Managing Uncertain Data: Probabilistic Approaches

Wenjie Zhang [#1], Xuemin Lin [#2], Jian Pei [*3], Ying Zhang [#4]

[#] *University of New South Wales & NICTA, Australia*
[1] zhangw@cse.unsw.edu.au
[2] lxue@cse.unsw.edu.au
[4] yingz@cse.unsw.edu.au

[*] *Simon Fraser University, Canada*
[3] jpei@cs.sfu.ca

***Abstract*— Uncertain data are inherent in many important applications. Recently, considerable research efforts have been put into the field of managing uncertain data. In this paper, we summarize existing techniques to query and model uncertain data and systems that effectively manage uncertain data, mainly from a probabilistic point of view.**

I. Introduction

Managing uncertain data has been studied ever since the eighties last century from the database society. With the emergence of many recent important and novel applications involving uncertain data, there has been a great deal of research attention dedicated to this field. The applications include data cleaning, data integration, information extraction, sensor networks, economic decision making, market surveillance, trend prediction, moving object management, etc. Uncertainty is inherent in such applications due to various factors such as data randomness and incompleteness, limitation of equipment, and delay or loss in data transfer.

Note that in this paper, we do not distinguish between *imprecise* and *uncertain* data and use the term *uncertain* for the reason of simplicity. To be precise, imprecision means information available is not specific enough, for instance, the temperature outside is *between* 35 *and* 38 centigrade (interval); it is 35 *or* 38 centigrade (disjunction); it is *not* 20 centigrade (negative); or we simply do not know the outside temperature. On the other hand, uncertainty indicates it is impossible to determine wether information available is true or not. For instance, the temperature *may be* 38 centigrade [34].

Managing uncertain data is not well supported by conventional database systems. A number of technical issues in traditional databases have been reinvestigated recently under uncertain semantics, including modeling uncertainty, query evaluation, indexing, query processing against relational and spatial uncertain data. Many important results have been obtained in system and theory. Figure 1 summaries the recent breakthroughs in this field regarding three categories, modeling, querying and systems.

In addition to the two survey papers by Dalvi and Suciu [30] and by Aggarwal and Yu [5] on querying and mining uncertain databases, a tutorial on querying uncertain data is presented by Pei *et al* in [58]. In this paper, we present a comprehensive, concise survey that covers relational and multi-dimensional spatial uncertain data management as well as existing database systems that support uncertain data management.

The rest of this paper is organized as follows. In Section II, we introduce existing models to represent uncertainty. Section III summarizes various query types defined under uncertain semantics and corresponding techniques. Section IV briefly overviews available systems that support uncertainty management. We conclude the paper in Section V with a brief discussion of future research work in the field.

II. Modeling Uncertainty

The uncertainty of an object can be specified by three models [73]: fuzzy model [34], evidence-oriented model [47], [49] and probabilistic model [63]. In fuzzy models, fuzzy entities, fuzzy attributes, fuzzy relationship, fuzzy aggregation, fuzzy constraints, etc are used to model uncertainty and imprecision. In evidence-oriented models, the Dempster-Shafer Theory of Evidence is applied to model uncertainty and imprecision. Probabilistic models specify uncertainty with probability values and are widely used. In this paper we mainly discuss probabilistic models.

Most frequently used granularities to specify uncertainty are group-based (or table-based), object-based (or record-based) and attribute-based [73]. A group-based approach concerns the "coverage" of the group such as how much percent of objects in this group is present; an object-based approach assigns appearance probability to each object in the group; in the attribute level, an attribute of a tuple is associated with probability distribution information describing a set of possible values. Object-level uncertainty is more attractive for various reasons such as it results in relations that are usually 1NF and it is easier to store and operate on [67]. In this section, we firstly introduce object-based uncertainty specification using *possible world* semantics [1], [39], followed by advanced models capturing uncertain characteristics in databases.

A. Object-based Uncertainty Model

Independent Model. Suppose in an uncertain data set D an object (record) R has probability $P(R)$ ($P(R) > 0$) to occur and all objects are independent. A *possible world* W is a subset of D and each object $R \in D$ with $P(R) = 1$ must be included in W. Clearly, the occurrence probability of a possible world is $P(W) = \prod_{R \in W} P(R) \cdot \prod_{R \notin W} (1 - P(R))$. Let $\mathcal{W}$ be the

