Clustering Uncertain Data Based on Probability Distribution Similarity

Bin Jiang, Jian Pei, *Senior Member, IEEE*,
Yufei Tao, *Member, IEEE*, and Xuemin Lin, *Senior Member, IEEE*

Abstract—Clustering on uncertain data, one of the essential tasks in mining uncertain data, posts significant challenges on both modeling similarity between uncertain objects and developing efficient computational methods. The previous methods extend traditional partitioning clustering methods like k-means and density-based clustering methods like DBSCAN to uncertain data, thus rely on geometric distances between objects. Such methods cannot handle uncertain objects that are geometrically indistinguishable, such as products with the same mean but very different variances in customer ratings. Surprisingly, probability distributions, which are essential characteristics of uncertain objects, have not been considered in measuring similarity between uncertain objects. In this paper, we systematically model uncertain objects in both continuous and discrete domains, where an uncertain object is modeled as a continuous and discrete random variable, respectively. We use the well-known Kullback-Leibler divergence to measure similarity between uncertain objects in both the continuous and discrete cases, and integrate it into partitioning and density-based clustering methods to cluster uncertain objects. Nevertheless, a naïve implementation is very costly. Particularly, computing exact KL divergence in the continuous case is very costly or even infeasible. To tackle the problem, we estimate KL divergence in the continuous case by kernel density estimation and employ the fast Gauss transform technique to further speed up the computation. Our extensive experiment results verify the effectiveness, efficiency, and scalability of our approaches.

Index Terms—Clustering, uncertain data, probabilistic distribution, density estimation, fast Gauss transform

1 Introduction

Clustering uncertain data has been well recognized as an important issue [21], [22], [27], [36]. Generally, an uncertain data object can be represented by a probability distribution [7], [29], [35]. The problem of clustering uncertain objects according to their probability distributions happens in many scenarios.

For example, in marketing research, users are asked to evaluate digital cameras by scoring on various aspects, such as image quality, battery performance, shotting performance, and user friendliness. Each camera may be scored by many users. Thus, the user satisfaction to a camera can be modeled as an uncertain object on the user score space. There are often a good number of cameras under a user study. A frequent analysis task is to cluster the digital cameras under study according to user satisfaction data.

One challenge in this clustering task is that we need to consider the similarity between cameras not only in terms of their score values, but also their score distributions. One camera receiving high scores is different from one receiving low scores. At the same time, two cameras, though with the same mean score, are substantially different if their score variances are very different.

As another example, a weather station monitors weather conditions including various measurements like temperature, precipitation amount, humidity, wind speed, and direction. The daily weather record varies from day to day, which can be modeled as an uncertain object represented by a distribution over the space formed by several measurements. Can we group the weather conditions during the last month for stations in North America? Essentially, we need to cluster the uncertain objects according to their distributions.

1.1 Limitations of Existing Work

The previous studies on clustering uncertain data are largely various extensions of the traditional clustering algorithms designed for certain data. As an object in a certain data set is a single point, the distribution regarding the object itself is not considered in traditional clustering algorithms. Thus, the studies that extended traditional algorithms to cluster uncertain data are limited to using geometric distance-based similarity measures, and cannot capture the difference between uncertain objects with different distributions.

Specifically, three principal categories exist in literature, namely partitioning clustering approaches [27], [18], [24], density-based clustering approaches [21], [22], and possible world approaches [36]. The first two are along the line of the categorization of clustering methods for certain data [14], the possible world approaches are specific for uncertain data following the popular possible world semantics for uncertain data [8], [31], [17].

- *B. Jiang and J. Pei are with the School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, BC V5A 1S6, Canada. E-mail: {bjiang, jpei}@cs.sfu.ca.*
- *Y. Tao is with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Sha Tin, New Territories, Hong Kong SAR, China. E-mail: taoyf@cse.cuhk.edu.hk.*
- *X. Lin is with the School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia, and the School of Software, East China Normal University, China. E-mail: lxue@cse.unsw.edu.au.*

Manuscript received 11 June 2010; revised 10 Sept. 2011; accepted 27 Sept. 2011; published online 14 Oct. 2011.
Recommended for acceptance by C. Jermaine.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2010-06-0328.
Digital Object Identifier no. 10.1109/TKDE.2011.221.

1041-4347/13/$31.00 © 2013 IEEE Published by the IEEE Computer Society

As these approaches only explore the geometric properties of data objects and focus on instances of uncertain objects, they do not consider the similarity between uncertain objects in terms of distributions.

Let us examine this problem in the three existing categories of approaches in detail. Suppose we have two sets $\mathbb{A}$ and $\mathbb{B}$ of uncertain objects. The objects in $\mathbb{A}$ follow uniform distribution, and those in $\mathbb{B}$ follow Gaussian distribution. Suppose all objects in both sets have the same mean value (i.e., the same center). Consequently, their geometric locations (i.e., areas that they occupied) heavily overlap. Clearly, the two sets of objects form two clusters due to their different distributions.

Partitioning clustering approaches [27], [18], [24] extend the k-means method with the use of the expected distance to measure the similarity between two uncertain objects. The expected distance between an object P and a cluster center c (which is a certain point) is $ED(P, c) = \int_P f_P(x) dist(x, c) \mathrm{d}x$, where f_P denotes the probability density function of P and the distance measure $dist$ is the square of euclidean distance. In [24], it is proved that $ED(P, c)$ is equal to the $dist$ between the center (i.e., the mean) $P.c$ of P and c plus the variance of P. That is,

$$ED(P, c) = dist(P.c, c) + Var(P). \qquad (1)$$

Accordingly, P can be assigned to the cluster center $\text{argmin}_c\{ED(P, c)\} = \text{argmin}_c\{dist(P.c, c)\}$. Thus, only the centers of objects are taken into account in these uncertain versions of the k-means method. In our case, as every object has the same center, the expected distance-based approaches cannot distinguish the two sets of objects having different distributions.

Density-based clustering approaches [21], [22] extend the DBSCAN method [10] and the OPTICS method [3] in a probabilistic way. The basic idea behind the algorithms does not change—objects in geometrically dense regions are grouped together as clusters and clusters are separated by sparse regions. However, in our case, objects heavily overlap. There is no clear sparse regions to separate objects into clusters. Therefore, the density-based approaches cannot work well.

Possible world approaches [36] follow the possible world semantics [8], [31], [17]. A set of possible worlds are sampled from an uncertain data set. Each possible world consists of an instance from each object. Clustering is conducted individually on each possible world and the final clustering is obtained by aggregating the clustering results on all possible worlds into a single global clustering. The goal is to minimize the sum of the difference between the global clustering and the clustering of every possible world. Clearly, a sampled possible world does not consider the distribution of a data object since a possible world only contains one instance from each object. The clustering results from different possible worlds can be drastically different. The most probable clusters calculated using possible worlds may still carry a very low probability. Thus, the possible world approaches often cannot provide a stable and meaningful clustering result at the object level, not to mention that it is computationally infeasible due to the exponential number of possible worlds.

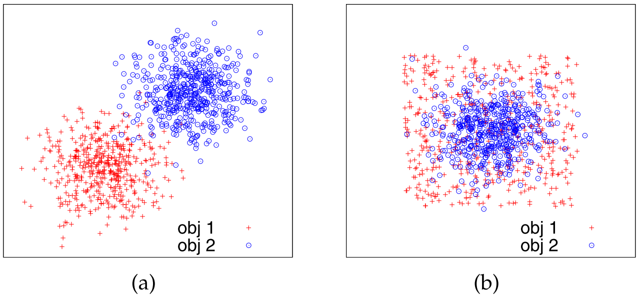


Fig. 1. Probability distributions and geometric locations.

1.2 Our Ideas and Contributions

Can we cluster uncertain objects according to the similarity between their probability distributions? Similarity measurement between two probability distributions is not a new problem at all. In information theory, the similarity between two distributions can be measured by the **Kullback-Leibler divergence** (**KL divergence** for short, also known as **information entropy** or **relative entropy**) [23].

The distribution difference cannot be captured by geometric distances. For example, in Fig. 1a, the two objects (each one is represented by a set of sampled points) have different geometric locations. Their probability density functions over the entire data space are different and the difference can be captured by KL divergence. In Fig. 1b, although the geometric locations of the two objects are heavily overlapping, they have different distributions (one is uniform and the other is Gaussian). The difference between their distributions can also be discovered by KL divergence, but cannot be captured by the existing methods as elaborated in Section 1.1.

In this paper, we consider uncertain objects as random variables with certain distributions. We consider both the discrete case and the continuous case. In the discrete case, the domain has a finite number of values, for example, the rating of a camera can only take a value in $\{1, 2, 3, 4, 5\}$. In the continuous case, the domain is a continuous range of values, for example, the temperatures recorded in a weather station are continuous real numbers.

