Computer and Information Science; Vol. 8, No. 3; 2015
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Handling Uncertainty in Database: An Introduction and Brief Survey

Ahmed Sharaf Eldin [1, 2], Doaa Saad EZanfaly [1, 3] & Nermin Abdelhakim Othman [1]

[1] Faculty of Computers & Information, Helwan Univ., Cairo, Egypt

[2] Faculty of Information Technology and Computer Science, Sinai Univ., Sinai, Egypt

[3] Faculty of Informatics & Computer Science, British University in Egypt, Cairo, Egypt

Correspondence: Nermin Abdelhakim Othman, Faculty of Computers & Information, Helwan Univ., Cairo, Egypt. E-mail: eng.nermin@gmail.com

Received: June 15, 2014 Accepted: July 21, 2014 Online Published: July 31, 2015

doi:10.5539/cis.v8n3p119 URL: http://dx.doi.org/10.5539/cis.v8n3p119

Abstract

In the last years, uncertainty management became an important aspect as the presence of uncertain data increased rapidly. Due to the several advanced technologies that have been developed to record large quantity of data continuously, resulting is a data that contain errors or may be partially complete. Instead of dealing with data uncertainty by removing it, we must deal with it as a source of information. To deal with this data, database management system should have special features to handle uncertain data. The aim of this paper is twofold: on one hand, to introduce some main concepts of uncertainty in database by focusing on different data management issues in uncertain databases such as join and query processing, database integration, indexing uncertain data, security and information leakage and representation formalisms. On the other hand, to provide a survey of the current database management systems dealing with uncertain data, presenting their features and comparing them.

Keywords: data uncertainty, database management systems, imperfect data, probabilistic data

1. Introduction

We live in an uncertain world, surrounded by data which is completely uncertain. Uncertainty exists when knowledge of the real world cannot be indicated with absolute confidence. Uncertainty might result from using unreliable information source, as faulty reading instrument or input forms that have been filled out incorrectly, or it can be a result of system errors, such as transmission errors and noise, delay in processing update transactions, and imperfections of software. Therefore uncertainty is considered an unavoidable result of information gathering methods that requires estimation or judgment. An increasingly large amount of uncertain data can be found in a variety of domains such as: data integration, experimental data, information extracted automatically from text, data from the physical world (Aggrawal, 2009; Rowe, 1994). Since most conventional databases, including relational databases are deterministic, and the ignorance of uncertainty is not an option in reliable applications, dealing with uncertain data becomes a must.

The broader taxonomy of the current research in managing uncertain data can be classified into two main areas: The first is modeling uncertain data. The main challenge in this area is handling the data while keeping it useful for data management or mining applications. The second area is managing and mining uncertain data where traditional data management techniques are adopted to deal with uncertain data, such as join processing, query processing, indexing, and data integration (Aggrwal, 2009). To apply traditional data mining techniques, uncertain data has to be summarized into atomic values (Dhandore & Ragha, 2014).

The aim of this paper is twofold: the first is an introduction to the main concepts of handling uncertainty in database. The second is surveying different data management issues in uncertain databases such as join and query processing, database integration, indexing uncertain data, security and information leakage and representation formalisms. As a matter of fact most of these issues; were discussed and surveyed by Aggrawal et al in (Aggrwal, 2009). In this paper we added two other issues which are: security and information leakage and representation formalisms and updated the others issues. To complete the study, we surveyed current uncertain database management systems such as Orion (Cheng, Singh, & Prabhakar, 2005), MystiQ (Boulos, Dalvi, Mandhani, Mathur, Chris, & Suciu, 2005), Trio (Benjelloun, Sarma, Halevy, & Widom, 2006; Widom, 2005), MayBMS (Huang, Antova, Koch, & Olteanu, 2009), MCDB (Jampani, Xu, Wu, Perez, Jermaine, & Haas, 2008), Bayes-Store (Wang, Michelakis, Garofalakis, & Hellerstein, 2008) and compared them according to the

operations handled.

This paper is organized as follows: Section two briefly describes some key concept in uncertainty in databases. Section three surveys various techniques of dealing with different data management issues on uncertain data. Section four discusses the differences between existing uncertain DBMSs. Section five contains the conclusion and summary.

2. Uncertainty in Database

Although uncertainty, vagueness, ambiguity, imprecision, and inconsistency are five terms that are sometimes used interchangeably, each term has its own meaning. In the database context, uncertainty refers to the data objects that cannot be assured with an obsolete confidence (Motro, 1994). Vagueness refers to a data item that belongs to a range of values without a clear determination of its exact value. For example when saying that a fish tail is long without specifying its exact length. Ambiguity means the incomplete description of a data item. For example, it may not be specified whether the fish tail is measured in cm or mm. Imprecision means not precise and it refers to the level of exactness. For example, the fish color is red or orange. Finally, Inconsistency happens when having conflicting items. For example, the fish tail is greater than 10 cm and the fish tail is greater than or equal 12 cm.

The common sources of uncertainty are unreliable information source such as faulty reading instrument or input forms filled incorrectly, and system errors that includes transmission noise, imperfection of system software and delay in processing update transaction (Motro, 1994).

In uncertain database systems, uncertainty is handled in two main dimensions, the uncertainty of data and the uncertainty of operations. Uncertainty of data has two levels: the first level is the attribute level where tuples exist for certain in the database but the attribute value is however uncertain. The second level is the tuple uncertainty where all attribute in the tuple are known precisely but the existence of the tuple itself in a relation is uncertain.

The degree of uncertainty differs according to the information form and the number of alternatives when uncertain data exists. The highest degree of uncertainty is found when there is a doubt about the existence of true value in the existing data, and it decreases when there is a range of values for an uncertain object. The uncertainty degree decreases when the uncertain value comes from a few set of alternatives. Uncertainty degree is further decreased when there is a probability attached with each alternative indicating their correctness (Motro, 1994; Motro, 1995).

The uncertainty of operations includes transformation and modification. Transformation is defined as the operation that gets new data from the stored one. Queries are considered the frequent transformation type used. Uncertain request from the user can occur due to several reasons; such as lacking the information already present, being not sure about what they need. After the answer is delivered to the user, the uncertainty level decreases when the user is more familiar with the answer he has got (Motro, 1994; Motro, 1995). Modification operation includes any operation that cause change in the data already present. The user is the one who defines the modification needed. The uncertainty here can be caused from several reasons such as; lacking system information, lacking database information or the uncertainty can be in the data to be modified. Few tools are present for solving the uncertainty in the modification process (Motro, 1994; Motro, 1995).

Processing uncertainty main reasons is the uncertainty about the tools used by the system in processing the request. So in case the description and transformation process are free of uncertainty still we need to check the processing uncertainty (Motro, 1994; Motro, 1995).

