Front. Comput. Sci., 2020, 14(1): 162–190
https://doi.org/10.1007/s11704-017-7063-z

REVIEW ARTICLE

A survey of uncertain data management

Lingli LI[1], Hongzhi WANG (✉)[2], Jianzhong LI[2], Hong GAO[2]

1 Department of Computer Science and Technology, Heilongjiang University, Heilongjiang 150001, China
2 Department of Computer Science and Technology, Harbin Institute of Technology, Heilongjiang 150001, China

© Higher Education Press and Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract Uncertain data are data with uncertainty information, which exist widely in database applications. In recent years, uncertainty in data has brought challenges in almost all database management areas such as data modeling, query representation, query processing, and data mining. There is no doubt that uncertain data management has become a hot research topic in the field of data management. In this study, we explore problems in managing uncertain data, present state-of-the-art solutions, and provide future research directions in this area. The discussed uncertain data management techniques include data modeling, query processing, and data mining in uncertain data in the forms of relational, XML, graph, and stream.

Keywords uncertain data, probabilistic database, probabilistic XML, semi-structured data, data stream

1 Introduction

With the rapid development of sensor networks, Web services, and radio frequency identification (RFID) techniques, uncertain data has become ubiquitous. Uncertain data exist for a variety of reasons, which are mainly divided into the following: uncertainty from original data, summary data, privacy preservation, and automatic/semi-automatic information extraction. In many applications, such as the economic, military, logistics, finance, telecommunications, meteorological, and oceanic fields, uncertain data plays an increasingly critical role. Typical uncertain data applications include sensor networks, RFID applications, Web applications, and location-based service (LBS).

In the last 30 years, great interest has been devoted to uncertain data management. In fact, uncertain data processing techniques concern two aspects of uncertainty:

- **Attribute-level uncertainty** In an uncertain database D, the database has attribute-level uncertainty if one or more attributes is uncertain in following cases: (a) Attribute values are from a set of discrete values, each of which is associated with an existence probability; (b) attribute values are consecutively distributed possible values, and they associate with a probability density function. When performing queries in such databases, each record extracts one possible value from its uncertain attributes or distribution to form an instance table. Many applications such as sensors, electronic labels, and GPS values have attribute-level uncertainty in data records. Fuhr and Rölleke [1] describes the probability ?-table model, which extends the relational model by uncertainty.
- **Record-level uncertainty** In an uncertain database D, if records do not have uncertain attributes, but each record in this database exists with some probability, the database contains record-level uncertainty. More complex record-level uncertainty also includes a group of generation rules, each of which contains a group of records that provides the constraint condition of this record group. Usually, there are two types of generation rules: (a) exclusive rules, which require that this group of records cannot appear at the same time, and (b) coexistence rules, which require that this group of records must exist together.

Received February 20, 2017; accepted June 23, 2017

E-mail: wangzh@hit.edu.cn

We can point out some of the steps that have marked the history of uncertain data management. One pioneering contribution is the proposition of the *lineage* approach by Imieliński and Lipski [2], which represents uncertain data by a combination of classical database relations and propositional formulas. The study on probabilistic databases has been continued since then; see details in [1, 3–5]. In 2007, Sen and Deshpande [6] modeled a probabilistic database as a large graphical network. In 2008, Siciu [7] carried out a remarkable survey on probabilistic databases with this approach.

In probabilistic databases, the tuple-independent model [8] has been widely applied owing to its mathematical simplicity. This model assumes tuples are independent and queries are transformed into Boolean expressions. In probabilistic databases with independent tuples, every result tuple is associated with a Boolean formula lineage [9]. A query evaluation is computable in PTIME if the Boolean formulas can be factorized into a form in which every variable appears at most once, called read-once functions [10]. Read-once expressions are helpful because they are computable in PTIME, in contrast to #P-complete for arbitrary Boolean functions. Since then, several other cases computable in PTIME and their extensions have been studies in [11–13].

For probabilistic database systems, there has been extensive work under the relational database scheme, such as Trio [14], MayBMS [15], UDBMS [16], and MystiQ [17]. SPROUT [18] is the state-of-art probabilistic database query engine, which is integrated in MayBMS. The Trio and MayBMS systems deal well with uncertainty of data in an extended relational model and support SQL-based query language. Moreover, the UDBMS system developed by Purdue University can handle attribute and tuple uncertainty with arbitrary correlations, as well as discrete and continuous probability density functions (PDFs).

In contrast with relational probabilistic database systems, full-fledged semi-structured probabilistic database systems are still missing. The EvalDP system by Kimelfeld et al. [19] and ProApproX by Souihli and Senellart [20] are two XML document query processing systems. Cascadia [21] is an RFID data management system. For uncertain data streams, the PODS [22] and CLARO [23] systems support stream processing for uncertain data using continuous random variables.

For semi-structured data and stream data, techniques on uncertain data have made great progress and researchers have achieved a wealth of research results. Some surveys have been proposed on uncertain data management [24, 25]. However, the existing surveys focus on uncertain relational data problems and involve only a few uncertain XML data problems. Kimelfeld and Senellart [26] proposed a survey on probabilistic XML data management in 2013. With the rapid progress made by research in this area, many novel techniques for uncertain data management have been proposed. We propose this survey to summarize the state-of-art research progress on both relational and semi-structured uncertain data, as well as to provide future research directions in the area of uncertain data management.

In this survey, we will cover the area of uncertain databases in a broad sense and give an overall view. The remaining sections of this paper are organized as follows: In Section 2, we present relational uncertain data management issues, i.e., uncertain models, basic data operators, query processing techniques, indexing techniques, and uncertain data mining. In Section 2.1, we summarize relational uncertain data models, such as the *possible world semantics* and the *probabilistic graphical model*. Possible world is the most popular system of uncertain querying semantics, which provides the theoretical basis for the state-of-art uncertain database, the probabilistic database. In Section 2.2, we present the basic operators, i.e., selection, aggregation, and join operation, for query processing of uncertain data. In Section 2.3, various types of queries are defined and studied by database researchers, and various indexes are designed according to the problem properties. In Section 3 and Section 4, we describe the uncertainty management issues with semi-structured data, i.e., XML documents and graph (RFID) data. Section 5 presents uncertain data stream management issues and techniques, and in Section 6, we introduce the quality issues of uncertain data.

2 Uncertain relational data management

Relational databases, as a widely used type of database system, have achieved great success in both academia and industry. Currently, a large amount of data is managed in relational databases, and thus uncertain data management mainly concerns uncertain relational databases, which have yielded a number of research results. In this section, we will survey uncertain relational data management techniques from aspects of uncertain data models, basic data operators, advanced query operators, and their processing techniques.

2.1 Uncertain relational data model

The data model is the base of data management. In this section, we will survey three types of data models for uncertain data: the uncertain relational data model, possible world

model, and probabilistic graphical model.

2.1.1 Uncertain data model

The most common uncertain data model is an extension of the relational model, including *c*-table [2], ?-table [1], [4], or-set table [4], or-set-? table [3], [4], [27], etc. A probabilistic relational data model is called *complete* if it allows for the representation of any finite world set. The first complete uncertain relational data model is *c*-table. *c*-table is composed of *c*-tuples, which have following properties: 1) some attributes are replaced by free variables, 2) each record has an attribute condition, which defines the relational paradigm of free variables in this record, and 3) the entire *c*-table may also have a global constraint.

When assigning all variables, each assignment that satisfies all the constraint conditions will form an instance of a possible table. The reason why *c*-table is complete is that it allows arbitrary constraints. In real applications, however, defining arbitrary constraints carries a high cost in reading and reasoning.

The ?-table model [1], [4] represents tuples' existence probability with an independent probability field. Suppose w_i is a possible world instance (discussed in detail in Section 2.1.2); w_i's probability is the joint probability of tuples. The ?-table model can describe existence-level uncertainty, whereas the or-set table [4] model tends to describe attribute-level uncertainty. A tuple's attribute value is described as the "or" relationship between multiple candidate values and can be seen as a discrete PDF. The or-set-? table [3], [4], [27] model can be seen as a combination of the models above. The definition of the *c*-table model is similar to that of the or-set table, and the difference is that it is derived from the *c*-table [2]. The or-?-set table [28], [27] containing both attribute-level and record-level uncertainty is called *x*-relation [4].

2.1.2 Possible world model

The mainstream uncertain database is the probabilistic database, and it is founded on the basis of the possible world model. A possible world space consists of a series of possible world instances $W = w_1, w_2, \ldots, w_n$. $P : W \to [0, 1]$ is a probability distribution on it. Each possible world instance corresponds to a certain database, where the uncertain attributes are certain values satisfying constraints. Possible world semantics is the starting point and basis of query processing techniques on uncertain data.

Definition 1 Probabilistic database. Given a schema with relation names $R_1, \ldots, R_k$, $sch(R_l)$ denotes the attributes of relation schema R_l. A *probabilistic database* is a finite set of structures

$$\mathbf{W} = \{\langle R_1^1, R_2^1, \ldots, R_k^1, p^{[1]}\rangle, \ldots, \langle R_1^n, \ldots, R_k^n, p^{[n]}\rangle\}$$

of relations $R_1^i, R_2^i, \ldots, R_k^i$, and numbers $0 < p^{[i]} \leqslant 1$ such that

$$\sum_{1 \leqslant i \leqslant n} p^{[i]} = 1.$$

We call an element $\langle R_1^i, R_2^i, \ldots, R_k^i, p^{[i]}\rangle \in \mathbf{W}$ a *possible world*, and $p^{[i]}$ its probability.

An uncertain database can have either attribute-level uncertainty or record-level uncertainty, or both; for an uncertain attribute, its value can either be discrete or continuous; for records existing at some probability, there can either exist generating rules or not; generating rules can either be exclusive, coexistence, or other rules [29].

According to different possible worlds, there are some typical uncertain databases and possible world instances: 1) those that contain attribute-level uncertainty, where uncertain attributes are discrete, 2) those that contain attribute-level uncertainty, where uncertain attributes are continuous, and 3) those that contain record-level uncertainty and have exclusive generating rules. In real applications, an uncertain database is one of the typical databases above or a combination thereof.

2.1.3 Probabilistic graphical model

The probabilistic graphical model is another powerful uncertain data model. As the dependence relationship between uncertain data can naturally be described as directed graphical models (Bayesian networks) and undirected graphical models (Markov networks), uncertain data can naturally be described in the probabilistic graphical model. Its basic idea is to describe uncertain attribute values by random variables and represent attribute values' uncertainty and existence by probabilities.

In the probabilistic graphical model, databases are represented as a two-tuples (*R*,*P*), where *R* is a set of relations and *P* is a probabilistic graphical model (either a Bayesian network or Markov network). Each tuple or attribute in *R* is associated with a random variable representing the existence probability. Each node of *P* is a random variable that may correspond to a certain tuple in *R*, and the edges represent the relationship between random variables. According to Chen et al. [30], a random variable may be associated with multiple tuples, because sometimes we can introduce extra random

variables to describe the relationships (dependencies) among several tuples.

2.2 Data operators

2.2.1 Basic data operators

Probabilistic world-set algebra

Extended relational algebra operations The operations of relational algebra (selection σ, projection π, product $\times$, union $\cup$, difference $-$, and attribute renaming ρ) are applied in each possible world independently. The semantics of operations Θ on a probabilistic database $\mathbf{W}$ is

$$\begin{aligned}&\Theta(R_l)(\mathbf{W})\\ &:= \{\langle R_1, R_2, \ldots, \Theta(R_l), p\rangle | \langle < R_1, R_2, \ldots, R_k, p, \rangle \in \mathbf{W}\},\end{aligned}$$

for unary operations ($1 \leqslant l \leqslant k$). For binary operations, the semantics is

$$\Theta(R_l, R_m) := \{\langle R_1, R_2, \ldots, R_k \Theta(R_l), \ldots, R_k, p\rangle \in \mathbf{W}\}.$$

Tuple confidence: conf An operation for computing the tuple confidence is given by

$$\begin{aligned}&conf(R_l)(\mathbf{W})\\ &:= \{\langle R_1, R_2, \ldots, R_k, S, p\rangle | \langle R_1, R_2, \ldots, R_k, p\rangle \in \mathbf{W}\},\end{aligned}$$

where w.l.o.g., $P \notin sch(R_l)$, and

$$S = \{\langle \overrightarrow{t}, P : Pr[\overrightarrow{t} \in R_l]\rangle | \overrightarrow{t} \in \bigcup_i R_l^i\},$$

with schema $sch(S) = sch(R_l) \cup \{P\}$. The result of conf$(R_l)$, the relation S, is the same in all possible worlds, i.e., it is a certain relation.

Uncertainty introducing: repair-key An uncertainty-introducing operation, $repair - key$, which can be thought of as sampling the maximum repair of a key for a relation. Let $\overrightarrow{A}, B \in sch(R_l)$. For each possible world $\langle R_1, R_2, \ldots, R_k, p\rangle \in \mathbf{W}$, let the column B of R contain only numerical values greater than 0 and let R_l satisfy the fd $sch(R_l) - B \rightarrow sch(R_l)$. Then,

$$\begin{aligned}&[repair - key_{\overrightarrow{A}@B}(R_l)](\mathbf{W}) :=\\ &\quad \{\langle R_1, \ldots, R_k, \pi_{sch(R_l)-B}(\hat{R}_l), \hat{p}\rangle | \langle R_1, \ldots, R_k, p\rangle \in \mathbf{W},\\ &\quad\quad \hat{R}_l \textit{ is a maximal repair of fd } \overrightarrow{A} \rightarrow sch(R_l),\\ &\quad\quad \hat{p} = p \cdot \prod_{\overrightarrow{t} \in \hat{R}_l} \frac{\overrightarrow{t}.B}{\sum_{\overrightarrow{s} \in R_l : \overrightarrow{s}.\overrightarrow{A} = \overrightarrow{t}.\overrightarrow{A}} \overrightarrow{s}.B}\}.\end{aligned}$$

2.2.2 Advanced data operators

In this section, the possible world model is assumed for these advanced query operators.

1) Selection

One-dimensional space Cheng et al. [31] first proposed the range selection problem over uncertain data in one-dimensional space and they designed two auxiliary index structures to visit uncertain ranges efficiently.

A database contains uncertain objects T_i ($i = 1, 2, \ldots, n$) and each object contains some attributes, of which an attribute c's value is uncertain. The value of $T_i.c$ distributes within an uncertain interval $[L_i, R_i]$, and the PDF is $f_i(x)$. Cheng et al. [31] mainly solves the *probabilistic threshold queries* (PTQ) problem with a probabilistic range query. Given an interval $[a, b]$, the PTQ query result is a set of T_i whose probability p_i for attribute c's value in interval $[a, b]$ is greater than or equal to p_q, where p_q is a manually set limit. The most direct way to solve the PTQ problem is to look up all uncertain objects in database. If the object's uncertain interval overlaps the query interval, the probability is computed according to Formula (1):

$$P_i = \int_R f_i(x)dx. \tag{1}$$

If the probability is greater than the given limit, this object should be put into result set. However, this method has a high I/O cost, for it traverses all uncertain objects and the cost of computing formula (1) is also very high. In order to improve the query efficiency, Cheng et al. [31] proposes the probability threshold indexing (PTI). PTI revises one-dimensional R-tree structure by adding probability information to intermediate nodes to accelerate the pruning.