978-0-7695-3185-4/08 $25.00 © 2008 IEEE
DOI 10.1109/WAIM.2008.42

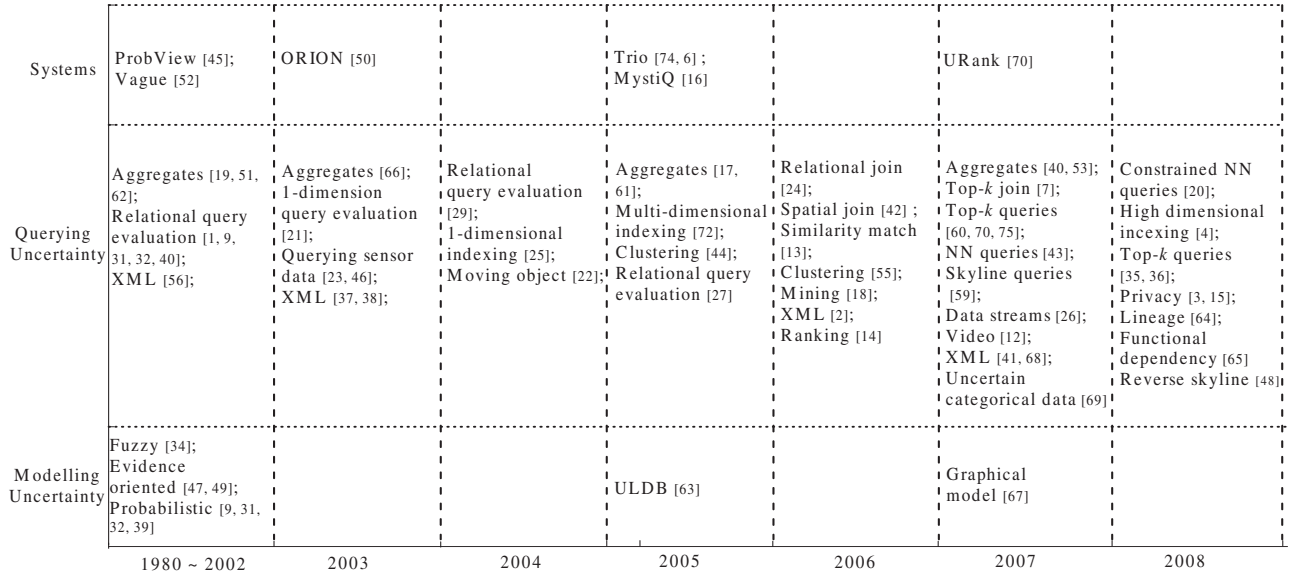

Fig. 1. Research breakthroughs in managing uncertainty

set of all possible worlds of D and N be the number of objects with occurrence probability smaller than 1, then $|\mathcal{W}| = 2^N$. The sum of the membership probabilities of all possible worlds in $\mathcal{W}$ equals to 1; that is, $\sum_{W \in \mathcal{W}} P(W) = 1$.

General Model. In a general case, records in a data set may be correlated. A comprehensive study of possible world semantics is conducted by Hua *et al* in [35], [71], [75]. A set of records R_1, ..., R_m are *exclusive* if at most one of them could appear in a possible world and $\sum_{1 \leq i \leq m} P(R_i) \leq 1$ where $P(R_i)$ is the occurrence probability of R_i. A set of exclusive records are also called a *generation rule* $\mathcal{R}$. Occurrence probability of a generation rule $\mathcal{R}$ is the sum of probabilities of all the records involved in $\mathcal{R}$; that is, $P(\mathcal{R}) = \sum_{R \in \mathcal{R}} P(R)$. Note that a generation rule (virtually regarded as an object) could contain only one record and different generation rules are independent. Given a set of m generation rules – $\mathcal{G}_D = \{\mathcal{R}_1, ..., \mathcal{R}_m\}$, a possible world W is defined as an element in $\prod_{\mathcal{R} \in \mathcal{G}'} \mathcal{R}$ where $\mathcal{G}'$ is a subset of $\mathcal{G}_D$, $\mathcal{G}'$ contains every generation rule $\mathcal{R}$ such that $P(\mathcal{R}) = 1$. Let $|\mathcal{R}|$ be the number of records in $\mathcal{R}$. The number of all possible worlds with respect to $\mathcal{G}_D$ is:

$$|\mathcal{W}| = \prod_{\mathcal{R} \in \mathcal{G}_D, P(\mathcal{R})=1} |\mathcal{R}| \prod_{\mathcal{R} \in \mathcal{G}_D, P(\mathcal{R})<1} (|\mathcal{R}| + 1) \quad (1)$$

Occurrence probability of a possible world W is:

$$P(W) = \prod_{\mathcal{R} \in \mathcal{G}_D, \mathcal{R} \cap W \neq \phi} P(\mathcal{R} \cap W) \times \prod_{\mathcal{R} \in \mathcal{G}_D, \mathcal{R} \cap W = \phi} (1 - P(\mathcal{R})) \quad (2)$$

where $P(\mathcal{R} \cap W)$ refers to the occurrence probability of a record which is in both $\mathcal{R}$ and W.

As an example, Table I records the ID of speeding vehicles (*Vehicle* in the table) and speed (*Speed*) captured by sensor nodes (*SID*) at certain location (*Loc.*) and time (*Time*). Each record is given an occurrence probability (*P*) representing its confidence to be true. In this example, records R1 and R2 can not appear in the same possible world; that is, R1 and R2 are exclusive. R3 is independent with them; this means the generation rule containing R3 only is independent with generation rule $\{R1, R2\}$. There are 6 possible worlds in all for this uncertain database, as shown in Table II, along with corresponding occurrence probabilities.

RID	SID	Time	Loc.	Vehicle	Speed	P
$R1$	$S1$	$2:00PM$	$L1$	$HB1235$	120	0.7
$R2$	$S1$	$2:00PM$	$L1$	$HB1238$	150	0.2
$R3$	$S6$	$3:45PM$	$L17$	$HA2568$	170	0.9

TABLE I
SPEEDING VEHICLES RECORDS.

Possible World	Occurrence Probability
$W_1 = \{\phi\}$	0.01
$W_2 = \{R1\}$	0.07
$W_3 = \{R2\}$	0.02
$W_4 = \{R3\}$	0.09
$W_5 = \{R1, R3\}$	0.63
$W_6 = \{R2, R3\}$	0.18

TABLE II
POSSIBLE WORLDS OF TABLE I.

B. Advanced Models

There are a number of advanced models. In this subsection, we introduce three representatives.