Directly computing KL divergence between probability distributions can be very costly or even infeasible if the distributions are complex, as will be shown in Section 3. Although KL divergence is meaningful, a significant challenge of clustering using KL divergence is how to evaluate KL divergence efficiently on many uncertain objects.

To the best of our knowledge, this paper is the first to study clustering uncertain data objects using KL divergence in a general setting. We make several contributions. We develop a general framework of clustering uncertain objects considering the distribution as the first class citizen in both discrete and continuous cases. Uncertain objects can have any discrete or continuous distribution. We show that distribution differences cannot be captured by the previous methods based on geometric distances. We use KL divergence to measure the similarity between distributions, and demonstrate the effectiveness of KL divergence in both partitioning and density-based clustering methods. To tackle the challenge of evaluating the KL divergence in the continuous case, we estimate KL divergence by kernel density estimation and apply the fast Gauss transform to boost the computation. We conducted experiments on real and synthetic data sets to show clustering uncertain data in probability distribution is meaningful and our methods are efficient and scalable.

The rest of the paper is organized as follows. Section 2 reviews the related work. In Section 3, we define uncertain objects and the similarity using KL divergence, and show how to evaluate KL divergence in both discrete and continuous cases. In Section 4, we present the partitioning and density-based clustering methods using KL divergence. In Section 5, we develop implementation techniques to speed up the clustering and introduce the fast Gauss transform to boost the evaluation of KL divergence. We conduct an extensive empirical study in Section 6, and conclude the paper in Section 7.

2 Related Work

Clustering is a fundamental data mining task. Clustering certain data has been studied for years in data mining, machine learning, pattern recognition, bioinformatics, and some other fields [14], [16], [19]. However, there is only preliminary research on clustering uncertain data.

Data uncertainty brings new challenges to clustering, since clustering uncertain data demands a measurement of similarity between uncertain data objects. Most studies of clustering uncertain data used geometric distance-based similarity measures, which are reviewed in Section 2.1. A few theoretical studies considered using divergences to measure the similarity between objects. We discuss them in Section 2.2.

2.1 Clustering Based on Geometric Distances

Ngai et al. [27] proposed the UK-means method which extends the k-means method. The UK-means method measures the distance between an uncertain object and the cluster center (which is a certain point) by the expected distance. Recently, Lee et al. [24] showed that the UK-means method can be reduced to the k-means method on certain data points due to (1) described in Section 1.

Kriegel and Pfeifle [21] proposed the FDBSCAN algorithm which is a probabilistic extension of the deterministic DBSCAN algorithm [10] for clustering certain data. As DBSCAN is extended to a hierarchical density-based clustering method referred to as OPTICS [3], Kriegel and Pfeifle [22] developed a probabilistic version of OPTICS called FOPTICS for clustering uncertain data objects. FOPTICS outputs a hierarchical order in which data objects, instead of the determined clustering membership for each object, are clustered.

Volk et al. [36] followed the possible world semantics [1], [15], [8], [31] using Monte Carlo sampling [17]. This approach finds the clustering of a set of sampled possible worlds using existing clustering algorithms for certain data. Then, the final clustering is aggregated from those sample clusterings.

As discussed in Section 1, the existing techniques on clustering uncertain data mainly focus on the geometric characteristics of objects, and do not take into account the probability distributions of objects. In this paper, we propose to use KL divergence as the similarity measure which can capture distribution difference between objects. To the best of our knowledge, this paper is the first work to study clustering uncertain objects using KL divergence.

2.2 Clustering Based on Distribution Similarity

We are aware that clustering distributions has appeared in the area of information retrieval when clustering documents [37], [5]. The major difference of our work is that we do not assume any knowledge on the types of distributions of uncertain objects. When clustering documents, each document is modeled as a multinomial distribution in the language model [30], [34]. For example, Xu and Croft [37] discussed a k-means clustering method with KL divergence as the similarity measurement between multinomial distributions of documents. Assuming multinomial distributions, KL divergence can be computed using the number of occurrences of terms in documents. Blei et al. [5] proposed a generative model approach—the Latent Dirichlet Allocation (LDA for short). LDA models each document and each topic (i.e., cluster) as a multinomial distribution, where a document is generated by several topics. However, such multinomial distribution-based methods cannot be applied to general cases where the type of distributions are not multinomial.

There are also a few studies on clustering using KL divergences.

Dhillon et al. [9] used KL divergence to measure similarity between words to cluster words in documents in order to reduce the number of features in document classification. They developed a k-means like clustering algorithm and showed that the algorithm monotonically decreases the objective function as shown in [9], and minimizes the intracluster Jensen-Shannon divergence while maximizing intercluster Jensen-Shannon divergence. As their application is on text data, each word is a discrete random variable in the space of documents. Therefore, it is corresponding to the discrete case in our problem.

Banerjee et al. [4] theoretically analyzed the k-means like iterative relocation clustering algorithms based on Bregman divergences which is a general case of KL divergence. They summarized a generalized iterative relocation clustering framework for various similarity measures from the previous work from an information theoretical viewpoint. They showed that finding the optimal clustering is equivalent to minimizing the loss function in Bregman information corresponding to the selected Bregman divergence used as the underlying similarity measure. In terms of efficiency, their algorithms have linear complexity in each iteration with respect to the number of objects. However, they did not provide methods for efficiently evaluating Bregman divergence nor calculating the mean of a set of distributions in a cluster. For uncertain objects in our problem which can have arbitrary discrete or continuous distributions, it is essential to solve the two problems in order to scale on large data sets, as we can see in our experiments.

Ackermann et al. [2] developed a probabilistic $(1+\epsilon)$-approximation algorithm with linear time complexity for the k-medoids problem with respect to an arbitrary similarity measure, such as squared euclidean distance, KL divergence, Mahalanobis distance, etc., if the similarity measure allows the 1-medoid problem being approximated within a factor of $(1+\epsilon)$ by solving it exactly on a random sample of constant size. They were motivated by the problem of compressing Java and C++ executable codes which are modeled based on a large number of probability distributions. They solved the problem by identifying a good set of representatives for these distributions to achieve compression which involves nonmetric similarity measures like KL divergence. The major contribution of their work is on developing a probabilistic approximation algorithm for the k-medoids problem.

The previous theoretical studies focused on the correctness of clustering using KL divergence [9], the correspondence of clustering using Bregman divergence in information theory [4], and the probabilistic approximation algorithm [2]. However, they did not provide methods for efficiently evaluating KL divergence in the clustering process, neither did they experimentally test the efficiency and scalability of their methods on large data sets. Different to them, our work aims at introducing distribution differences especially KL divergence as a similarity measure for clustering uncertain data. We integrate KL divergence into the framework of k-medoids and DBSCAN to demonstrate the performance of clustering uncertain data. More importantly, particular to the uncertain objects in our problem, we focus on efficient computation techniques for large data sets, and demonstrate the effectiveness, efficiency, and scalability of our methods on both synthetic and real data sets with thousands of objects, each of which has a sample of hundreds of observations.

3 Uncertain Objects and KL Divergence

This section first models uncertain objects as random variables in probability distributions. We consider both the discrete and continuous probability distributions and show the evaluation of the corresponding probability mass and density functions in the discrete and continuous cases, respectively. Then, we recall the definition of KL divergence, and formalize the distribution similarity between two uncertain objects using KL divergence.

3.1 Uncertain Objects and Probability Distributions

We consider an uncertain object as a random variable following a probability distribution in a domain $\mathbb{D}$. We consider both the discrete and continuous cases.

If the domain is discrete (e.g., categorical) with a finite or countably infinite number of values, the object is a *discrete random variable* and its probability distribution is described by a *probability mass function* (pmf for short). Otherwise, if the domain is continuous with a continuous range of values, the object is a *continuous random variable* and its probability distribution is described by a *probability density function* (pdf for short). For example, the domain of the ratings of cameras is a discrete set $\{1, 2, 3, 4, 5\}$, and the domain of temperature is continuous real numbers.

In many case, the accurate probability distributions of uncertain objects are not known beforehand in practice. Instead, the probability distribution of an uncertain object is often derived from our observations of the corresponding random variable. Therefore, we associate each object with a sample of observations, and assume that the sample is finite and the observations are independent and identically distributed (i.i.d. for short).

By overloading the notation, for an uncertain object P, we still use P to denote the corresponding random variable, the probability mass/density function, and the sample.

For discrete domains, the probability mass function of an uncertain object can be directly estimated by normalizing the number of observations against the size of the sample. Formally, the pmf of object P is

$$P(x) = \frac{\left|\{p \in P | p = x\}\right|}{|P|}, \tag{2}$$

where $p \in P$ is an observation of P and $| \cdot |$ is the cardinality of a set.