Still, PTI cannot avoid the intrinsic R-tree problem, i.e., one range query can traverse many large intervals' minimum bounding rectangles (MBRs). To deal with this situation, Cheng et al. [31] proposes another approach. Every interval $[x, y]$ is mapped to a point (x, y) in two-dimensional space. Assuming that the PDF distribution is uniform, transfer PTQ into a two-sided orthogonal query and the 2D R-tree is used to accelerate query processing. For any arbitrary distribution, the intervals are placed with similar means and standard deviations into one MBR.

Multi-dimensional space Tao et al. [32] studies the multidimensional space range query problem for objects with any arbitrary probability density. They propose some pruning strategies and a new storage method $U - tree$ to optimize the I/O cost and CPU time.

Different from the problem in [31], the object T_i in Tao et al. [32] contains two parts: 1) PDF $Ti.pdf(x)$, where x is a point from d-dimensional space, and 2) an uncertain region $T_i.ur$ in d-dimensional space. The probability range query means that given a super rectangle r_q and limit value p_q, if p_i is greater or equal than p_q, T_i is put into the result set.

The size of an uncertain object's *Probabilistically Constrained Regions* (PCR) is determined by the parameter p, with a range of $[0, 0.5]$. $Ti.pcr(p)$ is enclosed by four lines l_1^+, l_1^-, l_2^+, and l_2^-. l_1^+ divides $T_i.ur$ into two parts, with the probability of p for T_i to occur in the right of l_1^+. Accordingly, l_2^+ and l_2^- divide $T_i.ur$ in the horizonal direction.

Tao et al. [32] proposes the notion of *Conservative Functional Boxes* (CFB) to restrict the PCR and design the corresponding algorithms.

2) Aggregation operation and data analysis

Aggregation is an important type of operation in statistical data analysis. For statistical analysis of uncertain data, there exists some work on aggregation query operations. Some of them are designed to return the estimation of aggregation values [33, 34], which are I/O effective aggregation algorithms over uncertain data. In [35], the authors consider the expectation value and ALL_SUM value. [36] and [37] focus on approximate algorithms for probabilistic aggregation queries. Kanagal and Deshpande [38] uses the central limit theorem to approximately estimate the distribution of sufficiently large numbers of uncertain values. Burdick et al. [39] investigates query processing with HAVING restrictions, which has a connection with the 0–1 pack problem. Ré and Suciu [40] studied the complexity of evaluating a HAVING query and established a set of trichotomy results for conjunctive queries with HAVING predicates: 1) the exact problem has P-time data complexity. 2) The exact evaluation problem is #P-hard, but the approximate evaluation problem has P-time data complexity. 3) The exact evaluation problem is #P-hard, and the approximate evaluation problem is also hard.

Fink et al. [41] considers the problem of exact probability computation for positive relational algebra queries. Their knowledge compilation technique compiles arbitrary semimodule and semiring expressions into a decomposition tree, which is a technique that reflects structural decompositions of expressions into independent and mutually exclusive sub-expressions. They built the connection between query tractability and polynomial-time decomposition trees.

Based on aggregation operations, we can perform an OLAP analysis on uncertain data. [33, 39] deal with the imprecise OLAP query problem based on the possible world model. According to the semantics of the possible world model, a database with imprecise facts can be described as multiple possible worlds. Each of them only has precise facts, and the final query result can be obtained from these possible world instances. For example, given an uncertain database D, all possible world instances $w_1, w_2, \ldots, w_n$ are first generated. The probability for each possible world instance w_i to occur is $p(w_i)$. A query Q is executed on every possible world instance and we get $Q(w_i)$. Finally, these query results are aggregated and we obtain $\Sigma p(w_i) * Q(w_i)$.

When we have a large dataset, the method that enumerates all possible worlds is infeasible. [33] first studied the optimization method when the data are independent. This method first constructs the *extended database* (EDB), and then scans EDB in a single pass and calculates the query result. Burdick et al. [39] improves the method to compute EDB, which further improves the query efficiency.

The case of independent data is easy to process, but it becomes more complex when there exist constraints in the data. Burdick et al. [33] proposes an optimizing method that constructs a *marginal database* (MDB) based on the original database D, and then gets the query result by a single pass scan of MDB. In MDB, all facts' edge probabilities are precomputed in order to accelerate the query processing.

2.2.3 Join operation

Join is a database operator that combines two datasets into one single set of data objects based on certain predicates. Given two relations R and S composed of uncertain objects, a join over uncertain data $J(R, S)$ is a Cartesian product $J(R, S) = (a, b, s(a, b)) \in [0 \ldots 1]$, where $s(a, b)$ represents the score calculated from two uncertain objects a and b by given a compare operator and join predicates.

A join over uncertain data can be classified according to query types, join predicates, and score functions. Classified according to query types, join operations include: probabilistic join query (PJQ), probabilistic threshold join query, probabilistic Top-k join query, V-join, and D-join. To be more specific, given two uncertain datasets, PTJQ retrieves all pairs of objects satisfying certain given join predicates and with greater probability than the given threshold. PJQ is a special case of PTJQ, in which the threshold is 0. PTop$k$JQ returns k pairs uncertain objects which satisfy the join predicate and with highest probabilities. V-join [42] returns join tuple pairs with the join attribute values' discrepancy less than some threshold; D-join [42] returns tuple pairs with the join attribute values' variation distance less than the given threshold.

Classified by join predicates, join queries over uncertain data include the ϵ-range join query and k-nearest-neighbor (k-NN) query. Classified by score functions, join queries over uncertain data include: Boolean join, similarity distance join, and spatial distance join.

The traditional join operator is implemented by the nest-loop cooperation, block-based nest-loop comparison, sort-merge, hash-merge, and index-based methods. Upon uncertain data join implementation, researchers design join query processing technique in two ways. One is to inherent methods for processing certain data. The other is to revise existing methods and develop new algorithms considering the probability dimensions over uncertain data. To summarize the preceding well-known join processing algorithms, we can observe that their differences are mainly shown in uncertain data representation, distance measure type, join query type, query predicates, and result representation. We can classify them into three categories: confidence-based join query, probability similarity join query, and probability spatial join query. We will discuss these join operations in detail.

Confidence-based join Confidence-based join query algorithms reduce the search space by the confidence of the join query result tuples. It considers neither join-related object attributes nor join predicates. The general approach is to sort the tuple objects in relations R and S in descending order [43]. Assuming that the objects are independent of each other, the possibility of the two objects' pairs is at most their confidence product. In this way, we can greatly accelerate the process efficiency of queries such as PTJQ and Ptop$k$JQ. Additionally, they can be implemented simply by block loop comparisons.

Probability similarity join The probability similarity join query needs to consider both the confidence between objects and similarity relation between objects. In general, only a small fraction of objects satisfy the join predicate. Therefore, a highly efficient pruning method can improve the query efficiency in a large scale. For a continuous uncertain model, we need a continuous PDF to represent the similarity score between two uncertain objects. This ensures the similarity probability between two objects is a continuous PDF. Cheng et al. [44] proposes the item-level pruning, page-level pruning, and index-level pruning. Item-level pruning avoids the costly probability estimation process by building lower and upper bounds for join predicates' probabilities. The main idea of page-level pruning's is to avoid page visits by adding an x-bound for each node. Index-level pruning further organizes the pages into a tree structure in order to improve join query's I/O processing performance. For discrete uncertain models, probability similarity query processing techniques mainly include clustering, sampling, and a lower–upper bound method based on spatial distance.

Probability spatial join Besides the probability attributes in uncertain data, the probability spatial join query also needs to consider their spatial relations. Spatial join queries use spatial predicates (for example, intersect, overlap, etc.) and distance predicates (for example, distance range, nearest neighbor, etc.) to process queries. The most common query processing method is to use a spatial index (e.g., R-tree) and spatial position relations (e.g., the maximum and minimum distance between spatial objects).

The earliest probability similarity join method based on discrete uncertain objects [45] represents each uncertain object O_i as m data points $o_{i1}, o_{i2}, \ldots, o_{im}$ in d-dimensional space. In this way, each data point o_{ij} represents object O_i's alternate position. This method uses the k-means clustering algorithm and sampling method to divide each object's sample point into k groups, each of which is approximately represented by a minimal bounding super rectangle. For each cluster C of uncertain object O, assume the probability for O matching a sample point of C is known and each sample point has the same probability to match O's position. This multi-level approximate method greatly reduces the computational complexity of join processing. Kriegel et al. [45] takes use of the filter-and-refine querying paradigm on the distance join query and nearest neighbor query. First, a cluster representation is used to compute the lower and upper bound of the join probability to filter join pairs. Then, in the refine phase, the final results are acquired using an uncertain distance and the probability of candidate pairs' matching the query predicates.

The probability similarity join method based on continuous uncertain model [44] takes each uncertain object as a random variable, and each entry is represented by a possible value interval and probability distribution. The main contribution of [44] is proposing and extending the equal (unequal) join operation as well as three join pruning strategies based on the item level, page level, and index level. In the continuous uncertain model, the probability that two uncertain objects are at some position equal is 0. Thus, two uncertain objects a and b can only be equal with a certain fineness c. Ljosa and Singh [46] first proposed the probabilistic spatial join query method for processing geographical images and biomedical image data. The plane sweep algorithm proposed by Ljosa and Singh [46] has a time complexity of $O(n(logn+k))$, where n is the number of uncertain data points and k is the number of results. Its main idea is to transform d-dimensional object points into the $(d+1)$-dimensional space, where the $(d+1)$th

dimension is the object's confidence in logarithmic space. In this way, a triangle pruning area is constructed in order to quickly distinguish whether objects satisfy spatial predicates.

Jestes et al. [47] first defines string join problem in probabilistic string, taking the *expected edit distance* (EED) as a similarity measure. In detail, they develop several effective pruning techniques for string-level and character-level models, including: lower bound filters and upper bound filters. Lian and Chen [48] introduces the *probabilistic set similarity join* (PS2J) problem based on possible world semantics, Jaccard distance pruning, probabilistic upper bound pruning, etc.

2.3 Query processing

In this section, we discuss processing methods for various types of queries on uncertain data. At the beginning, we list them in Table 1 for a brief description.

Table 1 Query types

Query type	Reference	Summary
SQL-like	[49]	Query rewrite
	[6], [50]	Probabilistic graphical model
	[51]	Threshold query optimization
	[52]	Query containment
	[53]	Array database system for scientific and engineering applications
TOP-K	[54]	Sort tuples according to the semantics worlds
	[55], [56]	U-Top*k* and U-*k*Rank query based on *x*-relation
	[57]	Global-Top*k* query
	[58]	Approximation algorithm for tuples' aggregation probabilities
	[59]	E-Score and E-Rank
	[60]	*C*Typical-Top*k*
	[54] [61] [62]	position probability U-*i*Ranks
K nearest neighbor	[63] [64] [65]	Constraint *k* nearest neighbor query
	[66]	Group *k* nearest neighbor query
	[67]	Reverse *k* nearest neighbor query
	[66] [68] [69] [70]	Superseding *k* nearest neighbor query
Skyline	[71] [72]	Object's dominance
	[57]	Data stream skyline query
	[73]	Probabilistic tpo-*k* dominating

2.3.1 SQL-like query

To begin with, we investigate the SQL-like query, which is quite a basic query type. SQL-like queries are essentially like traditional queries in certain database queries, with the addition of probability information into the query sentences.

Andritsos et al. [49] first brings the query rewrite technique to query processing. To be specific, given a SPJ query: "select $A_1, A_2, \ldots, A_n$ from $R_1, R_2, \ldots, R_m$ where W", we can transform this query into the following form: "select $A_1, A_2, \ldots, A_n, sum(R_1.prob. * \cdots * R_m.prob)$ from $R_1, R_2, \ldots, R_m$ where W group by $A_1, A_2, \ldots, A_n$." In this way, probabilities of results are introduced in the query sentences.

During the processing of SQL-like queries on relational database, calculating the probability of the query results is a major research problem. It is hard to completely calculate the query results. Research results have shown that, in a probabilistic database, a query can either be computed in PTIME or be a #P-hard problem [74, 75].

Sen and Deshpande [6] processes SQL sentences with a graphical model. The query evaluation problem is treated as an inference problem in an appropriately constructed probabilistic graphical model. The joint probability distribution is decomposed by applying a probabilistic graphical model. The probabilities of query results are computed and returned by the conditional probability distribution from factor expressions after the decomposition. Wick et al. [50] encodes the possible world using the graphical model. First, possible worlds are sampled by the Markov chain Monte Carlo (MCMC) method and local changes in worlds are inferred to generate a factor graph. According to the factor graph, queries only need to run on changed worlds, instead of all sampled possible worlds.

Qi et al. [51] studies the probabilistic threshold query optimization problem, in which the optimization method supports query plan enumeration. To support query optimization, Qi et al. [51] proposes general optimization rules as well as rules specifically for selections, projections, and joins. This paper introduces a threshold operator to the query plan and shows that it is generally desirable to push down the operator as much as possible.

Moore et al. [52] studies the query containment problem over uncertain relational databases, which allows queries such as "is the possibility that q_1 holds greater than 0.2 and the probability that q_2 holds greater than 0.6." The containment problem is one of the core problems in query optimization, which can be used to pick up an index or apply a view. Moore et al. [52] proves that the containment problem is decidable, and gives an EXPSPACE-algorithm based on linear programming.

Ge et al. [53] studies an array database system for scientific and engineering applications. In these applications, the value of a cell is often imprecise and uncertain. Queries on these uncertain data are represented by Monte Carlo queries. This

type of query supports a graphical model in representing correlation between data and complex operations in science and engineering domains. In this paper, they use Markov random fields combined with an array's chunking or tiling mechanism to model correlated data. In order to process these queries, Ge et al. [53] proposes the array join algorithm and optimization algorithm of stop conditions of Monte Carlo query processing.

2.3.2 TOP-K query

The efficient processing of top-k queries is a crucial requirement in many interactive environments that involve massive amounts of data. In this section, we discuss top-k query processing over uncertain data.

When performing sort and query operations over an uncertain dataset, we need to consider the correlation and semantics between probability values and their attribute values. Soliman et al. [54] first sorts tuples according to the semantics world constituted by probability values and attribute values. They introduce two algorithms U-Top$k$ and U-kRank, with different query semantics. On this basis, other ranking methods such as PT-k, E-Rank, and cTypical-Top$k$ are proposed. The U-Top$k$ query algorithm returns the tuple vector with the greatest probability to become top-k in all possible worlds. The U-kRank query algorithm computes the tuple's rank in all possible worlds and then performs the query and returns the tuple with the ith greatest probability. To perform query processing, Soliman et al. [54]'s main idea is to map each tuple to a state and maintain a state diagram composed of all tuples' rules. Each tuple is sorted decreasingly according to their scoring functions.

The costs of both time and space are expensive in Soliman et al. [54]. To speed up the query processing, Yi et al. [55] takes use of the data model based on x-relation to deal with U-Top$k$ and U-kRank queries. For a U-Top$k$ query, one stack is used to store the probabilities of the maximum k tuples, which reduces the time complexity to $O(nlogk)$, and space complexity to $O(n)$. For an uncertain sort query, the algorithm develops a method based on dynamic programming, which calculates the probability for tuple t ranking ith according to its former $(i-1)$ tuples. Its time complexity is $O(n^2k)$, and space complexity is $O(n)$. If each tuple could only have one possible value (i.e., one alternative), the computation cost could be reduced to $O(nk)$ [56].