Finally, a probabilistic database is an uncertain database in which the possible worlds have associated probabilities. Each data item, tuple and value that an attribute can take is associated with a probability $\in (0, 1)$, with 0 representing that the data is certainly incorrect, and 1 representing that it is certainly correct (Motro, 1995). There is also the research area of fuzziness in database systems, which has resulted in a number of models aimed at the representation of imperfect information in DB. Fuzzy relational database is an extension of the relational database in order to treat, store, and interrogate imprecise data. This extension introduce fuzzy predicates under shapes of linguistic expressions that, at the time of flexible querying, permits to have a range of answers in order to offer the user all intermediate variations between completely satisfactory answers and those completely dissatisfactory (Touzi & Hassine, 2009).

3. Handling Uncertainty in Databases

The survey done by Aggrawal et al. in (Aggrwal, 2009) is considered to be a corner stone for researchers in the area of managing uncertain data. Hence, we took it as a starting point for this paper. In this section, we discuss a

number of data management applications on uncertain data. Aggrawal et al. in (Aggrwal, 2009) have included most of the applications and techniques that are handling the management issues on uncertain data such as join processing, query processing, data integration, and indexing. We have added in this paper other recent techniques that handle the same issues. Moreover, we have included two other important management issues to our paper which are security and information leakage, and Representation Formalisms. This survey covers almost all the management issues on uncertain data and shows how techniques can handle it.

3.1 Indexing

Indexing uncertain data is the key technique for efficient query evaluation over uncertain data. The problem of indexing uncertain data is challenging because the diffuse probabilistic nature of the data can reduce the effectiveness of index structure and makes the cost of queries execution a concern. Index structures for deterministic data are not appropriate for uncertain data determining the suitable index structure for uncertain data depends on two main factors: the nature of uncertainty in data that depends mainly on the application domain and the type of required queries (Aggarwal, 2009).

In index structures and their associated algorithms are developed to effectively answer Probabilistic Threshold Queries (PTQs). The Index scheme called probability threshold indexing (*PTI*), is based on the idea of augmenting uncertain information to an R-tree. The one-dimensional intervals are mapped to a two-dimensional space to show that the problem of interval indexing with probabilities is significantly harder than interval indexing (indexing on interval queries which is a complex query). A technique called variance-based clustering is used to overcome the limitation of this problem. The index structure can answer the queries for various kinds of uncertain information, in an almost optimal sense.

The problem of range searching was introduced by (Tao, Cheng & Xiao, 2007), solved by considering a small histogram consisting of one piece. In (Tao, Cheng, Xiao, Ngai, Kao & Prabhakar , 2005; Tao, Cheng & Xiao, 2007) this problem is considered in two higher dimensions and presented some index structures based on space partitioning heuristics. In indexing categorical uncertain data is handled, using a heuristic solution, namely, each random object take a value from a discrete, unordered domain.

(Agarwal, Cheng, Tao, & Yi, 2009)Presents linear or near linear size indexing schemes for both the fixed and variable threshold version of the problem, with logarithmic or poly-logarithmic query times. An optimal index is presented for answering queries on uncertain data where the probability threshold is fixed. In (Qi, Singh, Shah, & Prabhakar, 2008) the Probabilistic Nearest Neighbor (PPN) query is studied with probability threshold (PPNT) which returns all uncertain objects with NN probability greater than the threshold. An augmented R-tree index is proposed with additional probabilistic information to facilitate pruning as well as global data structures for maintaining the current pruning status.

The indexing algorithms proposed in (Singh, Mayfield, Prabhakar, Shah, & Hambrusch, 2007; Qi, Singh, Shah, & Prabhakar, 2008), are not considered a general indexing algorithm. As in (Singh, Mayfield, Prabhakar, Shah, & Hambrusch, 2007) only categorical uncertain data is considered. In (Qi, Singh, Shah, & Prabhakar, 2008) they only focused on indexing the nearest neighbor queries. The indexing algorithm used in (Tao, Cheng, & Xiao, 2007) is the most effective way to solve index challenge when dealing with probabilistic queries, as it can provide correct query answers for different uncertain data.

3.2 Security and Information Leakage

When dealing with the problem of security and information leakage challenge, solving this problem is based on appropriate data modeling usage. For better understanding of the models used, considered the two main security properties in Table 1, the quantitative and qualitative security properties (Ngo & Huisman, 2013). The quantitative security property is based on Shannon Entropy H(X) to measure the information content of a random variable. Where Information Leaked = initial uncertainty - remaining uncertainty (McCamant & Ernst, 2008). Shannon entropy proves superior to guessing entropy that only guarantees the non-negative property of leakage for deterministic programs. (Ngo & Huisman, 2013) Propose a novel model of quantitative analysis for multi-threaded program that also take into account the effect of observables in intermediates states along the trace.

Mainly a probabilistic data model has been used to deal with the information leakage in views and in data exchanges. Usually in this case the data is private and only a certain view is published by the owner. This view usually has shortage of private information in its data. Many approaches have been working on this problem. From these approaches is modeling the attacker's background knowledge as a probability space, to be able to check whether the posteriori probability of the secret is in fact different from the priori probability: if the two are

the same then it's a perfect security case, otherwise practical security is satisfied if they are only close to each other (Re & Suciu, 2007). This process appears to be extremely difficult when the input probabilities are not known.

Designing security policies in the case of data uncertainty represent another big challenge. Nowadays a common practice is to define access control rules by specify them in terms of certain credentials offered by a user (Re & Suciu, 2007). The process of defining the right semantics for such access control policies when credential is probabilistic is still an open problem for researchers till now. In (Chothia, Kawamoto, Novakovic, & Parker, 2013) they develop an information leakage model that can measure the leakage between arbitrary points in a probabilistic program. There model does not detect information leakage that occur between variables that have not been annotated. They believe that detecting leakage at selected points is more practical than one that attempt to detect all possible leaks. They base their framework on a simple probabilistic, imperative language that they call CH-IMP.

Table 1. Security Property

	Qualitative Security Property	Quantitative Security Property
Features	Used for applications where private data need strict protection. Based on the idea; that private data shouldn't have access from public data. It prevents any flow from high security level to low level.	Used in application where we want to get information that depends on private data. Determine how much secrete information is leaked. This leaked amount is expressed in quantitative term. Offer a method to compute bounds on how much information is leaked.
Applications	Internet backing, e-commerce, and medical information systems.	Password checker program.
Drawback	Reject s any program that contain leakage, even if this leakage is unavoidable.	Not suitable for some application, as for the multi-threaded program.

3.3 Query Processing

The existence of data uncertainty in many real-worlds made the importance of the uncertain query processing increases. The incorporation of probabilistic information affects the correctness and computability of the query plan. Having a query over an uncertain database requires computation or aggregation over a large number of possibilities. The answer to a standard SQL query over probabilistic database is a set of probabilistic tuple, each tuple returned by the system has a probability of being in the query's answer set. Computing these probabilities is difficult and is an open research area. To process large scale probabilistic data we need to develop specific probabilistic inference techniques that can be integrated well with SQL query processors and optimizers, and that scale to large volumes of data.

