Minimizing the Variance of Cluster Mixture Models for Clustering Uncertain Objects

Francesco Gullo
Yahoo! Research
Barcelona, Spain
Email: gullo@yahoo-inc.com

Giovanni Ponti
ENEA - Portici Research Center
Portici (NA), Italy
Email: giovanni.ponti@enea.it

Andrea Tagarelli
DEIS Department
University of Calabria
Rende (CS), Italy
Email: tagarelli@deis.unical.it

***Abstract*—In recent years, there has been a growing interest in clustering *uncertain objects*. In contrast to traditional, "sharp" data representation models, uncertain objects are modeled as probability distributions defined over uncertainty regions. In this context, a major issue is related to the poor efficiency of existing algorithms, which is mainly due to expensive computation of the distance between uncertain objects.**

In this work, we extend our earlier work [16] in which a novel formulation to the problem of clustering uncertain objects is defined based on the minimization of the variance of the mixture models that represent the clusters being discovered. Analytical properties about the computation of variance for cluster mixture models are derived and exploited by a partitional clustering algorithm, called *MMVar*. This algorithm achieves high efficiency since it does not need to employ any distance measure between uncertain objects when computing local minima of the objective function at the basis of the proposed formulation. Experiments have shown that MMVar is scalable and outperforms state-of-the-art algorithms in terms of efficiency, while achieving better average performance in terms of accuracy.

I. Introduction

Uncertainty in data naturally arises from a variety of real-world phenomena, such as implicit randomness in a process of data generation/acquisition, imprecision in physical measurements, and data staling [1]. For instance, sensor measurements may be imprecise at a certain degree due to the presence of various noisy factors (e.g., signal noise, instrumental errors, wireless transmission) [9], [11]. Another example is given by data representing moving objects, which continuously change their location so that the exact positional information at a given time instant may be unavailable [33]. In data integration, uncertainty can arise for several reasons, such as the approximation assumptions on the semantic mappings between the data sources and the mediated schema, poor knowledge about what the exact mappings are, or the transformation between keyword queries and a set of candidate structured queries [4], [10]. The biomedical research domain abounds of examples of data inherently affected by uncertainty. As an example, in the context of gene expression microarray data [31], handling the so-called probe-level uncertainty represents a key aspect that enables more expressive data representation and more accurate processing [25], [26]. Further examples of uncertain data come from distributed applications, privacy preserving data mining, and forecasting or other statistical techniques used to generate data attributes [3].

In general, uncertainty can be considered at *table*, *tuple* or *attribute* level [32], and is formally specified by *fuzzy models* [12], *evidence-oriented models* [24], or *probabilistic models* [30]. This work focuses on data containing attribute-level uncertainty modeled according to probabilistic models. In particular, we are interested in probabilistic representations that make use of *probability distributions* aimed at describing the likelihood that any given object appears at each position in a multidimensional domain region [7], [15], [19], [22], [23]. Probabilistic models may in principle involve alternative ways of representing uncertainty, such as exploiting some statistical properties (e.g., error percentage or deviation from an expected value); however, although these properties provide a concise information about an uncertain set of values, probability distributions offer a more accurate representation. We hereinafter refer to data objects described in terms of probability distributions defined over multidimensional domain regions as *uncertain objects* (cf. Fig. 2).

Dealing with uncertain objects has raised several issues in data mining and knowledge discovery. In order to produce meaningful results, algorithms for mining uncertain objects should carefully take into account uncertainty. Consider for instance a scenario of similarity detection in the simple uncertain dataset depicted in Fig. 1, where, for simplicity, uncertain objects are represented in terms of their domain regions only, as the reasoning holds regardless of any specific probability distribution. 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., $\{o'_1, o'_2\}$, $\{o''_1, o'''_1\}$, $\{o''_2, o'''_2\}$ in Fig. 1(b) that are substantially different from the ideal ones which would be identified by considering the true representations (i.e.,

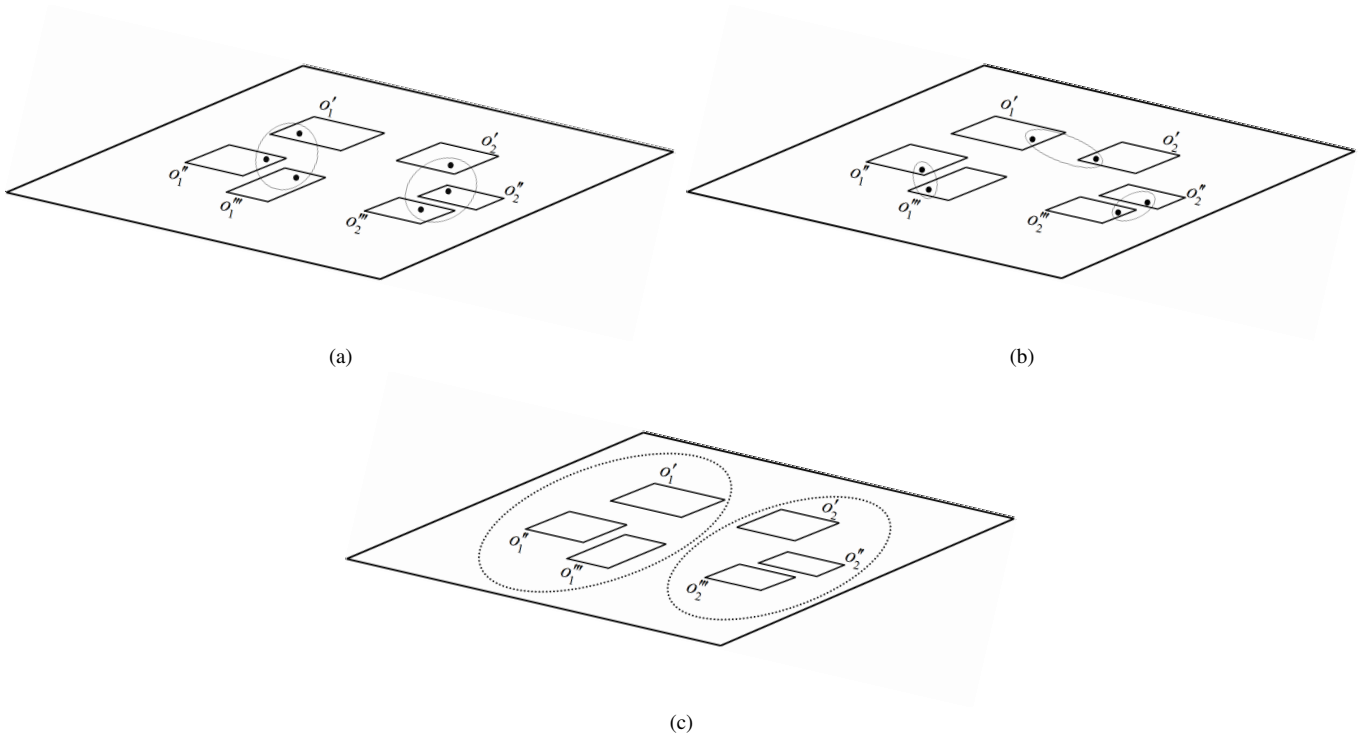

Figure 1. Similarity detection in an uncertain dataset: (a) true representations of objects and their desired grouping, (b) observed representations which may lead to unexpected groupings, (c) desired grouping identified by considering the object uncertainty (domain regions).

$\{o'_1, o''_1, o'''_1\}$, $\{o'_2, o''_2, o'''_2\}$ in Fig. 1(a). Instead, taking into account in a proper way uncertainty, i.e., considering the whole domain regions (and pdfs) of the uncertain objects, may help to recognize the correct grouping (Fig. 1(c)).

Among data mining tasks, *clustering* uncertain objects has been attracting increasing interest in recent years [7], [15], [17], [19], [22], [23], [27]. There are two main issues that make clustering uncertain objects a very challenging problem. First, existing algorithms require some notion of distance between uncertain objects, whose definition is usually a non-trivial task. Indeed, existing approaches fall into two main categories, which have both some weaknesses in their own. The first approach consists in computing the distance between aggregated values extracted from the probability distributions of the uncertain objects (e.g., expected values), and has a complexity linear w.r.t. the number S of statistical samples used for representing distributions. The second approach involves the computation of the so-called *expected distance* (ED) between distributions [15], which aims to exploit the whole information available from the distributions and works in $\mathcal{O}(S^2)$. Though pretty efficient, the first approach (i.e., distance between aggregated values) has clearly an accuracy issue, since all the information available from the distributions is collapsed into a single, representative numerical value. Conversely, the ED-based approach is more accurate but also inefficient.

A further, more critical issue concerns the efficiency of existing algorithms. This is partially related to the need for a distance measure between uncertain objects discussed above, since the use of a slow measure clearly leads to poor efficiency. However, more generally, it intrinsically depends on the specific formulations at the basis of existing algorithms, which constrain such heuristics to continuously execute critical operations, such as access to the samples of distributions (needed, e.g., for integral approximations) and/or the computation of distances between uncertain objects.

In this paper, we extend a novel formulation to the problem of clustering uncertain objects, which was previously introduced in our earlier work [16]. In the attempt of overcoming the above discussed issues, the key idea is to take into account the *mixture model* of the set of random variables representing the uncertain objects within a cluster in order to define a clustering objective criterion based on the minimization of the *variance* of the cluster mixture models. Such a criterion is designed to fulfil efficiency and accuracy requirements. High-quality clusters can in fact be discovered since the compactness of a set (cluster) of uncertain objects increases as the variance of the cluster mixture models decreases. At the same time, the proposed criterion enables the definition of a fast heuristic that does not require any distance measure between uncertain objects.

Within this view, the main contributions of this work can be summarized as follows. We revise a recently pre-

sented formulation to the problem of clustering uncertain objects [16], which essentially relies on the minimization of the variance of the mixture models that represent the clusters of uncertain objects to be discovered. In particular, we provide further theoretical insights by analytically investigating relations between the computation of mixture models and their variances, and consequently deriving some properties of the objective function at the basis of the proposed formulation. In this regard, we come up with the *MMVar* algorithm, whose definition improves upon the algorithm firstly introduced in [16]. The *MMVar* algorithm proposed here shares with the early algorithm in [16] the nice capability of being at the same time effective, as it is able to discover local minima of the proposed objective function, as well as very efficient, as it does not need any distance measure between uncertain objects. Moreover, the proposed MMVar aims to mitigate one of the major weaknesses of the early algorithm, i.e., the initialization phase. In [16], in fact, the initialization was simply based on either a random procedure or an external clustering algorithm operating on the "deterministic" versions of the uncertain objects. Here, we define an incremental clustering step to initialize a k-way clustering solution, which can lead to initializations that better reflect the semantics of the proposed objective function and, therefore, may significantly help the proposed algorithm to reach local minima closer to global minima. We have conducted an extensive experimental evaluation on both benchmark and real datasets to assess our MMVar in terms of accuracy, efficiency and scalability. Compared to prominent state-of-the-art algorithms, MMVar revealed to be always faster and on average more accurate, particularly achieving the best maximum accuracy in more than half cases we have considered and being generally better than its early counterpart in [16]. Moreover, MMVar has shown to scale linearly with the dataset size, also offering better scalability than the fastest competing method.

The rest of the paper is organized as follows. Section II briefly describes the state-of-the-art in clustering uncertain objects. Section III provides the notion of uncertain object used throughout the paper. Section IV discusses our proposal in detail. Experimental settings and clustering results are presented in Section V, and Section VI concludes the paper. In Appendix, proofs of main theoretical results are finally reported.

II. Related Work

We briefly review the main state-of-the-art algorithms for clustering uncertain objects. Table I summarizes the computational complexities of prominent methods, according to the following notation: n is the size of the input set of uncertain objects, m is the dimensionality of the uncertain objects (i.e., number of features), k is the desired number of clusters, I is the number of iterations to convergence required by partitional clustering algorithms, and S denotes the number of statistically independent samples employed for representing probability distributions.

One of the earliest attempts to solve the problem of clustering uncertain objects is the partitional algorithm *UK-means* [7], which is an adaptation of the popular K-means to the context of uncertain objects. UK-means relies on the expected distance between uncertain objects and (deterministic) cluster centroids, at each iteration. Since this is an expensive computation (the cost of the integral approximation based on S is not negligible), the complexity of UK-means is $\mathcal{O}(I\,S\,k\,n\,m)$. In order to improve the efficiency of the basic UK-means, [27] and [21] propose some pruning techniques to avoid the calculation of redundant object-to-centroid distances, based on the computation of bounding-boxes and Voronoi diagrams, respectively. However, such techniques do not guarantee an asymptotic complexity gain w.r.t. the basic UK-means. In [23], the *CK-means* algorithm is proposed as a variant of UK-means that exploits the moment of inertia of rigid bodies in order to reduce the execution time for computing object-to-centroid distances. CK-means essentially comprises two steps: in the first one (*offline* phase), the distances between each object and its mass center are computed in $\mathcal{O}(S\ n\ m)$, 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 working in $\mathcal{O}(I\ k\ n\ m)$.