For continuous domains, we estimate the probability density function of an uncertain object by kernel density estimation.

Kernel density estimation [33], [32] is a nonparametric way of estimating the probability density function of a continuous random variable. Given a sample of a continuous random variable P, the kernel density estimate of the probability density function is the sum of $|P|$ kernel functions. In this paper, we use the popular Gaussian kernels.

Each Gaussian kernel function is centered at a sample point $p \in P$ with variance h. h is called the bandwidth, and is used to control the level of smoothing. A popular choice of the bandwidth is the Sliverman approximation rule [33] for which $h = 1.06 \times \delta |P|^{-\frac{1}{5}}$, where δ is the standard deviation of the sample points of P. In 1D case, the density estimator is

$$P(x) = \frac{1}{|P|\sqrt{2\pi}h} \sum_{p \in P} e^{-\frac{(x-p)^2}{2h^2}}.$$

For the d-dimensional ($d \geq 2$) case, the kernel function is the product of d Gaussian functions, each with its own bandwidth h_j ($1 \leq j \leq d$), and the density estimator is

$$P(x) = \frac{1}{|P|(2\pi)^{d/2} \prod_{j=1}^{d} h_j} \sum_{p \in P} \prod_{j=1}^{d} e^{-\frac{(x.D_j - p.D_j)^2}{2h_j^2}}, \tag{3}$$

where we denote a d-dimensional point p by $(p.D_1, \ldots, p.D_d)$.

3.2 KL Divergence

In general, KL divergence between two probability distributions is defined as follows:

Definition 1 (Kullback-Leibler Divergence [23]). *In the discrete case, let f and g be two probability mass functions in a discrete domain $\mathbb{D}$ with a finite or countably infinite number of values. The Kullback-Leibler diverge between f and g is*

$$D(f \parallel g) = \sum_{x \in \mathbb{D}} f(x) \log \frac{f(x)}{g(x)}. \tag{4}$$

In the continuous case, let f and g be two probability density functions in a continuous domain $\mathbb{D}$ with a continuous range of values. The Kullback-Leibler divergence between f and g is

$$D(f \parallel g) = \int_{\mathbb{D}} f(x) \log \frac{f(x)}{g(x)} \mathrm{d}x. \tag{5}$$

In both discrete and continuous cases, KL divergence is defined only in the case where for any x in domain $\mathbb{D}$ if $f(x) > 0$ then $g(x) > 0$. By convention, $0 \log \frac{0}{p} = 0$ for any $p \neq 0$ and the base of $\log$ is 2.

Note that, KL divergence is not symmetric in general, that is, $D(f \| g) \neq D(g \| f)$.

3.3 Using KL Divergence as Similarity

It is natural to quantify the similarity between two uncertain objects by KL divergence. Given two uncertain objects P and Q and their corresponding probability distributions, $D(P \| Q)$ evaluates the relative uncertainty of Q given the distribution of P. In fact, from (4) and (5), we have

$$D(P \parallel Q) = \mathbf{E}\left[\log \frac{P}{Q}\right], \tag{6}$$

which is the expected log-likelihood ratio of the two distributions and tells how similar they are. The KL divergence is always nonnegative, and satisfies Gibbs' inequality. That is, $D(P \parallel Q) \geq 0$ with equality only if $P = Q$. Therefore, the smaller the KL divergence, the more similar the two uncertain objects.

In the discrete case, it is straightforward to evaluate (4) to calculate the KL divergence between two uncertain objects P and Q from their probability mass functions calculated as (2).

In the continuous case, given the samples of P and Q, by the law of large numbers and (6), we have

$$\lim_{s \to \infty} \frac{1}{s} \sum_{i=1}^{s} \log \frac{P(p_i)}{Q(p_i)} = D(P \parallel Q),$$

where we assume the sample of $P = \{p_1, \ldots, p_s\}$. Hence, we estimate the KL divergence $D(P \parallel Q)$ as

$$\hat{D}(P \parallel Q) = \frac{1}{s} \sum_{i=1}^{s} \log \frac{P(p_i)}{Q(p_i)}. \tag{7}$$

It is important to note that the definition of KL divergence necessitates that for any $x \in \mathbb{D}$ if $P(x) > 0$ then $Q(x) > 0$. To ensure that the KL divergence is defined between every pair of uncertain objects, we smooth the probability mass/density function of every uncertain object P so that it has a positive probability to take any possible value in the domain. In other words, we assume that there is always a small nonzero probability for an uncertain object to observe any unseen value. As in the camera rating example, if we only observe ratings 2, 3, 4, and 5 of a camera, but not 1, we still assume that the camera has a small probability to be rated as 1, even we have no such observations.

The smoothing is based on the following two assumptions about the uncertain objects to be clustered in our problem:

1. We assume that the probability distribution of every uncertain object to be clustered is defined in the same domain $\mathbb{D}$.
2. We assume that the domain $\mathbb{D}$ is bounded.

If $\mathbb{D}$ is discrete, we assume it has a finite number of values. For the continuous domain, we only consider $\mathbb{D}$ as a bounded range of values. In most applications, possible values are within a sufficiently large bounded range, as we can use the lowest and highest recorded values to define the range. For example, cameras can be rated at five possible grades, and temperatures on Earth's surface usually range between $-90°\text{C}$ to $60°\text{C}$.

We smooth a probability distribution P as follows:

$$\hat{P}(x) = \frac{P(x) + \delta}{1 + \delta|\mathbb{D}|}, \tag{8}$$

where $0 < \delta < 1$, $|\mathbb{D}|$ is the number of possible values in $\mathbb{D}$ if $\mathbb{D}$ is discrete and the area of $\mathbb{D}$ (i.e., $|\mathbb{D}| = \int_{\mathbb{D}} dx$) if $\mathbb{D}$ is continuous.

Clearly, the sum/integral of $\hat{P}(x)$ over the entire domain remains 1. For two uncertain objects P and Q, after smoothing, $\hat{P}(x) = \hat{Q}(x)$ for any $x \in \mathbb{D}$ still holds if and only if $P(x) = Q(x)$.

The error incurred by such an approximation is

$$\left|\hat{P}(x) - P(x)\right| = \left|\frac{1 - P(x)|\mathbb{D}|}{1/\delta + |\mathbb{D}|}\right| \in \left[0, \frac{\max\{1, \left|1 - |\mathbb{D}|\right|\}}{1/\delta + |\mathbb{D}|}\right].$$

As the domain $\mathbb{D}$ is bounded, $|\mathbb{D}|$ is bounded. By choosing a sufficient small δ, we can provide an approximation at arbitrary accuracy level.

In summary of this section, we model every uncertain object to be clustered as a random variable in the same bounded domain $\mathbb{D}$ and represent it with a sample of i.i.d. observations. The probability distribution of an uncertain object is estimated from its sample by (2) in the discrete case and (3) using kernel density estimation in the continuous case. In both cases, the probability mass/density function is smoothed by (8).

We define the similarity between two uncertain objects as the KL divergence between their probability distributions. The KL divergence is calculated by (4) and (7) in the discrete and continuous cases, respectively.

4 Clustering Algorithms

As the previous geometric distance-based clustering methods for uncertain data mainly fall into two categories, partitioning and density-based approaches, in this section, we present the clustering methods using KL divergence to cluster uncertain objects in these two categories. In Section 4.1, we present the uncertain k-medoids method which extends a popular partitioning clustering method k-medoids [19] by using KL divergence. Then, we develop a randomized k-medoids method based on the uncertain k-medoids method to reduce the time complexity. Section 4.2 presents the uncertain DBSCAN method which integrates KL divergence into the framework of a typical density-based clustering method DBSCAN [10]. We describe the algorithms of the methods and how they use KL divergence as the similarity measure.

4.1 Partitioning Clustering Methods

A partitioning clustering method organizes a set of n uncertain objects $\mathbb{O}$ into k clusters $\mathbb{C}_1, \ldots, \mathbb{C}_k$, such that $\mathbb{C}_i \subseteq \mathbb{O}$ $(1 \leq i \leq k)$, $\mathbb{C}_i \neq \emptyset$, $\bigcup_{i=1}^{k} \mathbb{C}_i = \mathbb{O}$, and $\mathbb{C}_i \cap \mathbb{C}_j = \emptyset$ for any $i \neq j$. We use C_i to denote the representative of cluster $\mathbb{C}_i$. Using KL divergence as similarity, a partitioning clustering method tries to partition objects into k clusters and chooses the best k representatives, one for each cluster, to minimize the total KL divergence as below

$$TKL = \sum_{i=1}^{k} \sum_{P \in \mathbb{C}_i} D(P \parallel C_i). \tag{9}$$

For an object P in cluster $\mathbb{C}_i$ ($1 \leq i \leq k$), the KL divergence $D(P \parallel C_i)$ between P and the representative C_i measures the extra information required to construct P given C_i. Therefore, $\sum_{P \in \mathbb{C}_i} D(P \parallel C_i)$ captures the total extra information required to construct the whole cluster $\mathbb{C}_i$ using its representative C_i. Summing over all k clusters, the total KL divergence thus measures the quality of the partitioning clustering. The smaller the value of TKL, the better the clustering.

k-means [25], [26] and k-medoids [19] are two classical partitioning methods. The difference is that the k-means method represents each cluster by the mean of all objects in this cluster, while the k-medoids method uses an actual object in a cluster as its representative. In the context of uncertain data where objects are probability distributions, it is inefficient to compute the mean of probability density functions. k-medoids method avoids computing the means. For the sake of efficiency, we adopt the k-medoids method to demonstrate the performance of partitioning clustering methods using KL divergence to cluster uncertain objects.