There are two broad semantic approaches used: Intentional semantics approach that is based on modeling the uncertain database in event models or possible worlds then use tree-like structures of inferences on these event combinations. Using the tree-like structure make it possible to get all the possibilities enumerated over which the query may be evaluated and subsequently aggregated. This semantics results are complex in term of the evaluation time which represent its drawback, but usually lead to correct results (Dalvi & Suci , 2007). The other approach is the Extensional semantics approach, instead of performing the whole enumeration process to the tree of inferences, this semantics attempts to design a plan which can approximate these queries. When dealing with simple expression, extensional semantics will be the best choice. But it's not preferred when dealing with complex expression as the dependencies in the underlying query results cannot be evaluated easily that why dealing with complex expression appears to be this semantic drawback (Dalvi & Suciu, 2007).

Query evaluation is one of the important factors that should be taken into consideration when dealing with queries. This issue became more complicated in the case of uncertain data or probabilistic data. One of the techniques for adding probabilistic information into query evaluation is a generalization of the standard relational model which was discussed in (Fuhr & Rolleke, 1997). The probabilistic relations are treated as generalizations of deterministic relations. Then a modification is made to the operators of relational algebra in order to take the tuple weights into account during query processing. In (Dalvi & Suciu, 2007) the presence of a correct extensional plan was the focus. But for queries which don't admit correct extensional plan, two techniques are proposed to construct results which yield approximately correct answers. A fast heuristic is

designed which can avoid large errors, and a sampling-based Monte-Carlo algorithm is designed which is more expensive, but can guarantee arbitrarily small errors. In (Dalvi & Suciu, 2007) a solution to the case of uncertain predicates on deterministic data is extended. We note that the work in this technique assumes tuple independence which is often not the case for a probabilistic database. In (Dalvi & Suciu, 2005) the data statistics and explicit probabilities at the data sources are used. Probabilistic database with complex tuples correlation is used to deal with the imprecision.

Tuple correlation is also one of the important issues that should be taken into consideration in query processing on uncertain data. As it's the case in most of recent applications. Such as sensor data which is highly correlated in space and time (Deshpande, Guestrin, Madden, Hellerstein, & Hong, 2004). Even in the cases that assume that tuples are independent; many intermediate query results may contain correlations. Statistical modeling technique where used in (Sen & Deshpande, 2007) on querying correlated tuples. Then this method built a framework which presents uncertainties and correlations through the use of joint probability distribution.

Ranked queries are very useful in decision making applications, and data mining tasks (Dhandore & Ragha, 2014). In particular, in database D a ranking query retrieves k objects in the database that have the highest scores. Ranked queries on uncertain database were discussed in (Lian & Chen , 2008) introducing two effective pruning methods, spatial and probabilistic, to help reduce the ranked query search space. Inverse ranking query was proposed in (Lian & Chen, 2011) by introducing a query named the *probabilistic inverse ranking queries* (PIR) which retrieves the possible rank of a given query object in an uncertain database with confidence above the probability threshold and they also include effective pruning methods to reduce the search space. In additional to that a study of three interesting aggregate *PIR* queries which are (max, top-m, avg.), was made but unfortunately they did not cover wider scale of aggregates.

The use of possible worlds semantics present another challenge as it allows complex correlations among tuples in the database. In (Soliman, Ilyas, & Chang, 2007) the generalization rules are used to deal with this issue, which are logical formulas that determine valid worlds. The interaction between both the possible world's semantics and top-k queries need careful redefinition of the query semantics.

The work in (Re, Dalvi, & Suciu, 2007), (Yi, Li, Kollios, & Srivastava, 2008), (Hua, Pei, Zhang, & Lin, 2008) studied the top-k queries in the probabilistic databases. In (Re, Dalvi, & Suciu, 2007) the main focus was on reducing the difficulty of getting the k uncertain objects that satisfy the query predicates in all possible worlds with the highest probabilities.AVG aggregate function is not supported in (Re, Dalvi, & Suciu, 2007) .The U-Top*k* query was proposed in (Yi, Li, Kollios, & Srivastava, 2008) which get set of k uncertain objects such that this set is also the top-k answer set appearing in some possible worlds with the highest probability, and the U-*k*Ranks, which finds k objects such that the *i*-th object ($1 \leq i \leq k$) has the *i*-th highest rank in some possible worlds with the highest probability. (Yi, Li, Kollios, & Srivastava, 2008) Improved the U-Top*k* and U-*k*Ranks queries efficiency by including early stopping conditions.

A *probabilistic threshold top-k* (PT-*k*) query was proposed in (Hua, Pei, Zhang, & Lin, 2008) which gets the k objects so that there is a top-k query answer in some possible worlds with the highest probabilities. In (Peng, Diao, & Liu, 2011) the threshold query processing for uncertain data was optimized. Cormode et al. (Cormode, Li, & Yi, 2009) used the expected ranks as a way to rank objects in a probabilistic database. In (Lian & Chen, 2009) the probabilistic top-k dominating (PTD) query was discussed which was then improved in (Lian & Chen, 2013). In (Lian & Chen, 2011) the probabilistic top-k star (PT*k*S) query was proposed, which gets k objects in an uncertain database that are near to a static/ dynamic query point, taking both distance and probability aspects into consideration.

In addition to ranked queries, other types of uncertain database queries are handled such as nearest neighbor query (Kriegel, Kunath, & Renz, 2007), group nearest neighbor (Lian & Chen, 2008), reverse nearest neighbor (Lian & Chen, 2009), Top-k nearest neighbor (Dallachiesa, 2014) and range query (Li, 2014).

3.4 Representation Formalisms

In most cases the probabilistic database is a probability space over all possible instances of the database, called possible worlds. We cannot numerate these possible instances; instead a concise representation formalism that describes all possible worlds and their probabilities is needed. The most common technique used, use conditional independence between variables and represent a probability space in term of a graphical model (Pearl, 1988). For an efficient query evaluation a trade-off is required between the succinctness of representation formalism and the complexity of evaluating interesting queries (Antova, Jansen, Koch, & Olteanu, 2008).

In the case of a probabilistic database, the lineage as well needs to be represented to know the reason of

uncertainty. The trio project has discussed the problem of representing both the uncertainty and the lineage in (Benjelloun, Sarma, Halevy, & Widom, 2006). Lineage is usually expressed in some form of boolean expressions (Afrati & Vasilakopoulos, 2010).

In (Parsons & Saffiotti, 1993) a method that enables systems that use different uncertainty handling formalisms to qualitatively integrate their uncertain information, and argues that this makes it possible for distributed intelligent systems to achieve tasks that would otherwise be beyond them. This paper approach is grounded on the notion of degrading, given a representation of uncertainty, they degrade its information content to a level that can be shared between all the different formalism; this degraded information is then communicated between agents.