- Fuhr and Rolleke model uncertainty based on non-first-normal-form (NF2) [33] where records in a relation are assigned probabilistic weights. Imprecise attribute values are modelled as a probabilistic sub-relation. Moreover, a probabilistic relational algebra (PRA) is proposed as a generalization of standard relational algebra.
  Tuple t in a probabilistic relation modeled by NF2 contains three aspects, its attribute values, an event expression $t.\eta$ and event probability $t.\beta$. As shown in Figure 2 [33], the relation *BOOK* consists of atomic attributes *BNO*, *YEAR*, and attributes *PRICE*, *INDEX*, *AUTHOR* which are modeled by subrelations. Types of probabilistic relations include "deterministic", "independent", "disjoint" and "dependent". For example, subrelation *price* is *disjoint* meaning that one and only one event between $B\hat{E}P1$ and $B\hat{E}P2$ can be true. *INDEX* is *independent* meaning that both $B\hat{E}I1$ and $B\hat{E}I2$ can be true with different confidence and *AUTHOR* is *deterministic* indicating that both values for *NAME* in this subrelation takes the same event probability as the tuple it belongs to, namely, 1.0.
  Clearly, general model introduced earlier can also be used to model such NF2 probabilistic relations, in a clearer and more concise way.
- Sarma *et al* integrates *lineage* to model uncertainty [63]. Linage is associated with a data item carrying information about its derivation. A model ULDBs (Uncertainty-Lineage Databases) is developed by extending standard SQL relational model with the following four aspects [10].
  1) *alternatives* capturing the uncertainty of contents of a tuple.
  2) *maybe* annotations "?" representing the uncertainty about the presence of a tuple.
  3) *confidence* values quantifying the degree of above two types of uncertainties.
  4) *lineage* recording derivation information of tuple alternatives.

  In fact, besides a new ingredient *lineage*, this model is almost identical to the independent model in object-level uncertainty.

  Table III gives an example of ULDBs. Record R1 can be either of the two tuples with different confidence values. R2 exists in this table with confidence 0.9. Table IV captures vehicle ID and driver names. We join these two tables and project on the driver attribute, clearly obtaining only one tuple (John). We call this tuple R5. *Lineage* captures how R5 is derived from the original two tables by a function λ over the alternatives of tuples. $\lambda(R5, 1) = ((R1, 2), (R3, 1))$ means that the first alternative of R5 is derived from joining of the second alternative of R1 and first alternative of R3.
- Sen and Deshpande utilize a probabilistic graphical model [57] to facilitate query evaluation over uncertain data with general forms of correlations [67]. Besides *independence* and *mutual exclusivity*, *implies* and *nxor* are also explored; that is, the presence of one tuple implies absence of other tuples and high positive correlation between two tuples, respectively. Each tuple is associated with a boolean valued random variable X_t, namely *false* and *true*. In the probabilistic graphical model, nodes represent random variables while edges represent correlations. Thus different types of correlations, such as complete independence, mutual exclusivity, positive correlation can be modelled. Query evaluation problem with correlations is then transformed into equivalent problem under probabilistic graphical model and can be solved using existing techniques such as inference algorithms.

RID	(Vehicle ID, Speed)
R1	(HB1235, 120): 0.7 || (HB1238, 120): 0.2
R2	(HA2568, 170): 0.9 ?

TABLE III
VEHICLE AND SPEED.

RID	(Vehicle ID, Driver)
R3	(HB1238, John)
R4	(HC2457, Wendy)

TABLE IV
VEHICLE AND DRIVER.

A lot of research work aims to represent uncertainty besides what we introduced above, for instance [1], [8], [9], [31], [32], [39]. We omit details from the paper due to the space limit.

III. ANALYZING UNCERTAIN DATA

In this section, we introduce existing work on analyzing uncertain data, which can be divided into two groups: relational uncertain data and spatial uncertain objects.

Note that an uncertain object may be described by either a continuous or a discrete case. In *continuous* cases, an uncertain object U may be described by a probability density function (PDF) f_U such that $\int_{u \in U} f_U(u)du = 1$; Nevertheless, in many applications PDFs are not always available. Instead, an uncertain object U is represented by a set of *instances* such that each instance $u \in U$ has a probability $P(u)$ to appear. Such a representation is also referred as a *discrete* case, has the property that $0 < P(u) \leq 1$ and $\sum_{u \in U} P(u) = 1$.

A. Relational Uncertain Data

Query Evaluation. Cheng *et al* present a broad classification of probabilistic queries over one-dimensional uncertain data as well as techniques for evaluating probabilistic queries [21]. There are four types in all, value-based non-aggregates, entity-based non-aggregates, value-based aggregates and entity-based aggregates according to two aspects: 1) the query requires qualifying objects or values and 2) the query is aggregate-based or not. An example of entity-based non-aggregate query is: given an interval $[l, r]$ where $l < r$, return a set of tuples (T_i, P_i) where attribute a of T_i is within the range $[l, r]$ with non-zero probability P_i. Bounding and pruning techniques are deployed to evaluate these queries. In [25], Cheng *et al*

BOOK												
η	β	BNO	YEAR	PRICE			INDEX			AUTHOR		
				η	β	VAL	η	β	TERM	η	β	NAME
$B\hat{E}$	1.0	1	92	$B\hat{E}P1$ $B\hat{E}P2$	0.6 0.4	30 25	$B\hat{E}I1$ $B\hat{E}I2$	0.9 0.8	IR DB	$B\hat{E}A\hat{E}$ $B\hat{E}A\hat{E}$	1.0 1.0	Smith Jones
$B\hat{E}$	1.0	2	93	$B\hat{E}P3$	1.0	29	$B\hat{E}I3$ $B\hat{E}I4$	0.9 0.8	AI DB	$B\hat{E}A\hat{E}$	1.0	Miller
$B\hat{E}$	1.0	3	92	$B\hat{E}P4$ $B\hat{E}P5$	0.7 0.3	28 25	$B\hat{E}I5$	0.9	DB	$B\hat{E}A\hat{E}$	1.0	Jones
$B\hat{E}$	1.0	4	90	$B\hat{E}P6$ $B\hat{E}P7$	0.5 0.5	32 28	$B\hat{E}I6$	0.9	DB	$B\hat{E}A\hat{E}$	1.0	Jones

Fig. 2. Relation BOOK

explore access methods to also support range search for one dimensional data only.

A series of work has been done by Dalvi and Suciu from University of Washington to evaluate probabilistic queries. In [29], they tackle the problem of evaluating queries with uncertain predicates. Optimization algorithms that can evaluate efficiently most queries are presented. They also show that the evaluation of some queries is $\#P\text{-}complete$; these queries are approached in two different methods: a heuristic avoiding significant errors and a Monte-Carlo simulation algorithm with precision guarantees. In [27], they propose to answer queries from statistic and probabilistic views. In [28] a very clean and complete theoretical result is provided that the complexity of evaluating conjunctive queries over uncertain data set is either $PTIME$ or $\#P\text{-}complete$.