In Section 4.1.1, we first present the uncertain k-medoids method which integrates KL divergence into the original k-medoids method. Then, we develop a randomized k-medoids method in Section 4.1.2 to reduce the complexity of the uncertain one.

4.1.1 Uncertain K-Medoids Method

The uncertain k-medoids method consists of two phases, the building phase and the swapping phase.

Building phase. In the building phase, the uncertain k-medoids method obtains an initial clustering by selecting k representatives one after another. The first representative C_1 is the one which has the smallest sum of the KL divergence to all other objects in $\mathbb{O}$. That is,

$$C_1 = \underset{P \in \mathbb{O}}{\operatorname{argmin}} \left(\sum_{P' \in \mathbb{O} \setminus \{P\}} D(P' \parallel P) \right).$$

The rest $k - 1$ representatives are selected iteratively. In the ith ($2 \leq i \leq k$) iteration, the algorithm selects the representative C_i which decreases the total KL divergence (9) as much as possible. For each object P which has not been selected, we test whether it should be selected in the current round. For any other nonselected object P', P' will be assigned to the new representative P if the divergence $D(P' \parallel P)$ is smaller than the divergence between P' and any previously selected representatives. Therefore, we calculate the contribution of P' to the decrease of the total KL divergence by selecting P as

$$\max\left(0, \min_{j=1}^{i-1} \big(D(P' \parallel C_j)\big) - D(P' \parallel P)\right).$$

We calculate the total decrease of the total KL divergence by selecting P as the sum over the contribution of all nonselected object, denoted by $DEC(P)$. Then, the object to be selected in the ith iteration is the one that can incur the largest decrease, that is,

$$C_i = \underset{P \in \mathbb{O} \setminus \{C_1, \ldots, C_{i-1}\}}{\operatorname{argmax}} \big(DEC(P)\big).$$

We end the building phase as long as k representatives are selected and proceed to the swapping phase.

Swapping phase. In the swapping phase, the uncertain k-medoids method iteratively improves the clustering by swapping a nonrepresentative object with the representative to which it is assigned. For a nonrepresentative object P, suppose it is assigned to cluster $\mathbb{C}$ whose representative is C. We consider the effect of swapping P and C in two cases on every nonselected object P' other than P:

- If P' currently belongs to C, when C is replaced by P, we will reassign P' to P or one of the other $k - 1$ existing representatives, to which P' is the most similar.
- If P' currently belongs to a representative C' other than C, and $D(P' \parallel P) < D(P' \parallel C')$, P' is reassigned to P.

When a reassignment happens, the decrease of the total KL divergence by swapping P and C is recorded. After all nonrepresentative objects are examined, we obtain the total decrease of swapping P and C. Then, we select the object P_{max} which can make the largest decrease. That is,

$$P_{max} = \underset{P \in \mathbb{O} \setminus \{C_1, \ldots, C_k\}}{\operatorname{argmax}} \big(DEC(P)\big).$$

We check that whether the swapping of P_{max} can improve the clusters, i.e., whether $DEC(P_{max}) > 0$. If so, the swapping is carried into execution. Otherwise, the method terminates and report the final clustering.

Complexity. The building phase in the uncertain k-medoids method requires evaluating the KL divergences between $O(kn^2)$ pairs of objects, where n is the number of objects. In the swapping phase, in each iteration, it evaluates $O(n^2)$ KL divergences to find the optimal swapping. In total, the uncertain k-medoids method has complexity $O((k + r)n^2E)$, where r is the number of iterations in the swapping phase and E is the complexity of evaluating the KL divergence of two objects. The method cannot scale on big data sets due to it quadratic complexity with respect to the number of objects. Next, we present a randomized k-medoids method to bring down the complexity.

4.1.2 Randomized K-Medoids Method

The randomized k-medoids method, instead of finding the optimal nonrepresentative object for swapping, randomly selects a nonrepresentative object for swapping if the clustering quality can be improved. We follow the simulated annealing technique [20], [6] to prevent the method from being stuck at a local optimal result.

The randomized k-medoids method follows the building-swapping framework. At the beginning, the building phase is simplified by selecting the initial k representatives at random. Nonselected objects are assigned to the most similar representative according to KL divergence. Then, in the swapping phase, we iteratively replace representatives by nonrepresentative objects.

In each iteration, instead of finding the optimal nonrepresentative object for swapping in the uncertain k-medoids method, a nonrepresentative object P is randomly selected to replace the representative C to which P is assigned. To determine whether P is a good replacement of C, we examine the two cases as described in the swapping phase in Section 4.1.1. After all nonrepresentative objects are examined, the total decrease of the total KL divergence by swapping P and C is recorded as DEC.

Then, we compute the probability of swapping as

$$Pr_{swap}(DEC) = \begin{cases} 1 & \text{if } DEC > 0 \\ e^{DEC \times \log(r_{cur})} & \text{if } DEC \leq 0, \end{cases}$$

where r_{cur} is the current number of iterations. Basically, if $DEC > 0$, the swapping can improve the clustering, then the swapping is put into practice. In the cases where $DEC \leq 0$, the swapping may not improve the clustering. However, the swapping may still be carried out with probability $e^{DEC \times \log(r_{cur})}$ to prevent the algorithm from being stuck at a local optimal result. We repeat the iterative process until a swapping is not executed.

The randomized k-medoids method has time complexity $O(rnE)$ where r is the number of iterations in the swapping phase and E is the complexity of evaluating the KL divergence of two objects. The cost of the building phase in the uncertain k-medoids method is removed since the representatives are randomly initialized. The object selected for swapping in each iteration in the swapping phase is also randomly picked, so the randomized k-medoids method evaluates $O(n)$ KL divergences in each iteration. We will see in our experiments that the randomized k-medoids method can scale well on data sets with a large number of objects.

4.2 Density-Based Clustering Methods

Unlike partitioning methods which organize similar objects into the same partitions to discover clusters, density-based clustering methods regard clusters as dense regions of objects that are separated by regions of low density.

DBSCAN [10] is the first and most representative density-based clustering method developed for certain data. To demonstrate density-based clustering methods based on distribution similarity, we develop the uncertain DBSCAN method which integrates KL divergence into DBSCAN. Different to the FDBSCAN method [21] which is based on geometric distances and finds dense regions in the original geometric space, the uncertain DBSCAN method transforms objects into a different space where the distribution differences are revealed.

The uncertain DBSCAN method finds dense regions through core objects whose ε-neighborhood contains at least μ objects. Formally, P is a core object, if

$$\left|\{Q \in \mathbb{O} | D(Q \parallel P) \leq \varepsilon\}\right| \geq \mu.$$

An object Q is said to be direct density-reachable from an object P if $D(Q \parallel P) \leq \varepsilon$ and P is a core object.

Initially, every core object forms a cluster. Two clusters are merged together if a core object of one cluster is density-reachable from a core object of the other cluster. A noncore object is assigned to the closest core object if it is direct density reachable from this core object. The algorithm iteratively examines objects in the data set until no new object can be added to any cluster.

Note that direct density-reachability described above is an asymmetric relationship. This is not caused by the asymmetry of KL divergence. The relationship is also asymmetric on certain data where similarity between points are measured by the square of euclidean distance.

The quality of the clustering obtained by the uncertain DBSCAN method depends on the parameters ε and μ. We will show the performance in Section 6.

The complexity of the uncertain DBSCAN method is $O(n^2E)$ where n is the number of uncertain objects and E is the cost of evaluating the KL divergence of two objects. Essentially, the KL divergence of any pair of objects is evaluated because we do not assume any index on the data set.

5 Boosting Computation

The uncertain k-medoids method, the randomized k-medoids method, and the uncertain DBSCAN method all require evaluation of KL divergences of many pairs of objects. As the number of uncertain objects and the sample size of each object increase, it is costly to evaluate a large amount of KL divergence expressions. In this section, we first show the implementation technique to save the computation in Section 5.1. Then, Section 5.2 introduces fast Gauss transform to provide a fast approximation of KL divergence in the continuous case.