3.5 Join Processing

This section aims at surveying the currently followed research directions concerning joins on uncertain data. It presents the most prominent representatives of the join categories. The problem of join processing is challenging in the context of uncertain data, because the join-attribute is probabilistic in nature. The approaches mainly differ in the representation of the uncertain data, the distance measure or other type of object comparison, the types of queries and query predicates and the representation of the result. Join methods can be classified into:

3.5.1 Confidence-Based Join Methods

Most confidence-based join methods depend on reducing the search space based on the confidence values of the input data. For the candidate selection neither the join-relevant attribute of the object nor the join predicates are taken into account. (Agrawal & Widom, 2007) Propose efficient confidence- based join approach for all query types (as the stored, stored-threshold). Assume that the stored relations provide efficient sorted access by confidence and that neither joins relation fit into main memory. Assume also the uncertainty of the objects and their independence. This approach can be applied regardless of the join-predicate and the type of score function. The probabilistic top-k join queries are also handled. The same way is used to handle the sorted and sorted-threshold join queries.

3.5.2 Probabilistic Similarity Join Methods

A recognized short come in the confidence based join methods is that the knowledge about the relevant attributes for the join predicate was not incorporated. The previous confidence-based join methods return the pairs of objects regardless of their distance, as long as their combined confidence is sufficient. Similarity join are very selective queries, where only very few candidates satisfy the query predicate. That is why an effective pruning technique is needed for an efficient similarity join processing. Similarity join applications benefit from pruning those candidate whose attributes do not likely satisfy the join predicate. This way guarantees that the candidates having a very low probability are avoided.

In (Cheng, Singh, Prabhakar, Shah, Vitter, & Xia, 2006) the similarity joins over uncertain data are studied based on the continuous uncertainty model. An uncertainty interval is accomplished for each uncertain object attribute by an assigned uncertainty probability distribution function (pdf). For two uncertain objects, each represented by a continuous pdf, their score in turn lead to a continuous pdf representing the similarity probability distribution of both objects. The probabilistic similarity join consists of an uncertainty interval and an uncertainty pdf. The probabilistic join queries are defined through the probabilistic predicate defined on the uncertain pairs. Also two join queries are proposed, the *probabilistic join query (PJQ)*, and the probabilistic *Threshold Join Query (PTJQ)*.

3.5.3 Probabilistic Spatial Join Methods

Spatial joins are applied on spatial objects, which are objects that have a certain position is space and a spatial extension. This spatial joins depend mainly on spatial predicates that refers to spatial topological predicates. The probabilistic spatial join is evaluated in two steps: filtering and refinement. In (Ni, Ravishankar, & Bhanu, 2003) evaluating probabilistic spatial joins was the focus, dealing with the object pairs, and the intersection probability between them. The probabilistic R-tree (PrR-tree) index was proposed, which supports a probabilistic filter step. An efficient algorithm was proposed to obtain the intersection probability between two candidate polygons for the refinement step.

In (Burdick, Deshpande, Jayram, Ramakrishnan, & Vaithyanthan, 2005) a probabilistic spatial join approach was proposed based on uncertainty model. Where the uncertain spatial objects are composed of primitive volume elements with confidence values assigned to each of them. Then the score function is used to evaluate the join predicate for each pair. Based on this score function, a Probabilistic Threshold Join Query (PTJQ) and a Probabilistic Top-k Join Query (PTopkJQ) were proposed. (Ljosa & Singh, 2008) Presents algorithm for two kinds of probabilistic spatial join queries, the first is the (PSJ) threshold PSJ queries, which return all pairs that

score above a given threshold. The second kind is called top-k PSJ queries, which return the k top-scoring pairs. This algorithm mainly focuses on speeding up the queries.

3.6 Data Integration

Data integration is an important application in the context of uncertain data. It is the general process of providing single information source out of some local information sources. The term data integration is often used to refer to information integration applied to structure data (both schema and instances). From the basic data integration tasks is a comparing local data source to identify *matching entities*, e.g., two columns with telephone and home-telephone from two company databases both containing customer telephone. The information about matching entities, usually called *mapping*, is then used to merge the input data sources by including, for example, all of the customers' telephone into a single column. Unfortunately, automated tools may still fail in identifying all the correct mappings, e.g., because of variations in columns.

Data integration systems need to handle uncertainty at three levels (Aggarwal, 2009):

- Uncertain mediated schema: mediated schema is defined as a set of schema terms in which queried are posed. Mediated schema doesn't include all the attributes present in the sources, but rather the aspects of the domain that the application builder wishes to expose to the users. There are several reasons for uncertainty arising in the schema mapping. First is the mediated schema is known directly from sources that will cause uncertainty in the results. Second, when domains get broad, there will be some uncertainty about how to model the domain.
- Uncertain schema mapping: data integration systems depend on schema mapping to define the semantic relationships between the data in the source and terms used in the mediated schema. Taking into consideration that schema mapping can be inaccurate. In practice, schema mapping are often generated by semi-automatic tools and not necessarily verified by domain experts.
- Uncertain data: reasons for uncertain data are various, such as the extraction from unstructured or semi-structured sources by automatic methods. Other reason is that data may come from sources that are unreliable or not up to date.
- Uncertain queries: In some cases queries are given as keywords rather than structured query over well defined schema. In this case the system need to transform this query into structured one with respect to the data sources.

One of the ways to handle the integration is to explicitly represent the uncertainty produced by the data integration system and to consider it an important result of the integration process. In a survey made on 2003 about data integration, the problem of uncertain data management was not mentioned; it was stated that the main difficulty was the discovery of correct semantic relationships between schema objects (Halevy, 2003). After that, the problem of dealing with *imprecise mappings* was mentioned in another survey paper (Doan & Halevy, 2005). However, it was noticed that we will never be able to find all correct matches and that we should therefore be aware of possible errors and find ways to use partially incorrect results.

As for uncertainty management within the data integration process. Uncertain data integration goal is using the uncertainty available in the data sources and/or generated during the matching phase, to create an uncertain integrated view of the data. There are several methods to represent uncertainty. One of these ways is by using the quantitative methods, e.g., specifying the probability that a mapping is correct, or qualitative methods, e.g., using fuzzy sets and possibility theory to represent preferences about the correctness of a mapping. Quantitative models are the most frequently used in recent data integration methods. Qualitative approach is used to reduce the complexity of the manipulation of uncertainty.

In particular, as many mediated databases consistent with the sources are possible, there can also be many alternative query answers. Thus they define two categories (correct answer and strongest correct answer) to characterize good and best query answers. An answer is good if it is contained in the answers of all mediated databases consistent with the sources. Many approached worked on reducing the number of mappings thus increasing the efficiency of the process.

In (Nottelmann & Straccia, 2007) several methods were used like the ad hoc threshold, top-k to remove some discovered rules. In (Gal, 2006) it's showed that the analysis of the top-k mappings can be used as a selection criterion (keeping the relationships that are more stable in high-likelihood mappings). Both (Nottelmann & Straccia, 2005) and (Keulen, Keijzer, & Alink, 2005) remove some possibilities using thresholds and constraints; however checking these constraints may become an additional source of complexity. In (Keijzer & Keulen, 2007) the authors suggest user feedback can be used to reduce the number of possible worlds. In (Sarma, Dong, &

Halevy, 2008) some uncertainty was removed by categorizing all the mapping with a probability greater than a predefined threshold as certain and those with probability less than the predefined threshold as wrong.