Sen and Deshpande utilize probabilistic graphical model to approach the same problem in uncertain data sets with correlated tuples as introduced in Section II [67].

Aggregate Queries. A most recent work on aggregate query processing over relational uncertain data is from Stanford InfoLab as a function supported by their system Trio [53]. Five types of aggregate operators are tackled, *COUNT*, *MIN*, *MAX*, *SUM* and *AVG*. Among them, *COUNT*, *MIN* and *MAX* are relatively easy and there exist polynomial algorithms [35]. However, it is shown in [30] that results for *SUM* and *AVG* may be different in each possible world and computing *SUM* or *AVG* is #P-complete. Three approximate alternatives are proposed to avoid exhaustively materialize all possible results caused by "exact" aggregation in uncertain databases: lowest possible value, highest possible value and expected value. For instance, lowest possible value of SUM (*LSUM*) is defined as the sum of lowest value from each uncertain object (e.g., sets of tuples from probabilistic table that are governed by a generation rule). Specifically, expected-average (*EAVG*) value is approximated using expected-sum (*ESUM*) divided by expected-count (*ECOUNT*). Transformed aggregate queries are processed using TriQL techniques used in Trio which is an extension of SQL.

A thorough and fundamental study of OLAP against uncertain and imprecise data has been conducted in [17]. Other major work may be found in [19], [40], [51], [61], [62], [66].

Join Queries. Join queries over one dimensional uncertain data are defined by Cheng *et al* in [24] in a continuous case. Uncertainty over a data item a is parameterized with an uncertainty interval $a.U$ and PDF $a.f(x)$. Uncertainty comparison operators, *equality*, *inequality*, *greater than* and *less than* are defined in a continuous fashion. Take *equality* between two uncertain items a and b as an example. Since the PDFs for both a and b are continuous, the probability that a equals b could be infinitesimally small. A new parameter $resolution$ (c) is introduced to avoid this: a equals b if they are within c distance i.e., $|a - b| \leq c$. The probability that a equals b with resolution c is defined as:

$$P(a =_c b) = \int_{-\infty}^{+\infty} a.f(x) \cdot (b.F(x + c) - b.F(x - c))dx \qquad (3)$$

where $b.F(x)$ denotes the cumulative distribution function (CDF) of b.

Denote θ_u as a uncertainty comparison operator and R and S are uncertain data sets; probabilistic join query (PJQ) returns all pairs of tuples (R_i, S_j) with $P(R_i \theta S_j) > 0$, where $R_i \in R, S_j \in S$. Probabilistic threshold join query (PTJQ) further imposes a probability threshold and only uncertain item pairs with matching probability value no less than this threshold satisfy PTJQ. Based on this threshold, pruning techniques in different indexing levels are proposed to answer PTJQ.

A recent work on join queries on uncertain data is given in in a top-k fashion [7] by Agrawal and Widom. In such confidence-aware joins, only results with top-k matching confidence will be output.

Top-k Queries. Top-k queries are important in analyzing uncertain data. Unlike a top-k query over certain data which returns the k best alternatives according to a ranking function, a top-k query against uncertain data has inherently more sophisticated semantics. Soliman *et al* [71] first relate top-k queries with uncertain data. They define two types of important queries - $U\text{-}Topk$ and $U\text{-}kRank$, regarding discrete cases.

U-Top$k$ returns a set of k records which as a whole have the highest probability to be the top-k results in all possible

worlds. A precise definition is as follows [71].

> Let D be an uncertain data set with possible worlds space $\mathcal{W} = \{W_1, ..., W_n\}$. Let $\mathcal{T} = \{T^1, ..., T^m\}$ be a set of k-length record vectors, where for each $T^i \in \mathcal{T}$: (1) records of T^i are ordered according to scoring function $\mathcal{F}$, and (2) T^i is the top-k answer for a non empty set of possible worlds $W(T^i) \subseteq \mathcal{W}$. A *U-Topk* query, based on *F*, returns $T^* \in \mathcal{T}$, where T^* = $argmax_{T^i \in \mathcal{T}}(\sum_{w \in W(T^i)} P(w))$.

U-*k*Rank retrieves k ordered records where the i-th record has the highest probability of ranking in the i-th position among all possible worlds [71].

> Let D be an uncertain data set with possible worlds space $\mathcal{W} = \{W_1, ..., W_n\}$. For $i = 1, ..., k$, let $\{x_i^1, ..., x_i^m\}$ be a set of records, where each record x_i^j appears at rank i in a non empty set of possible worlds $W(x_i^j) \in \mathcal{W}$ based on scoring function $\mathcal{F}$. A *U-kRanks* query, based on $\mathcal{F}$, returns $\{x_i^*; i = 1, ..., k\}$, where x_i^* = $argmax_{x_i^j}(\sum_{w \in W(x_i^j)} P(w))$.

Methods proposed in [71] navigate all possible states of the search space, meanwhile minimizing the number of tuples accessed. Based on novel observations, Yi *et al* [75] significantly improve the efficiency while tackling the same queries.

Threshold based top-k queries defined by Hua *et al* [35], [36] aim to retrieve all records whose probability of being top-k results in all possible worlds is no less than a given probability threshold. Re *et al* [60] deal with query evaluation on probabilistic database and results are ranked according to the probability of satisfying a given query.

