Volume 2, Issue 7, July 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Uncertain Data Algorithms and Applications

Miss Pragati Pandey
Mtech [CT]-IV SEM,
Rungta College of Eng. & Tech
Bhilai, Chhattisgarh

Miss Prateeksha Pandey
ME [CSE]
CIMT College
Bhilai, Chhattisgarh

Mrs. Minu Choudhary
Lecturer [CSE]
Rungta College of Eng. & Tech.
Bhilai, Chhattisgarh

Abstract*—in recent years, a number of indirect data collection methodologies have led to the proliferation of uncertain data. Such databases are much more complex because of the additional challenges of representing the probabilistic information. In this paper, we provide uncertain data mining and management applications. We will explore the various models utilized for uncertain data representation. In the field of uncertain data management, we will examine traditional database management methods such as join processing; query processing, selectivity estimation, OLAP queries, and indexing. In the field of uncertain data mining, we will examine Traditional mining problems such as frequent pattern mining, outlier detection, classification, and clustering. We discuss different methodologies to process and mine uncertain data in a variety of forms.

***Index Terms*—Mining methods and algorithms, database applications, database management, information technology and systems.**

I. INTRODUCTION

IN recent years, many advanced technologies have been developed to store and record large quantities of data continuously. In many cases, the data may contain errors or may only be partially complete. For example, sensor networks typically create large amounts of uncertain data sets. In other cases, the data points may correspond to objects which are only vaguely specified, and are therefore considered uncertain in their representation. Similarly, surveys and imputation techniques create data which is uncertain in nature. This has created a need for uncertain data management algorithms and applications. In uncertain data management, data records are typically represented by probability distributions rather than deterministic values.

A. *Modeling of uncertain data*

A key issue is the process of modeling the uncertain data. Therefore, the underlying complexities can be captured while keeping the data useful for database management applications.

B. *Uncertain data management*

In this case, one wishes to adapt traditional database management techniques for uncertain data. Examples of such techniques could be joining processing, query processing, indexing, or database integration. Uncertain data mining. The results of data mining applications are affected by the underlying uncertainty in the data. Therefore, it is critical to design data mining techniques that can take such uncertainty into account during the computations

C. *Uncertain data mining*

The results of data mining applications are affected by the underlying uncertainty in the data. Therefore, it is critical to design data mining techniques that can take such uncertainty into account during the computations.

II. UNCERTAIN DATA REPRESENTATION AND MODELING

A database that provides incomplete information consists of a set of possible instances of the database. It is important to distinguish between incomplete databases and probabilistic data.

A. Probabilistic Database Definitions

A probabilistic database is defined as follows:

Definition 1. A probabilistic-information database is a finite probability space whose outcomes are all possible database instances consistent with a given schema. This can be represented as the pair (X, p), where X is a finite set of possible database instances consistent with a given schema, and p(I) is the probability associated with any instance I Є X. We note that since p(.) represents the probability vector over all instances in X, we have ∑IЄX p(I) = 1. We note that the above representation is a formalism of the "possible world's model". Probabilistic?-tables are a simple way of representing probabilistic data. In this case, one models the probability that a particular tuple is present in the database. Thus, the probability of a particular instantiation of the database can be

defined as the product of the probabilities of the corresponding set of tuples to be present in the database with the product of the probabilities of the complementary set of tuples to be absent from the database.

B. Simplifying Assumptions in Practical Applications

The above definitions are fairly general formalisms for probabilistic data. In many practical applications, one may often work with simplifying assumptions on the underlying database. One such simplifying assumption is that the presence and absence of different tuples is probabilistically independent. In such formalism, all possible probability distributions on possible worlds are not captured with the use of independent tuples. This is referred to as incompleteness. Furthermore, one needs to be careful in the application of such formalism, since it may result in inconsistency. Our discussions above can be summarized in terms of the major classes of uncertainty that most applications work with. Broadly, most applications work on two kinds of uncertainty:

1) Existential uncertainty

2) Attribute level uncertainty

A. Recent Projects

A number of recent projects have designed uncertain databases around specific application requirements. For example, the Conquer project introduced query rewriting algorithms to extract clean and consistent answers from unclean data under possible world's semantics. Methods are also proposed to derive probabilities of uncertain items. One of the key aspects of the Conquer project is that it permits real time and dynamic data cleaning in such a way that clean and consistent answers may be obtained for queries. Another example of such a database is the Orion project which presents query processing and indexing techniques in order to manage uncertainty over continuous intervals. Such application-specific databases are designed for their corresponding domain, and are not very effective in extracting information from "possible worlds" semantics. A recent and interesting line of models for uncertain data is derived from the Trio project at Stanford University. This work introduces the concept of Uncertainty- Lineage Database (ULDB), which is a database with both uncertainty and lineage. We note that the introduction of lineage as a first-class concept within the database is a novel concept which is useful in a variety of applications such as query processing. The basic idea in lineage is that the model keeps track of the sources from which the data was acquired and also keeps track of its influence in the database.