Let D be an uncertain database, and its possible world space is $W = w_1, w_2, \ldots, w_n$. $Q = q_1, q_2, \ldots, q_m$, where q_i is a set of records corresponding to the top-k records sorted by some scoring function f in each possible world. The probability for each record t in the Top-k on the basis of Q is called global-Top$K$ probability of this record. Global-Top$K$ query returns k records with most global-Top$K$ probabilities. A probability threshold query (PT-k query) and global-Top$k$ [57]'s main idea is to compute the aggregation possibility for a tuple to become top-k in all possible worlds.

Hua et al. [58] provides an approximate algorithm based on sampling and Poisson approximation rules to prune the calculation of tuples' aggregation probabilities. Because the PT-k query only returns tuples with greater probability than the threshold, taking use of TA [76] algorithm and reuse each tuple's probability result can avoid estimating all tuples and saving computation cost, which would improve the query result calculation efficiency in a large scale.

E-Score [59] is ranked totally by tuples' expectation, which has a minimum computation and storage cost, but the quality of results is heavily dependent on the distribution of scores and confidence values. In addition, the association between scores and confidence values has great influence on the computation. The expected rank (E-Rank) has totally different query semantics from the above above sorting query algorithm. The expected rank for a tuple in the possible world PW is the number of tuples whose attribute values are greater than its attribute values. As seen from its definition, the query results of the expectation rank are dependent on both the rank position and the probability of the tuple existing in the possible world. That is, the tuple's ranking depends on the probability distribution of its score.

Cormode et al. [59] proposes an effective algorithm to compute expected ranks. The algorithm has a time complexity of $O(NlogN)$ for any N tuples of uncertain relational data. As the algorithm uses expectation to calculate ranks and Markov inequality to approach the upper and lower bound in the computation of attributes' model, it can effectively prune out unknown tuples, but sacrifices precision.

CTypical-Top$k$ query [60] proposes typical tuple vectors to allow the user to effectively sample attribute distribution and probability distribution, and they try to solve the problem that the k tuples in U-Top$k$ query results cannot effectively express top-k semantics over uncertain data. In addition, Ge et al. [60] proposes the idea to use dynamic programming to solve the tuple combination optimization problem in the possible world. In summary, U-Top$k$ is the most natural rank method intuitively; U-kRank, Global-Top$k$, and PT-k are the reflection of tuple rank positions; E-Rank and E-Score are to find certain ranks over uncertain data to some extent; c-Typical-Top$k$'s goal is to query top-k tuples' typical attributes

over uncertain data, in order to meet certain requirements.

In this area, some researchers concentrate on proposing different query semantics, as well as the top-k query algorithm and rank query algorithm in different application environments to meet computation demand in certain environments. For example, Li et al. [77] uses a tuple's ranking in all possible worlds to represent features and use these features to determine the tuple's ranking. On this basis, Li et al. [77] proposes a general probability rank query framework. Li et al. [78] concentrates on how to reduce communication the cost of probability uncertain query in distributed environments.

On the basis of the position probability U-iRanks proposed in Soliman et al. [54], Soliman and Zlyas [61] proposes the optimization object based on top-k position records' position probability sum, and uses the bipartite graph matching method to find the optimal total order.

Currently, to solve the partial problem, people essentially make use of the probabilistic partial model mentioned in Soliman and Zlyas [61]. The probabilistic partial model sorts the continuous scores' domain by end points. Each record t has two end points: low and up. If some record t_i has a greater low_i than another record t_j's up_j, t_i totally dominates t_j; if the two records' domains intersect, they satisfy probability dominating relationship. The dominating probability of each other can be calculated by the joint density function's double integral in the intersection area. In Li and Deshpande [62], the position probability calculation problem is transformed into one-dimensional integral by generating functions. Even though, in continuous score condition, the calculation of the probability is still an integral problem, which is intuitively a complex operation. Soliman and Zlyas [61] uses Monte Carlo integration to calculate the dominating probability of uniform distributed scores.

2.3.3 K nearest neighbor query

A traditional neighbor is defined as an object with the shortest distance to a given query object q. An uncertain object has multiple data instances. Certain data's probability distribution can also be described as a probabilistic density function. Another uncertainty is existence uncertainty [79, 80], which assumes each object to exist in database with some probability $p(O)(0 < p(O) < 1)$. This uncertain object only has one instance.

Cheng et al. [81] first classifies uncertain data queries into value-based probability queries and entity-based probability queries. Value-based probability queries return an interval value and its corresponding probability or object and its corresponding probability. Entity-based probability queries return a group of objects and their probabilities. Entity-based queries include entity probability range query (ERQ), entity probability nearest neighbor query (ENNQ), and entity minimum value query (VMinQ). These two types of queries can be further divided into aggregation query and non-aggregation query, according to whether they contain aggregation properties.

Approximate query processing methods Uncertain data neighbor query process can generally be divided into four phases: projection, pruning, bounding, and evaluation. The projection phase computes the uncertain region for each object according to the application's uncertain model. The pruning phase prunes objects with zero probability of becoming a neighbor of the query object q to reduce the expensive neighbor probability computation cost. The bounding phase further prunes out uncertain object's regions that intersect with the query region but is not contained in the query region. Finally, the evaluation phase computes the probability for each object O_i.

Other nearest neighbor query over uncertain data includes: indexing [79], [67], clustering, and sampling [82]. Dai et al. [79] studies spatial queries where objects exist with some uncertainty and are associated with an existential probability.

Kriegel et al. [82] proposes a nearest neighbor query method based on clustering and uses a Monte Carlo sampling method to compute the probability density of each object.

Uncertain data neighbor queries are expended in multiple aspects. On one side, the query condition could be changed. Some examples include constraint neighbor query [63], which constrains the query results, k-nearest neighbor query [64], and group nearest neighbor query [66], which changes a single-query object q into a multiple-query object $Q = q_1, q_2, \ldots, q_n$. However, we can change the query semantics. For example, considering a probabilistic neighbor query in reverse [68–70], superseding neighbor query [67] which finds the best neighbor set, etc. Common processing techniques include: pruning method based on spatial relationship [66], [70], pruning method based on probability precomputation [63], [66] and probability bound pruning methods [64], [83].

Probabilistic constraint k-nearest-neighbor query As the probability neighbor query algorithm returns all possible neighbor objects with a probability greater than 0, the number of returned objects may be large, and the possibilities may be very small. One solution is to add a threshold constraint to possible neighbor objects. For example, return neighbor

objects with a probability higher than 30%. Sometimes, however, the query may not be concerned with the exact probability of neighbors, and only requires a high enough probability of the returned neighbor. The constrained probability nearest neighbor query [63] meets this type of application's requirements. In order to control the query's quality, we can add tolerance parameter constraint conditions. The general idea to deal with such queries is to divide distances $R_i = |O_i - q|$ into multiple subdistance partitions $S = S_1, S_2, \ldots, S_M$ (M is the number of subpartitions) according to the distance information and probability density changing point position information. Then, the probability for R_i to be in each partition S_j is precomputed and the probability is accumulated. Finally, based on this precomputed probability information, the upper and lower bound equations are proposed for assessing the neighbor probability.

In location-based services, sensor monitoring, and biology management systems, we usually need k-nearest neighbor queries to process such uncertain data. The probability threshold k-nearest neighbor query T-k-PNN [64] is this type of query. It returns a set R composed of k neighbor objects with the set probability $p(R)$ greater than the given threshold T. However, the challenge for solving the T-k-PNN problem lies in: 1) It has an exponential complexity to select k objects from n objects; 2) Calculating the set probability $p(R)$ has a high complexity. The T-k-PNN algorithm first removes the objects that cannot become q's neighbor by k-bound filtering, which is defined as the kth most remote distance f_k. Then, R's probability upper bound is computed by the lemma that $p(R) \leqslant \prod O_i \in S\, Pr(r_i \leqslant f_k)$, where r_i represent the distance between O_i and q, which further reduces the search space. Finally, in the evaluation phase, accumulated probability information is used to provide the upper and lower bound of $p(R)$ to further reduce the computation cost.

Zhang et al. [65] introduces the notion of median rank and proves that both the expected rank and median rank satisfy the top-k properties, including exact-k, containment, unique ranking, value invariance, stability, and fairness. For a given query q, this paper proposes an I/O and CPU effective expected (or median) based rank method over uncertain data. In order to deal with uncertain object relations and high I/O cost introduced by the large number of uncertain instances, this paper proposes a random approximation algorithm with guaranteed approximation ratio.

Probabilistic group nearest neighbor query When the reference point is expanded to multiple queries $Q = \{q_1, q_2, \ldots, q_m\}$, the uncertain nearest neighbor query becomes the probabilistic group nearest neighbor query (PGNN) [66]. PGNN returns a set of uncertain objects so that each object has a minimum distance to Q and its probability is greater than the threshold. PGNN plays an important role in applications such as forest fire suppression and multi-image feature nearest neighbor search.

Probabilistic superseding nearest neighbor query The probabilistic superseding nearest neighbor query is another important extended nearest neighbor query over uncertain data. Given a query object q and two candidate nearest neighbor objects o_1 and o_2. O_1 supersedes o_2 if o_1 is more likely to be closer to q than o_2. An object is a superseding nearest neighbor (SNN) of q, if it supersedes all the other NN-candidates. When no object can supersede all its neighbor candidate set objects, an SNN query returns a minimum neighbor query set (called SNN core), each object of which can supersede all neighbor candidate object outside the SNN core. SNN core can be seen as the set of best neighbor objects in uncertain data. SNN query can be applied to any neighbor queries over uncertain data. Yuen et al. [67] first proposes the SNN query, and introduces an effective algorithm to compute the SNN core by directed graph theory and R-tree index.

Probabilistic reverse nearest neighbor query The foregoing query algorithms over uncertain data all consider queries from a positive point of view, whereas another important research direction is reverse query methods [66, 68, 69]. These methods have a wide range of applications in terms of resource allocation, personalized marketing strategy, game development, and military strategy planning. The processing techniques mainly include: spatial pruning based on the perpendicular bisector [70], [68] and the probability upper and lower bound pruning methods [69]. The probabilistic reverse nearest neighbor query algorithm [70] returns the uncertain objects with a greater probability than the threshold to become query object q's reverse nearest neighbor (RNN). Given a query object q and perpendicular bisector $\perp$ of an uncertain object o and possible position point o', which divides the data space into two disjoint halfplanes: $HP_q(q, o)$ the halfplane close to q and $HP_o(q, o')$ the halfplane close to o'. If another uncertain object p's possible position point p' in halfplane $HP'_o(q, o')$ exists, p' can be safely pruned out, because o' is closer to q than p' and q's reverse neighbor cannot be p'. If we use a bounding hypersphere to bound an uncertain object, the bounder of the conservative pruning region (CPR) by o's all instance point o' and q's $\perp$ operation is a hyperbola. In this way, we can quickly determine whether uncertain object q is in $CPR(q, o)$ by this hyperbola, in order to decide whether to prune out p.

Processing PRNN queries from the perspective of possible

world semantics is another research direction [68]. Its pruning strategies include: 1) pruning other objects p using the halfplane pruning method, including the halfplane formed by query instance $q \in Q$ (Q is an uncertain query object) and R_f (the minimum bounding rectangle of f), the halfplane pruning region formed by R_Q and R_f; 2) pruning using distance measure inequality $maxdist(R_c, R_f) < mindist(R_c, R_Q)$; 3) pruning based on a control relationship between objects; 4) pruning based on aggregate probability bounds in searching progress.

In applications such as image data analysis, sensor data monitoring, multi-object decision, and business planning, we often need to identify a given object's importance for other objects. This is the probabilistic inverse ranking (PIR) query [69], [83], which retrieves all possible rankings for an uncertain query object q in an uncertain database with a probability greater than the given threshold.

2.3.4 Skyline query

Skyline query is an important class of decision support queries, whose aim is to find a subset of the object collection S in D-dimensional space. Any point in the subset cannot be dominated by any other nodes in S. The dominance relationship here refers to many objects in the given D-dimensional space; one of them, i.e., p, is better than another one q in one dimension at least and is not worse than it (better than or equal to) in other dimensions. Then, we can say p is able to dominate q [72]. The Skyline query can be extended to the implementation on uncertain data and its nature is an extension of the probabilistic nearest-neighbor query.

It is assumed that within the probabilistic skyline query model, the uncertain object usually contains multiple independent instances that appear in the same probability. The probability of an uncertain object U is defined to be the integral of multiplying the probability density of each instance $u \in U$ and the probability of those objects that cannot dominate u [71]. Then, it will retrieve all those whose probabilities are over the given threshold.

General processing techniques include: hierarchical partitioning techniques [71], trimming for the upper or lower probability bounds [71, 73], space partition methods to close neighbors [84], pre-computation methods based on the properties of probability distributions [84], and approximation techniques [73].

Pei et al. [71] proposes p-skyline query over discrete uncertain objects and develops two algorithms for it, i.e., bottom-up and top-down. Within the set of uncertain objects, there can be multiple instances to each one of them. In order to simplify the model for a better understanding, the author assumed that uncertain objects are independent of each other and carry the same probability of appearance.

In order to reduce the calculation of the probability of each uncertain object instance, we partition the instances into layers with decreasing probability. Based on the above strategy, the bottom-up algorithm can utilize heap and R-tree for data organization, whereas the top-down method will create a partition tree for each uncertain object. The partition tree is a binary tree whose process of construction is quite similar to that of the kd-tree. Each leaf node includes a series of instances and the corresponding MBB, and each internal node is used to maintain the MBR. For a node N in such one partition tree, we set $N_{\min}$ and $N_{\max}$ to represent the lower left and upper right corner of vertices separately. For any instance u in N, we can obtain its probability as follows: $Pr(N_{\min}) \geqslant Pr(u) \geqslant Pr(N_{\max})$.

The positive probability skyline query can be implemented with the idea of bottom-up or top-down, whereas that processed in the reverse perspective reflects the influence of our query object q [67]. With the given uncertain objects here, i.e., o and p, the farthest point to our query point q is F_o in o, and the middle point between them is M_o; then, any uncertain object in the trim area $Pr(q, o)$, such as, p, cannot become the reverse Skyline object of q. If the $1-\beta$ super-matrix of p, i.e., $UR1-\beta(p)$, is included in the trim area of the uncertain object o, it can be trimmed off, where the uncertain area of p is $UR(p)$ and the center is C_p. Zhang and Chomicki [57] introduced the probabilistic skyline query method into the data stream, and proved the need for defending for its set SN. Probabilistic top-k dominating (PTD) [73] retrieves k uncertain data and then use them to control the maximum of uncertain objects via query object q.

2.3.5 Indexing techniques

Many indexing techniques have been put forward in order to effectively support queries on uncertain relational data. In this section, we survey the index over uncertain data. A brief description of these indexes are listed in Table 2 below.