(Keijzer & Keulen, 2008) Use consistency rules that make part of the possible worlds to be removed. As this way reduces the number of possible worlds, the number of all alternative mappings is exponential on the number of pairs of schema objects; therefore even reducing it by a fixed percentage may not scale to real-world integration a task which is considered its drawback.

4. Uncertain Database Management Systems

During the past years, different releases of DBMSs for dealing with uncertain data have been emerged. In this section, we review most of these systems by highlighting their strategy, strength, and weakness.

4.1 MayBMS

MayBMS is a probabilistic Database Management system developed by Oxford and Cornell universities. The MayBMS system is considered to be a complete probabilistic database management system that leverages robust relational database technology. It was developed in 2005 as an extension of the open-source PostgreSQL server backend and has undergone several transformations. Its backend is easily accessible through multiple APIs (inherited from PostgreSQL), and has efficient internal operators for processing probabilistic data (Huang, Antova, Koch, & Olteanu, 2009).

MayBMS *main features are* (Huang, Antova, Koch, & Olteanu, 2009):

- A powerful query language for processing and transforming uncertain data
- Space-efficient representation and storage
- Efficient query evaluation based on mature relational technology
- Support for conditioning and data cleaning

MayBMS is known with its U-relational database where it stores its probabilistic data. Queries are represented in an extension of SQL with specialized constructs for probability computation and what-if analysis. The U-relations in the U-relational database are standard relations extended with condition and probability columns to encode correlations between the uncertain values and probability distribution for the set of possible worlds. Where the variables from finite set of independent random variables are stored in the conditional columns and the probabilities of the variables assignments occurring in the same tuple are stored in the probability columns. TheMayBMS query language extends SQL with uncertainty-aware constructs.

Extensions of relational algebra or SQL with limited constructs, such as certain or top-k, are not expressive enough. It is not allow for the convenient construction of new worlds or for the use of data correlations across worlds. MayBMS does not support several aspects such as the lineage; standard SQL aggregates such as sum or count on the uncertain relations only support expectation of the aggregation which is considered as its drawback.

4.2 Trio

Trio is developed at Stanford University in 2010 (Agrawal, Benjelloun, Das, Hayworth, Nabar, Sugihara, & Widom, 2006) for managing uncertain data and data lineage using an extended relational model and a SQL-based query language. Through this project, a new schema named ULDBs is introduced. The ULDBs adds uncertainty and lineage of the data as first-class concepts. In addition, a SQL-based query language for ULDBs called TriQL is developed where the semantics of the SQL are modified to take uncertainty and lineage into account, and some new constructs are added to query uncertainty and lineage directly. The first working prototype of Trio model and language was built on top of a conventional DBMS (Agrawal, Benjelloun, Das, Hayworth, Nabar, Sugihara, & Widom, 2006).

Trio data model semantics is based on the possible worlds which is a set of possible instances for the database. In a discrete uncertainty the uncertain database represent a finite set of possible instances with continuous uncertainty. The uncertain attribute value may be an arbitrary probability distribution function (pdf) over a continuous domain, describing the possible values for an attribute. In Trio they the semantic of standard query is defined naturally. When dealing with queries in the trio the query result on uncertain database must include the result of applying the query Q to each possible instance of U (Agrawal & Widom, 2009).

The Trio includes the lineage in query processing to define the data from which each result value was derived. Lineage is needed to properly represent uncertainty, and to compute result confidence values lazily. The lineage is generated at query time and for the results that involve pdfs, the lineage is extended to include relevant predicates and mappings. The Trio deals with expensive queries by using approximate answers, either using

sample function, or a histogram based on the weight function. When it come to integration, Trio data model include a confidence value for each tuple that represent the probability of the tuple existence. This confidence feature is very useful for pdf integration.

Trio main features are (Widom, 2005):

- Data values are uncertain, approximate, or incomplete. A record may include confidence that it actually belong in the database.
- Queries operate over uncertain data, may return uncertain results.
- Lineage is an internal part of the data model.
- Lineage and accuracy may be queried.
- Lineage can be used to enhance the data modifications.

Trio database management system is considered the most powerful database management system on uncertain data, which plenty of researches building their techniques based on its model.

4.3 MystiQ

MystiQ is a probabilistic database system developed at University of Washington. It uses a probabilistic data model to find answers in large numbers of data sources exhibiting various kinds of imprecision (Boulos, Dalvi, Mandhani, Mathur, Chris, & Suciu, 2005).

MystiQ *main features are* (Boulos, Dalvi, Mandhani, Mathur, Chris, & Suciu, 2005):

- Support for complex SQL queries with approximate match predicates, and their ranked result.
- Ability to return best matches when no tuples satisfies all the predicates.
- Support complex SQL queries over inconsistent data, global constraint definition, and the definition of a soft view in queries.

What makes MystiQ different from any other system is that it provides probabilistic semantics that makes it a middleware where data is normally stored in a relational database system. Being a middleware enable it to escalate the infrastructure of an existing database engine ex: query evaluation, query optimization, and indexes.

MystiQ focuses on efficient processing of SQL queries. It combines two query evaluation techniques:

First pushes the computation of the output probability in the DB engine using a technique called “safe plans”. Second runs a Monte Carlo simulation in the middle ware guiding the simulation steps to quickly identify and rank the top-k most probable answers. MystiQ can do the select-from –where-group by queries over large probabilistic databases. MystiQ allows users to define and materialize views over events which are an important feature when managing probabilistic data. MystiQ also handles sufficient lineage with minimum errors (Re & Suciu, 2008). However, MystiQ do not handle queries with a having clause and queries with self joins. It treats these queries as unsafe queries. As it also do not support the polynomial lineage. These unsupported features are considered to be shortage in the MystiQ that need to be covered in other work.

4.4 Orion

Orion database system (previously known as U-DBMS), is a state-of-the-art uncertain database management system with built-in support for probabilistic data as first class data type. In contrast to other uncertain database, Orion supports both attribute and tuple uncertainty with arbitrary correlations. This enables the database to handle both continuous and discrete uncertain values. It also provides various indexes for efficient query evaluation. It is implemented in C and PL/PGSQ (Cheng, Singh, & Prabhakar, 2005). It is built on top of PostgreSQL, an object-oriented relational open-source database system.

Orion main features include (Cheng, Singh, & Prabhakar, 2005):

- An integrated implementation of the "PDF Attributes" data model, which is consistent with Possible Worlds Semantics and supports both continuous and discrete uncertainty.
- Efficient access methods for querying uncertain data, including three index structures based on R-trees, signature trees, and inverted indexes.
- Improved query optimization, join algorithms, and selectivity estimation by gathering and exploiting additional statistics over probabilistic data types.
- Integration with PL/R for graphical visualization of and statistical inference over uncertain data.