B. Extensions to Semi structured and XML Data

Recently, uncertain data models have also been extended to semi structured and XML data. XML data poses numerous unique challenges. Since XML is structured, the probabilities need to be assigned to the structural components such as nodes and links. Furthermore, element probabilities could occur at multiple levels and nested probabilities within a sub tree must be considered. Furthermore, incomplete data should be handled gracefully since one may not insist on having complete probability distributions. Another unique issue in the case of XML data is that the probabilities in an ancestor descendent chain are related probabilistically. In the most general case, this can lead to issues of computational intractability. The approach is to model some classes of dependence (e.g., mutual exclusion) which are useful and efficient to model. The work also designs techniques for a restricted class of queries on the data. Another interesting approach to probabilistic XML data construction. In this technique, probabilistic XML trees are constructed in order to model the structural behavior of the data.

III. UNCERTAIN DATA MANAGEMENT APPLICATIONS

In this section, we will discuss the design of a number of data management applications with uncertain data. These include applications such as query processing, Online Analytical Processing, selectivity estimation, indexing, and join processing. We will provide an overview of the application models and algorithms in this section.

A. Query Processing of Uncertain Data

In traditional database management, queries are typically represented as SQL expressions which are then executed on the database according to a query plan. As we will see, the incorporation of probabilistic information has considerable effects on the correctness and computability of the query plan.

1) ***Intentional and Extensional Semantics:-*** A given query over an uncertain database may require computation or aggregation over a large number of possibilities. In some cases, the query may be nested, which greatly increases the complexity of the computation. There are two broad semantic approaches used:

a) Intentional semantics

b) Extensional semantics

1) *Queries with Correlations*

2) *Top-k Query*

3) *The OLAP Model*

a) Consistency

b) Faithfulness

4) *Correlation-preservation*

5) *Query allocation*

6) *Aggregating uncertain measures*

B. Indexing Uncertain Data

The problem of indexing uncertain data arises frequently in the context of several application domains such as moving trajectories or sensor data. In such cases, the data is updated only periodically in the index, and therefore the current attribute values cannot be known exactly; they can only be

estimated. There are many different kinds of queries which can be resolved with the use of index structures:

1) *Range queries*

2) *Nearest neighbor queries*

3) *Aggregate queries*

4)*Moving Object Environments*

Definition 1. An uncertainty region $U_i(t)$ of an object O_i at

Time t is a closed region such that O_i can be found only in this region.

Definition 2. The uncertainty density function $f_i(x, y, t)$ is the probability density function of the object O_i at location (x, y) and time t. This uncertainty function has a value of 0 outside $U_i(t)$.

The technique for processing a probabilistic nearest neighbor query involves evaluating the probability of each object being closest to the query point. One of the key challenges of the nearest neighbor query is that unlike the probabilistic range query, one cannot determine the probability for an object independent of the other points. The solution basically comprises the steps of projection, pruning, bounding, and evaluation. These steps are summarized as follows:

a) Projection

b) Pruning phase

c) Bounding phase

d) Evaluation phase

e). Probabilistic Threshold Queries

Definition 3. Given a closed interval [c, d], where c, d $\in$ R and $c \leq d$, a probabilistic threshold query returns a set of tuples T_i, such that the probability p_i that $T_i.a$ is inside [c, d], is greater than or equal to p, where $0 \leq p \leq 1$. We note that $T_i.a$ represents the probability attribute of tuple T_i. Thus, a probabilistic threshold query can be treated as a range query, which operates on probabilistic uncertainty.