UK-means and CK-means suffer from an accuracy issue. Indeed, cluster centroids are computed as deterministic objects using the expected values of the pdfs of the clustered objects. In [15], the *UK-medoids* algorithm is proposed to overcome the above issue. It employs distance functions properly defined for uncertain objects that are pre-computed offline in $\mathcal{O}(S^2\ n^2\ m)$; these distances are then employed in a classic K-medoids scheme working in $\mathcal{O}(I\ n^2)$.

The works in [8], [14] focus on the uncertain K-center problem, which is exploited during the execution of several well-known partitional uncertain object clustering algorithms. In particular, [8] shows a variety of different bicriteria approximation algorithms, which are however unable to preserve the number of centers. [14] overcomes that limitation by providing true approximation algorithms for a wider class of K-center-based problems.

Other recent works aim to address the high dimensionality in uncertain data by focusing on the problems of subspace clustering [18] and projected (or projective) clustering [2] over uncertain objects. The authors in [35] propose an approach based on the well-known 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 [13].

Density-based approaches to clustering uncertain objects have been defined in [19], [22]. In [22], the *$\mathcal{F}$DBSCAN* is proposed as a fuzzy version of the popular DBSCAN,

Table I
COMPUTATIONAL COMPLEXITIES OF MAIN STATE-OF-THE-ART ALGORITHMS FOR CLUSTERING UNCERTAIN OBJECTS

algorithm	*total*	*online*	*offline*
UK-means	$\mathcal{O}(I\ S\ k\ n\ m)$	$\mathcal{O}(I\ S\ k\ n\ m)$	—
CK-means	$\mathcal{O}(n\ m\ (I\ k + S))$	$\mathcal{O}(I\ k\ n\ m)$	$\mathcal{O}(S\ n\ m)$
UK-medoids	$\mathcal{O}(n^2(I + S^2\ m))$	$\mathcal{O}(I\ n^2)$	$\mathcal{O}(S^2\ n^2\ m)$
$\mathcal{F}$DBSCAN	$\mathcal{O}(S^2\ n^2\ m)$	$\mathcal{O}(S^2\ n^2\ m)$	—
$\mathcal{F}$OPTICS	$\mathcal{O}(S\ n^2\ m)$	$\mathcal{O}(S\ n^2\ m)$	—
UAHC	$\mathcal{O}(n^2(S\ m + \log n))$	$\mathcal{O}(n^2(S\ m + \log n))$	—

mainly based on the use of fuzzy distance functions to compute the *core object* and *reachability* probabilities. The authors propose an MBR-based heuristic that reduces the number of distances to be computed from $\mathcal{O}(S^2\ n^2)$ to a minimum $\mathcal{O}(n^2)$. Unfortunately, the pruning power of the proposed strategy is not ensured in general; therefore, the computational complexity of $\mathcal{F}$DBSCAN in the worst case is $\mathcal{O}(S^2\ n^2\ m)$. A similar approach is presented in *$\mathcal{F}$OPTICS* [19]. Like the well-known density-based clustering algorithm OPTICS, $\mathcal{F}$OPTICS produces an augmented ordering of the objects based on the novel notion of fuzzy object reachability-distance. By exploiting proper data structures (i.e., core object arrays and reachability lists), $\mathcal{F}$OPTICS can be executed in $\mathcal{O}(S\ n^2\ m)$.

The augmented ordering outputted by $\mathcal{F}$OPTICS makes that method falling into the category of hierarchical clustering algorithms as well. Another hierarchical method, i.e., *UAHC*, is proposed in [17]. UAHC exploits a cluster merging criterion based on an information-theoretic measure to compute the distance between cluster prototypes. The complexity of UAHC is $\mathcal{O}(n^2(S\ m + \log n))$.

III. MODELING UNCERTAINTY

Uncertain objects are typically represented according to *multivariate* or *univariate* uncertainty models [15]. In a multivariate uncertainty model, an uncertain object is defined in terms of an m-dimensional region and a multivariate pdf, which stores the probability that the exact representation of the object coincides with any multidimensional point in the region. In a univariate uncertainty model, any uncertain object has, for each of its m attributes, an interval and a univariate pdf that assigns a probability value to each point within the interval.

Although any uncertain object can be represented according to either model depending on the specific application context, the univariate model can be considered as a particular case of the multivariate one. In fact, a univariate uncertain object o can be treated as multivariate by either *(i)* explicitly transforming o into a multivariate object, provided that the conditional pdfs are known in advance or specific statistical assumptions can be made (e.g., statistical independence), or *(ii)* considering each attribute of o as a 1-dimensional multivariate object, in order to apply the functions/definitions/models designed for the multivariate case, and then exploiting simple methods for combining the results obtained for all attributes into a single one. For this reason, in the rest of the paper we will focus on the most general case of multivariate uncertainty modeling, whose formal definition is provided 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:

$$f(\vec{x}) = 0,\ \forall \vec{x} \in \Re^m \setminus \mathcal{R} \tag{1}$$

$$f(\vec{x}) > 0,\ \forall \vec{x} \in \mathcal{R} \tag{2}$$

□

The above definition refers to the most general case of uncertain objects modeled as continuous random variables and described by pdfs. Nevertheless, this definition also includes the case where uncertain objects are represented by discrete probability mass functions, as well as the case where the pdfs are approximated by a set of statistical samples. For the sake of brevity, we hereinafter refer to the continuous uncertainty model, as the corresponding discrete version can be trivially obtained by replacing integrals with sums.

The expected value ($\vec{\mu}$), second order moment ($\vec{\mu}_2$), and variance ($\vec{\sigma}^{\,2}$) vectors of any given uncertain object $o = (\mathcal{R}, f)$ are defined as follows:

$$\vec{\mu}(o) = \int_{\vec{x} \in \mathcal{R}} \vec{x}\ f(\vec{x})\ \mathrm{d}\vec{x} \tag{3}$$

$$\vec{\mu}_2(o) = \int_{\vec{x} \in \mathcal{R}} \vec{x}^{\,2} f(\vec{x})\ \mathrm{d}\vec{x} \tag{4}$$

$$\vec{\sigma}^{\,2}(o) = \int_{\vec{x} \in \mathcal{R}} (\vec{x} - \vec{\mu}(o))^2 f(\vec{x})\ \mathrm{d}\vec{x}\ =\ \vec{\mu}_2(o) - \vec{\mu}^{\,2}(o) \tag{5}$$

Each j-th component ($j \in [1..m]$) of the $\vec{\mu}$, $\vec{\mu}_2$ and $\vec{\sigma}^2$ vectors is as follows:

$$\mu_j(o) = \int_{\vec{x} \in \mathcal{R}} x_j\ f(\vec{x})\ \mathrm{d}\vec{x} \quad (\mu_2)_j(o) = \int_{\vec{x} \in \mathcal{R}} x_j^{\,2} f(\vec{x})\ \mathrm{d}\vec{x} \tag{6}$$

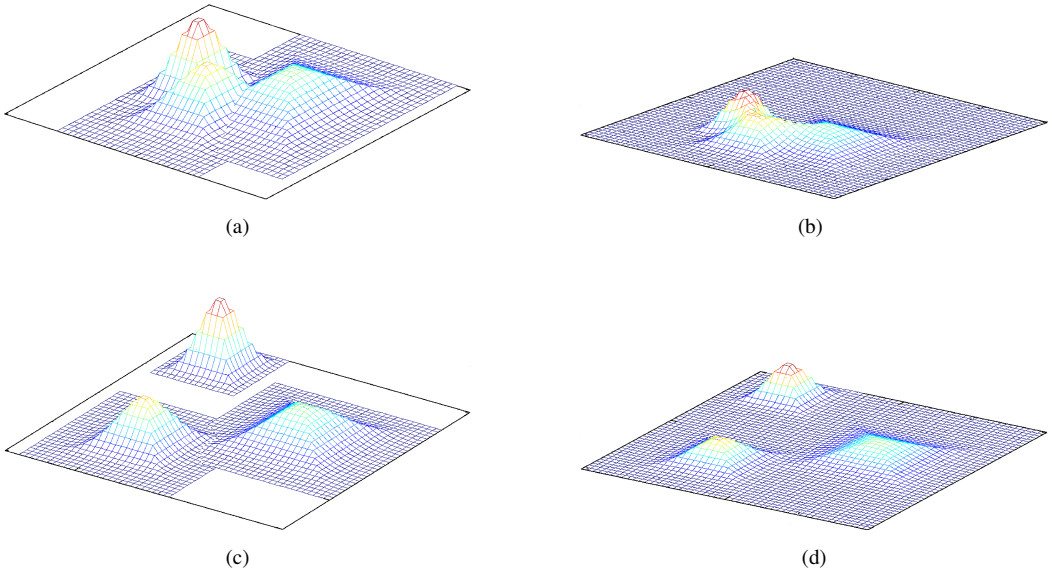

Figure 2. Uncertain prototype variance and cluster compactness: (a) compact set of objects and (b) its low-variance uncertain prototype, (c) less compact set of objects and (d) its higher-variance uncertain prototype.

$$(\sigma^2)_j(o) = \int_{\vec{x}\in\mathcal{R}} (x_j - \mu_j(o))^2 f(\vec{x})\ \mathrm{d}\vec{x} = (\mu_2)_j(o) - \mu_j^2(o) \quad (7)$$

Furthermore, given any vector $\vec{\sigma}^2$ of variances, the "global" variance expressed in terms of a single numerical value is defined as the sum of variances along each dimension:

$$\sigma^2(o) = \|\vec{\sigma}^2(o)\|_1 = \sum_{j=1}^{m} (\sigma^2)_j = \int_{\vec{x}\in\mathcal{R}} \|\vec{x} - \vec{\mu}(o)\|^2 f(\vec{x})\ \mathrm{d}\vec{x} \quad (8)$$

Note that if f is either a discrete probability mass function or approximated by a set $\mathcal{S}$ of statistical samples, (3) and (4) can be rewritten as follows:

$$\vec{\mu}(o) = \left(\sum_{\vec{y}\in\mathcal{S}} f(\vec{y})\right)^{-1} \sum_{\vec{y}\in\mathcal{S}} \vec{y}\ f(\vec{y}) \quad (9)$$

$$\vec{\mu}_2(o) = \left(\sum_{\vec{y}\in\mathcal{S}} f(\vec{y})\right)^{-1} \sum_{\vec{y}\in\mathcal{S}} \vec{y}^{\,2} f(\vec{y}) \quad (10)$$

IV. Clustering Uncertain Objects via Cluster Variance Minimization

A key notion in the proposed formulation to the problem of clustering uncertain objects is that of *uncertain prototype* (or simply *prototype*) of a set of uncertain objects. Any uncertain prototype is defined as the *mixture model* of the random variables representing the objects in a given set.

Definition 2: The *uncertain prototype* $\mathcal{P}_C$ of any given set C of uncertain objects is a pair $(\mathcal{R}_C, f_C)$, where

$$\mathcal{R}_C = \bigcup_{o=(\mathcal{R},f)\in C} \mathcal{R} \quad (11)$$

$$f_C(\vec{x}) = \frac{1}{|C|} \sum_{o=(\mathcal{R},f)\in C} f(\vec{x}) \quad (12)$$

□

It can be noted that any uncertain prototype defined as reported above is an uncertain object according to Def. 1.

Fact 1: The uncertain prototype $\mathcal{P}_C = (\mathcal{R}_C, f_C)$ of any set C of uncertain objects is a multivariate uncertain object satisfying Def. 1. □

Defining uncertain prototypes as mixture models has a number of crucial advantages. First, it 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 (or centroids) of uncertain objects, which collapse all the information about uncertainty into a single, representative numerical value, like that exploited by UK-means and CK-means algorithms (cf. Section II). 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. Finally, more importantly, focusing on statistical properties of the uncertain prototypes defined as mixture models enables the definition of an effective criterion to properly formulate clustering of uncertain objects.