One-dimensional indexing methods One-dimensional indexing methods are used for processing the range query on one-dimensional uncertain data. By means of computing the integral of the PDF, the method performs the calculation on the probability of the whole uncertain objects within a specified interval in advance, and then stores the interval and probability values in an index structure. The way in which

Table 2 Indexing method

Indexing method	Reference	Summary	Supported data model
One-dimensional index	[31]	Probabilistic threshold index (PTI)	S
Multi-dimensional index	[32] [85]	U-Tree, a multi-dimensional PTI	S
	[86]	Guass-Tree, based on probabilistic feature vectors (PVF)	S
	[87]	APLA-Tree, based on piecewise-linear approximations	S
	[88]	Uncertain Voronoi Diagram	S
	[89]	UP-Index	S
Inverted index	[90]	Probabilistic inverted index (PII)	S
Others	[91]	Junction Tree, built using tree partitioning	S

we establish index is quite similar to R-tree, where each intermediate node is made up of the boundary matrix of their child nodes and the pointers to them, while each child node also includes a table consisting of a plurality of intervals and the corresponding probability threshold, and the leaf node contains the spacing of uncertain data and its probabilistic density function. A typical method of indexing techniques, probabilistic threshold indexing (PTI) [31], can set multiple x-bounds if the attribute value of each tuple in the database corresponds to a one-dimensional interval $[a, b]$.

Multidimensional indexing methods Multidimensional indexing method is applied in processing the region query of high-dimensional uncertain data and mainly used in spatial–temporal databases, such as searching for an object belonging to a region whose probability is greater than a certain threshold.

A multidimensional version of extending PTI is U-Tree [32], [85]. With o representing an object, we set $o.ur$ for the expression of its region, along with $o.pdf(.)$ to express its PDF. We get $o.pdf(x) > 0$ if the arbitrary point x is in the region of $o.ur$, or else $o.pdf(x) = 0$. The method is based on the probabilistically constrained region techniques. We first divide the region into a plurality of regular rectangles, each of which corresponds to a probability. We then utilize U-tree (similar to R-tree) as the index for PCR. These PCR contribute effectively to the query operations, such as pruning and verification.

In the case in which the PDF presents as Gaussian distribution, we can specify the uncertain object representations by probabilistic feature vectors (pfv) [86]. In this method, we set the feature value of each object with its variance of the probability distribution. This index is also referred to as the Gauss-Tree, which is quite applicable to the probability range queries. We find that it significantly helps to improve the query results in quality and efficiency, compared to traditional feature vectors, and also can be applied to similarity search dependent on feature vectors in bioinformatics.

With the adaptive piecewise-linear approximations (APLA) techniques to summarize the probability histogram, we can represent the probability distribution of any object. We get the APLA-tree [87], which can answer probabilistic range queries of each level on the basis of such an idea, then define the neighbors relationship with the calculation of *expected nearest neighbors* (ENN) for a certain object q. This algorithm can help to avoid the high cost of probability calculation owing to its independence of the other inexact objects which are overlapped. Furthermore, a UV-diagram (uncertain Voronoi diagram) [88] extends convex polygons (Voronoi diagram) to support nearest neighbor query processing for uncertain data.

Angiulli and Fassetti [89] proposes a novel multidimensional index, i.e., UP-index, to answer range queries in the metric space. With the selection of a number of central points, it can help reduce the set of candidates that do not satisfy our conditions and implement pruning effectively due to the histogram established with the upper bounds between a point in the metric space and a center point. While taking advantage of this index, the candidates can be cut in the selection phase and the pruning can be completed independently of the dimension.

Inverted index The probabilistic inverted index comes from information retrieval, and applies to the query on uncertain classification data. Its basic idea is to maintain a list of lists. From the aspect of index structures, it is natural to use a random access method when the external list is small. Otherwise, it uses sequential access.

The inverted index is extended to deal with uncertain classification data, which comes to be probabilistic inverted index [90]. The element in its outer list corresponds to a domain element in one-to-one correspondence, whereas that of the inner list stores the documents where the tuple belongs, and is made up of the tuple-id along with its probability value.

In order to facilitate the query on the probability threshold value, methods such as row pruning, column pruning, and highest-prob-first have been developed.

Other indexes With the hypothesis of dependence of

data [91], the application of uncertain index technology is extended into the area of uncertain data. They represent the correlations in the probabilistic database with a junction tree, and an index structure is built using tree partitioning techniques.

2.4 Uncertain relational data mining

2.4.1 Clustering

The most general idea is to extend certain traditional data-oriented clustering methods into uncertain data. In traditional clustering algorithms, the input data are single-value data points, whereas in uncertain data clustering algorithms, each input point represents an object composed by multiple data points. These data points can be either continuous or discrete, but must follow a certain probability distribution. Note that uncertain data clustering and fuzzy clustering have similar concepts, but are essentially different. Fuzzy clustering is a soft partition and the processed data are certain, but the subordinate relationship of each data point is fuzzy, which allows a data point to belong to multiple clusters with different probabilities.

Partition-based uncertain clustering algorithms

Partition-based clustering algorithms organize a dataset with n data elements into $k(k \leqslant n)$ parts. Each part represents a cluster. Traditional partition algorithms mainly include the K-means algorithm and K-medoids algorithm. Chau et al. [92] first introduced an uncertain clustering algorithm, UK-means, when analyzing moving data clustering problems [93]. The UK-means algorithm is an iterative algorithm, which has similar idea as the K-means algorithm. It randomly chooses k center points $c_j(j = 1, 2, \ldots, k)$ to represent k clusters C_j, and the distance from data object $o_i(i = 1, 2, \ldots, n)$ to c_j is determined by computing the expected distance. Each data object is added to its nearest cluster, and then the cluster center is recomputed according to the clustering. This process iterates until the algorithm converges.

Ngai et al. [42] further improves the UK-means algorithm by applying sampling technique in a data modeling process to choose the data objects that meet certain PDF. In order to improve the algorithm's computational efficiency, they also introduce the MBR to describe the area where data points may occur. In addition, they propose multiple pruning strategies to reduce the quantity of the expected distance data to compute in the clustering process.

Because the main breakthrough point to improve the UK-means algorithm's efficiency is to reduce the computation of the expected distance, Lee et al. [94] proposes the a CK-means algorithm by simplifying the expected distance to the distance between data points. The idea comes from the simplified calculation method for the solid moment of inertia in a mechanical model, which simplifies the inertia moment to the measured relationship of a fixed point to the rotary axis by the parallel axis theorem. Thus, it removes the original integral form and reduces the computation amount. The CK-means algorithm's approach is: first calculate the center point of each data object, then perform traditional K-means algorithm for each of these center points. The theoretical analysis shows that CK-means algorithm's results are exactly the same as UK-means when using variance to define expected distance. Experiments show that, with the increase in the number of objects or the number of clusters, the CK-means algorithm has a significantly lower time cost than UK-means.

Another improvement [95] of the UK-means method is based on the Voronoi graph [96]. This method divides the space of k cluster centers into k cell structures $V(p_i)$, which is composed of perpendicular bisectors of the line connecting any two clusters. The position relationship between data object o_i's MBR and cell structure $V(p_i)$ are determined. Then, employ pruning by two pruning techniques, Voronoi-cell pruning and bisector pruning. At the same time, they combine the two strategies with [42]'s strategies, which can further improve the algorithm's efficiency.

Currently, the improvements on UK-means mainly consider how to reduce the computation of expected distance. However, UK-means and its improved algorithms have two shortcomings. First, the cluster center is represented by a single certain data point, which will lose uncertain information and affect the final clustering results' precision. Second, it produces a new cluster center in each iteration, and we need to compute the expected distances from every data object to this new cluster center, which is costly. Therefore, Gullo et al. [97] proposes the UK-medoids algorithm by extending the traditional K-medoids algorithm. First, calculate the expected distances between objects, for the cluster center set is generated from the input data objects. Thus, these distances only need to be computed once, and can be employed in every iteration. Then, select k medoid points using the PAM algorithm and use them to cluster iteratively. For each iteration, choose a real data object as a cluster center. In addition, Gullo et al. [97] also designs a novel measure for uncertain data distances, i.e., fuzzy distance function (FDF), which is adaptable to both discrete and continuous PDFs.

Cormode and McGregor [98] proposes a method that employs a function to compute the expected distance from an uncertain point to any center, then cluster with traditional

methods.

Density-based uncertain clustering algorithms

As discussed above, partition-based methods only find globular clusters. The clustering of arbitrary shape brings difficulties. Moreover, noise points have great impact on the cluster results. Density-based methods are designed to solve this problem, where clusters are treated as high-density regions separated by low-density regions.

Kriegel and Pfeifle [99] proposes the uncertain data FDBSCAN algorithm by improving the DBSCAN algorithm. The original DBSCAN algorithm's approach is to select one object that meets the core condition and does not belong to any cluster, and then create a new cluster on it. Based on this core object in this cluster, core objects are collected with reachable density into this cluster until no new core object is added. The process above is repeated to formulate the final clustering result. The FDBSCAN algorithm's clustering process is very similar to that of the DBSCAN algorithm. The difference is that they use different similarity measures, i.e., FDBSCAN employs a distance distribution function that reflects the probability density instead of the expected Euclidean distance. Experiments show that this measure is more precise on clustering results. With the same similarity measure, they propose the FOPTICS algorithm to improve the OPTICS algorithm [100]. This is an algorithm that is suitable for large datasets, visualizes output cluster results in an intuitive way, and clusters uncertain data more accurately.

The two algorithms are very sensitive to parameters, in order to overcome this shortcoming, Xu and Li [101] propose the uncertain data clustering algorithm P-DBSCAN based on DBCSAN and that probability index. First, they take full advantage of the data objects' probability distribution information in the definition of the core object and density-reachable, which make use of the MBR to select the minimum and maximum distance as a similarity measure instead of the expected distance. Second, in the initial step of the algorithm, build an R*-tree on the whole uncertain dataset using multidimensional indexing techniques. In the traversal process, start from the root node, objects that do not meet the parameters are pruned.

Other clustering algorithms

Hamdan and Govaert [102] propose a mixture model clustering algorithm of uncertain data, i.e., the improved expectation maximum (EM) algorithm. In this paper, data are multiple uncertain regions composed of multi-interval-value data. On this basis, compute the distribute parameters and find the maximum likelihood estimation of the mixture model parameters. Divide the datasets to get the fuzzy clusters. However, this method cannot be applied to other clustering methods.

Xiao and Hung [103] and Xu and Li [101] investigate how to select different methods to describe the uncertainty of data specific to different PDFs. For example, for uniformly distributed and Gaussian distributed data objects, we can use analytic solution (AS); for arbitrary distributed data objects, we can use distance between means (DM), pair-wise between random samples (PRS), grid approximation and pair-wise comparison between random samples(GAPS), pair-wise comparison between Gaussian mixture (PGM), and approximation by single Gaussian (ASG).

2.4.2 Classification

Classification is a supervised learning method to build a model based on a set of input data. Its mission is to obtain an objective function (also called classification model) by learning, and to map each set of attributes into a predefined class label set. Classification algorithms for uncertain data mainly come to the following two branches.

Uncertain classification of support vector machine Support vector machine (SVM) is a classification technique based on statistical learning theory, which has numerous applications in areas such as text classification, and handwriting number recognition. For the SVM algorithm of uncertain data, it takes outliers in input data (such as text or image attributes) as uncertainty. It is based on the assumption that no matter whether the outliers occur in training set or test set, they have similar influence on the training and testing process. Therefore, many classification methods only consider the influence by outliers in the training set or the testing set.

Inspired by the total least squares regression method, Bi and Zhang [104] propose the total support vector classification (TSVC) method for uncertain data. Its training set carries noise input, and the noise point is selected by a simple bounded model with uniform priors instead of general Gaussian noise. Meanwhile, this method also gives an intuitive geometric interpretation, which can be used to optimize the probability distance between any two classes on the edge. Experiments show that the TSVC method is superior to the standard SVM for both artificial and real datasets.

Moreover, Bhattacharyya et al. [105] proposes a method to build a binary classifier using second-order cone program (SOCP) and Chebyshev inequalities. This classifier applies to both classification problems and regression problems. In classification problems, separation of training set data is done by setting the outliers influence parameters in the worst case. In regression problems, given two improvements to SOCP

by Chebyshev inequalities, they are used to construct corresponding linear regression functions. The two methods above apply to simple linear models. For the more complex nonlinear models, Yang and Gunn propose three iterative algorithms in [106], [107]. One is the uncertainty support vector classification (USVC) method based on a linear model, which can be seen as a generalization of the SOCP problem. The other two are both iterative algorithms based on nonlinear models, taking noise-specific covariance information as input. The two methods are adaptive uncertainty support vector classification (AUSVC), which combines TSVC and USVC, and minmax probability support vector classification (MPSVC), which combines MPM and SVC. The latter two methods are empirically shown to be a more effective, more robust uncertain data classification methods.

Extended bayesian methods The Bayesian theorem is a statistical theory, which combines prior knowledge and new evidence collected from data. It mainly solves the uncertain relationship of attribute set and class label in classification. The reason why the Bayesian theorem can play an important role in classification problems is that its class condition probability together with prior probability and evidence can represent posterior probability. Traditional Bayesian classification methods mainly include two types, the Naive Bayesian and Bayesian Belief Networks.

Bayesian classification method is widely applied in bioinformatics. When studying tumor tissue gene and protein variability categories, Demichelis et al. [108] proposes a hierarchical naive Bayesian model classifier. This algorithm is based on the Bayesian hierarchical model, which provides a simple representation of sample probability distribution by concrete appropriate conditional independent structure and conditional probability distribution sets. Experiments show that when the unevenness of the sample classes' distributions is not the same, the hierarchical Bayesian method is more effective than the standard Bayesian method.

2.4.3 Frequent itemset mining over uncertain data

1) Apriori-based algorithms

The Apriori algorithm has played an important role in static database frequent itemset mining. In order to perform frequent itemset mining on uncertain data, Chui et al. [109] proposes the U-Apriori algorithm by improving the Apriori algorithm. In the Apriori algorithm, the candidate support is computed by accumulating a certain support counter. If the candidate occurs in a transaction, its support is incremented by 1. In uncertain databases, the support of every item is represented by probability, and the support of every itemset X is acquired by accumulating the probability product of every item x belonging to t_i.

Similar to Apriori, U-Apriori is also based on a generate-and-check framework, so it cannot apply to larger databases. In order to solve this problem, Chui et al. [109] proposes a pruning strategy to remove low-probability events in the original database. This forms a new database DT, where the U-Apriori algorithm is applied. In order to solve the threshold selection problem above, Chui and Kao [110] proposes a substitute method: Decremental Pruning.

2) Tree-structure-based algorithms

The tree-structure-based FP-growth algorithm is an effective traditional frequent itemset mining algorithm. Similar to the U-Apriori algorithm based on the Apriori algorithm, [111–113] propose a tree-structure-based frequent itemset mining algorithm UF-growth algorithm for uncertain databases. Similar to the FP-growth algorithm, the UF-growth algorithm is divided into two steps: 1) construct a UF-tree; 2) perform frequent itemset mining in the UF-tree. In the UF-tree constructing process, the most important aspect is how to store every item's information such that it is beneficial to the mining process later on. In order to save memory space, Leung et al. [113] propose two improvement schemes. By the two improvements, the UF-growth algorithm can save memory space and improve the algorithm's efficiency compared with Apriori-based algorithms.

In the evidential database area, mining frequent itemset on uncertain data also attracts concern. Evidential databases require the evidence theory to model data, i.e., using basic belief assignment to describe the uncertainty of every attribute. Hewawasam et al. [114] proposes the belief itemset tree (BIT) method. Tobji et al. [115] proposes a new data structure, record identifier lists (RidList), which speed up itemset support counting using a vertical description of an evidential database. Later, they improve the BIT method by FIM method [116]. This method calculates the maximum frequent itemset MF and its subsets, and gets the final result set F on this basis. In addition, feasible subsets in F can be quickly computed, which greatly improves the original algorithm's efficiency.