4.5 MCDB: Monte Carlo Database System

This is a prototype system that proposes a new approach for handling enterprise-data uncertainty (Jampani, Xu, Wu, Perez, Jermaine, & Haas, 2008). Within the MCDB the uncertainty is not included in the data model, and the query processing is performed on the classical relational data model. MCDB enable the user to declare arbitrary variable generation (VG) function that embodies the database uncertainty. This is then used by the MCDB to generate random values for the uncertain attributes, and to run queries.

MCDB main features are (Jampani, Xu, Wu, Perez, Jermaine, & Haas, 2008):

- Representing uncertainty via "VG functions". This is used to randomly generate realized values for uncertain attributes.
- Handle arbitrary joint probability distributions over discrete or continuous attribute.
- Use novel query processing techniques, executing a query plan exactly once, over tuple bundles instead of ordinary tuples.

However MCDB has several points that need improvement as the query optimization, error control, risk assessment and lineage.

4.6 BayesStore: Probabilistic Data Management Architecture

Most recent approach that develops a probabilistic data base management system depends on simplistic model of uncertainty which can be easily mapped onto existing relational architectures: Probabilistic information is associated with individual data tuples. But unfortunately that introduce a gap between the statistical model which is used by the analysts and the model in the probabilistic DB, this is the case in the Trio and MayBMS.

BayesStore project solve this "model-mismatch" by supporting statistical models, evidence data and inference algorithms as first-class in the probabilistic data base management system (Wang, Michelakis, Garofalakis, & Hellerstein, 2008). BayesStore ia a probabilistic data management architecture built on the principle of handling statistical models and probabilistic inference tool.

BayesStore main features (Wang, Michelakis, Garofalakis & Hellerstein, 2008):

- Encode the correlation patterns between uncertain data.
- Enhance probabilistic inference and statistical model manipulation as part of the standard DBMS.
- Represents model and evidence data as relational tables.
- Implement inference algorithm efficiently in SQL.
- Add probabilistic relational operators to the query engine.
- Optimizes query with both relational and inference operators.

The BayesStore goals can be summed up as; supporting query processing efficiently, supporting extensible API for plugging in new models and inference algorithms, and scaling up to large datasets.

4.7 PrDB: Probabilistic Data Base

PrDB goal is to design a probabilistic database model that can capture the uncertainties and complex correlations that appear in real world application. And also capture the probabilistic regularities. PrDB unifies ideas from large-scale structures graphical model like relational model (PRMs), and probabilistic query processing. (Sen & Deshpande, 2007)

Its framework is based on the notion of "shared factors", which not only allow the expression and manipulation of uncertainties at various levels of abstractions, but also support capturing rich correlations among uncertain data. PrDB support declarative SQL-like language for specifying uncertain data and correlations among them.

PrDB main features are(Sen & Deshpande, 2007):

- Support capturing rich correlations among uncertain data.
- Support exact and approximate evaluation of wide range of queries, including references, SQL queries, and decision queries.

Finally, these systems can be summed up as follows: Trio project (Benjelloun, Sarma, Halevy, & Widom, 2006) focused on the study of uncertainty and lineage in incomplete database. MystiQ (Boulos, Dalvi, Mandhani, Mathur, Chris, & Suciu, 2005) supports various constructs for handling uncertainty that include tuples associated with probabilities. MystiQ is mainly a middle ware that leverage infrastructure of existing DB engines. The

MayBMS project (Huang, Antova, Koch, & Olteanu, 2009) focused on representation problems, query language design, and query evaluation on uncertain data. A fundamental design choice that set MayBMS apart from Trio and MystiQ is that it's an extension of the open-source PostgresSQL server backend, and not a front-end application of PostgreSQL .MCDB (Jampani, Xu, Wu, Perez, Jermaine, & Haas, 2008) focused on complex probabilistic model with native Monte Carlo simulation. Orion project (Cheng, Singh, & Prabhakar,2005) focused on tuple and attribute uncertainty with attribute correlation given by continuous value probably distribution. BayesStore (Wang, Michelakis, Garofalakis, & Hellerstein, 2008) efficiently express and reason about correlation among uncertain data items, in a concise and statistical way. PrDB (Sen & Deshpande, 2007) focus on managing and exploiting rich correlations in probabilistic databases .Other group has also studied correlation in probabilistic database (Sen & Deshpande, 2007). Table 2 presents a comparison between these uncertain management systems.

5. Conclusion

The field of uncertain data management has become one of the most vital topics in recent years. That caused a lot of techniques to be introduced to handle the different management issues of uncertainty. This paper surveys broad areas of work in uncertainty management issues. We presented the important management techniques along with the key representational issues in uncertain data management. The field of the uncertainty management will expand over time, so we hope that this survey will be a good starting point to researchers focusing on the important and emerging issues in this field. In this paper we also gave an overview of the DBMS that handle uncertain data, shown its features and weakness.

Uncertain DBMS can be enhanced by taking the probability of all instances into account in data management. For example, taking the instances probability in the aggregate queries and events can have a great effect on the accuracy of the DBMS. As considering probabilities is indispensable when dealing with uncertain data, this probability usage need to be improved. Enhancing the aggregate queries on uncertain data is the main scope for our future work.

Table 2. Uncertain Database Systems Comparison

Uncertain Systems	**Developed at (University)**	**Representation System**	**Operations (supported or not supported)**				**Lineage**
			Select	*Join*	*View*	*Agg.Q*	
Trio	Stanford	ULDB Model	TriQL	Yes	Yes	Yes	Yes
MystiQ	Washington	Probabilistic Data Model	Extend SQL. Queries *with having* not supported	Self Join not support	Yes	(Sum ,Count) only	Sufficient lineage but, no polynomial lineage
Orion	Purdue	PDF attributes Data Model	Yes	Yes	Yes	Yes	No
MayBMS	Cornell	U-Relation	I-SQL (Extend SQL with uncertain constructs)	Yes	No	Expectation of aggregates	No
MCDB	Wisconsin	Probabilistic Data Model with VG functions.	Yes	Yes	Yes	Yes	Need to Improve.
BayesStore	California	First-order statistical Model	Yes	Yes	Yes	No	No
PrDB	Maryland	Graphical, state of art probabilistic Model	Yes	Yes	No	No	No

References

Afrati, F. N., & Vasilakopoulos, A. (2010). Managing Lineage and Uncertainty under a data exchange setting. *In Proceedings of the 4th international conference on Scalable uncertainty management* (pp. 28-41). http://dx.doi.org/10.1007/978-3-642-15951-0_9

Agarwal, P. K., Cheng, S., Tao, Y., & Yi, K. (2009). Indexing uncertain data. In proc.Association for computing machinery ,(pp.137-146). http://dx.doi.org/10.1145/1559795.1559816

Aggarwal, C. C. (2009). *Managing and Mining Uncertain Data.*

Aggrwal, C. C. (2009). A survey of uncertain data algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering* (pp.609-623). http://dx.doi.org/10.1109/TKDE.2008.190

Agrawal, P., & Widom, J. (2007). *Confidence-aware join in large uncertain database.* Retrieved from In http://dbpuds.stanford.edu/pub/2007-14

Agrawal, P., & Widom, J. (2009). Continuous uncertainty in trio. *In Metropolitan underwriting discussion* .