Definition 4. An x-bound of an MBR/uncertainty interval M_j is a pair of lines, namely left-x-bound (denoted by $M_j.lb(x)$) and right-x-bound (denoted by $M_j.rb(x)$). Every uncertain object contained in this MBR is guaranteed to have a probability of at most x (where $0 \leq x \leq 1$) of being left of the left-x-bound and also guaranteed to have a probability of at most x of being right of the right-x-bound. We note that this kind of bound is a generalization of the concept of the MBR. This is because the MBR of an internal node can be viewed as a 0-bound. This is because it guarantees that all intervals in the node are contained in it with probability 1.The purpose of storing the information of the x-bound of a node is to avoid investigating the contents of a node. This saves I/O costs during index exploration. The presence of the x-bound allows us to decide whether an internal node contains any qualifying MBRs without further probing into the sub trees of this node. Let p be the threshold probability for the query. The two necessary pruning conditions (both conditions must hold) for node M_j to be pruned with the use of the x-bound are as follows:

- Mj can be pruned if [a, b] does not intersect leftx-bound or right-x-bound of M_j, i.e., either $b < M_j.lb(x)$ or $a > M_j.rb(x)$.
- $P \geq x$.

In the event that the above conditions do not hold, the internal contents of node Mj are examined and further exploration of the tree is resumed. It has been that the probability threshold query (PTQ) index is quite efficient when the threshold p is fixed a priori across all queries. When the threshold p varies, then the index continues to be experimentally efficient on the average, though the actual behavior mat vary quite a bit across different queries.

C. Uncertain Categorical Data

The definition used for the categorical data domain is as follows:

Definition 1. Given a discrete categorical domain $D = \{d_1 \ldots d_N\}$, an uncertain discrete attribute (UDA) u is a probability distribution over D. It can be represented by the probability vector $u.P = \{p_1, \ldots, p_N\}$ such that $Pr(u = d_i) = u.p_i$. The probability that two uncertain attribute values are equal can be computed by calculating the corresponding equality probability over all possible uncertain values. Therefore, we have the following.

Observation 1. Given two UDAs u and v, the probability that they are equal is given by $Pr(u = v) = \sum u.pi \times v:pi$. Analogous to the notion of equality of value is distributional similarity. The distance function may be defined in terms of the L1 function, the L2 function, or the Kullback-Leibler distance function. The kinds of queries resolved by the technique are as follows:

1) Probabilistic equality query (PEQ):- Given a UDA q, and a relation R with a UDA a, the query returns all tuples t from R along with probability values, such that the probability value $Pr(q = t.a) \geq 0$.

2) Probabilistic equality threshold query (PETQ):- Given a UDA q, a relation R with UDA a and a threshold r, $r \geq 0$. The answer to the query is all tuples t from R such that Pr(q = t.a).

3) Distributional similarity threshold query (DSTQ):- Given a UDA q, a relation R with UDA a, a threshold r_d, and a divergence function F, DSTQ returns all tuples t from R such that $F(q, t.a) \leq r_d$.

4) Probabilistic equality threshold join (PETJ):- Given two uncertain relations R, S, both with UDAs a, b, respectively; relation $R \times_{Ra = Sb} S$ consists of all pairs of tuples r, s from R, S, respectively, such that $Pr(r.a = s.b) \geq r$.

5)*Probabilistic Distribution R-Tree:* - Next, we will discuss the probabilistic distribution R-Tree which is an alternative for indexing UDAs. The broad approach is to index the vector of probability values of the possible attribute values. Thus, if there is N possible probability values then, data points are created in RN. One distinction from traditional R-Trees is that the underlying queries have very different semantics. The uncertain queries are hyper plane queries on the N-dimensional cube. The MBRs of this R-Tree are thus defined in terms of the corresponding Probability values. This ensures that the essential pruning properties of R-Trees are maintained. For example, for the case of probabilistic threshold query, one can compute the maximum probability of equality for any node in the sub tree by taking the maximum dot product of the target object probabilities with the corresponding probability vector from the MBR. When this value is less than the user-specified threshold, the corresponding sub tree can be pruned. The results suggest that neither of the two techniques emerges as a clear winner, and either of the techniques may perform.

D. Join Processing on Uncertain Data

In the case of join processing, techniques have been developed for probabilistic join queries and similarity joins. In the case of probabilistic join queries, it is assumed that each item is associated with a range of possible values and a probability density function, which quantifies the behavior of the data over that range. The range of values associated with the uncertain variable is denoted by a.U = [a.l, a.r]. Thus, a.l is the lower bound of the range and a.r is the upper bound of the range. By incorporating the notion of uncertainty into data values, imprecise answers are generated. Each join-pair is associated with a probability to indicate the likelihood that the two tuples are matched. A second kind of join is the similarity join. Similarity is measured by the distance between the two feature vectors. The join is performed based on this distance.