3 Uncertain XML data management

Uncertain hierarchical information can be formalized in terms of a probabilistic XML space (abbr. px-space), i.e., a probabilistic distribution over a set of ordinary XML docu-

ments. Typically, XML documents are modeled as unranked and unordered trees. A number of probabilistic XML models have been proposed to describe those px-spaces and various problems of managing probabilistic XML data have been studied, such as query evaluation and algebraic manipulation. Abiteboul et al. [117] presented a unified view of these different models in terms of p-documents that are trees with two types of nodes: ordinary and *distributional*. P-documents and types of distributional nodes are discussed in detail in Section 3.1.

3.1 Probabilistic XML data model

Abiteboul et al. [117] defined different families of p-documents in terms of the types of distributional nodes that are allowed. $PrXML^C$, where $C \in ind, mux, det, exp, cie$, denotes the family of p-documents that use the types appearing in the subset C.

Definition 2 Probabilistic XML space. A probabilistic XML space is a probability distribution over a space of ordinary documents. It is a pair (D, p), where D is a nonempty, finite set of documents and $p : D \rightarrow R^+$ maps every document $d \in D$ to a positive real number $p(d)$, such that $\sum_{d \in D} p(d) = 1$.

Definition 3 P-document. A p-document is a tree P that consists of two types of nodes. *Ordinary* nodes have labels, and they may appear in documents. *Distributional* nodes are only used for defining the probabilistic process that generates random documents.

For the purpose of introducing different distributions for both discrete and continuous types, we classify the distribution node into six types:

1) Independent type, *ind* [118, 119]. The child nodes of ind in the probabilistic XML document tree appear to be independent and do not affect each other. If the probability for ind node v to choose its child w is $p(w)$s, then the probability of selecting a subset C of node v is $\Sigma_{w \in C} p(w).E$

2) Mutually exclusive type, *mux* [118, 119]. The child nodes of mux can either occur once or do not appear at all. If the appearance probability of the mutually exclusive child nodes $w_1, w_2, \ldots, w_n$ are $p(w_1), p(w_2), \ldots, p(w_n)$ respectively.

3) Event-driven type, *cie* [120, 121]. The existence of cie nodes depends on the independent external event variables, such as $e_1, e_2, \ldots, e_m$. In other words, the occurrence or non-occurrence of external event variables determines the presence condition of the cie node.

4) Combination type, *exp* [122, 123]. There is a plurality of child nodes of exp. Therefore, we can select different child nodes to make up its collection. The probabilities of exp child nodes collections are $p(c_1), p(c_2), \ldots, p(c_n)$.

5) Determined type, *det*. Its child nodes must appear entirely.

6) Continuous probability distribution type, *cont* [122]. What the cont node describes is that its node must obey a continuous probability distribution, such as Binomial distribution, Poisson distribution, Gaussian distribution, Normal distribution, etc.

The former five types above introduce the probability of discrete distributions, and the last describes probability of a continuous distribution. According to the actual applications' needs, researchers use the combination of these six types of nodes and propose a new probability XML data model, PrXML.

3.1.1 Uncertain representation of junctions

In order to satisfy the needs of applications, researchers put forward the probability XML data model, which includes nodes of different distribution types. We express the general model with PrXML$\{type_1, type_2, \ldots\}$, where $type_1$ and $type_2$ represent any type of the above six distribution nodes. For example, PrXML$\{ind, mux\}$ represents a model that contains only independent nodes and distribution nodes. Apart from the node content uncertainty, probability XML data model describes structural uncertainty in a probabilistic XML model.

In the example of the previous model, PrXML$\{ind, mux\}$, the probability of any node depends on its father, and each relationship between them is either *mux* or *ind*. Then, in the next example of PrXML$\{cie\}$ [120, 121], we do not set values on the nodes or any edges to describe its uncertainty, but attach a series of events on each node variable. Then, the existence of a node depends on whether the related external event occurs.

In the model of PrXML$\{exp\}$ [122, 123], every node can make any combination for its child nodes. On the basis of [122] and [123], this text constrains the structure of the graph into a tree, and transforms the probability of a range into a point.

Kimelfeld et al. [19] propose a probability XML data model of PrXML$\{cie, exp\}$ which includes the probability of *cie* and *exp* nodes. [124] extends the probabilistic XML data model described in advance, allowing the leaf nodes to be counted to express continuous probability distributions, such

as the Binomial distribution, Poisson distribution, or Gaussian distribution. Therefore, the model could be generalized.

3.1.2 Representation of structured uncertainty

The model of possible worlds has a great application in uncertain databases. The probabilistic XML data model introduced above, i.e., PrXML$\{type1, type2, \dots\}$, establishes a possible world for the probabilistic XML database. On the basis of such a model, we can get an instance of the possible world through two procedures, in which the probability value can be calculated by the inherent characteristics of the distribution node.

- Select legitimate child nodes for each distribution node at random and delete all the unselected child nodes and their descendants.
- Delete all the distribution nodes. If an ordinary node v does not have its father, it can pick the nearest and the most common ancestor to its father node. We can get all the instances in a possible world by repeating the above two steps, which satisfy the condition its sum of probability is 1. In general, the number of instances in possible world is far higher than the scale of probability XML database, and may be even exponential to the latter. This is the key difficulty for probabilistic XML database management.

According to the needs to solve practical problems and the different combinations of the above five types of distribution nodes, a new probabilistic XML data model is proposed.

Kimelfeld et al. [125] investigate the conversion among these models based on their ability of expression. PrXML$\{ind\}$ and PrXML$\{mux\}$ cannot convert to each other, and the same for PrXML$\{exp\}$ and PrXML$\{cie\}$. However, all of them can be transformed into the form of PrXML$\{exp, cie\}$, which is the strongest in expressive power, in contrast to PrXML$\{ind\}$ and PrXML$\{mux\}$, whose ability comes to the weakest. Contrary to the six types of distribution node described above, we need to extend the conversion relationship among probabilistic XML data models to further enhance the expression power.

3.2 Querying over uncertain XML data

3.2.1 Query algebra of uncertain XML data

In accordance with the theory of relational algebra, XML query algebra is an operation collection of XML files that follow a certain data model. Then there are mainly two ideas in the implementation of probabilistic XML algebra. One is to extend XML Algebra (extended XML Algebra, shorted as e-XML Algebra), which increases actions to probabilistic data on the basis of established XML generation. The other is to implement operations of probabilistic XML based on relational algebra, which is suitable for probabilistic XML data management on XML databases.

e_XML Algebra The method of extending XML algebra has been applied first in [119]. The algebra system is based on the implementation of TIMBER and adds functions of managing probabilistic XML data in the query parser and query execution. This method was applied directly in the TAX algebra system, which is relatively simple on implementation, but not flexible in the aspects of modifying and extending XML algebra.

SP_algebra [126–130] introduce PXML algebra, SP_algebra, based on the relational algebra in connection with different PXML data models. Its manipulate objects are semi-structured probability ones, i.e., SPO. SP algebra includes standard set operations, such as union, and intersect. It also extends the operators of relational algebra (for example, selection, projection, join, etc.) for the purpose of managing probabilistic data. In addition, it defines a new conditionalization, which is relevant to probability calculation. The procedure for this method is as follows. It first transforms SPO into relational tables, and then applies PXML algebra SP algebra to them. This work does not involve the definition of aggregation function and the completeness proof of PXML algebra.

Hung et al. [123] define algebraic calculus for the data model of PSD, which are discussed in detail in the following, such as projection (ancestor projection), selection, and Cartesian product. In particular, the projection includes ancestor projection, descendant projection and the projection of a single node. The selection operators can be classified into object selection operator and value selection operator according to the selection of object and value.

3.2.2 Query semantics

According to the property of query results, we can classify the query semantics of PXML into the following types. One is object-oriented, whose results are node objects and attributes on the probabilistic XML tree. The other is a value-oriented query, whose results are the labels of nodes. In the second semantics, in the case of query results being probability numeric, the accuracy of probability calculation depends on the type distribution node [131].

3.2.3 Query method

At present, researchers have proposed many XML database query processing methods, of which the twig query is one of the popular query forms. Kimelfeld and Sagiv [132], Kimelfeld et al. [125] follows the query ideas of the twig pattern in PXML. Because the twig can be processed on PXML, its query program can be classified into the following aspects.

One is that to execute the query on the entire random file corresponding to the PXML file. In general, there are many real random files corresponding to one PXML file. According to the integrated approach of the PXML file, we can list the entire random XML file and the probability related to it. Queries are performed on a file [129]. Obviously, it is a really simple method but not effective in query processing, owing to the cost of listing the whole random XML files.

The other is to process the query on probabilistic files directly [118, 119]. This is a feasible method. The key point is the selection of probabilistic nodes needed in it. The implementation in [126] and [130] add probability operation functions in the mature XML prototype system. Kimelfeld and Sagiv [132] match twigs in probabilistic XML from the perspective of query semantics. We classify the semantics into three types as follows, while taking the different query modes or the conditions of their results into account. The first is Boolean, which is to check the satisfiability of the query. In other words, its work is the inspection of whether there exist solutions to meet the query conditions for a certain file and inquiry. The second is the complete semantics. Under the constraint of the former, it will return all the nodes that satisfy the exact mapping constraints. The last is the incomplete ones whose results do not fully meet the requirements. In all three types of semantics, the quest for obtaining the largest matching subtree becomes the core issue.

4 Uncertain graph data management

It was found that with the increasing scale of graph data, owing to the error in the data collection technique, out-of-date data, and data privacy conservation, a large number of graph data contain uncertainty. Here are some examples for graph uncertainty:

1) In bioinformatics, protein interaction networks are represented by graphs, where vertices represent protein and edges represent protein interactions. Owing to the intrinsic random errors and deviations in high-throughput biological assay techniques (such as the yeast two-hybrid technique), whether the experimentally measured protein interaction exists in reality is uncertain. The famous biological database STRING stores this uncertainty in the database.
2) The topology structure of wireless sensor network is a graph, where vertices represent sensors and edges represent the wireless communication links between sensors. Owing to the dynamic nature of wireless sensor networks, mobility of nodes, failure, and dormancy mechanism, whether there exists a communication link between two sensor nodes is uncertain.
3) In an intelligent transportation system, the road network is represented as a weighted graph, where vertices represent road intersections, edges represent road, and the edges' weights represent the traffic flow on the road. Owing to the random transmission delay of GPS, the current road network's data cannot accurately reflect the actual situation, which contains uncertainty.
4) In information networks, some links may fail, such that the links will be associated with probability. [133] describes the uncertainty in social networks and how to use uncertain relational database techniques to solve it.

As uncertain graph data has been widely employed, researchers have carried out research on uncertain graph models, basic data operation, and mining techniques. In this section, we will present a survey on these aspects.

4.1 Uncertain graph model

The uncertain graph semantic model plays an important role in uncertain graph management research. It provides the representation of uncertainty in graphs and its meaning, and thus lays the foundation for the formalized management and mining of uncertain graph data.

Hintsanen et al. [134, 135] propose probabilistic graph to plot uncertain graph data model. Given a graph $G(V, E, P)$, where V is the vertex set in graph G, E is the edge set in graph G, and each edge $e \in E$ has an associated probability $P(e)$ for functioning; each edge can fail with probability 1- $P(e)$. For example, in a transportation network, each edge has an associated blocking probability (due to the snowstorm, traffic accidents, etc.).

Zou et al. [136–138] expand the semantics of possible worlds in uncertain data management by providing the semantic model of possible worlds on graph data. In this model, an uncertain graph has a real number in [0, 1], denoting the existence possibility of this edge. Assume that all existence possibilities of edges are independent. By indepen-

dently and randomly selecting an edge based on the existence probability, we can get one certain graph implicated by the uncertain graph (i.e., possible world). This certain graph is called an implicated graph of the uncertain graph. It is proved theoretically that an uncertain graph expresses a probability distribution of all implicated graphs it contains. Since then, [72, 139–142] have all used this semantic model.

Different from the previous probability for edge, Zou et al. [143, 144] expand the semantic model of possible worlds by introducing a real number in [0, 1] on each vertex, denoting the existence possibility of vertex. At the same time, the real number on one edge denotes the existence possibility of this edge when the two endpoints both exist. This model assumes the existence possibilities on vertices are mutually independent, and the existence possibilities of edges are also mutually independent.

4.2 Basic operators over uncertain graph data

Potamias et al. [139] investigate the k-nearest neighbor selection operator over uncertain data, i.e., finding k nearest vertices with a given vertex in an uncertain graph. Owing to the existence of uncertainty, three distance concepts are proposed, the expected distance, median distance, and majority distance. In order to implement these operations, Potamias et al. [139] proposed a general algorithm framework and random walk algorithms for computing each distance.

In 2010, Yuan et al. [140] study the probability threshold-based shortest path computation on uncertain graphs, i.e., given an uncertain graph, vertices s, t, and threshold 0 <p< 1, compute all paths P between s and t, such that P has no less probability than p to become the shortest path between s and t.

Yuan and Wang [72] propose a probability reachable query operator, i.e., given an uncertain graph, vertex s and t, compute the probability for existing at least one path between s and t (i.e., reachable). They also design a random algorithm to perform probability reachable query operation in polynomial time.

In 2011, Yuan et al. [145] investigates the subgraph isomorphism problem on probabilistic graph databases. Yuan et al. [146] investigates the similarity search problem over probabilistic graph database. These two works both assume that edges among graphs are relative. They are proved to be #P-complete problems, and both are solved on a filter-and-verify framework. For subgraph isomorphism problem, Yuan et al. [145] proposes probabilistic inverted index (PIndex) based on subgraph optimal feature selection. In the filter phase, the Pindex is used to filter out the results that do not meet certain conditions. In the verification phase, precise bounded algorithm is used to check the obtained results. For similarity search problem, in order to speed up the query process, a probabilistic matrix index (PMI) based on lower and upper bounds of subgraph isomorphism are built. In the filter phase, the lower and upper bound based on PMI subgraph similarity is used to filter intermediate results. A large number of probabilistic graphs can be sorted by PMI to maximize the pruning efficiency. In the verification phase, a sampling method is proposed to verify the candidate subgraphs.

4.3 Uncertain graph data mining

4.3.1 Frequent subgraph pattern mining algorithms

Frequent subgraph pattern mining on an uncertain graph is to find all subgraph patterns with high possibility and high frequency from an uncertain graph. This pattern mining plays an important role for the study of common sub-pattern in evolution in a biological network with multiple species [147].

Zou et al. [136–138] investigate a frequent subgraph pattern mining algorithms on uncertain graph in expectation semantics. Owing to the existence of uncertainty in a graph, the frequency measurement will change. Zou et al. [136, 138] propose the notion of expected support, used to measure the expected frequency for a subgraph pattern to appear in uncertain graphs. Thus, this mining problem's objective is to quickly discover all frequent subgraph patterns with no less support than p from a set of uncertain graphs, where p is a real number in (0, 1). Zou et al. [136] proves this problem is #P-hard [148], then provides an approximate expected frequent subgraph pattern mining algorithm. This is an approximate algorithm that allows a small fraction of infrequent subgraph patterns in the mining result. However, the expected support should be no less than $(1 - \epsilon) \times p$, where $0 < \epsilon < 1$ is an extremely small number. This algorithm is the first frequent subgraph pattern mining algorithm. Inspired by this algorithm, Papapetrou et al. [141] proposes an indexing technique to further improve the efficiency of the algorithm.