In this respect, an interesting remark about the variance of mixture models can be drawn based on the following

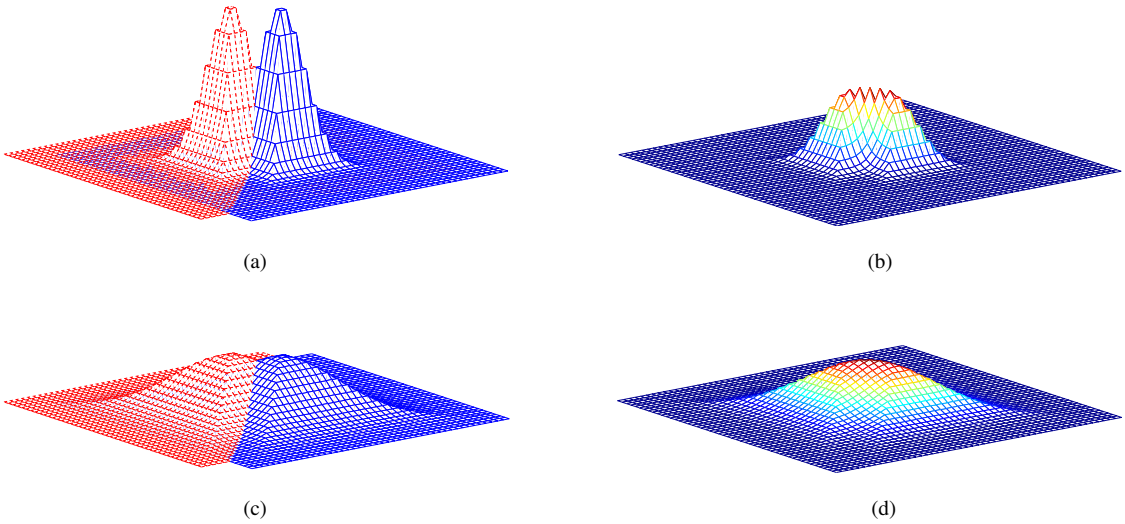

Figure 3. Impact of variance on the object uncertainty: (a) set of low-variance objects and (b) its low-variance uncertain prototype implying low uncertainty, (a) set of higher-variance objects and (b) its higher-variance uncertain prototype implying higher uncertainty, though the closeness between the object expected values.

intuition: the lower the variance of the mixture model (i.e., uncertain prototype) of a set of uncertain objects, the higher the compactness of that set, i.e., the higher the probability that the set is a high-quality cluster. This concept is graphically depicted in Fig. 2, where two sets of 2-dimensional uncertain objects are shown ((a) and (c)), along with their corresponding uncertain prototypes ((b) and (d)). It is easy to observe that the set in Fig. 2-(a) should be recognized as a better cluster than the set on Fig. 2-(c), as the compactness of the former is clearly higher. This remark is confirmed if one takes care of the variance of the uncertain prototypes that describe the two sets. Indeed, the variance of the prototype of the first set (Fig. 2-(b)) is clearly lower than the variance of the other prototype (Fig. 2-(d)), as the values of the pdf depicted in Fig. 2-(b) are close to the expected value more than those of the pdf in Fig. 2-(d).

Moreover, it should be noted that the variance of the mixture model of a set of uncertain objects is strictly related to the uncertainty of those objects; thus, it allows for taking into account uncertainty carefully, increasing the probability of achieving higher accuracy in a clustering process. This intuition is supported by the example illustrated in Fig. 3, where two pairs of uncertain objects along with their corresponding mixture models are depicted (Fig. 3(a)-(b) and Fig. 3(c)-(d), respectively). For both pairs of objects, the distance between their expected values is very small, while the variance of the objects in Fig. 3(c) is much larger than the variance of the objects in Fig. 3(a). If the uncertainty of the objects is not taken into account, i.e., if one considers the expected values only, both pairs would represent good clusters. Nevertheless, the difference in the variances implies that the uncertainty of the objects in Fig. 3(c) is greater than that of objects Fig. 3(a), as for the former pair of objects the "spreading" of values w.r.t. the expected value is larger; thus, even if the distance between the corresponding expected values is the same, the objects in Fig. 3(a) should represent a higher-quality cluster than the objects in Fig. 3(c). It easy to see that considering the variance of the mixture models of the two pairs allows for correctly discriminating between the two cases; indeed, the variance of the mixture model in Fig. 3(b) is evidently smaller than that of the mixture model in Fig. 3(d).

Based on the above considerations, we propose to formulate the problem of clustering uncertain objects by minimizing the variance of the uncertain prototypes of the clusters to be identified. Formally, given a set $\mathcal{D}$ of uncertain objects, the objective is to find a partition of $\mathcal{D}$ that minimizes the following objective function:

$$J(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sigma^2(\mathcal{P}_C) \tag{13}$$

where $\sigma^2(\mathcal{P}_C)$ is the variance of the prototype $\mathcal{P}_C$ of cluster C, which is computed according to (8).

A. The MMVar Algorithm

We show next some analytical properties about the computation of the variance of mixture models. Such properties are at the basis of the MMvar heuristic proposed in this work.

Lemma 1: Let C be a set of uncertain objects, where each $o \in C$ is a pair $(\mathcal{R}, f)$, and $\mathcal{P}_C$ be the uncertain prototype of

C. The expected value $\vec{\mu}(\mathcal{P}_C)$ and the second order moment $\vec{\mu}_2(\mathcal{P}_C)$ of prototype $\mathcal{P}_C$ are as follows:

$$\vec{\mu}(\mathcal{P}_C) = \frac{1}{|C|} \sum_{o \in C} \vec{\mu}(o)$$

$$\vec{\mu}_2(\mathcal{P}_C) = \frac{1}{|C|} \sum_{o \in C} \vec{\mu}_2(o)$$

□

Lemma 2: Let $\mathcal{P}_C$ be the uncertain prototype of any set C of uncertain objects, C' (resp. C'') be the set defined by deleting (resp. adding) object o' (resp. o'') from (resp. to) C, i.e., $C' = C \setminus \{o'\}$, $C'' = C \cup \{o''\}$. The expected values $\vec{\mu}(\mathcal{P}_{C'})$, $\vec{\mu}(\mathcal{P}_{C''})$ and second order moments $\vec{\mu}_2(\mathcal{P}_{C'})$, $\vec{\mu}_2(\mathcal{P}_{C''})$ of prototypes $\mathcal{P}_{C'}$, $\mathcal{P}_{C''}$ of sets C', C'' are as follows:

$$\vec{\mu}(\mathcal{P}_{C'}) = \frac{|C| \times \vec{\mu}(\mathcal{P}_C) - \vec{\mu}(o')}{|C| - 1}$$

$$\vec{\mu}_2(\mathcal{P}_{C'}) = \frac{|C| \times \vec{\mu}_2(\mathcal{P}_C) - \vec{\mu}_2(o')}{|C| - 1}$$

$$\vec{\mu}(\mathcal{P}_{C''}) = \frac{|C| \times \vec{\mu}(\mathcal{P}_C) + \vec{\mu}(o'')}{|C| + 1}$$

$$\vec{\mu}_2(\mathcal{P}_{C''}) = \frac{|C| \times \vec{\mu}_2(\mathcal{P}_C) + \vec{\mu}_2(o'')}{|C| + 1}$$

□

Proposition 1: Let $\mathcal{D}$ be a set of m-dimensional uncertain objects, $\mathcal{C}$ be a partition of $\mathcal{D}$, $\mathcal{P}_C$ be the prototype of any cluster $C \in \mathcal{C}$, and $\vec{\mu}(\mathcal{P}_C)$, $\vec{\mu}_2(\mathcal{P}_C)$ and $\sigma^2(\mathcal{P}_C) = \|\vec{\mu}_2(\mathcal{P}_C) - \vec{\mu}^2(\mathcal{P}_C)\|_1$ the expected value, second order moment and variance of $\mathcal{P}_C$, respectively. Let us consider a new partition $\mathcal{C}'$ of $\mathcal{D}$ obtained from $\mathcal{C}$ by moving an object o from cluster $C \in \mathcal{C}$ to cluster $\widehat{C} \in \mathcal{C}$; it holds that the value $J_{C,\widehat{C}}(\mathcal{C})$ of the objective function J for the new partition $\mathcal{C}'$ is the following:

$$J_{C,\widehat{C}}(\mathcal{C}) = J(\mathcal{C}) - (\sigma^2(\mathcal{P}_C) + \sigma^2(\mathcal{P}_{\widehat{C}})) + (\sigma^2(\mathcal{P}_{C'}) + \sigma^2(\mathcal{P}_{\widehat{C}'})) \tag{14}$$

where

$$C' = C \setminus \{o\} \qquad \widehat{C}' = \widehat{C} \cup \{o\}$$

$$\sigma^2(\mathcal{P}_{C'}) = \|\vec{\mu}_2(\mathcal{P}_{C'}) - \vec{\mu}^2(\mathcal{P}_{C'})\|_1$$

$$\sigma^2(\mathcal{P}_{\widehat{C}'}) = \|\vec{\mu}_2(\mathcal{P}_{\widehat{C}'}) - \vec{\mu}^2(\mathcal{P}_{\widehat{C}'})\|_1$$

and

$$\vec{\mu}(\mathcal{P}_{C'}) = \frac{|C| \times \vec{\mu}(\mathcal{P}_C) - \vec{\mu}(o)}{|C| - 1}$$

$$\vec{\mu}_2(\mathcal{P}_{C'}) = \frac{|C| \times \vec{\mu}_2(\mathcal{P}_C) - \vec{\mu}_2(o)}{|C| - 1}$$

$$\vec{\mu}(\mathcal{P}_{\widehat{C}'}) = \frac{|\widehat{C}| \times \vec{\mu}(\mathcal{P}_{\widehat{C}}) + \vec{\mu}(o)}{|\widehat{C}| + 1}$$

Algorithm 1 MMVar

Input: A set $\mathcal{D}$ of uncertain objects; $k = \sup(\mathbb{N})$, or $k \leq |\mathcal{D}|$ as a user-specified parameter (i.e., number of output clusters).
Output: A partition $\mathcal{C}$ of $\mathcal{D}$.

1: compute $\vec{\mu}(o)$, $\vec{\mu}_2(o)$, $\forall o \in \mathcal{D}$ {*(3)–(4), (9)–(10)*}
2: select $o^* \in \mathcal{D}$, randomly or by a predefined criterion
3: $\mathcal{C} \leftarrow \{\{o^*\}\}$
4: **repeat**
5: **for all** $o \in \mathcal{D}$ **do**
6: **if** o is not assigned to any cluster yet **then**
7: **if** $|\mathcal{C}| < k$ **then**
8: $C \leftarrow \{o\}$, $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$
9: compute $\vec{\mu}(\mathcal{P}_C)$, $\vec{\mu}_2(\mathcal{P}_C)$ {*Lemma 1*}
10: **else**
11: select $C \in \mathcal{C}$, randomly or by a predefined criterion
12: $C \leftarrow C \cup \{o\}$
13: recompute $\vec{\mu}(\mathcal{P}_C)$, $\vec{\mu}_2(\mathcal{P}_C)$ {*Lemma 2*}
14: **end if**
15: $V \leftarrow J(\mathcal{C})$ {*(13)*}
16: **end if**
17: $C^* \leftarrow \arg\min_{\widehat{C}} J_{C,\widehat{C}}(\mathcal{C})$, with $o \in C$ {*(14)*}
18: **if** $C^* \neq C \ \wedge (|C| > 1 \vee k = \sup(\mathbb{N}))$ **then**
19: $V \leftarrow J_{C,C^*}(\mathcal{C})$ {*(14)*}
20: recompute $\mathcal{C}$: move o from C to C^*, and remove C from $\mathcal{C}$ in case $C = \emptyset$
21: recompute $\vec{\mu}(\mathcal{P}_C)$, $\vec{\mu}_2(\mathcal{P}_C)$, $\vec{\mu}(\mathcal{P}_{C^*})$, $\vec{\mu}_2(\mathcal{P}_{C^*})$ {*Lemma 2*}
22: **end if**
23: **end for**
24: **until** no $o \in \mathcal{D}$ is relocated

$$\vec{\mu}_2(\mathcal{P}_{\widehat{C}'}) = \frac{|\widehat{C}| \times \vec{\mu}_2(\mathcal{P}_{\widehat{C}}) + \vec{\mu}_2(o)}{|\widehat{C}| + 1}$$

□

Proposition 1 hence states that, given any partition $\mathcal{C}$ of $\mathcal{D}$ and any other partition $\mathcal{C}'$ obtained from $\mathcal{C}$ by moving a single object from a cluster to another one, the value of the objective function J for $\mathcal{C}'$ can be computed in $\mathcal{O}(m)$ from $J(\mathcal{C})$ according to (14). This result puts the basis for our proposed heuristic algorithm, called *MMVar*, whose major feature lies in the capability of efficiently finding local minima of function J, without requiring any distance measure between uncertain objects.

The outline of MMVar is reported in Alg. 1. MMVar can either automatically discover the number of clusters or, alternatively, can be constrained to form a user-specified number of clusters. In the former case, k should be set to $\sup(\mathbb{N})$. To exploit the result of Proposition 1, MMVar only needs to maintain expected values and second order moments of the objects within $\mathcal{D}$ (i.e., vectors $\vec{\mu}(o)$ and $\vec{\mu}_2(o)$) and of the prototypes that are identified at each iteration (i.e., vectors $\vec{\mu}(\mathcal{P}_C)$ and $\vec{\mu}_2(\mathcal{P}_C)$). Expected values and second order moments are computed according to either the exact (cf. (3)-(4)) or approximated (cf. (9)-(10)) formulas (Line 1).