B. Multi-dimensional Spatial Uncertain Data

Range Queries. The first index structure supporting range queries on multi-dimensional spatial uncertain data with arbitrary PDFs is U-tree [72]. U-tree is a novel modification of R-tree to facilitate a set of new pruning and validating techniques. A d-dimensional uncertain object U is modeled using a d-dimensional uncertain region $U.ur$ and probability density function $U.pdf(x)$. Suppose the query region of a range query Q is r_Q, the appearance probability of U in r_Q is defined as:

$$P_{app}(U, Q) = \int_{U.ur \cap r_Q} U.pdf(x)dx \tag{4}$$

where $U.ur \cap r_Q$ is the intersection of $U.ur$ and r_Q. Given a probability threshold p, uncertain objects with $P_{app}(U, Q) \geq p$ are retrieved by the range query.

The basic idea to build a U-tree is illustrated in Figure 3 where polygon $U.ur$ is the uncertain range of 2-dimensional uncertain object U. For a given probability p_1, in each dimension, two lines are calculated. In the horizontal dimension, U has the probability p_1 to occur on the left side of line l_{1-}, also the probability p_1 to occur on the right side of line l_{1+}. Similarly, l_{2-} and l_{2+} are calculated in the vertical dimension. The shadowed region forms the probability constrained region (PCR) of U with respect to p_1. Such a region is used to prune or validate objects. There are multiple PCRs computed beforehand to facilitate range query processing, as shown in Figure 4. To tradeoff between space costs and pruning/validating abilities, a U-tree structure is constructed based on the approximation of such polygons.

In [12], [14], Böhm *et al* study range queries with the constraint that instances of uncertain objects follow Gaussian distribution. Results are ranked according to the probability of satisfying range queries. A more recent work addressing indexing high dimensional uncertain data is [4].

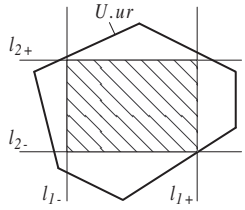

Fig. 3. Pruning/Validating in U-tree.

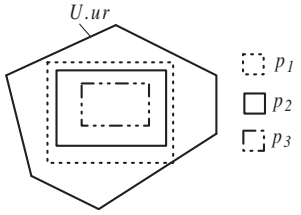

Fig. 4. Multiple PCRs.

Nearest Neighbor Queries. The problem of nearest neighbor query on uncertain objects is tackled in [43]. In a discrete case, both uncertain objects and a query object are represented by a set of s sampled instances. The probability that uncertain object U is the nearest neighbor of query object Q $P_{nn}(U, Q)$ is defined based on the instance pairs from U and Q.

$$P_{nn}(U, Q) = \frac{\sum_{i,j \in 1...s} P_{nn}(u_i, q_j)}{s^2}$$

where $P_{nn}(u_i, q_j)$ is the probability that instance $u_i \in U$ is the nearest neighbor of instance $q_j \in Q$. $P_{nn}(U, Q)$ in continuous cases is computed based on the probabilistic distance between Q and U and the probabilistic distance between Q and other objects except U.

To facilitate query processing, instances inside an object are clustered into several groups bounded by minimal bounding boxes (MBRs) and indexed by R-tree. Thus higher level pruning and validating measures can be applied.

Constrained nearest neighbor query is studied in [20] with a pre-given probability threshold. Only objects with probability no less than this threshold of being nearest neighbor of the query object will be output.

Skyline Queries. For two points u and v in a multi-dimensional space, u *dominates* v ($u \prec v$) if in each dimension the coordinate of u is not greater than that of v and there is one dimension in which the coordinate of u is smaller than v. For a given data set, the *skyline* operator returns all points in the data set which are not dominated by other points. As illustrated in Figure 5, skyline points are a, b and c since they are not dominated by any other points. Skyline operator over uncertain objects is more complex since it involves sophisticated analysis of probability distribution of each uncertain object. As in Figure 6, generally instances in each uncertain object have different dominating ability. This problem is firstly approached

by Pei *et al* in [59]. In a continuous case, suppose that f is the PDF of uncertain object U in the data space $\mathcal{D}$, the probability for U to be a skyline object is:

$$Pr(U) = \int_{u \in \mathcal{D}} f(u) \prod_{\forall V \neq U} (1 - \int_{v \prec u} f'(v)dv)du \quad (5)$$

Here $\prod_{V \neq U}(1 - \int_{v \prec u} f'(v)dv)$ is the probability that the point $u \in U$ is not dominated by any uncertain objects. f' denotes the PDF of V. In a discrete case, the skyline probability of U is:

$$Pr(U) = \sum_{u \in U}(P(u) \times \prod_{\forall V \neq U} (1 - \sum_{v \in V, v \prec u} P(v))) \quad (6)$$

$\prod_{V \neq U}(1 - \sum_{v \in V, v \prec u} p(v))$ is the probability that $u \in U$ is not dominated by any other objects. Recall that $P(u)$ denotes the appearance probability of instance u.

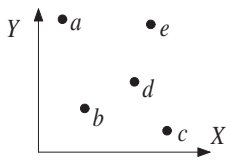

Fig. 5. Certain objects.

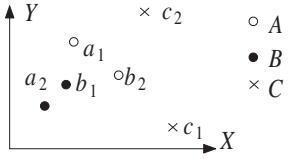

Fig. 6. Uncertain objects.

Bounding-pruning-refining iteration is deployed to achieve efficiency. Two algorithms, bottom-up and top-down, are developed. The bottom-up algorithm computes $P(U)$ from instance level. After calculating skyline probabilities of some selected instances, these values are used to prune other instances and objects. Top-down algorithm, on the other hand, partitions instances of one uncertain object into several groups and apply pruning techniques in the group and object level.

A variation of uncertain skyline, monochromatic and bichromatic reverse skyline search over uncertain objects, is studied in [48].