1) Probabilistic Join Queries:- Since each tuple-pair is probabilistic in nature, the join may contain a number of false positives which are typically those pairs which are associated with probability values. Each tuple-pair is associated with a probability that indicates the likelihood of the join. In order to compute these probability values, the notions of equality and inequality need to be extended to support uncertain data. We note that those join-pairs which have low probability can be discarded. This variant of probabilistic join queries is referred to as Probabilistic Threshold Join Queries. We note that the use of thresholds reduces the number of false positives, but it may also result in the introduction of false negatives. Thus, there is a tradeoff between the number of false positives and false negatives depending upon the threshold which is chosen. The reformulation of the join queries with thresholds is also helpful in improving the performance requirements of the method. A number of pruning techniques are developed in order to improve the effectiveness of join processing. These Pruning techniques are as follows:

a) Item-level pruning: - In this case, two uncertain values are pruned without evaluating the probability.

b) Page-level pruning: - In this case, two pages are pruned without probing into the data stored in each page.

c) Index-level pruning: - In this case, the data which is stored in a sub tree is pruned.

We note that a key operator in the case of joins is that of equality, since a join is performed only when the corresponding attribute values are equal. For the case of continuous data with infinitesimal resolution, this is never the case since any of the pair of attributes can take on an infinite possible number of values. Therefore, a pair of attributes is defined to be equal to one another within acceptable resolution c, if one attribute value is within c of another. Let a and b be the two join attributes. Let a.f(x) and b.f(x) represent the corresponding probability density and cumulative density functions, respectively. Correspondingly, the probability can be calculated as follows:

$P(a =_c b) = \int a.f(x) \,.\, (b.F(x + c) - b.F(x - c))dx$ ----------1.

For the case of the > and < operators, it is not necessary to use the resolution, and it is possible compute the corresponding probability of inequality P(a > b) and P(a < b) in a straightforward way. In order to evaluate the join, common block-nested-loop and indexed-loop can be used. The advantage of these algorithms is that they have been implemented in most database systems, and therefore only a small amount of modification is required in order to support the joins. The main difference is to use the uncertainty information in order to compute the probability of equality. For the use of probability density functions such as the uniform or the Gaussian function, closed form formulas may be obtained in order to determine the probability of equality. Subsequently, those pairs with probability less than the required threshold can be pruned. We note that the computations of the probability of a join can sometimes be expensive when the probabilistic computations cannot be expressed in closed form. Therefore, it is often useful to be able to develop quick pruning conditions in order to exclude certain tuple pairs from the join. Suppose a and b are uncertain valued variables and $a.U \cap b.U \neq \emptyset$. Let la, b, c, be max{a.l- c, b.l- c}, and let ua, b, c be min{a.r + c, b.r + c}. For equality and inequality, the following pruning conditions hold true:

- $P(a =_c b)$ is at most $\min\{a.F(u_{a,b,c}) - a.F(l_{a,b,c}), b.F(u_{a,b,c}) - b.F(l_{a,b,c})\}$.
- Correspondingly, it is easy to see that $P(a \neq_c b)$ is at least equal to the complement of the above expression.

We note that the above expressions can be computed easily as long as the cumulative density function of the expression is available either in closed or numerical form. We note that a tuple pair can be eliminated when the probability of equality is less than the user-defined threshold. We further note that in some cases, it may not be necessary to report the explicit probabilities of the tuple joins, as long as all tuples whose join

probability is above the user-defined threshold are reported. For such cases, it is only necessary to determine whether the required probability lies above a given threshold. For such cases, we can use another pruning condition. For a pair of uncertain-valued variables a and b, it is possible to compute a bound on the corresponding probability that one is greater than the other. Specifically, the bounds are as follows:

- If $a.l \leq b.r < a.r$, $P(a > b) \geq 1 - a.F(b.r)$.
- If $a.l \leq b.l \leq a.r$, $P(a > b) \leq 1 - a.F(b.l)$.

The above two inequalities can be used for those join tuples which satisfy the preconditions described above? Depending upon the direction of the inequality, one can immediately include or exclude the corresponding join tuples from the inequality. We note that in many of these join processing algorithms, the unit of retrieval is a page from an index structure. In such cases, one can prune the entire node of the index tree by constructing bounds on the join behavior of the nodes in the tree. By using this approach, either page-level pruning can be achieved, or index-level pruning can be achieved by using an inner level node in the index tree. A concept called the x-bound is used to augment the nodes of the underlying index structure.