The main cycle of the algorithm (Lines 4–24) is in charge of iteratively process all objects within $\mathcal{D}$, until

no decrement in the objective function has been observed. We can distinguish two phases: the initialization phase and the refinement of the clustering produced at the previous iteration. The initialization phase consists in an incremental scheme where the initial cluster of any single object is chosen in a greedy fashion according to the assignments already performed for the objects that have been processed earlier. The selection of the first object to allocate (Line 2) is either randomly performed or based on any other procedure provided that it guarantees linearity with the number of objects—one reasonable choice is to select o^* whose variance is closest to the average variance per object (i.e., $\sigma^2(o^*) \approx \text{avg}_{o \in \mathcal{D}} \sigma^2(o)$). The initial cluster C of any (unclustered) object o is chosen in two steps. First, C is initialized either as a singleton cluster containing o (Lines 7–9) or as a cluster among the ones already existing (Lines 10–13), depending whether the desired number of clusters k has been already reached or not. In case of an existing cluster is chosen as C, this choice could be made either randomly or according to a specific criterion (Line 11)—for instance, select $C \in \mathcal{C}$ such that $\sigma^2(\mathcal{P}_C) \geq \sigma^2(o)$, i.e., choose a cluster for which the insertion of an object will not increase the variance of the cluster prototype. Afterwards, the choice of C is refined by looking at the best cluster for o given the cluster assignments of the objects processed before o (Lines 18–23). Note that the initialization step based on an incremental procedure represents a point of difference between this formulation of MMVar and the earlier one presented in [16], in which the clustering initialization was devised either as a random generation or as the output of an external clustering algorithm.

After the initialization phase, a refinement step is carried out, where objects are relocated until cluster stability (Line 24) is reached. Note that once all objects have been clustered, the only steps of the main cycle actually performed are the ones in Lines 17–22, while Lines 6–16 are entirely skipped. At any iteration of the refinement phase, for each object $o \in \mathcal{D}$, the cluster C^* is discovered (Line 17) as the one leading to the maximum decrease in the objective function J if o is moved to it. C^* is discovered by applying (14) to the current partition $\mathcal{C}$; note that (14) is computed by taking into account the value V of the objective function J for the current partition $\mathcal{C}$. If C^* does not coincide with the cluster C to which o currently belongs (i.e., there exists at least one cluster but C such that function J decreases if o is moved to it), o is moved to C^* and both the new value V of function J and the expected values and second order moments of the prototypes of clusters C and C^* are recomputed, according to (14) and Lemma 2, respectively (Lines 18-22). Note that, if $k = \sup(\mathbb{N})$, i.e., if MMvar is required to discover the number of clusters by itself, the best move for any object o might imply that the cluster where o originally belongs to becomes empty. If this happens, the empty cluster is removed and the number of output clusters changes accordingly, as required (Line 20).

The proposed MMVar can be proved to converge to a local optimum of the objective function therein involved, and work linearly w.r.t. both the size of the input dataset and the dimensionality of the input uncertain objects, as shown in the following.

Proposition 2: The MMVar algorithm converges to a local minimum of function J defined in (13) in a finite number of steps. □

Proposition 3: Given a set $\mathcal{D}$ of n m-dimensional uncertain objects, the number k of output clusters, the number S of statistical samples used for representing the uncertain objects, and denoting by I the number of iterations to convergence, the computational complexity of the MMVar algorithm (Alg. 1) is $\mathcal{O}(n\ m\ (I\ k + S))$. □

Note that the MMVar computational complexity reduces to $\mathcal{O}(I\ k\ n\ m)$ when the moments of the various uncertain objects may be computed according to some closed-form expression in $\mathcal{O}(m)$; it typically happens in a wide number of real cases. Thus, looking at Table I, it is easy to note that the complexity of the proposed MMVar algorithm is at worst equal to and very often (far) lower than that of any other prominent algorithm for clustering uncertain objects; such a result proves a major claim of this work, which concerns the efficiency in solving the problem of clustering uncertain objects.

V. Experimental Evaluation

The proposed MMVar algorithm was evaluated in performing effective and efficient clustering of uncertain objects. The experimental evaluation was also conducted to compare MMVar with existing partitional algorithms, i.e., UK-means (UKM), CK-means (CKM), and UK-medoids (UKmed), density-based algorithms, i.e., $\mathcal{F}$DBSCAN ($\mathcal{F}$DB) and $\mathcal{F}$OPTICS ($\mathcal{F}$OPT), and the hierarchical algorithm UAHC. We also compare the MMVar algorithm proposed in this work with its early version defined in [16]. In the following, we discuss the evaluation methodology used in this work, whereas in Section V-B, we present the main experimental results from accuracy, efficiency and scalability viewpoints.

A. Experimental methodology

1) Datasets: Experiments were carried out on benchmark and real datasets. We selected eight benchmark datasets from [5], whose main characteristics are summarized in Table II-(a). Iris contains measurements on different iris plants. Wine refers to results of a chemical analysis on Italian wines derived from three different cultivars. In Glass, each glass instance is described by the values of its chemical components. Ecoli contains data on the Escherichia Coli bacterium, which are identified with values coming from

Table II
DATASETS USED IN THE EXPERIMENTS

(a) Benchmark datasets

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
KDDCup99	4,000,000	42	23

(b) Real datasets

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

different analysis techniques. Yeast objects describe main features and localization of various proteins. Image contains objects that were randomly drawn from a database of seven outdoor images; the images (3x3 regions) were hand-segmented to create a classification for each pixel. Abalone describes different types of abalone shells. Letter contains character images corresponding to the capital letters in the English alphabet,

As regards real data, we used datasets that describe gene expressions in biological tissues. These datasets were generated by *microarray analysis*, which is a technique widely used in genomics to enable detection of the expression levels of thousands of genes simultaneously [31]. We collected two microarray datasets available from [6], namely Neuroblastoma and Laukaemia, each describing the expressions of about 22,000 genes in cancer tissues (Table II-(b)). In particular, the Leukaemia dataset describes the transformation process of leukaemia stem cells initiated by MLL-AF9 fusion gene, while Neuroblastoma contains expression-based screening results for neuroblastoma differentiation. We recall that the genomics and, in general, biomedical domains are examplary real-life scenarios of data uncertainty (cf. Introduction).

Moreover, specifically for the scalability study, we used a very large dataset (4 million objects, last row of Table II-(a)), which was employed for the KDD Cup '99 contest and now available from the UCI repository [5].

2) Cluster Validity Criteria: 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 employ the well-known *F-measure* (F) [34], 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 and with $\mathcal{C} = \{C_1, \ldots, C_k\}$ a clustering solution, F-measure is defined as:

$$F(\mathcal{C}, \widetilde{\mathcal{C}}) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{h} |\widetilde{C}_i| \max_{j \in [1..k]} F_{ij}$$

where

$$F_{ij} = \frac{2\ P_{ij}\ R_{ij}}{P_{ij} + R_{ij}} \qquad P_{ij} = \frac{|C_j \cap \widetilde{C}_i|}{|C_j|} \qquad R_{ij} = \frac{|C_j \cap \widetilde{C}_i|}{|\widetilde{C}_i|}$$

for each $i \in [1..h]$, $j \in [1..k]$.

We also use two internal cluster validity approaches to evaluate the quality of the obtained clustering solutions. The first approach is based on *intra-cluster* ($intra(\mathcal{C})$) and *inter-cluster* ($inter(\mathcal{C})$) distances [20] (for a given clustering solution $\mathcal{C}$) which express cluster cohesiveness and cluster separation, respectively. Such distance values are 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 are normalized within $[0,1]$, Q ranges within $[-1,1]$. Formally:

$$intra(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} \frac{2}{|C|(|C|-1)} \sum_{o_u \in C} \sum_{\substack{o_v \in C, \\ u<v}} ED(o_u, o_v)$$

$$inter(\mathcal{C}) = \frac{2}{|\mathcal{C}|(|\mathcal{C}|-1)} \sum_{\substack{C_i, C_j \in \mathcal{C}, \\ i<j}} \frac{1}{|C_i| \times |C_j|} \sum_{o_u \in C_i} \sum_{o_v \in C_j} ED(o_u, o_v)$$

where ED denotes the expected distance between any two uncertain objects [15].

The second internal criterion is the *Silhouette index* [28]. This criterion is still based on the comparison of cluster tightness and separation, which are however calculated as the "silhouette width" for each object, average silhouette width for each cluster, and overall average silhouette width for the whole dataset. Formally, the Silhouette index $S(o, C)$ of an object $o \in C$ is:

$$S(o, C) = \frac{b(o, C) - a(o, C)}{max\{a(o, C), b(o, C)\}}$$

where $a(o, C)$ denotes the average distance between o and each other object in C, $b(o, C)$ denotes the minimum average distance between o and the objects in the other clusters. The Silhouette $S(C)$ of a cluster $C \in \mathcal{C}$ is defined as $S(C) = |C|^{-1} \sum_{o \in C} S(o, C)$. Finally, the global Silhouette $S(\mathcal{C})$ of the whole clustering solution $\mathcal{C}$ is computed as:

$$S(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} S(C)$$

Since $S(o, C)$ ranges within $[-1,1]$, $S(C)$ and $S(\mathcal{C})$ range in the same interval as well. Like Q, larger overall average silhouette indicates better clustering.

Table III
ACCURACY RESULTS ON BENCHMARK DATASETS: EXTERNAL EVALUATION

		F-measure ($\Theta \in [-1, 1]$)							
data	*pdf*	UKM	CKM	UKmed	$\mathcal{F}$DB	$\mathcal{F}$OPT	UAHC	**MMVar [16]**	**MMVar**
Iris	U	-.062	.028	.023	-.102	.005	.032	.073	.021
	N	-.010	.013	.010	-.063	.044	-.009	.030	-.011
	B	-.249	-.380	-.045	-.383	.023	-.339	-.012	-.012
Wine	U	-.179	.047	.175	-.179	.174	.254	.151	.134
	N	-.184	.024	-.085	-.185	.030	.076	.039	.092
	B	-.208	-.127	-.104	-.208	.006	-.015	-.212	-.179
Glass	U	-.066	-.053	-.048	-.430	-.120	.008	-.107	.010
	N	-.025	.012	-.070	-.040	-.136	.176	.035	.141
	B	-.231	-.302	.009	-.334	-.182	-.150	.181	.183
Ecoli	U	.199	.332	.223	-.136	.023	.088	.324	.286
	N	.131	.272	.045	.061	.015	.047	.286	.286
	B	-.160	-.303	-.034	-.383	-.239	-.179	-.017	-.017
Yeast	U	.220	.279	.315	-.085	.252	.182	.340	.339
	N	.159	.145	-.035	.079	-.001	.167	.398	.404
	B	-.098	-.201	-.055	-.311	-.195	-.109	.211	.211
Image	U	.278	.274	.241	-.283	-.113	.046	.071	.215
	N	.122	.132	-.061	-.251	-.081	.127	.028	.220
	B	-.024	-.204	.087	-.307	-.137	-.020	.144	.230
Abalone	U	.120	.092	.379	-.092	.291	.084	.539	.545
	N	.034	-.031	.009	.095	-.039	.059	.138	.180
	B	.080	-.084	.025	-.182	.315	.013	.496	.496
Letter	U	.008	.113	.237	-.338	-.201	.026	.165	.191
	N	-.076	-.082	-.039	-.340	-.203	.037	.127	.213
	B	-.202	-.399	.033	-.431	-.294	-.041	.033	.132
avg. score	U	.065	.139	.193	-.206	.039	.090	.195	.218
	N	.019	.061	-.028	-.081	-.046	.085	.135	.191
	B	-.137	-.250	-.011	-.317	-.088	-.105	.103	.131
overall avg. score		-.018	-.017	.051	-.201	-.032	.023	.144	.180
overall avg. gain		+.198	+.197	+.129	+.381	+.212	+.203	+.036	—

3) Uncertainty generation in benchmark datasets: 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 have *probe-level* uncertainty, which can easily be modeled in the form of Normal pdfs according to the *multi-mgMOS* method [25].[1] According to an approach already employed by previous works [7], 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 $\mathcal{D}$. In particular, we considered the *Uniform*, *Normal* and *Binomial* distributions, as they are commonly encountered in real uncertain data scenarios [1]. Every $f_{\vec{w}}$ was defined in such a way that its expected value corresponded 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 or the standard deviation of the Normal pdfs) were randomly chosen. We exploited the pdfs $f_{\vec{w}}$ to simulate what actually happens in typical real contexts for uncertain data, like the one depicted in Fig. 1. 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 $f_{\vec{w}}$ according to the classic Monte Carlo and Markov Chain Monte Carlo methods.[2] As a result, $\mathcal{D}'$ still contains deterministic data, which can be interpreted as observed representations of the input uncertain objects. In our evaluation, each of the selected clustering methods was carried out on $\mathcal{D}'$ so that it produced an output clustering solution denoted by $\mathcal{C}'$. A score $F(\mathcal{C}', \widetilde{\mathcal{C}})$ was hence obtained by comparing the output clustering $\mathcal{C}'$ to the reference classification of $\mathcal{D}$ (denoted by $\widetilde{\mathcal{C}}$) by means of the F-measure cluster validity criterion.