Similarity Joins. Kriegel *et al* study similarity joins on uncertain spatial objects in [42]. The probability that distance between two uncertain objects U and V is within a range $[a, b]$ is defined as,

$$P(a \leq d(U, V) \leq b) = \int_a^b f_d(U, V)(x)dx \quad (7)$$

where $f_d(U, V)$ is the probabilistic distance function between U and V. Although $f_d(U, V)$ may be computed directly for some uncertain object representations, for efficiency reasons, Kriegel *et al* propose algorithms based on Monte-Carlo sampling technique where each uncertain object is represented by a set of s sampled instances. In this case, the similarity join probability between U and V is defined as follows. Assume all instances in an uncertain object take the same probability to appear

$$P(d(U, V) \leq \epsilon) = \frac{|\{(u_i, v_j)|d(u_i, v_j) \leq \epsilon, 1 \leq i, j \leq s\}|}{s^2}$$

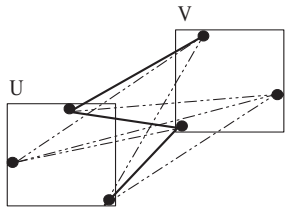

Fig. 7. Probabilistic similarity join

Here $P_{d(u_i,u_j) \leq \epsilon}$ denotes the probability that the distance between u_i and v_j is not greater than ϵ, where $u_i \in U$ and $v_j \in V$. As shown in Figure 7 where there are totally 9 pairs of instances, only the distances of three pairs of instances connected with solid line are smaller than a given distance threshold; consequently $P(d(U, V) \leq \epsilon)$ = 1/3. In their algorithms, instances are also grouped and indexed using R-tree. Then effective pruning techniques based on ϵ are applied. For any two input uncertain objects and distance threshold ϵ, the similarity join probability between these two objects regarding ϵ will be output by their algorithms.

In [13], a similar problem – *similarity matching* is investigated. In their settings, uncertainty of feature vectors follow Gaussian distribution. A novel index structure, Gauss-tree, is developed for similarity matching processing.

Clustering and Mining. Clustering uncertain objects is addressed in [44]. The distance between two uncertain objects is the same as Equation(7). Key concepts in density-based clustering on uncertain objects, such as *core objects*, *core object probability*, and *reachability probability* among objects are defined where a core object has a dense neighborhood and both core object probability and reachability probability are derived based on the Equation(7), respectively. Novel density-based clustering algorithm $\mathcal{F}DBSCAN$ is then developed based on these new concepts.

Ngai *et al* address the same problem in [55] using clustering algorithm based on the traditional K-mean algorithm. Different from the probabilistic distance functions in [44], distance values used between a pair of uncertain objects or between an uncertain object and a cluster are *expected* values. For arbitrary PDF, such *expected values* often involve expensive numerical integration calculation. Pruning techniques are also proposed to avoid such an expensive step. Chau *et al* tackle the problem of mining uncertain data in [18] as an extension of the techniques proposed in [55].

C. Other Aspects in Analyzing Uncertain Data

Query evaluation over uncertain data has also been studied against other applications, such as data streams [26], sensor networks [23], [46], moving objects [22], video retrieval [12], XML data [2], [38], [37], [41], [56], [68], categorical data [69], etc. Theoretical problems such as functional dependency [65] and confidence computation [64] analysis over uncertain relation data have also been addressed. Privacy issues in uncertain semantics are studied in [3], [15]

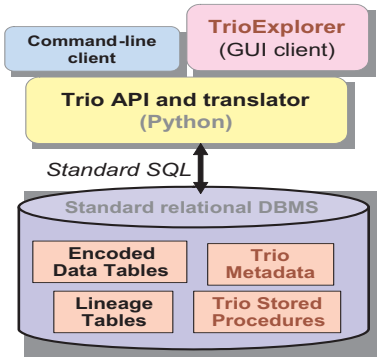

Fig. 8. Trio System Architecture

IV. Existing Systems

Many systems have been developed and implemented to support uncertain data management. We briefly summarize three representatives below.

Trio. Trio is developed by Standford InfoLab [6], [11], [54], [74]. As a database system, it not only tackles modelling and analyzing data but also the accuracy and lineage of data. Trio is developed based on data model ULDB which is introduced in Section II. It is implemented on the top of traditional relational DBMS (*PostgreSQL*). Query language in Trio is an extension of SQL, TrioQL. TrioQL handles queries, as well as accuracy and lineage of data. Figure 8 illustrates the system architecture of Trio [54]. The Trio API accepts TrioQL as well as regular SQL queries from client and translates them into standard SQL queries; in the relational DBMS, data tables are encoded, namely, are integrated with uncertainty information such as confidence and alternatives as introduced in Section II. Trio Stored Procedures handles confidence and lineage information.

MystiQ. MystiQ is a database system managing uncertainty in a probabilistic view developed by the University of Washington [16]. The system contains four main components, data modelling language (mDML), data definition language (mDDL), preprocessor and query translation engine. mDDL defines *approximate match operators* and allows users to specify *confidence* in query predicates. Below is an example query in mDDL [16].

```
SELECT F.title, D.name
FROM   Director D, Films F
WHERE D.did = F.did
AND    D.name ∼ 'Copolla'   [CONFIDENCE = 0.9]
AND    F.year ∼ 1975        [CONFIDENCE = 0.7]
```

This query retrieves film title and director name where the film was produced in approximately 1975 with confidence 0.7 and the director name is approximately 'Copolla' with confidence 0.9, from tables Director and Film. mDDL is integrated with the new components to manage uncertainty; the new components include predicate functions to specify measure used to generated similarity probabilities, global constraints for detecting and resolving inconsistency, etc. Based on mDDL specification, the preprocessor generates additional relational tables integrating probability values. The query translation engine is a critical component in MystiQ, translating queries written in mDML into regular SQL queries. Query evaluation techniques in MystiQ are introduced in [29], [60].