5.1 Efficient Implementation

In both the uncertain and randomized k-medoids methods but not DBSCAN, the most used operation is computing the difference between two KL divergence expressions, $D(P \parallel C) - D(P \parallel C')$, for example, to decide whether P should be assigned to representative C or C', and to compute the DEC of replacing a representative.

We can evaluate the divergence difference $D(P \parallel C) - D(P \parallel C')$ more efficiently than evaluating the two divergences separately and then computing the subtraction.

In the discrete case, applying (4), we have

$$\begin{aligned} D(P \parallel C) &- D(P \parallel C') \\ &= \sum_{p \in P} P(p) \log \frac{P(p)}{C(p)} - \sum_{p \in P} P(p) \log \frac{P(p)}{C'(p)} \\ &= \sum_{p \in P} P(p) \big(\log C'(p) - \log C(p)\big). \end{aligned} \tag{10}$$

In the continuous case, applying (7), we have

$$\begin{aligned} D(P \parallel C) &- D(P \parallel C') \\ &= \frac{1}{|P|} \sum_{p \in P} \log \frac{P(p)}{C(p)} - \frac{1}{|P|} \sum_{p \in P} \log \frac{P(p)}{C'(p)} \\ &= \frac{1}{|P|} \sum_{p \in P} \big(\log C'(p) - \log C(p)\big). \end{aligned} \tag{11}$$

By directly computing the difference between the two divergence expressions, we simplify the computation in both the discrete and continuous cases. More importantly, we avoid evaluating the term $\log P(p)$ in the continuous case, thus substantially reduce the computation in density estimation.

5.2 Fast Gauss Transform for Efficiently Calculating Probability Density Functions

The probability mass functions of uncertain objects in the discrete case can be directly calculated as shown in (2). The complexity is linear to the sample size of the uncertain object. Moreover, as the domain is finite, we can precompute the probability mass function of every object and store them in a hash table either in-memory or on-disk.

However, in the continuous case, the complexity of calculating the probability density functions is quadratic to the sample size of the uncertain object. Precomputation is not feasible since the domain is uncountably infinite. No matter we evaluate KL divergences directly or evaluate the divergence differences as described in Section 5.1, the major cost is on evaluating the following expression:

$$\sum_{p\in P}\log\sum_{q\in C}\prod_{j=1}^{d} e^{-\frac{(p.D_j-q.D_j)^2}{2h_{C,j}^2}}. \tag{12}$$

It requires to evaluate the sum of N Gaussian functions at M points, where N is the sample size of C and M is the sample size of P. Straightforwardly, $N \times M$ Gaussian functions are evaluated, and the computational complexity is $O(N \times M)$. Clearly, it is costly for large N and M. To make the computation practical for large data sets, we resort to approximate (12).

The main idea of approximating the sum of a series of Gaussian functions is to make use of the fact that the probability density of the Gaussian function decays exponentially and is negligible outside a certain distance to the center of the Gaussian. The fast Gauss transform [13] is one of the best techniques to do the job. It reduces the computational complexity from $O(N \times M)$ to $O(N + M)$ using a divide-and-conquer strategy and combined with the manipulation of Hermite polynomial expansions and Taylor series. However, as pointed out by Yang et al. [38], the constant factor in $O(N + M)$ grows exponentially with respect to the dimensionality d, which makes the technique impractical for $d > 3$. Yang et al. [38] developed an improved fast Gauss transform to reduce the constant factor to asymptotically polynomial order. We adopt their improved Gauss transform to boost the efficiency of evaluating (11).

The improved fast Gauss transform takes advantages of the exponential decay of the Gaussian. Another key technique is to expand the sum of Gaussian functions into a multivariate Taylor expansion and truncate the series after degree p, where p is chosen to be sufficiently large such that the bounded error is no more than the precision parameter ϵ.

To evaluate (12), the algorithm works in the following steps:

- Step 1: Approximate the sample of C by clustering. Partition the N sample points of C into k clusters $S_1, \ldots, S_k$ using the farthest point clustering algorithm [12], [11] such that the maximum radius of all clusters is less than $h\rho_1$, here h is the bandwidth of the Gaussian function and ρ_1 is a controlling parameter.
- Step 2: Choose parameter p for truncating Taylor expansion. Choose p sufficiently large such that the error estimate is less than the precision ϵ. Here, the error at any point of P is bounded by $N(\frac{2^p}{p!}\rho_1\rho_2 + e^{-\rho_2^2})$, where ρ_2 is a controlling parameter.
- Step 3: Compute the coefficients of the Taylor expansion. For each cluster S_i $(1 \le i \le k)$, let c_i denote the center found by the farthest point clustering algorithm, compute the coefficients below

$$C_\alpha^i = \frac{2^{|\alpha|}}{\alpha!}\sum_{q\in S_i}\prod_{j=1}^{d} e^{\frac{-(q.D_j-c_i.D_j)^2}{h^2}}\left(\frac{q-c_i}{h}\right)^{\alpha}.$$

  Here, $\alpha = (\alpha_1, \ldots, \alpha_d)$ is a multiindex which is a d-dimensional vector of nonnegative integers. For any multiindex α and any real value vector $x = (x.D_1, \ldots, x.D_d)$, we define the following operations:

$$\begin{aligned} x^\alpha &= x.D_1^{\alpha_1}\cdots x.D_d^{\alpha_d},\\ |\alpha| &= \alpha_1 + \cdots + \alpha_d,\\ \alpha! &= \alpha_1!\cdots\alpha_d!. \end{aligned}$$

- Step 4: Compute the sum of approximated Gaussian functions. For each point $s \in P$, find the clusters whose centers lie within the range $h\rho_2$. Then, the sum of Gaussian functions in (12) is evaluated as

$$\begin{aligned} &\sum_{q\in C}\prod_{j=1}^{d} e^{-\frac{(p.D_j-q.D_j)^2}{2h_{C,j}^2}}\\ &= \sum_{\substack{dist(s,c_i)\\ \le h\rho_2}}\sum_{|\alpha|<p} C_\alpha^i \prod_{j=1}^{d} e^{\frac{-(p.D_j-c_i.D_j)^2}{h^2}}\left(\frac{p-c_i}{h}\right)^{\alpha}, \end{aligned}$$

  where $dist(s, c_i)$ is the distance between p and c_i.

The tradeoff between the computational cost and the precision is controlled by parameters p, ρ_1, and ρ_2. The larger p and ρ_2, and the smaller ρ_1, the better precision, but the computation and storage cost increases. In our experiments, as suggested in [38], we set $p = \rho_2 = 10$ and ρ_1 is indirectly set by k (see Step 1) which is set to $\sqrt{N}$.

Our empirical study in Section 6 shows that the fast Gauss transform boosts the efficiency of our algorithms dramatically with only a small decrease of the clustering quality.

6 Empirical Study

We conducted extensive experiments on both synthetic and real data sets to evaluate the effectiveness of KL divergence as a similarity measure for clustering uncertain data and the efficiency of the techniques for evaluating KL divergences.

Our programs were implemented in C++ and compiled by GCC 4.3.3. The experiments were conducted on a computer with an Intel Core 2 Duo P8700 2.53 GHz CPU and 4 GB main memory running Ubuntu 9.04 Jaunty. All programs ran in-memory. The I/O cost is not reported.

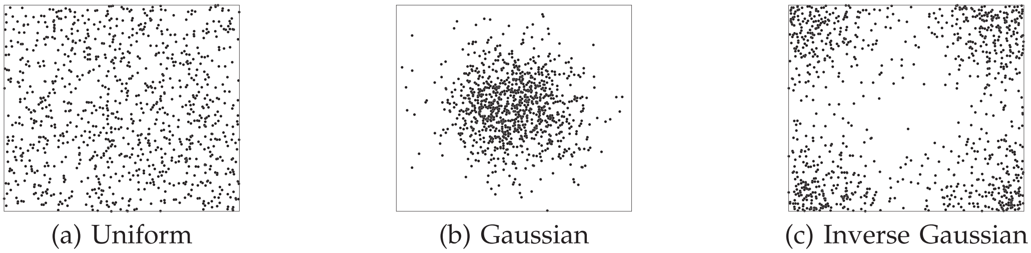

(a) Uniform (b) Gaussian (c) Inverse Gaussian

Fig. 2. Three types of distributions.

6.1 Synthetic Data

We generated data sets in both continuous and discrete domains. In the continuous case, an uncertain object is a sample drawn from a continuous distribution. In the discrete case, a data set is generated by converting a data set in the continuous case. We discretized the continuous domain by partitioning it into a grid. Every dimension is equally divided into two parts. So, a d-dimensional space is divided into 2^d cells of equal size. We use the central points of cells as values in the discrete domain. The probability of an object in a cell is the sum of the probabilities of all its sample points in this cell.