In Case 2, when uncertainty is taken into account, we further created an *uncertain dataset* $\mathcal{D}''$ from $\mathcal{D}$ which is the one designed to contain uncertain objects. In particular, for each $\vec{w} \in \mathcal{D}$, we derived an uncertain object $o = (\mathcal{R}, f)$ in such a way that $f = f_{\vec{w}}$, 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 clustering methods on $\mathcal{D}''$ as well, in

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

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

order to obtain 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 $\Theta(\mathcal{C}', \mathcal{C}'', \widetilde{\mathcal{C}}) = F(\mathcal{C}'', \widetilde{\mathcal{C}}) - F(\mathcal{C}', \widetilde{\mathcal{C}})$; 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 taken into account w.r.t. the case where no uncertainty is employed. Note that Θ ranges within $[-1, 1]$.

4) Setup of the clustering methods: We derived the lists of samples from the pdfs by employing the classic *Monte Carlo* and *Markov Chain Monte Carlo* sampling methods. 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 for each of the algorithms were averaged over 50 runs.

We performed a tuning phase for parameters ϵ (i.e., the threshold for the distance of the neighbors of any object) and μ (i.e., the minimum number of points within the neighborhood of any object) required by the density-based approaches $\mathcal{F}$DBSCAN and $\mathcal{F}$OPTICS. We set these parameters to the values that allowed each method to achieve the best accuracy results.

For UAHC and $\mathcal{F}$OPTICS, we considered the partitions obtained by cutting the dendrogram to the desired number of output clusters. For $\mathcal{F}$OPTICS, we initially derived a cluster hierarchy by means of the procedure described in [29].

B. Results

1) Accuracy on Benchmark Datasets: Tables III-IV show accuracy results on benchmark datasets for Uniform (U), Normal (N), and Binomial (B) distributions, in terms of external (Θ) and internal (Q) cluster validity criteria, respectively. In both tables, we also report, for each method, *(i)* the score for each type of pdf averaged over all datasets (for short, average score), *(ii)* the score averaged over all datasets and pdfs (for short, overall average score), and *(iii)* the overall average gain of our MMVar computed as the difference between the overall average score of MMvar and the overall average scores of the other algorithms.

The overall average scores and gains in terms of both Θ and Q, show that the proposed MMVar was more accurate than any other competing method, including the earlier MMVar [16]. In particular, the maximum gains were equal to 0.381 Θ and 0.215 Q, both obtained with respect to $\mathcal{F}$DB; the latter revealed to be the least accurate method, probably because of negative effects that rely on the difficulty in setting parameters ϵ and μ. Among the competitors, with the exception of the earlier MMVar, UKmed achieved the best results in terms of Θ (gap from MMVar equal to 0.129), whereas the best performing methods in terms of Q were $\mathcal{F}$OPT and UKmed (gap from MMVar equal to 0.155 and 0.158, respectively). UAHC overall performance (gaps from MMVar of 0.203 Θ and 0.195 Q) was closer to that of UKM and CKM than to the performance of density-based algorithms. UKM and CKM were comparable to each other in terms of Θ, and performed better than UAHC and the density-based methods; however, for the internal evaluation, CKM obtained the second worst overall average accuracy (gap from MMVar equal to 0.203). As concerns the comparison between the two versions of MMVar, although always better than the other competing methods, the earlier MMVar obtained lower performance than the current MMVar, with overall average gaps of 0.036 Θ and 0.022 Q.

Considering the average scores on single distributions, accuracy of MMVar remained on average higher than those of all other methods, including the earlier MMVar. Maximum and minimum average gains over all algorithms except the earlier MMVar were as follows: 0.424, 0.025 by Uniform, 0.272, 0.106 by Normal, and 0.448, 0.241 by Binomial, for Θ; 0.34, 0.228 by Uniform, 0.085, 0.035 by Normal, and 0.218, 0.167 by Binomial, for Q. Moreover, MMVar improved upon its earlier version up to 0.056 Θ and 0.032 Q, both on Normal pdfs.

Results obtained on the single dataset-by-pdf configurations further confirm the high accuracy obtained by MMVar. Indeed, according to Θ, MMVar achieved the best absolute results on 16 out of 24 dataset-by-pdf configurations; for additional 5 configurations, it remained comparable to the best method (gap lower than or equal to 0.05). Similarly, considering Q, MMVar was the best method on 14 configurations and achieved results comparable to the best ones in additional 2 configurations. Finally, we point out that MMVar was in general much more accurate than the method having the lowest computational complexity among the competitors, which is CKM (cf. Table I). Indeed, MMVar achieved Θ (resp. Q) results better than those of CKM on 19 (resp. 20) out of 24 dataset-by-pdf configurations.

2) Accuracy on Real Datasets: Table V 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 performed multiple tests by varying the number of clusters and assessed the results based on the internal criteria Q and S.

Looking at the Q average scores, MMVar was the best performing method on both datasets, with improvements of 0.030 on Neuroblastoma and of 0.022 on Leukaemia, both upon UAHC. Moreover, MMVar achieved the best overall average performance, with maximum and minimum gains (over the competing algorithms) of 0.523 (w.r.t. $\mathcal{F}$DB), and 0.027 (w.r.t. UAHC), respectively. In general, UAHC (resp. $\mathcal{F}$DB) was found as the best (resp. worst) method, whereas the performance of $\mathcal{F}$OPT and UKmed significantly decreased w.r.t. the benchmark datasets. For $\mathcal{F}$DB, in particular, its results remained constant even varying the number

Table IV
ACCURACY RESULTS ON BENCHMARK DATASETS: INTERNAL EVALUATION

		Quality ($Q \in [-1, 1]$)							
data	*pdf*	UKM	CKM	UKmed	$\mathcal{F}$DB	$\mathcal{F}$OPT	UAHC	**MMVar [16]**	**MMVar**
Iris	U	.151	.145	.148	.197	.093	.153	.147	.137
	N	.263	.194	.194	.238	.135	.231	.187	.167
	B	.118	-.001	.081	-.004	.202	-.001	.692	.716
Wine	U	-.001	-.002	.012	-.002	.128	-.001	-.001	.004
	N	-.020	.119	.417	.216	.093	.298	.124	.125
	B	.000	.011	.114	.000	.001	.000	.011	.012
Glass	U	.001	.001	.060	-.013	.001	.001	.226	.294
	N	.057	.062	.041	.042	.006	.142	.008	.079
	B	.004	.001	.006	-.002	.000	.000	.140	.134
Ecoli	U	.101	.031	.187	.000	.449	.008	.592	.591
	N	.141	.060	.029	.086	.284	.127	.151	.240
	B	.001	.000	.003	.000	.000	.000	.187	.187
Yeast	U	.041	.016	.193	.000	.289	.001	.566	.571
	N	.053	.031	.005	.040	.222	.150	.253	.269
	B	.003	.000	.001	.000	.002	.000	.184	.184
Image	U	.000	.000	.000	.000	.000	.000	.725	.736
	N	.065	.074	.010	-.001	.004	.130	.004	.021
	B	.015	.000	.159	.000	.000	.000	.008	.014
Abalone	U	.040	.025	.071	-.018	.101	.006	.226	.237
	N	.103	.055	.031	.086	.054	-.003	.057	.079
	B	.017	.005	.035	.000	.011	.000	.226	.226
Letter	U	.022	.006	.044	.000	.000	.054	.279	.319
	N	.352	.303	.357	-.022	.207	.015	.331	.387
	B	.000	.000	.000	.000	.000	.013	.147	.261
avg. score	U	.044	.028	.089	.021	.133	.028	.345	.361
	N	.127	.112	.136	.086	.126	.136	.139	.171
	B	.020	.002	.050	-.001	.027	.002	.199	.217
overall avg. score		.064	.047	.092	.035	.095	.055	.228	.250
overall avg. gain		+.186	+.203	+.158	+.215	+.155	+.195	+.022	—

of output clusters; this is motivated since this method automatically discovers the output clusters, therefore, setting any specific number does not have any impact on it. Again, our MMVar generally outperformed CKM: indeed, it was on average more accurate than CKM on both datasets (with gains of 0.025 and 0.146), while achieving better results on 14 out of 16 dataset-by-number-of-cluster configurations (i.e., all cases but those corresponding to a number of clusters equal to 2). Moreover, the proposed MMVar outperformed the earlier version defined in [16] in 12 out of 16 dataset-by-number-of-cluster configurations, while also achieving better average accuracy on both Neuroblastoma (0.030 average gain) and Leukaemia (0.022 average gain).

As far as the Silhouette cluster validity criterion, the average accuracy gains achieved by MMvar w.r.t. UKM, CKM, UKmed, and UAHC were even larger than those observed in terms of the Q scores. Density-based methods (i.e., $\mathcal{F}$DB and $\mathcal{F}$OPT) achieved instead generally better S results than Q; this is mainly motivated since the Silhouette index naturally tends to evaluate as better those clustering solutions in which clusters are quite dense and noisy objects are kept isolated, which are the kind of solution typically discovered by density-based clustering algorithms. However, our MMVar performed better than both $\mathcal{F}$DB and $\mathcal{F}$OPT on Laukaemia, while being better/comparable than/to $\mathcal{F}$DB on three configurations on Neuroblastoma. Again, the proposed MMVar was on average better than its earlier version in [16] on both datasets also in terms of S, achieving an overall average gain of 0.012.

3) Efficiency: We evaluated time performance of our MMVar and competing algorithms on both benchmark and real datasets.[3] Note that, unlike the accuracy evaluation, we avoid here to distinguish between the MMVar algorithm proposed in this work and the earlier version in [16] as the two algorithms have the same running time.

Figure 4 shows total (i.e., offline plus online) execution times (milliseconds, logarithmic scale) of all algorithms, while details about online vs. offline contributions are reported in Table VI; more specifically, Figures 4 (a)–(h) refer to the results obtained on benchmark datasets for each pdf, whereas Figure 4 (i) regards the two real datasets. We recall that we could not vary the pdf for real datasets as the uncertainty inherently provided along with these datasets is modeled according to Normal pdfs (cf. Sect. V-A). Moreover, results here refer to the computationally most expensive version of our MMVar, where the moments of the distributions are approximated according to a set of statistical samples (cf. Sect. IV).

[3]Experiments were conducted on a quad-core platform Intel Pentium IV 3GHz with 4GB memory and running Microsoft WinXP Pro.