3) *Similarity Join:* - The most popular similarity join is the distance-range join. In the distance-range join, we perform the join between two records, if the distance between the two does not exceed a user-defined parameter ϵ. The natural generalization for the case of uncertain data is to compute the expected distance between two relations, and perform the join if this expected distance is less than the parameter ϵ. This may result in considerable inaccuracies in the join computation process. This is because the expected distances are often skewed by the behavior of the tail end behavior of the probability functions of different attributes. Thus, the expected distances may not reflect the true likelihood that a given pair of records may join on a particular attribute. The result is that different joins which have similar probability of lying within the range of ϵ may be treated inconsistently.

E. Data Integration with Uncertainty

An important application in the context of uncertain data is that of data integration. In order to do so, the work introduces the concept of probabilistic schema mappings. These are defined as a set of possible (ordinary) mappings between a source schema and a target schema, where each possible mapping has an associated probability. It is suggested that there are two possible interpretations to probabilistic schema mappings. The first (table-specific mapping) assumes that there is a single correct mapping, but we do not know which it is. This single correct mapping applies to all tuples. In the second interpretation (tuple-specific mapping), the mapping depends upon the tuple to which it is applied. A number of algorithms are described for answering queries in the presence of probabilistic schema mappings.

F. Probabilistic Skylines on Uncertain Data

A problem which is quite relevant to the case of uncertain data is that of probabilistic skyline computation. The problem of skyline computation is used in multi criteria decision-making applications. For example, consider the case when statistics of different NBA players are computed, such as the number of assists, rebounds, baskets, etc. It is unlikely that a single player will achieve the best performance in all respects.

Definition 1. For two d-dimensional points $u = (u_1, \ldots, u_d)$ and $v = (v_1, \ldots, v_d)$, u is said to dominate v, if for each $i \in \{1, \ldots, d\}$, we have $u_i \leq v_i$, and for some $i_0 \in \{1, \ldots, d\}$, We have $u_{i0} < v_{i0}$.

Definition 2. Given a set of points S, a point u is a skyline point if there exists no other point $v \in S$ such that v dominates u. The skyline on S is the set of all skyline points.

Definition 3. Given a probability threshold p ($0 \leq p \leq 1$), the P-skyline is the set of uncertain objects, such that each of them has probability of at least p to be in the skyline.

IV. MINING APPLICATIONS FOR UNCERTAIN DATA

Recently, a number of mining applications have been devised for the case of uncertain data. Such applications include clustering and classification. We note that the presence of uncertainty can affect the results of data mining applications significantly. For example, in the case of a classification application, an attribute which has lower uncertainty is more useful than an attribute which has a higher level of uncertainty. Similarly, in a clustering application, the attributes which have a higher level of uncertainty need to be treated differently from those which have a lower level of uncertainty.

A. Clustering Uncertain Data

The presence of uncertainty changes the nature of the underlying clusters, since it affects the distance function computations between different data points. A technique has been proposed in order to find density-based clusters from uncertain data. The key idea in this approach is to compute uncertain distances effectively between objects which are probabilistically specified. The fuzzy distance is defined in terms of the distance distribution function. This distance distribution function encodes the probability that the distances between two uncertain objects lie within a certain user-defined range. Let d(X,Y) be the random variable representing the distance between X and Y. The distance distribution function is formally defined as follows:

Definition 1. Let X and Y are two uncertain records, and let P(X,Y) represent the distance density function between these Objects. Then, the probability that the distance lies within the Range (a, b) is given by the following relationship:

$$P(a \leq d(X,Y) \leq b) = \int p(X,Y)(z)dz \quad \text{----------2.}$$

Based on this technique and the distance density function, the method defines a reach ability probability between two data points. This defines the probability that one data point is

directly reachable from another with the use of a path, such that each point on it has density greater than a particular Threshold.

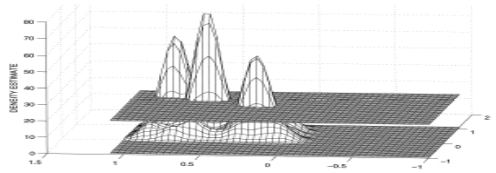

Fig 1. Density-based profile with lower density threshold.

B. Classification of Uncertain Data

A closely related problem is that of classification of uncertain data in which the aim is to classify a test instance into one particular label from a set of class labels. A method was proposed for support vector machine classification of uncertain data. This technique is based on a discriminative modeling approach which relies on a total least squares method. This is because the total least squares method assumes a model in which we have additive noise. However, instead of using Gaussian noise, the technique uses a simple bounded uncertainty model. Such a model has a natural and intuitive geometric interpretation. Note that the support vector machine technique functions by constructing boundaries between groups of data records. Then, the margin created by the support vector machine can be modified by using the uncertainty of the points which lie on the boundary. For example, if points on one side of the boundary have greater uncertainty, this influences the way in which the margins are adjusted by the classifier.