The data space was restricted to $[0,1]^d$. We use three different types of distributions shown in Fig. 2, the uniform distribution, the Gaussian distribution, and the inverse Gaussian distribution. An inverse Gaussian distribution was generated from a Gaussian distribution. Given a point x in a Gaussian distribution, we generated a point y for the inverse Gaussian distribution according to the following formula for $1 \leq i \leq d$,

$$y.D_i = \begin{cases} 1.5 - x.D_i & \text{if} x.D_i \in [0.5, 1]; \\ 0.5 - x.D_i & \text{if } x.D_i \in [0, 0.5). \end{cases}$$

We consider four major factors in our experiments, the dimensionality d of the data space (from 2 to 10), the cardinality (i.e., the number of objects) n of the data set (from 50 to 10,000), the sample size s (from 50 to 900), and the number of clusters k (from 3 to 25).

Given the number of clusters k for a data set, we generated k probability density functions, in which there is one uniform distribution, $(k-1)/2$ Gaussian distributions with different variances, and $(k-1)/2$ inverse Gaussian distributions with different variances. The variance of the ith Gaussian/inverse Gaussian distribution is $0.05i$. For each distribution, we generated a group of samples, each of which forms one object. Thus, objects in the same group are sampled from the same probability density function. The number of objects of a group was randomly picked with an expectation of $\frac{n}{k}$. We made sure that the total number of objects is equal to n. As we generated a synthetic data set in this way, we have the ground truth of the clustering in the data set. We used the ground truth to evaluate the clustering quality of our algorithms.

Our experiments include three parts. Section 6.1.1 first evaluates the effect of KL divergence used in clustering methods presented in Section 4. Then, Section 6.1.2 shows the speedup by the implementation techniques and fast Gauss transform introduced in Section 5. Last, Section 6.1.3 shows the scalability and feasibility of our algorithms on large data sets.

6.1.1 Effectiveness of KL Divergence in Clustering

We first compare the clustering quality of KL divergences with geometric distances in both partitioning clustering methods and density-based clustering methods. For partitioning clustering methods, we implemented the UK-means method [24] (denoted by UK) using geometric distances as well as the uncertain k-medoids method (denoted by KM-KL) and the randomized k-medoids method (denoted by RKM-KL) using KL divergences. For density-based clustering methods, we implemented the FDBSCAN method [21] (denoted by FD) using geometric distances and the uncertain DBSCAN method (denoted by DB-KL) using KL divergences.

We use precision and recall as the quality measurements. Let |G denote the ground truth clustering generated by the synthetic data generator, $\mathbb{C}$ is the clustering obtained by a clustering method. Two objects are called a pair if they appear in the same cluster in a clustering. We define

- *TP* true positive, the set of common pairs of objects in both $\mathbb{G}$ and $\mathbb{C}$;
- *FP* false positive, the set of pairs of objects in $\mathbb{C}$ but not $\mathbb{G}$;
- *FS* false negative, the number of pairs of objects in $\mathbb{G}$ but not $\mathbb{C}$.

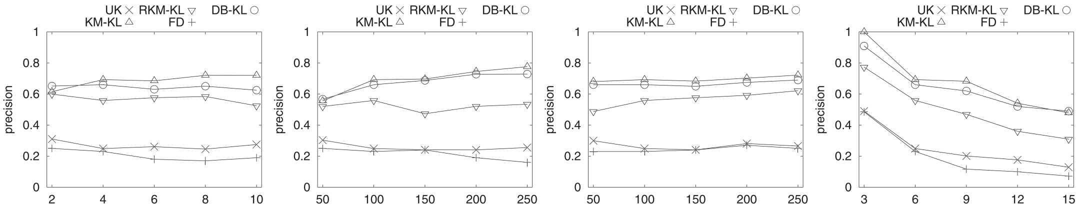
Then, the precision and recall of a clustering $\mathbb{C}$ are calculated, respectively, as

$$precision(\mathbb{C}) = |TP|/(|TP| + |FP|),$$
$$recall(\mathbb{C}) = |TP|/(|TP| + |FN|).$$

By default a data set contains $n = 100$ objects, each of which has $s = 100$ sample points in a $d = 4$ dimensional space. The number of clusters is $k = 6$ by default.

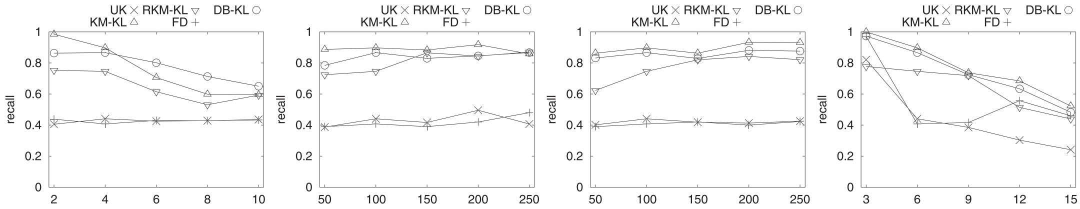
We note that density-based methods, FD and DB-KL, do not use k as a parameter. Instead, the clustering is controlled by the minimum number μ of points required to form a cluster and the neighborhood distance/divergence. As suggested in [21], we set $\mu = 5$ and varied ε so that FD and DB-KL can output approximately the specified number k of clusters.

Figs. 3 and 4 compare the precision and recall of UK, KM-KL, RKM-KL, FD, and DB-KL in the continuous case, respectively. Clearly, geometric distance-based methods UK and FD have very poor precision and recall, while the KL divergence-based methods KM-KL, RKM-KL, and DB-KL can obtain much better clustering. KM-KL has higher precision and recall than the randomized versions RKM-KL,



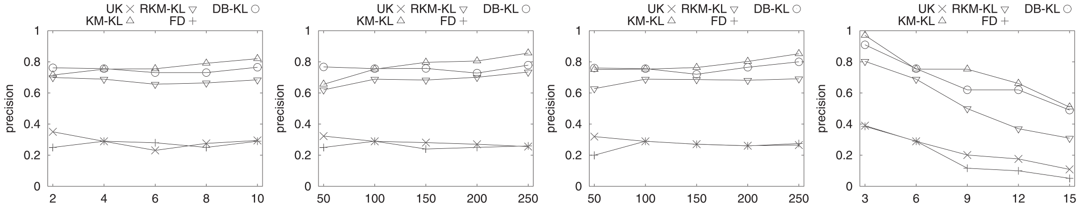
(a) dimensionality d (b) cardinality n (c) sample size s (d) # of clusters k

Fig. 3. Comparison in precision between KL divergences and geometric distances in the continuous case.



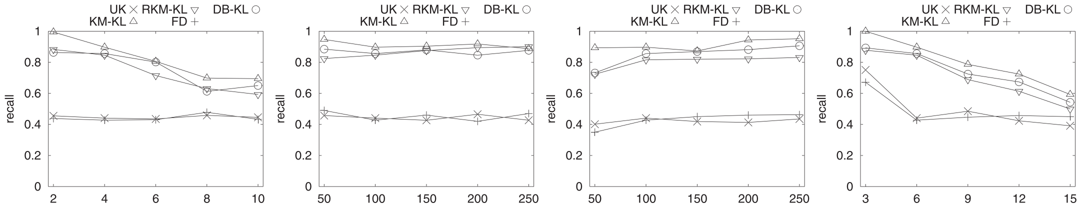
(a) dimensionality d (b) cardinality n (c) sample size s (d) # of clusters k

Fig. 4. Comparison in recall between KL divergences and geometric distances in the continuous case.



(a) dimensionality d (b) cardinality n (c) sample size s (d) # of clusters k

Fig. 5. Comparison in precision between KL divergences and geometric distances in the discrete case.



(a) dimensionality d (b) cardinality n (c) sample size s (d) # of clusters k

Fig. 6. Comparison in recall between KL divergences and geometric distances in the discrete case.

since the uncertain k-medoids method tries to find the optimal nonrepresentative object for swapping in each iteration. In the discrete case, as shown in Figs. 5 and 6, we see similar results as in the continuous case.

We also see that the precision and recall of all algorithms follow similar trends. They are not sensitive to the dimensionality. They increase as the sample size increases or the number of clusters decreases. Also, they drop when there are more clusters in a data set since the number of object pairs in a clustering decreases linearly as k increases; thus, the misclustered pairs incur large penalties on the precision and recall.