Table V
ACCURACY RESULTS (QUALITY AND SILHOUETTE) ON REAL DATASETS

		Quality ($Q \in [-1, 1]$)								*Silhouette* ($S \in [-1, 1]$)							
data	*# clusters*	UKM	CKM	UKmed	$\mathcal{F}$DB	$\mathcal{F}$OPT	UAHC	**MMVar [16]**	**MMVar**	UKM	CKM	UKmed	$\mathcal{F}$DB	$\mathcal{F}$OPT	UAHC	**MMVar [16]**	**MMVar**
Neuroblast.	2	.569	.589	.435	-.004	.100	.568	.592	.588	.670	.671	.682	.721	.901	.671	.643	.645
	3	.608	.584	.467	-.004	.169	.671	.600	.601	.583	.584	.585	.721	.883	.608	.466	.468
	5	.595	.619	.432	-.004	.093	.624	.678	.690	.594	.560	.429	.721	.921	.480	.406	.443
	10	.068	.066	.048	-.004	.008	.122	.098	.113	.394	.367	.255	.721	.955	.586	.553	.547
	15	.598	.622	.444	-.004	.096	.590	.675	.697	.364	.284	.181	.721	.960	.542	.661	.648
	20	.609	.599	.473	-.004	.092	.533	.582	.601	.295	.273	.139	.721	.970	.602	.701	.710
	25	.646	.567	.409	-.004	.090	.583	.596	.619	.282	.253	.106	.721	.975	.625	.772	.762
	30	.472	.527	.433	-.004	.082	.523	.532	.551	.199	.196	.088	.721	.979	.508	.782	.785
Leukaem.	2	.207	.266	.221	-.018	.068	.293	.212	.228	.667	.633	.523	.577	.827	.590	.543	.548
	3	.392	.316	.256	-.018	.080	.314	.305	.348	.440	.568	.563	.577	.567	.482	.290	.471
	5	.451	.372	.245	-.018	.061	.443	.481	.517	.571	.485	.415	.577	.687	.533	.490	.512
	10	.455	.368	.238	-.018	.213	.554	.405	.441	.350	.305	.233	.577	.600	.581	.667	.667
	15	.451	.320	.246	-.018	.192	.487	.501	.531	.327	.266	.140	.577	.674	.648	.748	.737
	20	.479	.322	.213	-.018	.186	.483	.492	.527	.240	.235	.122	.577	.554	.683	.792	.778
	25	.558	.296	.215	-.018	.353	.552	.588	.616	.258	.246	.093	.577	.538	.702	.813	.810
	30	.448	.296	.213	-.018	.369	.422	.483	.521	.190	.150	.073	.577	.882	.638	.841	.838
Neuroblastoma *avg. score*		.521	.522	.393	-.004	.091	.527	.544	.557	.423	.399	.308	.721	.943	.578	.623	.626
Leukaemia *avg. score*		.430	.320	.231	-.018	.190	.444	.433	.466	.380	.361	.270	.577	.665	.607	.648	.670
overall avg. score		.475	.421	.312	-.011	.141	.485	.489	.512	.401	.380	.289	.649	.805	.592	.636	.648
overall avg. gain		+.037	+.091	+.200	+.523	+.371	+.027	+.023	—	+.247	+.268	+.359	-.001	-.157	+.056	+.012	—

Table VI
ONLINE VS. OFFLINE EXECUTION TIMES (MILLISECONDS)

		UKM		CKM		UKmed		FDB		FOPT		UAHC		**MMVar**	
data	*pdf*	*online*	*offline*	*online*	*offline*	*online*	*offline*	*online*	*offline*	*online*	*offline*	*online*	*offline*	*online*	*offline*
Iris	U	226	—	2	16	29	1,468	29	—	375	—	734	—	2	16
	N	191	—	3	32	33	2,125	80	—	375	—	5,934	—	1	16
	B	1,080	—	2	62	32	5,391	105	—	406	—	4,889	—	1	63
Wine	U	369	—	3	31	49	3,718	335	—	672	—	4,640	—	3	16
	N	692	—	5	62	50	7,406	623	—	625	—	36,710	—	4	63
	B	2,720	—	5	281	52	27,360	309	—	844	—	45,267	—	4	265
Glass	U	490	—	7	16	66	3,844	92	—	875	—	5,527	—	3	16
	N	764	—	1	62	73	7,500	832	—	765	—	23,399	—	1	47
	B	4,633	—	2	219	73	26,047	378	—	937	—	52,866	—	5	203
Ecoli	U	2,869	—	4	15	122	7,344	87	—	1,938	—	10,150	—	7	15
	N	3,268	—	11	63	143	13,672	1,693	—	1,657	—	74,304	—	6	47
	B	5,198	—	5	266	168	48,219	415	—	1,859	—	123,613	—	7	250
Yeast	U	24,895	—	119	562	2,651	865,984	40,639	—	39,890	—	276,908	—	49	453
	N	26,875	—	98	1,188	2,840	1,775,172	122,207	—	32,563	—	12,064,828	—	66	1,187
	B	36,466	—	101	5,062	3,539	6,266,453	12,746	—	33,641	—	4,374,297	—	59	4,766
Image	U	21,453	—	267	78	8,939	165,625	1,784	—	130,625	—	1,602,809	—	142	94
	N	28,575	—	213	313	8,912	349,578	65,987	—	134,141	—	14,754,456	—	139	313
	B	110,452	—	294	1,344	8,850	1,087,203	11,258	—	138,688	—	13,945,281	—	149	1,250
Abalone	U	113,990	—	825	406	20,568	1,153,672	10,328	—	392,625	—	1,821,687	—	520	265
	N	119,500	—	898	719	25,178	2,140,875	258,498	—	323,172	—	54,930,953	—	543	735
	B	143,331	—	656	3,359	26,837	7,472,469	65,135	—	336,765	—	11,758,297	—	323	3,281
Letter	U	142,735	—	2,517	1,063	100,112	8,077,735	412,738	—	1,701,250	—	8,310,666	—	741	1,031
	N	186,965	—	2,424	8,328	102,553	16,209,734	1,296,300	—	1,696,312	—	124,999,197	—	1,079	3,328
	B	393,822	—	2,894	15,328	99,532	62,453,063	566,722	—	1,691,297	—	81,698,317	—	719	14,454
Real datasets	Neur.	167,226	—	806	11,956	26,627	3,913,531	216,513	—	402,451	—	6,689,207	—	486	10,577
	Leuk.	235,006	—	864	16,800	27,106	5,826,109	203,916	—	489,828	—	25,258,894	—	542	15,381

As we can observe, MMVar was always faster than all competing algorithms. In particular, with the exception of CKM, each competing algorithm was at least one order of magnitude slower than MMVar. Apart CKM, the fastest method (on average) among the competitors revealed to be $\mathcal{F}$DB, which was 1–2 orders slower than MMVar. $\mathcal{F}$DB was in general comparable to or faster than the other density-based algorithm $\mathcal{F}$OPT (which was, in turn, 1–3 orders slower than MMVar), even if the computational complexity of the former (in the worst case) is greater than that of $\mathcal{F}$OPT (cf. Table I); this indicates that the procedure employed by $\mathcal{F}$DB for pruning unnecessary distance calculations performed pretty well on the selected datasets. The partitional UKM was always 2 orders slower than our MMVar; moreover, with respect to the density-based algorithms, it was comparable to or slightly faster than $\mathcal{F}$OPT, and comparable to, one order slower than, or even slightly faster than $\mathcal{F}$DB. UAHC and UKmed were in general the slowest methods (up to 5 and 4 orders slower than MMVar, respectively); this was expected due to the intrinsic complexity of hierarchical approaches (UAHC) and the (slow) offline computation of expected distances between every pair of uncertain objects

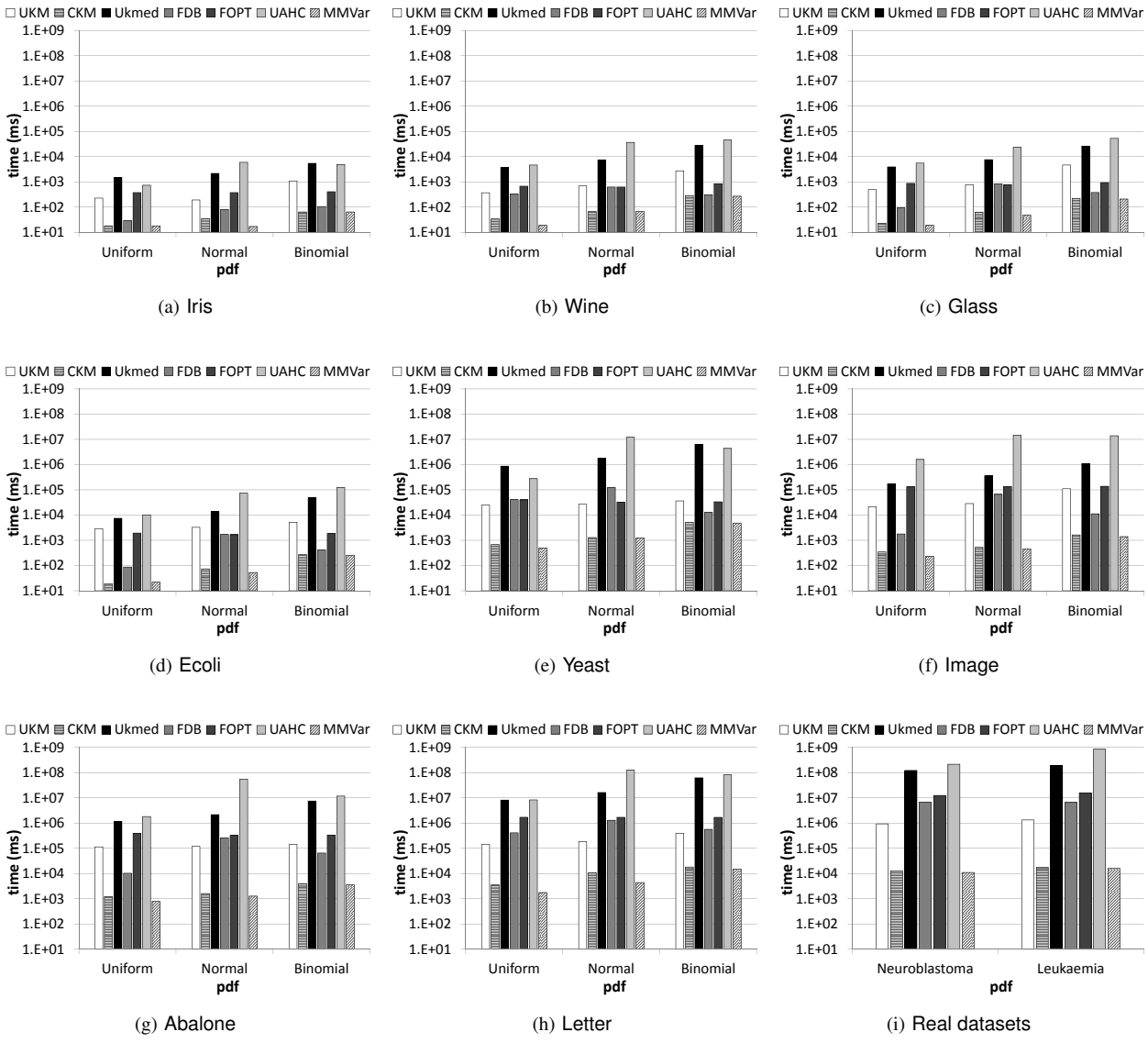

Figure 4. Efficiency results: all methods

(UKmed). As a general remark, the relative performances of the various algorithms were not affected by the form of pdf (for benchmark datasets). However, we noted that the times on Uniform were mostly lower than those on Normal and Binomial, which might be explained by the lower complexity in performing pdf sampling for Uniform than the other pdfs.

Figure 5 (times in linear scale) highlights a comparison between MMVar and CKM, which was the fastest competing method. All plots in the figure show that, though of the same order of magnitude, MMVar was always faster than CKM, which has already been recognized as much less accurate (cf. Sects. V-B1 and V-B2). In general, we note that greater improvements in the relative performance of MMVar with respect to CKM corresponded to larger datasets; this fact emphasizes the high efficiency of the proposed MMVar, particularly on large datasets. We also involved into this comparison the "fast" version of MMVar (denoted as *F-MMVar*), which exploits well-known closed-form expressions to fastly compute the moments of the distributions (i.e., without resorting to any pdf sampling); note that it makes sense to report results achieved by F-MMVar, since the moments of distributions can be efficiently computed according to closed-form formulas in most real cases (cf. Sect. IV). MMVar's performance gain with respect to CKM was even more evident by using the fast version F-MMVar; for most of the benchmark datasets, performance achieved by F-MMVar corresponded to only a few milliseconds regardless of the pdf form.