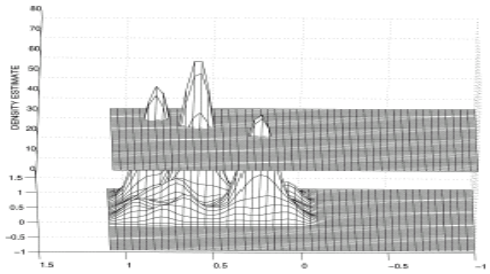

Fig. 2. Density-based profile with higher density threshold.

C. Frequent Pattern Mining

The problem of frequent pattern mining has also been explored in the context of uncertain data. In this model, it is assumed that each item has an existential uncertainty in belonging to a transaction. This means that the probability of an item belonging to a particular transaction is modeled in this approach. In this case, an item set is defined to be frequent, if its expected support is at least equal to a user specified threshold. In order to solve this version of the frequent pattern mining problem, the U-Apriori algorithm is proposed which essentially mimics the Apriori algorithm, except that it performs the counting by computing the Expected support of the different item sets. The expected support of a set of items in a transaction is obtained by simply multiplying the probabilities of the different items in the transaction. The approach can be made further scalable by using the concept of data trimming. In the data trimming approach, those items with very low existential probability are pruned from the data. The algorithm is then applied to the trimmed data. It has been shown that this approach can accurately mine the frequent patterns while maintaining efficiency.

D. Outlier Detection with Uncertain Data

The problem of outlier detection has also been extended to the case of uncertain data. In the case of the outlier detection problem, differing levels of uncertainty across different dimensions may affect the determination of the outliers in the underlying data. For example, consider the case in which the contours of uncertainty for two data points X and Y are illustrated in the form of elliptical shapes. The data point X seems to be further away from the overall data distribution as compared to the data point Y. However, the contours of uncertainty are such that the data point X has a greater probability of being drawn from the overall data distribution.

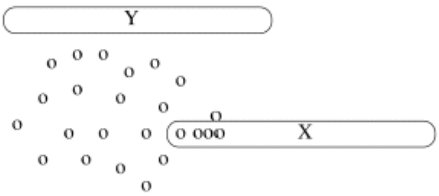

Fig. 3. Effect of uncertainty on outlier detection.

Correspondingly, it is possible to define the concept of an outlier in terms of the probability that a given data point is drawn from a dense region of the overall data distribution. In order to quantify the probability that a given uncertain data point is drawn from a dense region, we define the concept of n-probability. The n-probability of a data point Xi is defined as the probability that the uncertain data point lies in a region with (overall data) density at least n. Since the data point is uncertain, the n-probability may be computed by integrating the density of the data point along the contour of the intersection of the overall density function with threshold n. However, this can be computationally challenging from a numerical point of view. Therefore, the n-probability may be estimated with the use of sampling. The idea is to draw multiple samples from the data and compute the fraction of the samples over which the density threshold is specified. This can be used to define the concept of a (δ, n)-outlier.

V. CONCLUSION

The field of uncertain data management has seen a revival in recent years because of new ways of collecting data which have resulted in the need for uncertain representations. We presented the important data mining and management techniques in this field along with the key representational issues in uncertain data management.

REFERENCES

[1] Managing and Mining Uncertain Data, C. Aggarwal, ed. Springer, 2009.
[2] L. Antova, C. Koch, and D. Olteanu, "From Complete to Incomplete Information and Back," Proc. ACM SIGMOD, 2007.
[3] L. Antova, T. Jansen, C. Koch, and D. Olteanu, "Fast and Simple Relational Processing of Uncertain Data," Proc. 24th IEEE Int'l Conf. Data Eng. (ICDE), 2008.
[4] C.C. Aggarwal and P.S. Yu, "Outlier Detection with Uncertain Data," Proc. SIAM Int'l Conf. Data Mining (SDM), 2008.
[5] C.C. Aggarwal, "On Unifying Privacy and Uncertain Data Models," Proc. 24th IEEE Int'l Conf. Data Eng. (ICDE), 2008.
[6] C.C. Aggarwal and P.S. Yu, "A Framework for Clustering Uncertain Data Streams," Proc. 24th IEEE Int'l Conf. Data Eng. (ICDE), 2008.

