.5*
A-AHC/U-AHC	*157.1*	*237.6*	*212.2*	*182.1*	*173.1*	*260.7*	*145.0*	*216.3*	—	—

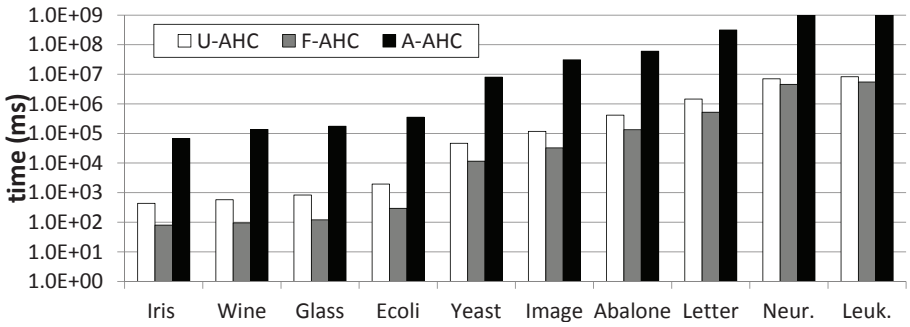


Figure 5: Efficiency results (ms).

Efficiency. It is widely known that the knowledge learned by hierarchical clustering algorithms comes with the cost of a time complexity generally higher than partitional or density-based schemes. For this reason, our efficiency evaluation was devised to focus on a comparison of the running times of our U-AHC algorithm with those of its naïve hierarchical counterparts only, i.e., F-AHC and A-AHC.[4] The main goal of this evaluation was to prove a major claim of this work: the proposed U-AHC outperforms A-AHC while performing closely to F-AHC.

The runtimes of all algorithms are summarized in Table 5 and displayed in Figure 5; in the table, details are also reported on the ratio of the U-AHC runtime to the F-AHC runtime (second last row) and the ratio of the A-AHC runtime to the U-AHC runtime (last row). Times refer to a number of samples $S = 500$ and to the Normal pdf, as we observed that the relative performances of the algorithms were never significantly affected by the form of distribution.

Looking at Table 5 and Figure 5, results confirm our time complexity analysis, as U-AHC and F-AHC were always much faster than A-AHC while U-AHC and F-AHC performed similarly to each other. Focusing on the ratios between the U-

[4]Experiments were carried out exploiting computing resources of CRESCO/ENEAGRID High Performance Computing infrastructure [19].

AHC runtime and F-AHC/A-AHC runtime, we found that A-AHC runtime was always two orders of magnitude slower than U-AHC, while the U-AHC runtime was always of the same order as F-AHC; Interestingly, the ratio U-AHC/F-AHC decreases for larger datasets, which is explained as, increasing n, the term n^2 becomes dominant over Sm, thus making the complexity of the main loops of the two algorithms ($\mathcal{O}(n^2(S\ m + \log n))$ for U-AHC, $\mathcal{O}(n^2 \log n)$ for F-AHC) comparable.

As concerns evaluation on KDDcup, efficiency analysis on this dataset represents a challenge because of its very large size that makes any hierarchical clustering process computationally expensive in practice. For this dataset we therefore devised a different stage of evaluation, which was based on an implementation of modified versions of our methods based on a parallel computing architecture.[5] We remark that all the methods involved in our comparison share the same underlying hierarchical scheme, and thus the parallel implementation was the same for all methods as well. This ensured a fair comparison. Results were in line with the ones observed for the other datasets: U-AHC was 2.7 times slower than F-AHC and 186 times faster than A-AHC.

7. Conclusion

We have provided a principled solution to the problem of hierarchical clustering of uncertain data. Starting from a revision of the method described in our earlier work [26], the key idea of this new approach lies in a well-founded linkage criterion (for the cluster merging step of the hierarchical algorithm) which takes into account information-theoretic properties of the probability distributions associated to the uncertain objects to be clustered. This prompted us to study the conditions that determine the suitability of using information-theoretic and expected distance measures in a combined way, in order to integrate their respective strengths. Our method has been experimentally shown to outperform major competing methods in terms of average accuracy on all datasets used in the evaluation. Also, from an efficiency viewpoint, our method outperforms the baseline group-average AHC algorithm equipped with the accurate expected distance, while being comparable to the fast baseline version of group-average AHC that computes the pair-wise distances of the uncertain objects as the difference between expected values.