URank. URank is a system mainly designed for answering *U-Topk* and *U-kRanks* queries from Waterloo University and UIUC [70]. Querying techniques are introduced in [71]. URank is also built on traditional relational DBMS with new components to cope with uncertainty. The system is composed of two layers, *Storage Layer* and *Processing Layer*. Physical data and generation rules are manipulated in *Storage Layer*, as well as different access methods, such as random access and sorted access. *Processing Layer* is mainly based on techniques in [71]. *Space Navigation* accesses data from the *Storage Layer*. Each accessed tuple is sent to the component *State Formulation* to calculate probabilities of newly generated states. *Rule Engine* is the part of the system to handle such state probability computation based on generation rules.

The other systems managing uncertain data may be found in [45], [50], [52].

V. Conclusion

Almost every problem in conventional databases needs to be reinvestigated under uncertain semantics since the uncertain nature poses great and unique technical challenges. In spite of significant amount of existing work to analyze uncertainty as introduced in this short survey, a large gap still exists to fully interpret uncertainty. Possible future research work includes uncertainty studies against various applications such as data streams, dominating queries, spatial queries in high dimensional space, XML data, graph data, data mining, statistics estimation, time series, etc.

Acknowledgement. Research of Wenjie Zhang, Xuemin Lin and Ying Zhang is supported in part by ARC discovery grants DP0881035 and DP0666428, and a Google research award. Jian Pei's research is supported in part by NSERC discovery grant.

References

[1] S. Abiteboul, P. Kanellakis, and G. Grahne. On the representation and querying of sets of possible worlds. In *SIGMOD 1987*.
[2] S. Abiteboul and P. Senellart. Querying and updating probabilistic information in XML. In *EDBT 2006*.
[3] C. Aggarwal. On unifying privacy and uncertain data models. In *ICDE 2008*.
[4] C. Aggarwal and P. Yu. On high dimensional indexing of uncertain data. In *ICDE 2008*.
[5] C. C. Aggarwal and P. S. Yu. A survey of uncertain data algorithms and applications. In *IBM Research Report 2007*.
[6] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. Nabar, T. Sugihara, and J. Widom. Trio: A system for data, uncertainty, and lineage. In *VLDB 2006*.
[7] P. Agrawal and J. Widom. Confidence-aware joins in large uncertain databases. In *Stanford University Technical Report 2007*.
[8] L. Antova, C. Koch, and D. Olteanu. 10^{10^6} worlds and beyond: Efficient representation and processing of incomplete information. In *ICDE 2007*.
[9] D. Barbara, H. Garcia-Molina, and D. Porter. The management of probabilistic data. *IEEE TKDE*, 4(5):487–502, 1992.

[10] O. Benjelloun, A. D. Sarma, A. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In *VLDB 2006*.

[11] O. Benjelloun, A. D. Sarma, C. Hayworth, and J. Widom. An introduction to ULDBs and the Trio system. *IEEE Data Engineering Bulletin*, 29(1):5–16, 2006.

[12] C. Böhm, M. Gruber, P. Kunath, A. Pryakhin, and M. Schubert. Prover: Probabilistic video retrieval using the Gauss-tree. In *ICDE 2007*.

[13] C. Böhm, A. Pryakhin, and M. Schubert. The Gauss-tree: Efficient object identification in databases of probabilistic feature vectors. In *ICDE 2006*.

[14] C. Böhm, A. Pryakhin, and M. Schubert. Probabilistic ranking queries on Gaussians. In *SSDBM 2006*.

[15] F. Bonchi, O. Abul, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving object databases. In *ICDE 2008*.

[16] J. Boulos, N. Dalvi, B. Mandhani, S. Mathur, C. Re, and D. Suciu. MystiQ: A system for finding more answers by using probabilities. In *SIGMOD 2005*.

[17] D. Burdick, P. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan. OLAP over uncertain and imprecise data. In *VLDB 2005*.

[18] M. Chau, R. Cheng, B. Kao, and J. Ng. Uncertain data mining: An example in clustering location data. In *PAKDD 2006*.

[19] A. L. P. Chen, J. Chiu, and F. S. C. Tseng. Evaluating aggregate operations over imprecise data. *TKDE*, 08(2):273–284, 1996.

[20] R. Cheng, J. Chen, M. Mokbel, and C.-Y. Chow. Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data. In *ICDE 2008*.

[21] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *SIGMOD 2003*.

[22] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *TKDE*, 16(9):1112–1127, 2004.

[23] R. Cheng and S. Prabhakar. Managing uncertainty in sensor databases. *SIGMOD Record*, 32(4):41–46, 2003.

[24] R. Cheng, S. Singh, and S. Prabhakar. Efficient join processing over uncertain data. In *CIKM 2006*.

[25] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter. Effcient indexing methods for probabilistic threshold queries over uncertain data. In *VLDB 2004*.

[26] G. Cormode and M. Garofalakis. Sketching probabilistic data streams. In *SIGMOD 2007*.

[27] N. Dalvi and D. Suciu. Answering queries from statistics and probabilsitc views. In *VLDB 2005*.

[28] N. Dalvi and D. Suciu. The dichotomy of conjunctive queries on probabilistic structures. In *PODS 2007*.

[29] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *VLDB 2004*.

[30] N. Dalvi and D. Suciu. Management of probabilistic data: Foundations and challenges. In *PODS 2007*.

[31] D. Dey and S. Sarkar. A probabilistic relational model and algebra. In *TODS 1996*.

[32] N. Fuhr. A probabilistic framework for vague queries and imprecise information in databases. In *VLDB 1990*.

[33] N. Fuhr and T. Rolleke. A probabilistic NF2 relational algebra for imprecision in databases. In *Unpublished Manuscript 1997*.

[34] J. Galindo, A. Urrutia, and M. Piattini. Fuzzy databases: Modeling, design, and implementation. *Idea Group Publishing*.

[35] M. Hua, J. Pei, W. Zhang, and X. Lin. Efficiently answering probabilistic threshold top-k queries on uncertain data. In *ICDE 2008*.

[36] M. Hua, J. Pei, W. Zhang, and X. Lin. Ranking queries on uncertain data: A probabilistic threshold approach. In *SIGMOD 2008*.