In uncertain graph mining, uncertainty needs to be included in the measurement of the importance of knowledge, and there are a variety of methods to accomplish this. Zou et al. [143] investigates frequent subgraph pattern mining algorithms in probabilistic semantics, and proposes the notion of subgraph pattern ψ frequent probability, which is used to measure the probability for the subgraph pattern with frequency no less than threshold ψ in an uncertain graph. Thus, the mining problem is how to quickly discover all subgraph patterns with no less frequent probability than $p(0 < p < 1)$

from a set of uncertain graphs. Zou et al. [143] proves this problem is #*P*-hard, and then provides an approximate probability frequent subgraph pattern mining algorithm. The algorithm also takes an approximate mining method, i.e., allowing the mining result to have a small fraction of non-frequent subgraph pattern. However, its ψ frequency probability should be no less than $p - \epsilon$, where $0 < \epsilon < p$ is an extremely small number.

4.3.2 Feature subgraph pattern mining algorithms

One of the important applications of frequent subgraph patterns is playing as features in classification and clustering. However, there are usually many frequent subgraph patterns and it is common for them to overlap over each other. In 2010, Han et al. [142] proposes an algorithm for mining any k maximal frequent subgraph pattern with expected support of no less than p, $0 < p < 1$ from an uncertain graph. Subgraph pattern S is a maximal frequent subgraph pattern, iff S has an expected support no less than p, and there is no other frequent subgraph pattern S', such that S is a subgraph of S'. This algorithm is a new mining algorithm based on a random walk.

4.3.3 Dense subgraph mining algorithms

Dense subgraph mining on uncertain graphs is to find a family of vertex sets, such that a subgraph induced by any vertex subset has a high probability to be a dense subgraph. Depending on different application requirements, the definition of dense subgraph is not the same, mainly including clique (complete graph), quasi clique (quasi complete graph) and k-connected subgraph. The dense subgraph mining on uncertain graph problem plays an important role in studying functional modules in biological networks.

Aiming at the strictest dense subgraph definition clique, Zou et al. [144] investigates mining top-k maximal cliques in uncertain data. Zou et al. [144] proposes the notion of maximal clique probability for vertex subsets, i.e., the probability for a vertex subset to become the maximal clique in an uncertain graph. The objective of this mining problem is to quickly find k vertex itemsets with size of at least s and the greatest probability to become the maximal clique. Zou et al. [144] proves this problem is NP-hard and provides a two-phase bound-and-search algorithm to mine the top-k cliques.

5 Uncertain data stream management

Stream data is an important form of data. Unlike traditional databases, stream data often have an endless data amount. Thus, one-pass algorithm, online process and analysis algorithm should be investigated. In data stream applications such as wireless sensor networks and RFID, the data are uncertain. For the management of uncertain data in these applications, researchers investigate from the two aspects of modeling and query processing. In this section, we present a survey on related results of these two aspects.

5.1 Uncertain data stream model

In uncertain data streams, tuples have uncertainty. Jayram et al. [34] assumes the tuples' value range in a discrete domain; the value of each tuple is a PDF on these discrete domains. Part of the research work tends to study a special basic case, i.e., that each stream tuple only has occurrence and non-occurrence possibilities. This model is usually called the point probability model [36]. Data stream models include the landmark model, slide window model, and decay window model. The landmark model considers all tuples from a fixed start time point to now. The slide window model only considers the newest W tuples, whenever there reaches a new tuple, a new tuple is eliminated. The decay window model does not eliminate tuples, but the importance of each tuple reduces with time. In all these time models, the arrival of new tuple will cause rapid change in possible world instance.

In 2008, Jin et al. [149] propose the sliding window model. Existing research results usually do not consider the correlation between tuples, assuming that tuples arrived at different time have no causal connection. Ré C [150] investigates on the case of multiple data stream applications. In this case, correlations exist among multiple data streams.

5.2 Uncertain data steam query processing

Traditional certain data-stream-oriented methods can be applied to uncertain data stream applications after repairing. For example, AMS sketch [151] can be used to process aggregation query, especially the F_2 problem. FM sketch [152] can be used to find the number of unique elements in a data stream. Cluster Feature is widely used to design all kinds of online cluster algorithms [153, 154]. Cormode and Garofalakis [36] develops AMS sketch and FM sketch methods by introducing a probability parameter and constructing pAMS structure and pFM structure, which can perform queries on uncertain data streams [33]. Aggarwal and Yu [155] revised the CF method by providing an error-based CF (ECF) method, which solves the cluster problem on data streams.

Li and Ge [156] investigate uncertain data streams by map-

ping the data into strings with noise and probability. The slide window sequence matching query problem is defined as querying the subsequence with possibility of a possible world with matching a given pattern in a sliding window on an uncertain sequence larger than a given threshold. Li and Ge [156] devises an exact algorithm, EXACT-MATCH, of complexity $O(lw)$, where l is the length of the pattern, and w is the window size threshold. For many applications with large w, Li and Ge [156] gives a randomized approximation algorithm with a time complexity of $O(lw^{1/2})$. To further accelerate query processing, it uses a technique called adaptive filtering (AF) that only tries to match a prefix of the pattern. The parameters of AF are chosen based on the contents of the sequence stream using the gradient descent optimization technique. Li and Ge [156] also proposes algorithms that handle forms of negation.

The problem of join on uncertain data streams (USJ) [157] is performed by incrementally maintaining USJ query results. The basic idea in [157] is two heuristic pruning methods, i.e., object-level pruning and sample-level pruning. The former's main principle is a spatial edge pruning method and the latter is a pre-computation-based probability pruning method.

Ge and Liu [158] is the first work to consider the accuracy fo an uncertain stream database, in which accuracy is taken into consideration all the way from the learned distributions based on raw data samples to the query results. Ge and Liu [158] performs an initial study of various components in an accuracy-aware uncertain stream database system, including the representation of accuracy information and how to compute query results' accuracy. In addition, Ge and Liu [158] proposes novel predicates based on hypothesis testing for decision-making using data with limited accuracy.

Peng et al. [159] investigates a class of complex queries including selection and join predicates. They require the returned query answers' existence probabilities to pass a certain threshold. Peng et al. [159] optimize this query in three aspects. 1) Expediting joins using new indexes on uncertain data, 2) expediting selections by reducing the dimensionality of integration and using faster filters, and 3) optimizing a query plan using a dynamic, per-tuple-based approach.

The PODS [22] and CLARO [23] systems support stream processing for uncertain data using continuous random variables. CLARO employs a unique data model that is flexible and allows efficient computation. Based on this model, the system provides effective support for complex relational operation including aggregation and join by exploring statistical theory and approximation.

Jayram et al. [34] first studies the problem of computing aggregation functions on data stream, especially AVG function. The method is complex and mainly employs generation functions, which are difficult to extend to solve other problems. Their follow-up [160] can solve more aggregation query problems, including F0 and F2, and improves the query efficiency of AVG function query processing. Zhang et al. [161] defines the frequent element problem on uncertain data stream, and designs a solution. In order to solve dimensionality problem faced by high-dimensional data clustering, Aggarwal et al. [162] proposes an algorithm for high-dimension projected clustering of uncertain data streams. Similarly, to break through the restriction of some specific uncertain models by UMicro, Zhang et al. [163] proposes a clustering algorithm for uncertain streams based on information entropy. It uses information entropy to measure the uncertainty information in tuples and considers the influence from both uncertainty and distance aspects.

Jin et al. [149] first proposes a query processing method for the sliding window model. As previously mentioned, there exist many types of top-k queries. They propose a top-k query framework. First, we can design compact sets for top-k queries, and each compact set contains a part of data. The compact set has two properties: 1) being able to compute top-k queries; 2) being able to be maintained incrementally. However, compact sets still cannot answer queries in the sliding window model. Therefore, multiple compact sets are compressed to obtain better spatial complexity and time complexity. Their SCSQ-buffer strategy is outstanding in both time complexity and space complexity.

Zhang et al. [164] introduces the probability Skyline query method into data streams, and theoretically proves that the set SN, $\epsilon = O \in WN \& Pnew(O) \geqslant \epsilon$ has the bound of $O(logN)$, where WN is a set composed of all elements in a sliding window with length N, $Pnew(O)$ denotes the probability that none of the new arrival elements dominates O, and ϵ is the probability threshold.

5.3 Data stream applications

Wireless sensor networks and RFID are two typical data stream applications. Next, we will introduce the newest application techniques on the two aspects.

5.3.1 Uncertain data in wireless sensor networks

Subramanian et al. [165] proposes an online outlier detection method in sensor data using nonparametric models. This framework can be extended in order to identify either distance or density-based outliers in a single pass over the data,

and with limited memory requirements. Cheng et al. [81] proposes a probabilistic query evaluation method for sensor networks and moving objects based upon uncertain data. First, a classification of queries is made based upon the nature of the result set: value-based non-aggregation, entity-based non-aggregation, entity-based aggregation and value-based aggregation. For each class, Cheng et al. [81] also develops algorithms for computing probabilistic answers. Deshpande et al. [166] proposes a model-driven data acquisition method in sensor networks. Because a sensor network is a typical data stream application, its data cannot be stored exhaustively in a database system. Therefore, Deshpande et al. [166] suggests a model-driven method to capture data and a probability model to calculate and control the error confidence. Cheng et al. [31] uses a PDF to describe the uncertainty of sensor data. Based on R-tree, Cheng et al. [31] proposes two index structures for uncertain data and their corresponding probability threshold query processing algorithms. Hida et al. [167] discusses the aggregation query problem under uncertainty in sensor networks. They also design an outlier detection algorithm, such that the aggregation query processing algorithm has sufficient fault tolerance to adapt the sensor network's failure or error data.

5.3.2 Uncertain RFID data management techniques

An RFID application system is actually a data stream system, especially an event stream management system whose most important data is the label locations. Welbourne et al. [168] builds a location model to describe the uncertainty from a label object's location, which is expressed with relation $At(time, tagID, loc, prob)$. *Loc* information is inferred by a particle filter on raw data. *Prob* is the probability that an event occurs, which quantifies the uncertainty of the location information. For example, quadruple (1:10pm, 10, room230, 0.75) denotes the simple event that label object 10 is at room 230 at 1:10 pm with a probability of 75%. This uncertainty only reflects the existence uncertainty at the tuple level, and tuples expressing the same time, same object, and same location information are independent of each other. In the *AT* quadruple, the reader identifier *r* is replaced by an inferred label location, which meets the applications. An *AT* quadruple can be seen as a basic event, i.e., an event occurs at a single time point. Multiple basic events after temporal operation constitute a complex event. A probability event extractor PEEX can continuously extract and store events defined by developers and users.

Kanagal and Deshpande [169] deal with query analysis on probability data streams, but their work concentrates on a few query types such as selection, mapping, and aggregation query. The Lahar system can deal with imprecise data streams, especially in RFID environments. It can process imprecise information such as misreading, collision data, and mismatch granularity.

6 Quality of uncertain data

Because uncertainty may cause low-quality results, quality issues on uncertain data [49, 170–175] have received increasing attention in recent years. Two major problems in this area have been studied: 1) how to integrate uncertain data sources? and 2) how to clean uncertain data? Current approaches to these problems are discussed in this section.

6.1 Uncertain data integration

The first approach to uncertain data integration has been proposed in [175]. As a crucial first step in data integration, probabilistic schema mappings are studied in [175]. There are two possible semantics for probabilistic schema mappings. One is the *by-table* semantics, which assumes the existence of a single correct mapping. The other is the *by-tuple* semantics, which assumes that the correct mapping depends on the particular tuples. The authors show that the query complexity for answering queries in the presence of probabilistic schema mappings is PTIME for *by-table* semantics and #P-complete for *by-tuple* semantic. Van Keulen and De Keijzer [173] studied the integration problem of uncertain XML data. User feedback was applied to improve the data quality by updating the database. In [173], two types of feedback are proposed. Positive/negative feedback indicates the corresponding answer should (or not) occur in the real world. Duplicate detection in uncertain data has attracted little attention. Panse et al. [172] is the first work on deduplication in probabilistic data. Two similarity metrics have been defined based on two different matching models, knowledge-based matching rules and probabilistic matching rules.

6.2 Uncertain data cleaning

In recent years, researchers studied the issues of cleaning uncertain data. An entropy-based quality metric, PWS-quality, has been proposed by Cheng et al. [174] to evaluate the ambiguity degree of probabilistic query answers (range and maximum queries) based on the Possible World Semantics (PWS). The authors also studied the problem of how to choose the set of uncertain objects to be cleaned in order to maximize the

quality of a query answer under a limited budget. However, the PWS-quality computing algorithm proposed by Cheng et al. [174] cannot be trivially extended to support probabilistic top-k queries efficiently. This motivates the need for designing more efficient PWS-quality computing algorithms for probabilistic top-k queries [171]. Moreover, without assuming the cleaning operation to be successful, Mo et al. [171] proposed a sophisticated cleaning model in which a cleaning operation is successful with a probability. Zhang et al. [170] studied how to leverage the power of crowdsourcing to improve the quality of uncertain data, where the error rate of the crowd has been taken into account. Zhang et al. [170] proved that it is NP-hard to find a set of k human intelligence tasks (HITs) such that the expected improvement of data quality is maximized. An effective approximation algorithm and an efficient heuristic method have been proposed to solve the problem.

7 Conclusion and future directions

The uncertain data management problem exists in incomplete databases, and fields such as machine learning and artificial intelligence. In recent years, with the requirements of many applications such as wireless sensor networks, RFID, Web information integration, and spatial–temporal data management, uncertain data management has become one of the hot issues in database literature. Researchers have studied various types of uncertain data and have produced a wealth of research results. In this paper, we provide an overview of these results according to the type of data and research issues. This work surveyed several important issues in uncertain data management, as they are described in the literature. Some of them are based on relational DBMS, and others are based on complex data structure. Nevertheless, a number of these data models, basic operators, and query processing techniques, address uncertainty in a similar way. Most of the models deal with uncertainty by attaching additional probability information to the original data. However, differences exist in their assumptions; some researchers assume uncertainty in only part of the data (for example, only in nodes of graphs, only in edges, or only contains structural uncertainty in XML data) for simplicity. Like many other database-related problems, uncertain data query processing is usually highly reliant on carefully designed indexes.

Although current research results can deal with various types of tasks, including query processing and data mining, there still exists some research work to do in the field of uncertain data management.

1) New uncertain models. Possible world is the most popular model; however, a significant problem of this model is its high complexity, especially when there are many associations between data objects, and the independence assumption does not hold for many efficient algorithms, which leads to no employment of these algorithms. Thus, we need to design new models to efficiently solve uncertain data query processing and mining issues.
2) Application-specific methods. Current methods usually require knowing the correlation and possibility of data; however, this is often closely related to applications. Thus, the need for uncertain data management methods closely related to specific applications is shown. Researchers can solve this issue systematically from application-oriented aspects such as the discovery, labeling, storage, and transmission of uncertain data.
3) Parallelization. Uncertainty exists in big data, and thus linear or sublinear algorithms are in demand to solve these tasks both efficiently and effectively. Because the cost of current solutions of uncertain data computing tasks is relatively high, they cannot be directly applied to big data. Query processing and mining on big data is a newly raised challenge. Knowing the fact that most query evaluation algorithms are effortlessly parallelizable, researchers could pay more attention to solving these problems in parallelization and developing more parallelizable algorithms.
4) Trade-off between efficiency and effectiveness. Even though many models and approaches for uncertain data management and mining have been proposed, currently most works of this area appear far from real applications. This is partially because of the high complexity of evaluating queries over uncertain data. Thus, approaches that balance efficiency and effectiveness are in demand to make the approaches for uncertain data management and mining practical.