[5] We used the 4864-core ENEAGRID CRESCO4 cluster [19] for this task.

References

[1] S. Abiteboul, P. Kanellakis, and G. Grahne. On the representation and querying of sets of possible worlds. In *Proc. ACM Int. Conf. on Management of Data (SIGMOD)*, pages 34–48, 1987.

[2] C. C. Aggarwal. *Managing and Mining Uncertain Data*. Springer, 2009.

[3] C. C. Aggarwal. On high dimensional projected clustering of uncertain data streams. In *Proc. IEEE Int. Conf. on Data Engineering (ICDE)*, pages 1152–1154, 2009.

[4] P. Agrawal, A. D. Sarma, J. D. Ullman, and J. Widom. Foundations of uncertain-data integration. *Proceedings of the VLDB Endowment (PVLDB)*, 3(1):1080–1090, 2010.

[5] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society*, 28(1):131–142, 1966.

[6] A. Asuncion and D.J. Newman. Uci machine learning repository, http://archive.ics.uci.edu/ml/.

[7] L. Douglas Baker and Andrew McCallum. Distributional clustering of words for text classification. In *Proc. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 96–103, 1998.

[8] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–110, 1943.

[9] Broad Institute of MIT and Harvard. Cancer program dataset page, http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

[10] M. Chau, R. Cheng, B. Kao, and J. Ng. Uncertain data mining: An example in clustering location data. In *Proc. PAKDD Conf.*, pages 199–204, 2006.

[11] X. Chen, S. Kar, and D. A. Ralescu. Cross-entropy measure of uncertain variables. *Information Sciences*, 201:53–60, 2012.

[12] G. Cormode and A. McGregor. Approximation algorithms for clustering uncertain data. In *Proc. ACM PODS Conf.*, pages 191–200, 2008.

[13] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *Proc. Int. Conf. on Very Large Data Bases (VLDB)*, pages 864–875, 2004.

[14] T. Denoeux. Maximum likelihood estimation from uncertain data in the belief function framework. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 25(1):119–130, 2013.

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

[16] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research (JMLR)*, 3:1265–1287, 2003.

[17] D. Dubois and H. Prade. Rough fuzzy sets and fuzzy rough sets. *Int. J. Gen. Syst.*, 17(2–3):191–209, 1990.

[18] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

[19] Giovanni Ponti et al. The role of medium size facilities in the HPC ecosystem: the case of the new CRESCO4 cluster integrated in the ENEAGRID infrastructure. In *Proc. Int. Conf. on High Performance Computing & Simulation (HPCS)*, pages 1030–1033, 2014.

[20] A. Gacek. Granular modelling of signales: a framework of granular computing. *Information Sciences*, 221:1–11, 2012.

[21] T. Green and V. Tannen. Models for incomplete and probabilistic information. *IEEE Data Engineering Bulletin*, 29(1):17–24, 2006.

[22] S. Guha and K. Munagala. Exceeding expectations and clustering uncertain data. In *Proc. ACM PODS conf.*, pages 269–278, 2009.

[23] F. Gullo, G. Ponti, and A. Tagarelli. Clustering uncertain data via k-medoids. In *Proc. Int. Conf. on Scalable Uncertainty Management (SUM)*, pages 229–242, 2008.

[24] F. Gullo, G. Ponti, and A. Tagarelli. Minimizing the variance of cluster mixture models for clustering uncertain objects. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 839–844, 2010.

[25] F. Gullo, G. Ponti, and A. Tagarelli. Minimizing the variance of cluster mixture models for clustering uncertain objects. *Statistical Analysis and Data Mining*, 6(2):116–135, 2013.

[26] F. Gullo, G. Ponti, A. Tagarelli, and S. Greco. A hierarchical algorithm for clustering uncertain data via an information-theoretic approach. In *Proc. IEEE ICDM Conf.*, pages 821–826, 2008.

[27] F. Gullo and A. Tagarelli. Uncertain centroid based partitional clustering of uncertain data. *Proceedings of the VLDB Endowment (PVLDB)*, 5(7):610–621, 2012.