[37] E. Hung, L. Getoor, and V. S. Subrahmanian. Probabilistic interval XML. In *ICDT 2003*.

[38] E. Hung, L. Getoor, and V. S. Subrahmanian. PXML: A probabilistic semistructured data model and algebra. In *ICDE 2003*.

[39] T. Imielinski and W. Lipski. Incomplete information in relational databases. *JACM*, 31(4), 1984.

[40] T. S. Jayram, S. Kale, and E. Vee. Efficient aggregation algorithms for probabilistic data. In *SODA 2007*.

[41] B. Kimelfeld and Y. Sagiv. Matching twigs in probabilistic XML. In *VLDB 2007*.

[42] H.-P. Kriegel, P. Kunath, M. Pfeifle, and M. Renz. Probabilistic similarity join on uncertain data. In *DASFAA 2006*.

[43] H.-P. Kriegel, P. Kunath, and M. Renz. Probabilistic nearest-neighbor query on uncertain objects. In *DASFAA 2007*.

[44] H. P. Kriegel and M. Pfeifle. Density-based clustering of uncertain data. In *KDD 2005*.

[45] L. V. S. Lakshmanan, N. Leone, and R. Ross. Probview: a flexible probabilistic database system. *TODS*, 22(3):419–469, 1997.

[46] K. Y. Lam, B. Y. L. R. Cheng, and J. Chau. Sensor node selection for execution of continuous probabilistic queries in wireless sensor networks. In *ACM VSSN 2004*.

[47] S. K. Lee. Imprecise and uncertain information in databases: an evidential approach. In *ICDE 1992*.

[48] X. Lian and L. Chen. Monochromatic and bichromatic reverse skyline search over uncertain databases. In *SIGMOD 2008*.

[49] E.-P. Lim, J. Srivastava, and S. Shekhar. An evidential reasoning approach to attribute value conflict resolution in database integration. *TKDE*, 8(5):707–723, 1996.

[50] C. Mayfield, S. Singh, R. Cheng, and S. Prabhakar. ORION: A database system for managing uncertain data. In *http://orion.cs.purdue.edu 2003*.

[51] S. McClean, B. Scotney, and M. Shapcott. Aggregation of imprecise and uncertain information in databases. *TKDE*, 13(6):902–912, 2001.

[52] A. Motro. Vague: a user interface to relational databases that permits vague queries. *ACM Trans. Inf. Syst.*, 6(3):187–214, 1988.

[53] R. Murthy and J. Widom. Making aggregation work in uncertain and probabilistic databases. In *Workshop on Management of Uncertain Data 2007*.

[54] M. Mutsuzaki, M. Theobald, A. de Keijzer, J. Widom, P. Agrawal, O. Benjelloun, A. D. Sarma, R. Murthy, and T. Sugihara. TrioOne: Layering uncertainty and lineage on a conventional dbms. In *CIDR 2007*.

[55] W. K. Ngai, B. Kao, C. K. Chui, R. Cheng, M. Chau, and K. Y. Yip. Efficient clustering of uncertain data. In *ICDM 2006*.

[56] A. Nierman and H. V. Jagadish. ProTDB: Probabilistic data in XML. In *VLDB 2002*.

[57] J. Pearl. Probabilistic reasoning in intelligent systems. In *1988*.

[58] J. Pei, M. Hua, Y. Tao, and X. Lin. Query answering technique on uncertain and probabilistic data. In *SIGMOD Tutorial 2008*.

[59] J. Pei, B. Jiang, X. Lin, and Y. Yuan. Probabilistic skyline on uncertain data. In *VLDB 2007*.

[60] C. Re, N. Dalvi, and D. Suciu. Efficient top-k query evaluation on probabilistic data. In *ICDE 2007*.

[61] R. Ross, V. S. Subrahmanian, and J. Grant. Aggregate operators in probabilistic databases. *JACM*, 52(1):54–101, 2005.

[62] E. A. Rundensteiner and L. Bic. Evaluating aggregates in possibilistic relational databases. *DKE*, 7(3):239–267, 1992.

[63] A. D. Sarma, O. Benjelloun, A. Halevy, and J. Widom. Working models for uncertain data. In *ICDE 2005*.

[64] A. D. Sarma, M. Theobald, and J. Widom. Exploiting lineage for confidence computation in uncertain and probabilistic databases. In *ICDE 2008*.

[65] A. D. Sarma, J. Ullman, and J. Widom. Functional dependencies for uncertain relations. In *ICDE 2008*.

[66] B. Scotney and S. McClean. Database aggregation of imprecise and uncertain evidence. *Inf. Sci. Inf. Comput. Sci*, 155(3):245–263, 2003.

[67] P. Sen and A. Deshpande. Representing and querying correlated tuples in probabilistic databases. In *ICDE 2007*.

[68] P. Senellart and S. Abiteboul. On the complexity of managing probabilistic XML data. In *PODS 2007*.

[69] S. Singh, C. Mayfield, S. Prabhakar, R. Shah, and S. E. Hambrusch. Indexing uncertain categorical data. In *ICDE 2007*.

[70] M. A. Soliman, I. F.Ilyas, and K. C.-C. Chang. Urank: Formulation and efficient evaluation of top-k queries in uncertain databases. In *SIGMOD 2007*.

[71] M. A. Soliman, I. F. Ilyas, and K. C. Chang. Top-k query processing in uncertain databases. In *ICDE 2007*.

[72] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao, and S. Prabhakar. Indexing multi-dimensional uncertain data with arbitrary probability density functions. *VLDB 2005*.

[73] Y. Tao, X. Xiao, and R. Cheng. Range search on multidimensional uncertain data. *TODS*, 32(3), 2007.

[74] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *CIDR 2005*.

[75] K. Yi, F. Li, D. Srivastava, and G. Kollios. Efficient processing of top-k queries in uncertain databases. In *Technical report, Florida State University, 2007*.