Acknowledgements This paper was partially supported by NSFC (61602159, U1509216, 61472099, 61133002), National Sci-Tech Support Plan (2015BAH10F01) and the Scientific Research Foundation for the Returned Overseas Chinese Scholars of Heilongjiang Province (LC2016026).

References

1. Fuhr N, Rölleke T. A probabilistic relational algebra for the integration of information retrieval and database systems. ACM Transactions

on Information Systems, 1997, 15(1): 32–66

2. Imieliński T, Lipski W. Incomplete information in relational databases. Journal of the ACM, 1984, 31(4): 761–791
3. Barbará D, Garcia-Molina H, Porter D. The management of probabilistic data. IEEE Transactions on Knowledge and Data Engineering, 1992, 4(5): 487–502
4. Lakshmanan L V, Leone N, Ross R, Subrahmanian V S. Probview: a flexible probabilistic database system. ACM Transactions on Database Systems, 1997, 22(3): 419–469
5. Zimányi E. Query evaluation in probabilistic relational databases. Theoretical Computer Science, 1997, 171(1): 179–219
6. Sen P, Deshpande A. Representing and querying correlated tuples in probabilistic databases. In: Proceedings of the 23rd International Conference on Data Engineering. 2007, 596–605
7. Suciu D. Probabilistic databases. SIGACT News, 2008, 39(2): 111–124
8. Cavallo R, Pittarelli M. The theory of probabilistic databases. In: Proceedings of the 13th International Conference on Very Large Data Bases. 1987, 71–81
9. Benjelloun O, Sarma A D, Halevy A, Widom J. ULDBS: databases with uncertainty and lineage. In: Proceedings of the 32nd International Conference on Very Large Data Bases. VLDB Endowment, 2006, 953–964
10. Sen P, Deshpande A, Getoor L. Read-once functions and query evaluation in probabilistic databases. Proceedings of the VLDB Endowment, 2010, 3(1–2): 1068–1079
11. Olteanu D, Huang J. Using OBDDs for efficient query evaluation on probabilistic databases. In: Proceedings of the International Conference on Scalable Uncertainty Management. 2008, 326–340
12. Roy S, Perduca V, Tannen V. Faster query answering in probabilistic databases using read-once functions. In: Proceedings of the 14th International Conference on Database Theory. 2011, 232–243
13. Kenig B, Gal A, Strichman O. A new class of lineage expressions over probabilistic databases computable in P-time. In: Proceedings of the 7th International Conference on Scalable Uncertainty Management. 2013, 219–232
14. Widom J. Trio: a system for integrated management of data, accuracy, and lineage. Stanford Infolab, 2004
15. Antova L, Koch C, Olteanu D. Maybms: managing incomplete information with probabilistic world-set decompositions. In: Proceedings of the 23rd International Conference on Data Engineering. 2007, 1479–1480
16. Cheng R, Singh S, Prabhakar S. U-DBMS: a database system for managing constantly-evolving data. In: Proceedings of the 31st International Conference on Very Large Data Bases. VLDB Endowment, 2005, 1271–1274
17. Boulos J, Dalvi N, Mandhani B, Mathur S, Re C, Suciu D. Mystiq: a system for finding more answers by using probabilities. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. 2005, 891–893
18. Olteanu D, Huang J, Koch C. Sprout: lazy vs. eager query plans for tuple-independent probabilistic databases. In: Proceedings of the 25th International Conference on Data Engineering. 2009, 640–651
19. Kimelfeld B, Kosharovsky Y, Sagiv Y. Query evaluation over probabilistic XML. The International Journal on Very Large Data Bases, 2009, 18(5): 1117–1140
20. Senellart P, Souihli A. Proapprox: a lightweight approximation query processor over probabilistic trees. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data. 2011, 1295–1298
21. Welbourne E, Khoussainova N, Letchner J, Li Y, Balazinska M, Borriello G, Suciu D. Cascadia: a system for specifying, detecting, and managing rfid events. In: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services. 2008, 281–294
22. Tran T T, Peng L, Li B, Diao Y, Liu A. PODS: a new model and processing algorithms for uncertain data streams. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. 2010, 159–170
23. Tran T T, Peng L, Diao Y, McGregor A, Liu A. Claro: modeling and processing uncertain data streams. The International Journal on Very Large Data Bases, 2012, 21(5): 651–676
24. Aggarwal C C, Yu P S. A survey of uncertain data algorithms and applications. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(5): 609–623
25. Zhou A Y. A survey on the management of uncertain data. Chinese Journal of Computers, 2009, 32(1): 1–16
26. Kimelfeld B, Senellart P. Probabilistic XML: Models and Complexity. Advances in Probabilistic Databases for Uncertain Information Management, Springer, Berlin, Heidelberg, 2013, 39–66
27. Sarma A D, Benjelloun O, Halevy A, Widom J. Working models for uncertain data. In: Proceedings of the 22nd International Conference on Data Engineering. 2006, 7
28. Green T J, Tannen V. Models for incomplete and probabilistic information. In: Proceedings of the International Conference on Extending Database Technology. 2006, 278–296
29. Sen P, Deshpande A, Getoor L. PRDB: managing and exploiting rich correlations in probabilistic databases. The International Journal on Very Large Data Bases, 2009, 18(5): 1065–1090
30. Chen R, Mao Y, Kiringa I. GRN model of probabilistic databases: construction, transition and querying. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. 2010, 291–302
31. Cheng R, Xia Y, Prabhakar S, Shah R, Vitter J S. Efficient indexing methods for probabilistic threshold queries over uncertain data. In: Proceedings of the 30th International Conference on Very Large Data Bases. VLDB Endowment, 2004, 876–887
32. Tao Y, Cheng R, Xiao X, Ngai W K, Kao B, Prabhakar S. Indexing multi-dimensional uncertain data with arbitrary probability density functions. In: Proceedings of the 31st International Conference on Very Large Data Bases. VLDB Endowment, 2005, 922–933
33. Burdick D, Deshpande P M, Jayram T, Ramakrishnan R, Vaithyanathan S. Olap over uncertain and imprecise data. In: Proceedings of the 31st International Conference on Very Large Data Bases. VLDB Endowment, 2005, 970–981
34. Jayram T, Kale S, Vee E. Efficient aggregation algorithms for proba-

bilistic data. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2007, 346–355

35. Dalvi N, Suciu D. Efficient query evaluation on probabilistic databases. In: Proceedings of the 30th International Conference on Very Large Data Bases. VLDB Endowment, 2004, 864–875
36. Cormode G, Garofalakis M. Sketching probabilistic data streams. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. 2007, 281–292
37. Ross R, Subrahmanian V, Grant J. Aggregate operators in probabilistic databases. Journal of the ACM, 2005, 52(1): 54–101
38. Kanagal B, Deshpande A. Efficient query evaluation over temporally correlated probabilistic streams. In: Proceedings of the 25th International Conference on Data Engineering. 2009, 1315–1318
39. Burdick D, Deshpande P M, Jayram T, Ramakrishnan R, Vaithyanathan S. Efficient allocation algorithms for olap over imprecise data. In: Proceedings of the 32nd International Conference on Very Large Data Bases. VLDB Endowment, 2006, 391–402
40. Ré C, Suciu D. The trichotomy of having queries on a probabilistic database. The International Journal on Very Large Data Bases, 2009, 18(5): 1091–1116
41. Fink R, Han L, Olteanu D. Aggregation in probabilistic databases via knowledge compilation. Proceedings of the VLDB Endowment, 2012, 5(5): 490–501
42. Ngai W K, Kao B, Chui C K, Cheng R, Chau M, Yip K Y. Efficient clustering of uncertain data. In: Proceedings of the 6th International Conference on Data Mining. 2006, 436–445
43. Agrawal P, Widom J. Confidence-aware join algorithms. In: Proceedings of the 25th International Conference on Data Engineering. 2009, 628–639
44. Cheng R, Singh S, Prabhakar S, Shah R, Vitter J S, Xia Y. Efficient join processing over uncertain data. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management. 2006, 738–747
45. Kriegel H P, Kunath P, Pfeifle M, Renz M. Probabilistic similarity join on uncertain data. In: Proceedings of the International Conference on Database Systems for Advanced Applications. 2006, 295–309
46. Ljosa V, Singh A K. Top-*k* spatial joins of probabilistic objects. In: Proceedings of the 24th International Conference on Data Engineering. 2008, 566–575
47. Jestes J, Li F, Yan Z, Yi K. Probabilistic string similarity joins. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. 2010, 327–338
48. Lian X, Chen L. Set similarity join on probabilistic data. Proceedings of the VLDB Endowment, 2010, 3(1–2): 650–659
49. Andritsos P, Fuxman A, Miller R J. Clean answers over dirty databases: probabilistic approach. In: Proceedings of the 22nd International Conference on Data Engineering. 2006, 30
50. Wick M, McCallum A, Miklau G. Scalable probabilistic databases with factor graphs and mcmc. Proceedings of the VLDB Endowment, 2010, 3(1–2): 794–804
51. Qi Y, Jain R, Singh S, Prabhakar S. Threshold query optimization for uncertain data. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. 2010, 315–326
52. Moore K F, Rastogi V, Ré C, Suciu D. Query containment of tier-2 queries over a probabilistic database. In: Proceedings of the VLDB Workshop on Management of Uncertain Data. 2010, 47–62
53. Ge T, Grabiner D, Zdonik S. Monte carlo query processing of uncertain multidimensional array data. In: Proceedings of the 27th International Conference on Data Engineering. 2011, 936–947
54. Soliman M A, Ilyas I F, Chang K C C. Top-*k* query processing in uncertain databases. In: Proceedings of the 23rd International Conference on Data Engineering. 2007, 896–905
55. Yi K, Li F, Kollios G, Srivastava D. Efficient processing of top-*k* queries in uncertain databases with x-relations. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(12): 1669–1682
56. Huang Y K, Chen C C, Lee C. Continuous k-nearest neighbor query for moving objects with uncertain velocity. GeoInformatica, 2009, 13(1): 1–25
57. Zhang X, Chomicki J. Semantics and evaluation of top-*k* queries in probabilistic databases. Distributed and Parallel Databases, 2009, 26(1): 67–126
58. Hua M, Pei J, Zhang W, Lin X. Ranking queries on uncertain data: a probabilistic threshold approach. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. 2008, 673–686
59. Cormode G, Li F, Yi K. Semantics of ranking queries for probabilistic data and expected ranks. In: Proceedings of the 25th International Conference on Data Engineering. 2009, 305–316
60. Ge T, Zdonik S, Madden S. Top-*k* queries on uncertain data: on score distribution and typical answers. In: Proceedings of the 35th ACM SIGMOD International Conference on Management of Data. 2009, 375–388
61. Soliman M A, Ilyas I F. Ranking with uncertain scores. In: Proceedings of the 25th International Conference on Data Engineering. 2009, 317–328
62. Li J, Deshpande A. Ranking continuous probabilistic datasets. Proceedings of the VLDB Endowment, 2010, 3(1–2): 638–649
63. Cheng R, Chen J, Mokbel M, Chow C Y. Probabilistic verifiers: evaluating constrained nearest-neighbor queries over uncertain data. In: Proceedings of the 24th International Conference on Data Engineering. 2008, 973–982
64. Cheng R, Chen L, Chen J, Xie X. Evaluating probability threshold k-nearest-neighbor queries over uncertain data. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. 2009, 672–683
65. Zhang Y, Lin X, Zhu G, Zhang W, Lin Q. Efficient rank based knn query processing over uncertain data. In: Proceedings of the 26th International Conference on Data Engineering. 2010, 28–39
66. Lian X, Chen L. Probabilistic group nearest neighbor queries in uncertain databases. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(6): 809–824
67. Yuen S M, Tao Y, Xiao X, Pei J, Zhang D. Superseding nearest neighbor search on uncertain spatial databases. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(7): 1041–1055

68. Cheema M A, Lin X, Wang W, Zhang W, Pei J. Probabilistic reverse nearest neighbor queries on uncertain data. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(4): 550–564
69. Lian X, Chen L. Probabilistic inverse ranking queries in uncertain databases. The International Journal on Very Large Data Bases, 2011, 20(1): 107–127
70. Lian X, Chen L. Efficient processing of probabilistic reverse nearest neighbor queries over uncertain data. The International Journal on Very Large Data Bases, 2009, 18(3): 787–808
71. Pei J, Jiang B, Lin X, Yuan Y. Probabilistic skylines on uncertain data. In: Proceedings of the 33rd International Conference on Very Large Data Bases. VLDB Endowment, 2007, 15–26
72. Yuan Y, Wang G. Answering probabilistic reachability queries over uncertain graphs. Chinese Journal of Computers, 2010, 33(8): 1378–1386
73. Lian X, Chen L. Top-*k* dominating queries in uncertain databases. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. 2009, 660–671
74. Grädel E, Gurevich Y, Hirsch C. The complexity of query reliability. In: Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. 1998, 227–234
75. Dalvi N, Suciu D. The dichotomy of conjunctive queries on probabilistic structures. In: Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. 2007, 293–302
76. Fagin R, Lotem A, Naor M. Optimal aggregation algorithms for middleware. In: Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. 2001, 102–113
77. Li J, Saha B, Deshpande A. A unified approach to ranking in probabilistic databases. Proceedings of the VLDB Endowment, 2009, 2(1): 502–513
78. Li F, Yi K, Jestes J. Ranking distributed probabilistic data. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data. 2009, 361–374
79. Dai X, Yiu M L, Mamoulis N, Tao Y, Vaitis M. Probabilistic spatial queries on existentially uncertain data. Advances in Spatial and Temporal Databases, 2005, 400–417
80. Yiu M L, Mamoulis N, Dai X, Tao Y, Vaitis M. Efficient evaluation of probabilistic advanced spatial queries on existentially uncertain data. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(1): 108–122
81. Cheng R, Kalashnikov D V, Prabhakar S. Evaluating probabilistic queries over imprecise data. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. 2003, 551–562
82. Kriegel H P, Kunath P, Renz M. Probabilistic nearest-neighbor query on uncertain objects. In: Proceedings of the International Conference on Database Systems for Advanced Applications. 2007, 337–348
83. Lian X, Chen L. Probabilistic inverse ranking queries over uncertain data. In: Proceedings of the International Conference on Database Systems for Advanced Applications. 2009, 35–50
84. Lian X, Chen L. Monochromatic and bichromatic reverse skyline search over uncertain databases. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. 2008, 213–226
85. Tao Y, Xiao X, Cheng R. Range search on multidimensional uncertain data. ACM Transactions on Database Systems, 2007, 32(3): 15
86. Bohm C, Pryakhin A, Schubert M. The gauss-tree: efficient object identification in databases of probabilistic feature vectors. In: Proceedings of the 22nd International Conference on Data Engineering. 2006, 9
87. Ljosa V, Singh A K. APLA: indexing arbitrary probability distributions. In: Proceedings of the 23rd International Conference on Data Engineering. 2007, 946–955
88. Cheng R, Xie X, Yiu M L, Chen J, Sun L. UV-diagram: a voronoi diagram for uncertain data. In: Proceedings of the 26th International Conference on Data Engineering. 2010, 796–807
89. Angiulli F, Fassetti F. Indexing uncertain data in general metric spaces. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(9): 1640–1657
90. Singh S, Mayfield C, Prabhakar S, Shah R, Hambrusch S. Indexing uncertain categorical data. In: Proceedings of the 23rd International Conference on Data Engineering. 2007, 616–625
91. Kanagal B, Deshpande A. Indexing correlated probabilistic databases. In: Proceedings of the 35th SIGMOD International Conference on Management of Data. 2009, 455–468
92. Chau M, Cheng R, Kao B, Ng J. Uncertain data mining: an example in clustering location data. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2006, 199–204
93. Li Y, Han J, Yang J. Clustering moving objects. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2004, 617–622
94. Lee S D, Kao B, Cheng R. Reducing UK-means to K-means. In: Proceedings of the 7th International Conference on Data Mining Workshops. 2007, 483–488
95. Kao B, Lee S D, Cheung D W, Ho W S, Chan K. Clustering uncertain data using voronoi diagrams. In: Proceedings of the 8th International Conference on Data Mining. 2008, 333–342
96. Dehne F, Noltemeier H. Voronoi trees and clustering problems. Information Systems, 1987, 12(2): 171–175
97. Gullo F, Ponti G, Tagarelli A. Clustering uncertain data via K-medoids. In: Proceedings of the International Conference on Scalable Uncertainty Management. 2008, 229–242
98. Cormode G, McGregor A. Approximation algorithms for clustering uncertain data. In: Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. 2008, 191–200
99. Kriegel H P, Pfeifle M. Density-based clustering of uncertain data. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. 2005, 672–677
100. Kriegel H P, Pfeifle M. Hierarchical density-based clustering of uncertain data. In: Proceedings of the 5th IEEE International Conference on Data Mining. 2005, 4