[28] S. Günnemann, H. Kremer, and T. Seidl. Subspace clustering for uncertain data. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 385–396, 2010.

[29] J. R. Hershey and P. A. Olsen. Variational bhattacharyya divergence for hidden markov models. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.

[30] E. Hung, L. Xiao, and R. Y. S. Hung. An efficient representation model of distance distribution between uncertain objects. *Computational Intelligence*, 28(3):373–397, 2012.

[31] T. Imielinski and W. Lipski Jr. Incomplete information in relational databases. *Journal of the ACM*, 31(4):761–791, 1984.

[32] Bin Jiang, Jian Pei, Yufei Tao, and Xuemin Lin. Clustering uncertain data based on probability distribution similarity. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 25(4):751–763, 2013.

[33] T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Trans. on Communication Technology*, 15(1):52–60, 1967.

[34] B. Kao, S. D. Lee, F. K. F. Lee, D. W. L. Cheung, and W. S. Ho. Clustering uncertain data using voronoi diagrams and r-tree index. *TKDE*, 22(9):1219–1233, 2010.

[35] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990.

[36] Jon M. Kleinberg. An impossibility theorem for clustering. In *Proc. Neural Information Processing Systems Conf. (NIPS)*, pages 446–453, 2002.

[37] S. Kotz and N. Johnson. *Encyclopedia of Statistical Sciences*. Wiley, 1981.

[38] H. P. Kriegel and M. Pfeifle. Density-based clustering of uncertain data. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 672–677, 2005.

[39] H. P. Kriegel and M. Pfeifle. Hierarchical density-based clustering of uncertain data. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 689–692, 2005.

[40] L. V. S. Lakshmanan, N. Leone, R. B. Ross, and V. S. Subrahmanian. Probview: A flexible probabilistic database system. *ACM Transactions on Database Systems (TODS)*, 22(3):419–469, 1997.

[41] S. D. Lee, B. Kao, and R. Cheng. Reducing uk-means to k-means. In *Proc. IEEE ICDM Workshops*, pages 483–488, 2007.

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

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

[44] F. Nielsen, S. Boltz, and O. Schwander. Bhattacharyya clustering with applications to mixture simplifications. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*, pages 1437–1440, 2010.

[45] W. Pedrycz. *Granular computing: analys and design of intelligent systems*. CRC Press, Francis Taylor, 2013.

[46] Y. H. Qian, J. Y. Liang, Y. Y. Yao, and C. Y. Dang. MGRS: a multi-granulation rough set. *Information Sciences*, 180(6):949–970, 2010.

[47] A. D. Sarma, O. Benjelloun, A. Halevy, and J. Widom. Working models for uncertain data. In *Proc. IEEE Int. Conf. on Data Engineering (ICDE)*, pages 7–18, 2006.

[48] E. Schubert, A. Koos, T. Emrich, A. Züfle, K. A. Schmid, and A. Zimek. A framework for clustering uncertain data. *Proc. PVLDB Endow. (PVLDB)*, 8(12):1976–1979, 2015.

[49] Y. Tao, X. Xiao, and R. Cheng. Range search on multidimensional uncertain data. *ACM Transactions on Database Systems (TODS)*, 32(3):15–62, 2007.

[50] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Trans. on Database Systems (TODS)*, 29:463–507, 2004.

[51] P. B. Volk, F. Rosenthal, M. Hahmann, D. Habich, and W. Lehner. Clustering uncertain data with possible worlds. In *Proc. IEEE Int. Conf. on Data Engineering (ICDE)*, pages 1625–1632, 2009.

[52] B. Yang and Y. Zhang. Kernel based k-medoids for clustering data with uncertainty. In *Proc. Int. Conf. on Advanced Data Mining and Applications (ADMA)*, pages 246–253, 2010.

[53] S. Zhao, L. Zhang, X. Xu, and Y. Zhang. Hierarchical description of uncertain information. *Information Sciences*, 268:133–146, 2014.

[54] A. Züfle, T. Emrich, K. A. Schmid, N. Mamoulis, A. Zimek, and M. Renz. Representative clustering of uncertain data. In *Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 243–252, 2014.