Agrawal, P., Benjelloun, O., Das, A., Hayworth, C., Nabar, S., & Sugihara, T., et al. (2006). Trio: a system for data, uncertainty, and lineage. *Proceedings of the 32nd international conference on Very large data base*, (pp. 1151-1154). http://dx.doi.org/10.1.1.108.9426

Antova, L., Jansen, T., Koch, C., & Olteanu, D. (2008). Fast and simple relational processing of uncertain data. *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering* (pp. 983-992). http://dx.doi.org/10.1109/ICDE.2008.4497507.

Benjelloun, O., Sarma, A. D., Halevy, A., & Widom, J. (2006). ULDBs: Databases with uncertainty and lineage. *Proceedings of the 32nd international conference on Very large data bases* (pp. 953-964). http://dx.doi.org/10.1.1.119.3771

Boulos, J., Dalvi, N., Mandhani, B., Mathur, S., Chris, R., & Suciu, D. (2005). Mystiq: a system for finding more answers by using probabilities. *Proceedings of the International Conference on Management of Data*, (pp. 891-893). http://dx.doi.org/10.1145/1066157.1066277

Burdick, D., Deshpande, P., Jayram, T. S., Ramakrishnan, R., & Vaithyanthan, S. (2005). OLAP over uncertain and imprecise data. *Proceedings.31st int'l Conf. Very Large Data Bases* (pp.123-144). http://dx.doi.org/10.1007/s00778-006-0033-y

Cheng, R., Singh, S., & Prabhakar, S. (2005). U-DBMS: a database system for managing constantly-evolving data. *Proceedings of the 31st International Conference on Very Large Data Bases* (pp.1271-1274). http://dx.doi.org/10.1.1.153.4956

Cheng, R., Singh, S., Prabhakar, S., Shah, R., Vitter, J. S., & Xia, Y. (2006). Efficient join processing over uncertain data. *Proceedings of the 15th international conference on Information and Knowledge management*, (pp. 738-747). http://dx.doi.org/10.1145/1183614.1183719

Cheng, R., Xia, Y., Prabhakar, S., Shah, R., & Scott, J. (2004). Efficient indexing methods for probabilistic threshold queries on uncertain data. *Proceedings of International conference on Very large data* (pp. 876-887). http://dx.doi.org/10.1.1.147.1107

Chothia, T., Kawamoto, Y., Novakovic, C., & Parker, D. (2013). Probabilistic Point-to-Point Information Leakage. *In Computer Security Foundations Symposium* (pp.193-205). http://dx.doi.org/10.1109/CSF.2013.20

Cormode, G., Li, F., & Yi, K. (2009). Semantics of ranking queries for probabilistic data and expected ranks. *Proceedings of the 25th International Conference on Data Engineering* (pp. 305 - 316). http://dx.doi.org/10.1109/ICDE.2009.75

Dallachiesa, M. I. (2014). Top-k Nearest Neighbor Search In Uncertain Data Series. *Proceedings of the (international conference on Very large data) Endowment, 8*(1), (pp. 13-24).

Dalvi, N., & Suciu, D. (2005). Query Answering Using Statistics and Probabilistic Views. *Proceedings 31st Int'l Conf. Very Large Data Bases* .

Dalvi, N., & Suciu, D. (2007). Efficient Query Evaluation on Probabilistic Databases. *The International Journal on Very Large Data Bases*,16, (pp.523-544). http://dx.doi.org/10.1007/s00778-006-0004-3

Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J., & Hong, W. (2004). Model-Driven Data Acquisition in Sensor Networks. *Proceedings of the Thirtieth international conference on Very large data bases* (pp. 588-599).

Dhandore, K., & Ragha, L. (2014). Performance Evaluation of Decision Trees for Uncertain Data Mining. *International Journal of Emerging Trend and Technology in computer science, 3*(6).

Doan, A., & Halevy, A. (2005). Semantic integration research in the database community: A brief survey. *AI*

Magazine , (pp.83–94). http://dx.doi.org/10.1609/aimag.v26i1.1801

Fuhr, N., & Rolleke, T. (1997). A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. *Association for Computing Machinery. Trans. Information Systems* (pp.32-66). http://dx.doi.org/10.1145/239041.239045

Gal, A. (2006). Managing uncertainty in schema matching with top-k schema mappings. *Journal on Data Semantics VI 4090* (pp. 90–114). http://dx.doi.org/10.1007/11803034_5

Halevy, A. (2003). Data integration: A status report. *In BTW. LNI, 26*, 24-29.

Hua, M., Pei, J., Zhang, W., & Lin, X. (2008). Ranking queries on uncertain data: a probabilistic threshold approach. *Proceedings of International Conference on Management of Data* (pp. 673-686). http://dx.doi.org/10.1.1.144.8130

Huang, J., Antova, L., Koch, C., & Olteanu, D. (2009). MayBMS: A Probabilistic Database Management System. *Proceedings of the 2009 SIGMOD International Conference on Management of Data* (pp. 1071-1074). http://dx.doi.org/10.1145/1559845.1559984

Jampani, R., Xu, F., Wu, M., Perez, L. L., Jermaine, C., & Haas, P. J. (2008). MCDB: a monte carlo approach to managing uncertain data. *Proceedings of the SIGMOD International Conference on Management of Data*, (pp. 687-700). http://dx.doi.org/10.1145/1376616.1376686

Keijzer, A. D., & Keulen, M. V. (2007). User feedback in probabilistic integration. *Proceedings of the 18th International Conference on Database and Expert Systems Applications* (pp.377 - 381). http://dx.doi.org/10.1109/DEXA.2007.97

Keijzer, A. D., & Keulen, M. V. (2008). Imprecise: Good-is-good-enough data integration,. *Proceedings of the 24th International Conference on Data Engineering* (pp. 1548-1551). http://dx.doi.org/10.1109/ICDE.2008.4497618

Keulen, M. V., Keijzer, A. D., & Alink, W. (2005). A probabilistic XML approach to data integration. *Proceedings of 21st International Conference on Data Engineering* (pp. 459–470). http://dx.doi.org/10.1109/ICDE.2005.11

Kriegel, H. P., Kunath, P., & Renz, M. (2007). Probabilistic nearest neighbor query on uncertain objects. *Proceedings of the 12th International Conference on Database Systems for Advanced Applications* (pp. 337-348). http://dx.doi.org/10.1007/978-3-540-71703-4_30

Li, J. W. (2014). Range Queries on Uncertain Data. *Springer* (pp.326-337). http://dx.doi.org/10.1007/978-3-319-13075-0_26

Lian, X., & Chen, L. (2008). Probabilistic group nearest neighbor queries in uncertain databases. *Knowledge and Data Engineering, IEEE, 20*(6), 809 – 824. http://dx.doi.org/10.1109/TKDE.2008.41