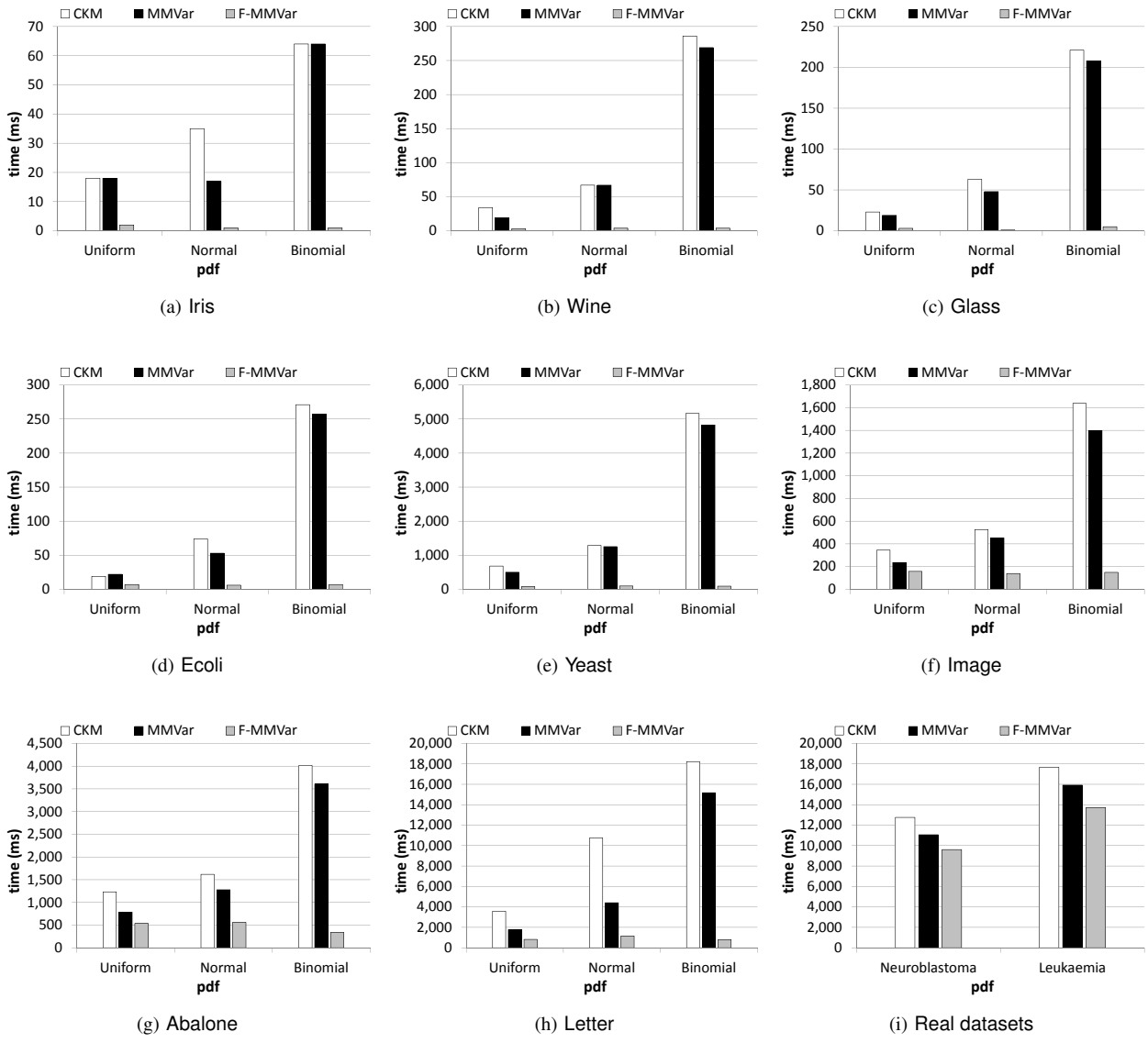

Figure 5. Efficiency results: CK-means vs. MMVar (and its fast version)

4) Scalability: We finally carried out a scalability study using the KDD Cup '99 dataset.[4] Figure 6 summarizes results obtained by MMVar and CKM, for which we varied the dataset size from 5% to 100% (increment of 5%). Note that, for each selected subset of the collection, we ensured that all 23 classes were covered by the objects within the subset. Thus, the number of clusters was conveniently fixed to 23 for all the algorithms under consideration.

As it can be noted in the figure, MMVar running times increased linearly with the dataset size. Moreover, MMVar exhibited an overall trend which increased more slowly than the linear trend shown by CKM: for instance, the ratio of the MMVar running time to the CKM running time was 0.6 at 50% of the dataset, and 0.76 at 100%. This confirmed the superiority of MMVar with respect to CKM also in terms of scalability.

[4]For this study, we carried out the algorithms on a CentOS 5.5 platform, with Linux 2.6.18 kernel, 64GB memory, 4 Intel(R) Xeon(R) CPU E7330, 2.40GHz quadcore

VI. Conclusion

We addressed the problem of clustering uncertain objects by focusing on the minimization of the variance of the mixture models that represent the clusters to be discovered. The rationale of the proposed approach is twofold: on the one hand, it allows for effectively recognizing clusters of uncertain objects, as the variance of the mixture model of a set of uncertain objects is inversely proportional to the compactness of that set; on the other hand, computing

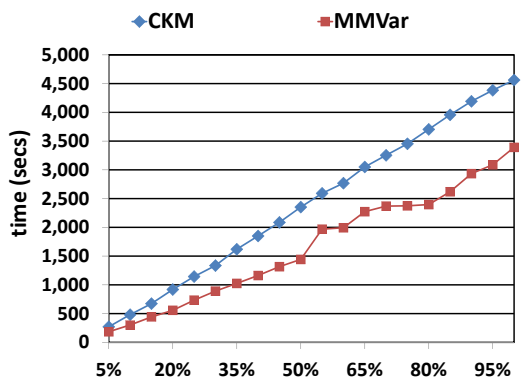

Figure 6. Scalability on the KDD Cup '99 dataset

the variance of mixture models can be carried out in a very efficient way by exploiting some analytical properties. This led us to the development of a fast heuristic, MMVar, to compute local optima of the objective function at the basis of the proposed formulation, which does not require any distance measure between uncertain objects. Based on experiments conducted on various benchmark and real datasets, MMVar turned out to be faster than prominent state-of-the-art algorithms for clustering uncertain objects, while achieving better average accuracy in terms of both external and internal cluster validity criteria.

As previously stated, this paper extends our earlier work [16], presenting theoretical findings for all the key notions underlying the proposed approach and performing a thorough experimentation to assess effectiveness, efficiency and scalability of MMVar, including comparison with state-of-the-art methods for clustering uncertain objects. The MMVar approach has the merit of taking into account the variance of the uncertain objects to model a cluster prototype, which brings the important benefit of avoiding the computation of uncertain object distances, and hence represents a notable advantage of higher accuracy with respect to existing distance-dependent centroid-based clustering methods for uncertain objects. We believe however that a purely variance-based notion of uncertain cluster prototype can still be inadequate to correctly summarize all cluster structures. Therefore, one direction of research could be identified in developing uncertain cluster prototypes in which central tendency and variability of the clustered objects are both taken into account in the definition of the cluster mixture models, while still avoiding to compute costly uncertain object-to-prototype distances.

REFERENCES

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

[2] 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.

[3] C. C. Aggarwal and P. S. Yu. A survey of uncertain data algorithms and applications. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 21(5):609–623, 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] A. Asuncion and D. Newman. Uci machine learning repository, http://archive.ics.uci.edu/ml/.

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

[7] M. Chau, R. Cheng, B. Kao, and J. Ng. Uncertain data mining: An example in clustering location data. In *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, pages 199–204, 2006.

[8] G. Cormode and A. McGregor. Approximation algorithms for clustering uncertain data. In *Proc. ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems (PODS)*, pages 191–200, 2008.

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

[10] X. L. Dong, A. Y. Halevy, and C. Yu. Data integration with uncertainty. *The VLDB Journal*, 18(2):469–500, 2009.

[11] A. Faradjian, J. Gehrke, and P. Bonnet. Gadt: A probability space adt for representing and querying the physical world. In *Proc. IEEE Int. Conf. on Data Engineering (ICDE)*, pages 201–211, 2002.

[12] J. Galindo, A. Urrutia, and M. Piattini. *Fuzzy Databases: Modeling, Design, and Implementation*. Idea Group Publishing, 2006.

[13] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007.

[14] S. Guha and K. Munagala. Exceeding expectations and clustering uncertain data. In *Proc. ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems (PODS)*, pages 269–278, 2009.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[21] B. Kao, S. D. Lee, D. W. Cheung, W.-S. Ho, and K. F. Chan. Clustering uncertain data using voronoi diagrams. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 333–342, 2008.

[22] 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.

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

[24] S. K. Lee. An extended relational database model for uncertain and imprecise information. In *Proc. Int. Conf. on Very Large Data Bases (VLDB)*, pages 211–220, 1992.

[25] 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.

[26] M. Milo, A. Fazeli, M. Niranjan, and N. D. Lawrence. A probabilistic model for the extraction of expression levels from oligonucleotide arrays. *Biochemical Society Transactions*, 31:1510–1512, 2003.

[27] W. K. Ngai, B. Kao, C. K. Chui, R. Cheng, M. Chau, and K. Y. Yip. Efficient clustering of uncertain data. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 436–445, 2006.

[28] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.

[29] J. Sander, X. Qin, Z. Lu, N. Niu, and A. Kovarsky. Automatic extraction of clusters from hierarchical clustering representations. In *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, pages 75–87, 2003.

[30] 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.

[31] M. Schena, D. Shalon, R. Davis, and P. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270:467–470, 1995.

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

[33] 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.

[34] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[35] 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.

APPENDIX

Fact 1: The uncertain prototype $\mathcal{P}_C = (\mathcal{R}_C, f_C)$ of any set C of uncertain objects is a multivariate uncertain object satisfying Def. 1.

Proof: Being C a set of multivariate uncertain objects, to prove that $\mathcal{P}_C$ is a multivariate uncertain object, we need to demonstrate that:

1) f_C is a pdf,
2) (1) of Def. 1 holds, and
3) (2) of Def. 1 holds.

Condition 1) is true since f_C represents a mixture of pdfs of the form $\sum_{o=(\mathcal{R},f)\in C} \alpha_i f(\vec{x})$, where $\alpha_i = 1/n$. As concerns condition 2), it results that:

$$\begin{aligned}\int\limits_{\vec{x}\in\Re^m\setminus\mathcal{R}_C} f_C(\vec{x})\mathrm{d}\vec{x} &= \int\limits_{\vec{x}\in\Re^m\setminus\mathcal{R}_C} \frac{1}{|C|}\sum_{o=(\mathcal{R},f)\in C} f(\vec{x})\mathrm{d}\vec{x} = \\ &= \frac{1}{|C|}\sum_{o=(\mathcal{R},f)\in C}\int\limits_{\vec{x}\in\Re^m\setminus\mathcal{R}_C} f(\vec{x})\mathrm{d}\vec{x}\end{aligned}$$

As $\mathcal{R}\subseteq\mathcal{R}_C$, $\forall o=(\mathcal{R},f)\in C$ (cf. (11)), it holds that:

$$\begin{aligned}&\frac{1}{|C|}\sum_{o=(\mathcal{R},f)\in C}\int\limits_{\vec{x}\in\Re^m\setminus\mathcal{R}_C} f(\vec{x})\mathrm{d}\vec{x} = \\ &= \frac{1}{|C|}\sum_{o=(\mathcal{R},f)\in C}\left(\int\limits_{\vec{x}\in\Re^m\setminus\mathcal{R}} f(\vec{x})\mathrm{d}\vec{x} - \int\limits_{\vec{x}\in\mathcal{R}_C\setminus\mathcal{R}} f(\vec{x})\mathrm{d}\vec{x}\right)\end{aligned}$$

which is equal to

$$\frac{1}{|C|}\sum_{o=(\mathcal{R},f)\in C}(0-0)=0$$

as, according to (1) of Def. 1, it holds that

$$f(\vec{x})=0,\ \forall\vec{x}\in\Re^m\setminus\mathcal{R} \quad\Rightarrow \int\limits_{\vec{x}\in\hat{\mathcal{R}}\setminus\mathcal{R}} f(\vec{x})\mathrm{d}\vec{x}=0,\ \forall\hat{\mathcal{R}}\subseteq\Re^m$$

To prove condition 3), it can be straightforwardly noted that, $f(\vec{x})>0$, $\forall\vec{x}\in\mathcal{R}$ (cf. (2)), and $f(\vec{x})\geq 0$, $\forall\vec{x}\in\Re^m, \forall o=(\mathcal{R},f)\in C$ (as each f is a pdf) clearly imply that

$$\sum_{o=(\mathcal{R},f)\in C} f(\vec{x})>0, \qquad \forall\vec{x}\in\bigcup_{o=(\mathcal{R},f)\in C}\mathcal{R}$$

accordingly, as $\mathcal{R}_C=\bigcup_{o=(\mathcal{R},f)\in C}\mathcal{R}_i$ (cf. (11)) and $|C|>0$, the following holds:

$$\frac{1}{|C|}\sum_{o=(\mathcal{R},f)\in C} f(\vec{x}) = f_C(\vec{x})>0,\ \forall\vec{x}\in\mathcal{R}_C$$

which proves condition 3). □

Lemma 1: Let C be a set of uncertain objects, where each $o\in C$ is a pair $(\mathcal{R},f)$, and $\mathcal{P}_C$ be the uncertain prototype of C. The expected value $\vec{\mu}(\mathcal{P}_C)$ and the second order moment $\vec{\mu}_2(\mathcal{P}_C)$ of prototype $\mathcal{P}_C$ are as follows:

$$\vec{\mu}(\mathcal{P}_C)=\frac{1}{|C|}\sum_{o\in C}\vec{\mu}(o)$$

$$\vec{\mu}_2(\mathcal{P}_C)=\frac{1}{|C|}\sum_{o\in C}\vec{\mu}_2(o)$$

Proof: According to (12) it holds that $\mathcal{P}_C=(\mathcal{R}_C,f_C)$, where $f_C(\vec{x})=|C|^{-1}\sum_{o\in C} f(\vec{x})$; thus:

$$\begin{aligned}\vec{\mu}(\mathcal{P}_C) &= \int\limits_{\vec{x}\in\mathcal{R}_C}\vec{x}\frac{1}{|C|}\sum_{o\in C} f(\vec{x})\mathrm{d}\vec{x} = \\ &= \frac{1}{|C|}\sum_{o\in C}\int\limits_{\vec{x}\in\mathcal{R}_C}\vec{x}f(\vec{x})\mathrm{d}\vec{x} = \\ &= \frac{1}{|C|}\sum_{o\in C}\left(\int\limits_{\vec{x}\in\mathcal{R}_C\setminus\mathcal{R}}\vec{x}f(\vec{x})\mathrm{d}\vec{x}+\int\limits_{\vec{x}\in\mathcal{R}}\vec{x}f(\vec{x})\mathrm{d}\vec{x}\right) \quad (15)\end{aligned}$$