6.1.2 Efficiency of Boosting Techniques

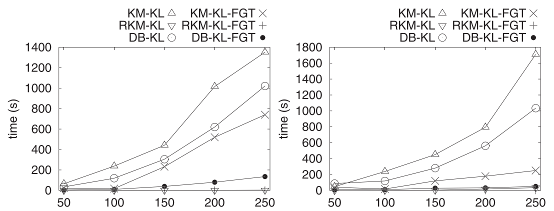
Fig. 7 shows the speedup of the boosting techniques, especially the fast Gauss transform. We plot the runtime of KM-KL, RKM-KL, and DB-KL and their corresponding methods implemented with the fast Gauss transform, annotated by KM-KL-FGT, RKM-KL-FGT, and DB-KL-FGT. Methods with the fast Gauss transform are significantly faster than their counterparts, since the fast Gauss transform brings down the complexity of evaluating the sum of KL divergences from quadratic to linear.

Figs. 8 and 9 show that the precision and recall of the methods with the fast Gauss transform are slightly lower than their counterparts which compute KL divergences directly due to the error incurred in the approximation.

In a word, the fast Gauss transform is a tradeoff between quality and time. The experiment results show that it can speed up the clustering a lot with an acceptable accuracy tradeoff.

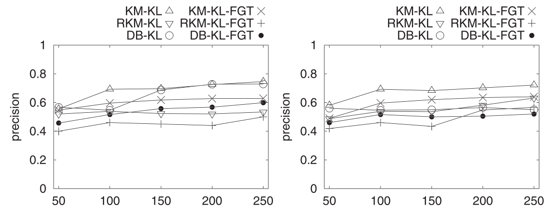
6.1.3 Scalability

As the uncertain k-medoids method and the uncertain DBSCAN method have quadratic complexity with respect to the number of objects, they cannot scale on data sets with a large number of objects. In this section, we test the scalability of the randomized k-medoids method on large data sets which consist of $4{,}000$ objects and 300 sample points in a 4D space by default. The number of clusters in a data set is $k = 10$ by default.



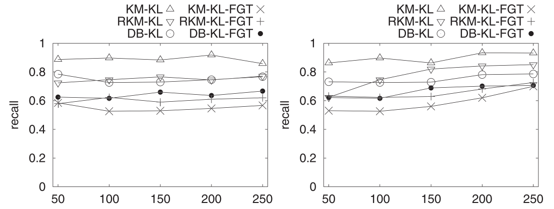
(a) cardinality n (b) sample size s

Fig. 7. Comparison in runtime between methods w/ and w/o the fast Gauss transform in the continuous case.



(a) cardinality n (b) sample size s

Fig. 8. Comparison in precision between methods w / and w/o the fast Gauss transform in the continuous case.



(a) cardinality n (b) sample size s

Fig. 9. Comparison in recall between methods with and without the fast Gauss transform in the continuous case.

Figs. 10 and 11 show similar trends of precision and recall of RKM-KL and RKM-KL-FGT as on small data sets.

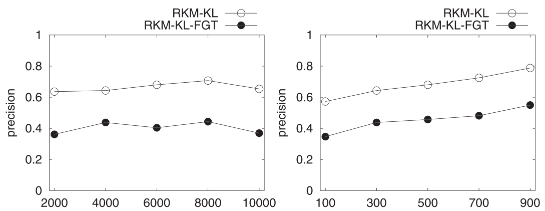
Fig. 12 shows the runtime of RKM-KL and RKM-KL-FGT on large data sets. RKM-KL scales well as the number of objects increases; however, it suffers from the quadratic complexity of evaluating the sum of KL divergences with respect to the sample size. Thanks to the fast Gaussian transform, RKM-KL-FGT is scalable on data sets with both a large number of objects and a large sample per object.

6.2 Real Data

We obtained a weather data set from the National Center for Atmospheric Research data archive (http://dss.ucar.edu/data sets/ds512.0/). The data set consists of 2,936 stations around the world. Each station contains 366 daily records in the year of 2008. Each record has three dimensions, average temperature, precipitation, and average humidity. We labeled the climate type of each station according to the Köppen-Geiger climate classification [28]. For the ground truth, stations with the same label are considered to be in the same cluster. In total, we have five clusters, tropical climate, dry climate, temperate climate, continental climate, and polar climate.

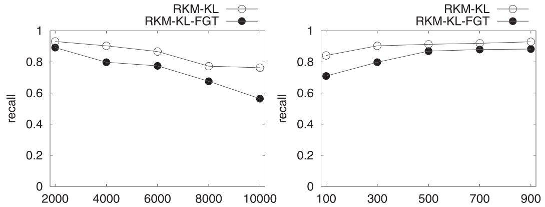
Fig. 13 shows an example distribution of each of the five climates. We only plot dimensions of average temperature and precipitation for better visualization. Clearly, we observe the difference among the five types of climates.

Fig. 14 shows the results of the UK, RKM-KL, RKM-KL-FGT, and a random clustering methods (denoted by RD) on the weather data set. The random clustering method randomly assigns an object to one of the k clusters, and it serves as the baseline of other methods. We see that RKM-KL and RKM-KL-FGT have much higher precision and recall than RD and UK. RKM-KL-FGT has feasible running time comparing to RKM-KL.



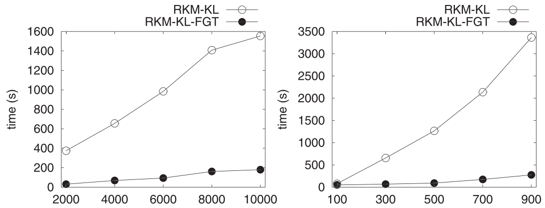
(a) cardinality n (b) sample size s

Fig. 10. Precision on large data sets in the continuous case.



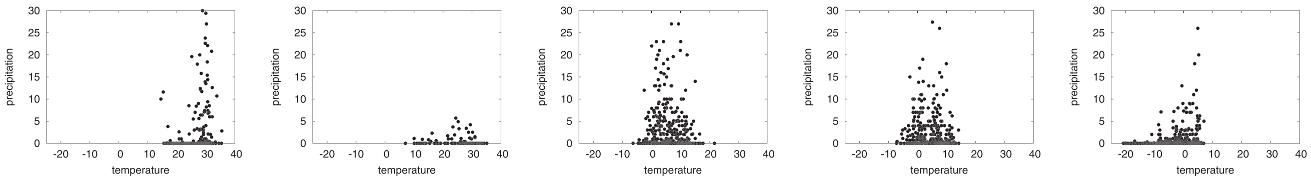
(a) cardinality n (b) sample size s

Fig. 11. Recall on large data sets in continuous case.



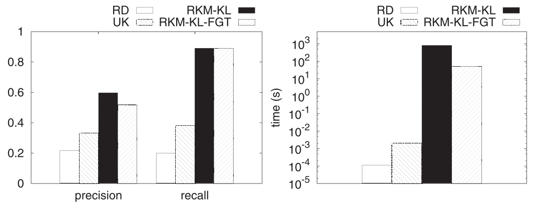
(a) cardinality n (b) sample size s

Fig. 12. Runtime on large data sets in continuous cases.



(a) tropical (b) dry (c) temperate (d) continental (e) polar

Fig. 13. Example distributions of the five types of climates.



(a) Precision and recall. (b) Time.

Fig. 14. Results on weather data.

6.3 Summary

In summary, both partitioning and density-based clustering methods have better clustering quality when using KL divergences as similarity than using geometric distances. The results confirm that KL divergence can naturally capture the distribution difference which geometric distance cannot capture.

To boost the computation in the continuous case to battle the costly kernel density estimation, the fast Gauss transform can speed up the clustering a lot with an acceptable accuracy tradeoff. To scale on large data sets, the randomized k-medoids method equipped with the Gauss transform has linear complexity with respect to both the number of objects and the sample size per object. It can perform scalable clustering tasks on large data sets with moderate accuracy.

7 Conclusions

In this paper, we explore clustering uncertain data based on the similarity between their distributions. We advocate using the Kullback-Leibler divergence as the similarity measurement, and systematically define the KL divergence between objects in both the continuous and discrete cases. We integrated KL divergence into the partitioning and density-based clustering methods to demonstrate the effectiveness of clustering using KL divergence. To tackle the computational challenge in the continuous case, we estimate KL divergence by kernel density estimation and employ the fast Gauss transform technique to further speed up the computation. The extensive experiments confirm that our methods are effective and efficient.

The most important contribution of this paper is to introduce distribution difference as the similarity measure for uncertain data. Besides clustering, similarity is also of fundamental significance to many other applications, such as nearest neighbor search. In the future, we will study those problems on uncertain data based on distribution similarity.