Lian, X., & Chen, L. (2008). Probabilistic Ranked Queries in Uncertain Database. *Proceedings of the 11th international conference on Extending database technology: Advances in database technology* (pp. 511-522). http://dx.doi.org/10.1145/1353343.1353406

Lian, X., & Chen, L. (2009). Efficient processing of probabilistic reverse nearest neighbor queries over uncertain data. *Proceedings of International Conference on Very Large Data* (pp.787-808). http://dx.doi.org/10.1007/s00778-008-0123-0

Lian, X., & Chen, L. (2009). Top-k Dominating Queries in Uncertain Databases. *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology* (pp. 660-671). http://dx.doi.org/10.1145/1516360.1516437

Lian, X., & Chen, L. (2011). Probabilistic Inverse Ranking Queries in Uncertain Databases. *Proceedings of International Conference on Very Large Data* (pp.107-127). http://dx.doi.org/10.1007/s00778-010-0195-5

Lian, X., & Chen, L. (2011). Shooting top-k stars in uncertain databases. *The International Journal on Very Large Data Bases* (pp.819-840). http://dx.doi.org/10.1007/s00778-011-0225-y

Lian, X., & Chen, L. (2013). Probabilistic Top-k Queries in Uncertain Database. *Information Sciences* .

Ljosa, V., & Singh, A. K. (2008). Top-k spatial joins of probabilistic objects. *Proceedings of the 24th International Conference on Data Engineering* (pp. 566 - 575). http://dx.doi.org/10.1109/ICDE.2008.4497465

McCamant, S., & Ernst, M. D. (2008). Quantitative Information Flow as network flow capacity. *Proceedings of*

the 2008 ACM SIGPLAN conference on Programming language design and implementation (pp. 193-205). http://dx.doi.org/10.1145/1375581.1375606

Motro, A. (1994). Managing of Uncertainty in Database Systems. *In Modern Database Systems, Addison-Wesley/ACM Press* (pp.457-476)

Motro, A. (1995). Imprecision and Uncertainty in database systems. *In Fuzziness in Database Management Systems, Physica-Verlag* (pp 3-22). http://dx.doi.org/10.1007/978-3-7908-1897-0_1

Ngo, T. M., & Huisman, M. (2013). Quantitative Security analysis for multi-threaded programs. *QALP* (pp.34-48). http://dx.doi.org/10.4204/EPTCS.117.3

Ni, J., Ravishankar, C. V., & Bhanu, B. (2003). Probabilistic Spatial Database Operations. *Springer* (pp.140-158). http://dx.doi.org/ 10.1007/978-3-540-45072-6_9

Nottelmann, H., & Straccia, U. (2005). splmap: A probabilistic approach to schema matching. *In European Conference on Information Retrieval* (pp.81–95).

Nottelmann, H., & Straccia, U. (2007). Information retrieval and machine learning for probabilistic schema matching. *Information Processing and Management: an International Journal , 3*(43), 552–576). http://dx.doi.org/10.1016/j.ipm.2006.10.014

Parsons, S., & Saffiotti, A. (1993). Integrating uncertainties handling formalisms in distributed artificial intelligence. *Proceedings of 2nd European Conference on symbolic and quantitative approach to reasoning and uncertainty* (pp.304-309). http://dx.doi.org/10.1007/BFb0028214

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems.*

Peng, L., Diao, Y., & Liu, A. (2011). Optimizing Probabilistic Query Processing on Continuous Uncertain Data. *Proceedings of International conference on Very large data*. http://dx.doi.org/10.1.1.220.7685

Qi, Y., Singh, S., Shah, R., & Prabhakar, S. (2008). Indexing probabilistic nearest neighbor threshold query. *QDB/MUD* (pp.87-102). http://dx.doi.org/10.1.1.147.4957

Re, C., & Suciu, D. (2007). Management of Data with Uncertainties. *CIKM'07*.

Re, C., & Suciu, D. (2008). Managing Probabilistic data with MystiQ: The can-do, the could-do, and the can't-do. *Proceedings of the 2nd international conference on Scalable Uncertainty Management* (pp. 5 - 18). http://dx.doi.org/10.1007/978-3-540-87993-0_3

Re, C., Dalvi, N., & Suciu, D. (2007). Efficient top-k query evaluation on probabilistic data. *Proceedings of the 23th International Conference on Data Engineering* (pp. 886 - 895). http://dx.doi.org/10.1109/ICDE.2007.367934

Rowe, W. D. (1994). *Managing Uncertainty.*

Sarma, A. D., Dong, X., & Halevy, A. (2008). Bootstrapping pay-as-you-go data integration Systems. *Proceedings of the ACM SIGMOD international conference on Management of Data* (pp. 861-874). http://dx.doi.org/10.1145/1376616.1376702

Sen, P., & Deshpande, A. (2007). Representing and Querying Correlated Tuples in Probabilistic Databases. *Proceedings of 23rd IEEE Int'l Conf. Data Eng. International Conference on Data Engineering* (pp. 596 - 605). http://dx.doi.org/10.1109/ICDE.2007.367905

Singh, S., Mayfield, C., Prabhakar, S., Shah, R., & Hambrusch, S. (2007). Indexing uncertain categorical data. *International Conference on Data Engineering* (pp.616-625). http://dx.doi.org/10.1109/ICDE.2007.367907

Soliman, M., Ilyas, I., & Chang, K. C. (2007). Top k-Query Processing in Uncertain Databases. *International Conference on Data Engineering*, (pp. 896 - 905). http://dx.doi.org/10.1109/ICDE.2007.367935

Tao, Y., Cheng, R., & Xiao, X. (2007). Range search on multidimensional uncertain data. *Association for Computing Machinery, Transactions on Database Systems*. http://dx.doi.org/10.1145/1272743.1272745

Tao, Y., Cheng, R., Xiao, X., Ngai, W. K., Kao, B., & Prabhakar, S. (2005). Indexing multi-dimensional uncertain data with arbitrary probability density function. *Proceedings of the 31st International Conference on Very Large Data Bases* (pp. 922 - 933). http://dx.doi.org/10.1.1.113.3540

Touzi, A. G., & Hassine, M. A. (2009). New Architecture of Fuzzy Database Management Systems. *International Arab Journal of information technology , 6*(3). http://dx.doi.org/10.1.1.182.5811

Wang, D. Z., Michelakis, E., Garofalakis, M., & Hellerstein, J. M. (2008). Bayesstore: managing large, uncertain

data repositories with probabilistic graphical models. *Very Large Database* (pp.340-351). http://dx.doi.org/10.1.1.140.6348

Widom, J. (2005). Trio A System for integrated management data, accuracy and lineage. *Conference on Innovative Data Systems Research.* http://dx.doi.org/10.1.1.153.9613

Yi, K., Li, F., Kollios, G., & Srivastava, D. (2008). Efficient processing of top-k queries in uncertain databases. *Proceedings of the 24th International Conference on Data Engineering* (pp. 1669 - 1682). http://dx.doi.org/s 10.1109/TKDE.2008.90

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).