As $f(\vec{x})=0$, $\forall\vec{x}\in\Re^m\setminus\mathcal{R}$ for any uncertain object $o=(\mathcal{R},f)$ (cf. Def. 1), it holds that $\int_{\vec{x}\in\mathcal{R}_C\setminus\mathcal{R}}\vec{x}\ f(\vec{x})\ \mathrm{d}\vec{x}=0$; therefore, considering again (15), it results that:

$$\vec{\mu}(\mathcal{P}_C) = \frac{1}{|C|}\sum_{o\in C}\int\limits_{\vec{x}\in\mathcal{R}}\vec{x}\ f(\vec{x})\ \mathrm{d}\vec{x} = \frac{1}{|C|}\sum_{o\in C}\vec{\mu}(o)$$

which proves the first part of the lemma. The second part $\vec{\mu}_2(\mathcal{P}_C)=|C|^{-1}\sum_{o\in C}\vec{\mu}_2(o)$ may be proved following an analogous reasoning. □

Lemma 2: Let $\mathcal{P}_C$ be the uncertain prototype of any set C of uncertain objects, C' (resp. C'') be the set defined by deleting (resp. adding) object o' (resp. o'') from (resp. to) C, i.e., $C'=C\setminus\{o'\}$, $C''=C\cup\{o''\}$. The expected values $\vec{\mu}(\mathcal{P}_{C'})$, $\vec{\mu}(\mathcal{P}_{C''})$ and second order moments $\vec{\mu}_2(\mathcal{P}_{C'})$, $\vec{\mu}_2(\mathcal{P}_{C''})$ of prototypes $\mathcal{P}_{C'}$, $\mathcal{P}_{C''}$ of sets C', C'' are as follows:

$$\vec{\mu}(\mathcal{P}_{C'})=\frac{|C|\times\vec{\mu}(\mathcal{P}_C)-\vec{\mu}(o')}{|C|-1}$$

$$\vec{\mu}_2(\mathcal{P}_{C'})=\frac{|C|\times\vec{\mu}_2(\mathcal{P}_C)-\vec{\mu}_2(o')}{|C|-1}$$

$$\vec{\mu}(\mathcal{P}_{C''})=\frac{|C|\times\vec{\mu}(\mathcal{P}_C)+\vec{\mu}(o'')}{|C|+1}$$

$$\vec{\mu}_2(\mathcal{P}_{C''})=\frac{|C|\times\vec{\mu}_2(\mathcal{P}_C)+\vec{\mu}_2(o'')}{|C|+1}$$

Proof: According to Lemma 1, it holds that:

$$\begin{aligned}\vec{\mu}(\mathcal{P}_{C'}) &= \frac{1}{|C'|}\sum_{o\in C'}\vec{\mu}(o) = \\ &= \frac{1}{|C|-1}\left(\sum_{o\in C}\vec{\mu}(o)-\vec{\mu}(o')\right) = \\ &= \frac{|C|\times\vec{\mu}(\mathcal{P}_C)-\vec{\mu}(o')}{|C|-1}\end{aligned}$$

The remaining statements of the lemma may be proved similarly. □

Proposition 1: Let $\mathcal{D}$ be a set of m-dimensional uncertain objects, $\mathcal{C}$ be a partition of $\mathcal{D}$, $\mathcal{P}_C$ be the prototype of any cluster $C \in \mathcal{C}$, and $\vec{\mu}(\mathcal{P}_C)$, $\vec{\mu}_2(\mathcal{P}_C)$ and $\sigma^2(\mathcal{P}_C) = \|\vec{\mu}_2(\mathcal{P}_C) - \vec{\mu}^2(\mathcal{P}_C)\|_1$ the expected value, second order moment and variance of $\mathcal{P}_C$, respectively. Let us consider a new partition $\mathcal{C}'$ of $\mathcal{D}$ obtained from $\mathcal{C}$ by moving an object o from cluster $C \in \mathcal{C}$ to cluster $\widehat{C} \in \mathcal{C}$; it holds that the value $J_{C,\widehat{C}}(\mathcal{C})$ of the objective function J for the new partition $\mathcal{C}'$ is the following:

$$J_{C,\widehat{C}}(\mathcal{C}) = J(\mathcal{C}) - (\sigma^2(\mathcal{P}_C)+\sigma^2(\mathcal{P}_{\widehat{C}})) + (\sigma^2(\mathcal{P}_{C'})+\sigma^2(\mathcal{P}_{\widehat{C}'})) \quad (14)$$

where

$$C' = C \setminus \{o\} \qquad \widehat{C}' = \widehat{C} \cup \{o\}$$

$$\sigma^2(\mathcal{P}_{C'}) = \|\vec{\mu}_2(\mathcal{P}_{C'}) - \vec{\mu}^2(\mathcal{P}_{C'})\|_1$$

$$\sigma^2(\mathcal{P}_{\widehat{C}'}) = \|\vec{\mu}_2(\mathcal{P}_{\widehat{C}'}) - \vec{\mu}^2(\mathcal{P}_{\widehat{C}'})\|_1$$

and

$$\vec{\mu}(\mathcal{P}_{C'}) = \frac{|C|\times\vec{\mu}(\mathcal{P}_C)-\vec{\mu}(o)}{|C|-1}$$

$$\vec{\mu}_2(\mathcal{P}_{C'}) = \frac{|C|\times\vec{\mu}_2(\mathcal{P}_C)-\vec{\mu}_2(o)}{|C|-1}$$

$$\vec{\mu}(\mathcal{P}_{\widehat{C}'}) = \frac{|\widehat{C}|\times\vec{\mu}(\mathcal{P}_{\widehat{C}})+\vec{\mu}(o)}{|\widehat{C}|+1}$$

$$\vec{\mu}_2(\mathcal{P}_{\widehat{C}'}) = \frac{|\widehat{C}|\times\vec{\mu}_2(\mathcal{P}_{\widehat{C}})+\vec{\mu}_2(o)}{|\widehat{C}|+1}$$

Proof: The $\vec{\mu}(\mathcal{P}_{C'})$, $\vec{\mu}_2(\mathcal{P}_{C'})$, $\vec{\mu}(\mathcal{P}_{\widehat{C}'})$ and $\vec{\mu}_2(\mathcal{P}_{\widehat{C}'})$ expressions follow directly from Lemma 2. This result along with (8) also implies the correctness of computing $\sigma^2(\mathcal{P}_{C'})$ and $\sigma^2(\mathcal{P}_{\widehat{C}'})$ as $\|\vec{\mu}_2(\mathcal{P}_{C'}) - \vec{\mu}^2(\mathcal{P}_{C'})\|_1$ and $\|\vec{\mu}_2(\mathcal{P}_{\widehat{C}'}) - \vec{\mu}^2(\mathcal{P}_{\widehat{C}'})\|_1$, respectively.

As $J(\mathcal{C})$ is defined in (13) as a sum of variances of the prototypes of clusters within $\mathcal{C}$, $J_{C,\widehat{C}}(\mathcal{C})$ can be computed by "replacing" the variances $\sigma^2(\mathcal{P}_C)$ and $\sigma^2(\mathcal{P}_{\widehat{C}})$ of the early clusters C, $\widehat{C}$ with the variances $\sigma^2(\mathcal{P}_{C'})$ and $\sigma^2(\mathcal{P}_{\widehat{C}'})$ of the new formed clusters C', $\widehat{C}'$. In this way, it holds that $J_{C,\widehat{C}}(\mathcal{C}) = J(\mathcal{C}) - (\sigma^2(\mathcal{P}_C) + \sigma^2(\mathcal{P}_{\widehat{C}})) + (\sigma^2(\mathcal{P}_{C'}) + \sigma^2(\mathcal{P}_{\widehat{C}'}))$, which proves the proposition. □

Proposition 2: The MMVar algorithm converges to a local minimum of function J defined in (13) in a finite number of steps.

Proof: Let us denote by $V^{(h)}$ the value $J(\mathcal{C}^{(h)})$, where $\mathcal{C}^{(h)}$ is the clustering computed, for all objects in $\mathcal{D}$, at the h-th iteration of MMVar. To prove the proposition, it is sufficient to show that $V^{(h)} \leq V^{(h-1)}$ at each iteration $h > 1$. Indeed, as the function J is bounded below (it is greater than or equal to zero), if the value of this function never increases at each iteration h, this would mean that the algorithm performs a descendent gradient over the function and necessarily terminates when a local minimum is reached.

At each iteration h, every object is processed and may in principle be responsible of the update of the current clustering $\mathcal{C}^{(h)}$. For this purpose, we denote by $\mathcal{C}_p^{(h)}$ the partition computed at the h-th iteration when p objects have been processed and by $V_p^{(h)}$ the corresponding value $J\left(\mathcal{C}_p^{(h)}\right)$ ($p \in [0..n]$). For each object o_p to be processed, it holds that the current clustering $\mathcal{C}_p^{(h)}$ is computed as $\mathcal{C}_p^{(h)} = \arg\min_{\mathcal{C}} J(\mathcal{C})$ over all $\mathcal{C}$ belonging to the set of all clusterings that can be obtained from $\mathcal{C}_{p-1}^{(h)}$ by moving the object o_p to any other possible cluster. In this respect, it holds that $V_p^{(h)}$ is smaller than or equal to any other value obtained by considering different clusterings built starting from $\mathcal{C}_{p-1}^{(h)}$, and, therefore, $V_p^{(h)} \leq J\left(\mathcal{C}_{p-1}^{(h)}\right) = V_{p-1}^{(h)}$, $\forall p, h$, i.e., $V_n^{(h)} \leq V_{n-1}^{(h)} \leq \cdots \leq V_1^{(h)} \leq V_0^{(h)}$, $\forall h$. As $V_n^{(h)} = V^{(h)}$ and $V_0^{(h)} = V_n^{(h-1)} = V^{(h-1)}$, it follows that $V^{(h)} \leq V^{(h-1)}$, $\forall h$, which proves the proposition. □

Proposition 3: Given a set $\mathcal{D}$ of n m-dimensional uncertain objects, the number k of output clusters, the number S of statistical samples used for representing the uncertain objects, and denoting by I the number of iterations to convergence, the computational complexity of the MMVar algorithm (Alg. 1) is $\mathcal{O}(n\ m\ (I\ k + S))$.

Proof: In the "offline" phase of the algorithm, the computation of the $\vec{\mu}$ and $\vec{\mu}_2$ values for all objects within $\mathcal{D}$ (Line 1) costs either $\mathcal{O}(n\ m)$ (if the pdfs of the uncertain objects are such that there exist closed-form expressions for computing moments), or $\mathcal{O}(S\ n\ m)$ (if the moments may be only approximated according to (9) by taking into account S samples from distributions).

Then, the k-way incremental initialization phase processes all objects one time, and, for each object, it looks at all clusters in order to find the best one given the assignments already computed. This leads to a $\mathcal{O}(k\ n\ m)$ time complexity for the entire initialization.

As far as the refinement phase, each iteration of the main cycle has the following computational cost. Computing expected value and second order moment of any cluster prototype according to Lemma 1 or Lemma 2 is $\mathcal{O}(m)$.

Once all objects have been assigned to k clusters, computing cluster C^* (Line 18) takes $\mathcal{O}(k\ m)$, since (14) needs to be evaluated for each cluster. All operations to be performed in case an object o is moved to C^* (Lines 20-22) take $\mathcal{O}(m)$, since their cost is dominated by that of computing $\vec{\mu}(\mathcal{P}_C)$, $\vec{\mu}_2(\mathcal{P}_C)$, $\vec{\mu}(\mathcal{P}_{C^*})$, $\vec{\mu}_2(\mathcal{P}_{C^*})$ according to Lemma 2. As the above operations are repeated for all objects in $\mathcal{D}$ until convergence, each iteration of the main cycle globally costs $\mathcal{O}(k\ n\ m)$. This leads to an overall complexity of $\mathcal{O}(n\ m\ (I\ k + S))$. □