Acknowledgments

The authors are grateful to the anonymous reviewers and the associate editor for their constructive and critical comments on the paper. The research of Bin Jiang and Jian Pei is supported in part by an NSERC Discovery Grant and an NSERC Discovery Accelerator Supplement Grant. The research of Yufei Tao is supported in part by grants GRF 4169/09, GRF 4166/10, and GRF 4165/11 from HKRGC. The research of Xuemin Lin is supported in part by grants ARCDP110102937, ARCDP0987557, ARCDP0881035, and NSFC61021004. All opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] S. Abiteboul, P.C. Kanellakis, and G. Grahne, "On the Representation and Querying of Sets of Possible Worlds," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD),* 1987.

[2] M.R. Ackermann, J. Blömer, and C. Sohler, "Clustering for Metric and Non-Metric Distance Measures," *Proc. Ann. ACM-SIAM Symp. Discrete Algorithms (SODA),* 2008.

[3] M. Ankerst, M.M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering Points to Identify the Clustering Structure," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD),* 1999.

[4] A. Banerjee, S. Merugu, I.S. Dhillon, and J. Ghosh, "Clustering with Bregman Divergences," *J. Machine Learning Research,* vol. 6, pp. 1705-1749, 2005.

[5] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[6] V. Cerny, "A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm," *J. Optimization Theory and Applications,* vol. 45, pp. 41-51, 1985.

[7] R. Cheng, D.V. Kalashnikov, and S. Prabhakar, "Evaluating Probabilistic Queries over Imprecise Data," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD),* 2003.

[8] N.N. Dalvi and D. Suciu, "Management of Probabilistic Data: Foundations and Challenges," *Proc. ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS),* 2007.

[9] I.S. Dhillon, S. Mallela, and R. Kumar, "A Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification," *J. Machine Learning Research,* vol. 3, pp. 1265-1287, 2003.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining (KDD),* 1996.

[11] T. Feder and D.H. Greene, "Optimal Algorithms for Approximate Clustering," *Proc. Ann. ACM Symp. Theory of Computing (STOC),* 1988.

[12] T.F. Gonzalez, "Clustering to Minimize the Maximum Intercluster Distance," *Theoretical Computer Science,* vol. 38, pp. 293-306, 1985.

[13] L. Greengard and J. Strain, "The Fast Gauss Transform," *SIAM J. Scientific and Statistical Computing,* vol. 12, pp. 79-94, 1991.

[14] J. Han and M. Kamber, *Data Mining: Concepts and Techniques.* Elsevier, 2000.

[15] T. Imielinski and W.L. Lipski Jr., "Incomplete Information in Relational Databases," *J. ACM,* vol. 31, pp. 761-791, 1984.

[16] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data,* Prentice Hall, 1988.

[17] R. Jampani, F. Xu, M. Wu, L.L. Perez, C.M. Jermaine, and P.J. Haas, "Mcdb: A Monte Carlo Approach to Managing Uncertain Data," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD),* 2008.

[18] B. Kao, S.D. Lee, D.W. Cheung, W.-S. Ho, and K.F. Chan, "Clustering Uncertain Data Using Voronoi Diagrams," *Proc. IEEE Int'l Conf. Data Mining (ICDM),* 2008.

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

[20] S. Kirkpatrick, C.G. Gelatt Jr., and M.P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, pp. 671-680, 1983.

[21] H.-P. Kriegel and M. Pfeifle, "Density-Based Clustering of Uncertain Data," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining (KDD),* 2005.

[22] H.-P. Kriegel and M. Pfeifle, "Hierarchical Density-Based Clustering of Uncertain Data," *Proc. IEEE Int'l Conf. Data Mining (ICDM),* 2005.

[23] S. Kullback and R.A. Leibler, "On Information and Sufficiency," *The Annals of Math. Statistics,* vol. 22, pp. 79-86, 1951.

[24] S.D. Lee, B. Kao, and R. Cheng, "Reducing Uk-Means to k-Means," *Proc. IEEE Int'l Conf. Data Mining Workshops (ICDM),* 2007.

[25] S.P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Information Theory,* vol. IT-28, no. 2, pp. 129-137, Mar. 1982.

[26] J.B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Statistics and Probability,* 1967.

[27] W.K. Ngai, B. Kao, C.K. Chui, R. Cheng, M. Chau, and K.Y. Yip, "Efficient Clustering of Uncertain Data," *Proc. Sixth Int'l Conf. Data Mining (ICDM),* 2006.

[28] M.C. Peel, B.L. Finlayson, and T.A. McMahon, "Updated World Map of the köppen-Geiger Climate Classification," *Hydrology and Earth System Sciences,* vol. 11, pp. 1633-1644, 2007.

[29] J. Pei, B. Jiang, X. Lin, and Y. Yuan, "Probabilistic Skylines on Uncertain Data," *Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB),* 2007.

[30] J.M. Ponte and W.B. Croft, "A Language Modeling Approach to Information Retrieval," *Proc. 21st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR),* 1998.

[31] A.D. Sarma, O. Benjelloun, A.Y. Halevy, and J. Widom, "Working Models for Uncertain Data," *Proc. Int'l Conf. Data Eng. (ICDE),* 2006.

[32] D.W. Scott, *Multivariate Density Estimation: Theory, Practical, and Visualization.* Wiley, 1992.

[33] B.W. Silverman, *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, 1986.

[34] F. Song and W.B. Croft, "A General Language Model for Information Retrieval," *Proc. Int'l Conf. Information and Knowledge Management (CIKM),* 1999.

[35] Y. Tao, R. Cheng, X. Xiao, W.K. Ngai, B. Kao, and S. Prabhakar, "Indexing Multi-Dimensional Uncertain Data with Arbitrary Probability Density Functions," *Proc. Int'l Conf. Very Large Data Bases (VLDB),* 2005.

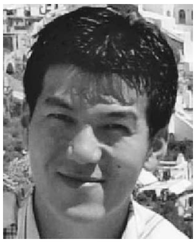
[36] P.B. Volk, F. Rosenthal, M. Hahmann, D. Habich, and W. Lehner, "Clustering Uncertain Data with Possible Worlds," *Proc. IEEE Int'l Conf. Data Eng. (ICDE),* 2009.

[37] J. Xu and W.B. Croft, "Cluster-Based Language Models for Distributed Retrieval," *Proc. 22nd Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR),* 1999.

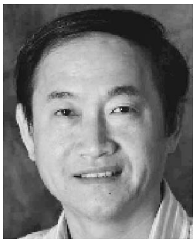
[38] C. Yang, R. Duraiswami, N.A. Gumerov, and L.S. Davis, "Improved Fast Gauss Transform and Efficient Kernel Density Estimation," *Proc. IEEE Int'l Conf. Computer Vision (ICCV),* 2003.

Bin Jiang received the BSc and MSc degrees in computer science from Peking University, China, and the University of New South Wales, Australia, respectively. He is currently working toward the PhD degree in the School of Computing Science at Simon Fraser University, Canada. His research interests lie in mining uncertain data.

Jian Pei is a professor in the School of Computing Science at Simon Fraser University, Canada. His research interests can be summarized as developing effective and efficient data analysis techniques for novel data intensive applications. He is currently interested in various techniques of data mining, web search, information retrieval, data warehousing, online analytical processing, and database systems, as well as their applications in social networks, health-informatics, business, and bioinformatics. His research has been supported in part by government funding agencies and industry partners. He has published prolifically and served regularly for the leading academic journals and conferences in his fields. He is an associate editor of the *ACM Transactions on Knowledge Discovery from Data (TKDD)* and new editor-in-chief of *IEEE Transactions of Knowledge and Data Engineering (TKDE)* beginning in 2013. He is the recipient of several prestigious awards. He is a senior member of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE).



Yufei Tao is a is a full professor at the Chinese University of Hong Kong. He also holds a visiting professor position, under the World Class University (WCU) program of the Korean government, at the Korea Advanced Institute of Science and Technology (KAIST). He is an associate editor of the *ACM Transactions on Database Systems* (*TODS*) and the *IEEE Transactions on Knowledge and Data Engineering* (*TKDE*). He is/was a PC cochair of ICDE 2014, a PC cochair of SSTD 2011, an area PC chair of ICDE 2011, and a senior PC member of CIKM 2010, 2011, and 2012. He is a member of the IEEE.



Xuemin Lin received the BSc degree in applied math from Fudan University in 1984 and the PhD degree in computer science from the University of Queensland in 1992. He is a professor in the School of Computer Science and Engineering, the University of New South Wales (UNSW). He has been the head of the database research group at UNSW since 2002. He is a concurrent professor in the School of Software, East China Normal University. Before joining UNSW, he held various academic positions at the University of Queensland and the University of Western Australia. During 1984-1988, he studied for the PhD degree in applied math at Fudan University. He is currently an associate editor of the *ACM Transactions on Database Systems*. His current research interests lie in data streams, approximate query processing, spatial data analysis, and graph visualization. He is a seniore member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.