101. Xu H, Li G. Density-based probabilistic clustering of uncertain data. In: Proceedings of the International Conference on Computer Science and Software Engineering. 2008, 474–477
102. Hamdan H, Govaert G. Mixture model clustering of uncertain data. In: Proceedings of the 14th IEEE International Conference on Fuzzy Systems. 2005, 879–884
103. Xiao L, Hung E. An efficient distance calculation method for uncertain objects. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining. 2007, 10–17
104. Bi J, Zhang T. Support vector classification with input data uncertainty. Advances in Neural Information Processing Systems, 2004, 17: 161–169
105. Bhattacharyya C, Pannagadatta K, Smola A J. A second order cone programming formulation for classifying missing data. Advances in Neural Information Processing Systems, 2005, 17: 153–160
106. Yang J, Gunn S. Exploiting uncertain data in support vector classification. In: Proceedings of the International Conference on Knowledge–Based Intelligent Information and Engineering Systems. 2007, 148–155
107. Yang J, Gunn S. Iterative constraints in support vector classification with uncertain information. Constraint-based Mining and Learning, 2007, 49
108. Demichelis F, Magni P, Piergiorgi P, Rubin M A, Bellazzi R. A hierarchical naive bayes model for handling sample heterogeneity in classification problems: an application to tissue microarrays. BMC Bioinformatics, 2006, 7(1): 514
109. Chui C K, Kao B, Hung E. Mining frequent itemsets from uncertain data. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2007, 47–58
110. Chui C K, Kao B. A decremental approach for mining frequent itemsets from uncertain data. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2008, 64–75
111. Leung C S, Carmichael C L, Hao B. Efficient mining of frequent patterns from uncertain data. In: Proceedings of the 7th International Conference on Data Mining Workshops. 2007, 489–494
112. Leung C K S, Brajczuk D A. Efficient mining of frequent itemsets from data streams. In: Proceedings of the British National Conference on Databases. 2008, 2–14
113. Leung C K S, Mateo M A F, Brajczuk D A. A tree-based approach for frequent pattern mining from uncertain data. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2008, 653–661
114. Hewawasam K, Premaratne K, Subasingha S, Shyu M L. Rule mining and classification in imperfect databases. In: Proceedings of the 8th International Conference on Information Fusion. 2005, 661–668
115. Tobji M A B, Yaghlane B B, Mellouli K. A new algorithm for mining frequent itemsets from evidential databases. Proceedings of Information Processing and Management of Uncertainty. 2008, 8: 1535–1542
116. Tobji M A B, Yaghlane B B, Mellouli K. Frequent itemset mining from databases including one evidential attribute. In: Proceedings of the International Conference on Scalable Uncertainty Management. 2008, 19–32
117. Abiteboul S, Kimelfeld B, Sagiv Y, Senellart P. On the expressiveness of probabilistic XML models. The International Journal on Very Large Data Bases, 2009, 18(5): 1041–1064
118. Li T, Shao Q, Chen Y. PEPX: a query-friendly probabilistic XML database. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management. 2006, 848–849
119. Nierman A, Jagadish H. ProTDB: probabilistic data in XML. In: Proceedings of the 28th International Conference on Very Large Data Bases. VLDB Endowment, 2002, 646–657
120. Abiteboul S, Senellart P. Querying and updating probabilistic information in XML. In: Proceedings of the International Conference on Extending Database Technology. 2006, 1059–1068
121. Senellart P, Abiteboul S. On the complexity of managing probabilistic XML data. In: Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. 2007, 283–292
122. Hung E, Getoor L, Subrahmanian V. Probabilistic interval XML. In: Proceedings of International Conference on Database Theory. 2003, 361–377
123. Hung E, Getoor L, Subrahmanian V. PXML: a probabilistic semistructured data model and algebra. In: Proceedings of the 19th International Conference on Data Engineering. 2003, 467–478
124. Abiteboul S, Chan T H H, Kharlamov E, Nutt W, Senellart P. Aggregate queries for discrete and continuous probabilistic XML. In: Proceedings of the 13th International Conference on Database Theory. 2010, 50–61
125. Kimelfeld B, Kosharovsky Y, Sagiv Y. Query efficiency in probabilistic XML models. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. 2008, 701–714
126. Zhao W, Dekhtyar A, Goldsmith J. Databases for interval probabilities. International Journal of Intelligent Systems, 2004, 19(9): 789–815
127. Zhao W, Dekhtyar A, Goldsmith J. A framework for management of semistructured probabilistic data. Journal of Intelligent Information Systems, 2005, 25(3): 293–332
128. Dekhtyar A, Goldsmith J, Hawkes S R. Semistructured probabilistic databases. In: Proceedings of the 13th International Conference on Scientific and Statistical Database Management. 2001, 36–45
129. Hung E. Managing uncertainty and ontologies in databases. UMD Theses and Dissertations, 2005
130. Magnani M, Montesi D. Management of interval probabilistic data. Acta Informatica, 2008, 45(2): 93–130
131. Cohen S, Kimelfeld B, Sagiv Y. Incorporating constraints in probabilistic XML. In: Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. 2008, 109–118
132. Kimelfeld B, Sagiv Y. Matching twigs in probabilistic XML. In: Proceedings of the 33rd International Conference on Very Large Data Bases. VLDB Endowment, 2007, 27–38
133. Adar E, Ré C. Managing uncertainty in social networks. IEEE Data Eng. Bull, 2007, 30(2): 15–22
134. Hintsanen P. The most reliable subgraph problem. In: Proceedings of the European Conference on Principles of Data Mining and Knowl-

edge Discovery. 2007, 471–478

135. Hintsanen P, Toivonen H. Finding reliable subgraphs from large probabilistic graphs. Data Mining and Knowledge Discovery, 2008, 17(1): 3–23
136. Zou Z, Li J, Gao H, Zhang S. Frequent subgraph pattern mining on uncertain graph data. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. 2009, 583–592
137. Zou Z, Li J, Gao H, Zhang S. Mining frequent subgraph patterns from uncertain graphs. Journal of Software, 2009, 20(11): 2965–2976
138. Zou Z, Li J, Gao H, Zhang S. Mining frequent subgraph patterns from uncertain graph data. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(9): 1203–1218
139. Potamias M, Bonchi F, Gionis A, Kollios G. K-nearest neighbors in uncertain graphs. Proceedings of the VLDB Endowment, 2010, 3(1–2): 997–1008
140. Yuan Y, Chen L, Wang G. Efficiently answering probability threshold-based shortest path queries over uncertain graphs. In: Proceedings of the International Conference on Database Systems for Advanced Applications. 2010, 155–170
141. Papapetrou O, Ioannou E, Skoutas D. Efficient discovery of frequent subgraph patterns in uncertain graph databases. In: Proceedings of the 14th International Conference on Extending Database Technology. 2011, 355–366
142. Han M, Zhang W, Li J Z. Raking: an efficient k-maximal frequent pattern mining algorithm on uncertain graph database. Chinese Journal of Computers, 2010, 33(8): 1387–1395
143. Zou Z, Gao H, Li J. Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2010, 633–642
144. Zou Z, Li J, Gao H, Zhang S. Finding top-k maximal cliques in an uncertain graph. In: Proceedings of the 26th International Conference on Data Engineering. 2010, 649–652
145. Yuan Y, Wang G, Wang H, Chen L. Efficient subgraph search over large uncertain graphs. Proceedings of the VLDB Endowment, 2011, 4(11): 876–886
146. Yuan Y, Wang G, Chen L, Wang H. Efficient subgraph similarity search on large probabilistic graph databases. Proceedings of the VLDB Endowment, 2012, 5(9): 800–811
147. Koyutürk M, Grama A, Szpankowski W. An efficient algorithm for detecting frequent subgraphs in biological networks. Bioinformatics. 2004, 20(Suppl 1): 200–207
148. Valiant L G. The complexity of enumeration and reliability problems. SIAM Journal on Computing, 1979, 8(3): 410–421
149. Jin C, Yi K, Chen L, Yu J X, Lin X. Sliding-window top-k queries on uncertain streams. Proceedings of the VLDB Endowment, 2008, 1(1): 301–312
150. Ré C, Letchner J, Balazinksa M, Suciu D. Event queries on correlated probabilistic streams. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. 2008, 715–728
151. Alon N, Matias Y, Szegedy M. The space complexity of approximating the frequency moments. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing. 1996, 20–29
152. Flajolet P, Martin G N. Probabilistic counting algorithms for data base applications. Journal of Computer and System Sciences, 1985, 31(2): 182–209
153. Zhang T, Ramakrishnan R, Livny M. Birch: an efficient data clustering method for very large databases. ACM Sigmod Record, 1996, 25(2): 103–114
154. Aggarwal C C, Han J, Wang J, Yu P S. A framework for clustering evolving data streams. In: Proceedings of the 29th International Conference on Very Large Data Bases. VLDB Endowment, 2003, 81–92
155. Aggarwal C C, Yu P S. A framework for clustering uncertain data streams. In: Proceedings of the 24th International Conference on Data Engineering. 2008, 150–159
156. Li Z, Ge T. Online windowed subsequence matching over probabilistic sequences. In: Proceedings of the International Conference on Management of Data. 2012, 277–288
157. Lian X, Chen L. Efficient join processing on uncertain data streams. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. 2009, 857–866
158. Ge T, Liu F. Accuracy-aware uncertain stream databases. In: Proceedings of the 28th International Conference on Data Engineering. 2012, 174–185
159. Peng L, Diao Y, Liu A. Optimizing probabilistic query processing on continuous uncertain data. Proceedings of the VLDB Endowment, 2011, 4(11): 1169–1180
160. Jayram T, McGregor A, Muthukrishnan S, Vee E. Estimating statistical aggregates on probabilistic data streams. In: Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. 2007, 243–252
161. Zhang Q, Li F, Yi K. Finding frequent items in probabilistic data. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. 2008, 819–832
162. Aggarwal C C, Han J, Wang J, Philip S Y. On high dimensional projected clustering of data streams. Data Mining and Knowledge Discovery, 2005, 10(3): 251–273
163. Zhang C, Gao M, Zhou A. Tracking high quality clusters over uncertain data streams. In: Proceedings of the 25th International Conference on Data Engineering. 2009, 1641–1648
164. Zhang W, Lin X, Zhang Y, Wang W, Zhu G, Xu Yu J. Probabilistic skyline operator over sliding windows. Information Systems, 2013, 38(8): 1212–1233
165. Subramaniam S, Palpanas T, Papadopoulos D, Kalogeraki V, Gunopulos D. Online outlier detection in sensor data using non-parametric models. In: Proceedings of the 32nd International Conference on Very Large Data Bases. VLDB Endowment, 2006, 187–198
166. Deshpande A, Guestrin C, Madden S R, Hellerstein J M, Hong W. Model-driven data acquisition in sensor networks. In: Proceedings of the 30th International Conference on Very Large Data Bases. VLDB Endowment, 2004, 588–599
167. Hida Y, Huang P, Nishtala R. Aggregation query under uncertainty in sensor networks. Technical Report, 2004
168. Welbourne E, Khoussainova N, Letchner J, Li Y, Balazinska M, Bor-

riello G, Suciu D. Cascadia: a system for specifying, detecting, and managing rfid events. In: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services. 2008, 281–294

169. Kanagal B, Deshpande A. Online filtering, smoothing and probabilistic modeling of streaming data. In: Proceedings of the 24th IEEE International Conference on Data Engineering. 2008, 1160–1169
170. Zhang C J, Chen L, Tong Y, Liu Z. Cleaning uncertain data with a noisy crowd. In: Proceedings of the 31st IEEE International Conference on Data Engineering. 2015, 6–17
171. Mo L, Cheng R, Li X, Cheung D W, Yang X S. Cleaning uncertain data for top-k queries. In: Proceedings of the 29th IEEE International Conference on Data Engineering. 2013, 134–145
172. Panse F, Van Keulen M, De Keijzer A, Ritter N. Duplicate detection in probabilistic data. In: Proceedings of the 26th International Conference on Data Engineering Workshops. 2010, 179–182
173. Van Keulen M, De Keijzer A. Qualitative effects of knowledge rules and user feedback in probabilistic data integration. Proceedings of the VLDB Endowment, 2009, 18(5): 1191–1217
174. Cheng R, Chen J, Xie X. Cleaning uncertain data with quality guarantees. Proceedings of the VLDB Endowment, 2008, 1(1): 722–735
175. Dong X L, Halevy A, Yu C. Data integration with uncertainty. Proceedings of the VLDB Endowment, 2009, 18(2): 469–500

Lingli Li is an associate professor at Heilongjiang University, China. She obtained her PhD degree from Harbin Institute of Technology in 2015. Her research interests include data management, data quality, entity resolution. She has published more than 10 papers in refereed journals and conferences such as IEEE Trans. of Knowledge and Data Engineering.

Hongzhi Wang is a professor and doctoral supervisor at Harbin Institute of Technology, China. His research area is data management, including data quality, XML data management, and graph management. He has published more than 100 papers in refereed journals and conferences. He is a recipient of the outstanding dissertation award of CCF, Microsoft Fellow, and IBM PhD Fellowship.

Jianzhong Li is a professor and doctoral supervisor at Harbin Institute of Technology, China. He is a senior member of CCF. His research interests include database, parallel computing, and wireless sensor networks, etc.

Hong Gao is a professor and doctoral supervisor at Harbin Institute of Technology, China. She is a senior member of CCF. Her research interests include data management, wireless sensor networks, and graph database, etc.