An Improved Data Model for Uncertain Data

UMAR HAYAT*, AND MUHAMMAD USMAN GHANI KHAN**

RECEIVED ON 13.02.2015 ACCEPTED ON 28.05.2015

ABSTRACT

Uncertain data can be categorized as imprecise data and probabilistic data. In each of these categories, the uncertainty can be found at different granularity levels. Conventional data models are developed for the purpose of storing, manipulating and retrieving certain data. These data models do not extend their support for the management of uncertain data. Thus, a standalone data model is required aimed at storing, manipulating and retrieving certain as well as uncertain data. In this paper we introduce UDM-relations, an uncertain data model for the management of uncertain data along with certain data. Vertical partitioning approach is used to translate an uncertain relation into UDM-relations. Our data model supports ALU (Attribute-Level Uncertainty) as well as TLU (Tuple-Level Uncertainty) for the finite sets of possible worlds. It follows the concept of standard relational database technology. With slight modifications to standard relational algebra operators, we have introduced four relational operators that are used to evaluate a query on UDM-relations.

Key Words: Uncertain Data, Data Model, Attribute-Level Uncertainty, Tuple-Level Uncertainty, Vertical Partitioning.

1. INTRODUCTION

In recent years, the field of uncertain databases has witnessed a revived interest due to the emergence of a wide range of indirect data gathering methodologies employed in various fields such as sensor data management, moving object management, web data integration, weather forecasting, and economic decision making etc. [1-2]. RFID (Radio Frequency Identification) systems have become much popular for the identification of moving objects. These methodologies or applications generate large amounts of uncertain data that needs to be managed carefully. It is to be noted that a lot of research for the management of uncertain data has been carried out in the past [3-8] and it does not study the nature of uncertain data generated as a result of new applications. Thus the nature of uncertain data generated by these new applications requires newer techniques for its management [9]. In recent years, some attempts [10-16] have been made in this direction.

Roughly, uncertain data can be defined as inexact data i.e. data which is not certain. Based on application area, we can categorize uncertain data as imprecise data and probabilistic data. In this paper we only deal with imprecise data. The uncertainty in imprecise data may exist at two levels i.e. ALU and TLU. ALU in an imprecise database is meant that an attribute may take more than one value for any given tuple. Whereas TLU questions about the presence of a tuple in a relation i.e. whether the specific

*Ph.D. Scholar, and **Associate Professor,
Department of Computer Science & Engineering, University of Engineering and Technology, Lahore.

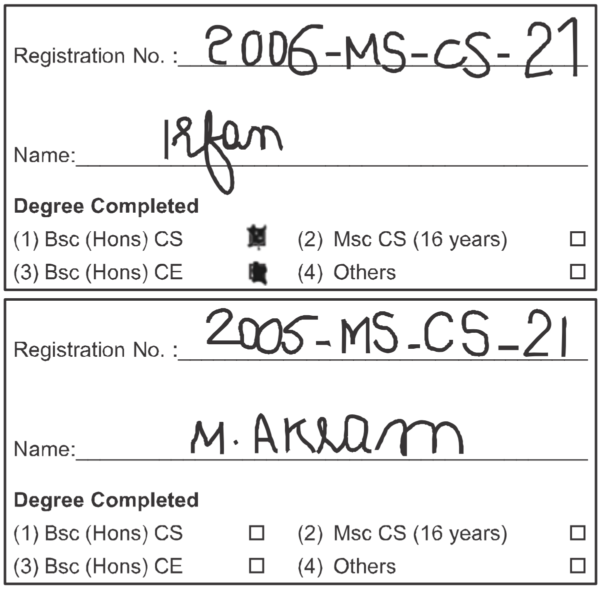
tuple is part of the database or not; this kind of uncertainty is also called existential uncertainty. Traditional DBMSs (Database Management Systems) only consider deterministic data at both attribute level and tuple level which means that either the data is definitely present or not [17]. They have no support for uncertain data in order to store and query it. In order to cope with this situation, researchers are left with three options at hand: (i) clean away the uncertainty present in data so that it can be managed by traditional DBMS [18-19]. This procedure results in loss of information that can be helpful in different decision making processes at later stages [20]. (ii) Deal with uncertainty present in data at application layer and let it be managed by conventional DBMS [21]. This strategy puts some additional burden on applications, thus, lowering their performance [20]. (iii) Develop some techniques that incorporate uncertainty as a first class citizen and handle it accordingly without losing useful information and also without exerting additional burden on applications. Thus the broader issue in the field of uncertain databases is to develop a data model (or simply model) that could provide a mechanism for storage, manipulation and retrieval of uncertain data. Based on our study of existing literature in respect of modeling uncertain data, all the proposed models have very obvious shortcomings of some kind or the other. Most of the existing data models only focus on ALU and they have no support for the TLU. In this paper, we propose a data model that stores, retrieves and manipulates both kinds of uncertainty.

Our model can be used for a range of applications. To illustrate, we give a real world example of a school or college database that depicts the snapshot of uncertain data. Let us consider an academic background scenario in which a significant number of students desirous to take admission in a graduate program at various institutions manually fill in admission forms in order to record their academic background. Shortly, due to its importance the data contained in these forms is to be filled in a database but uncertainty may be experienced about the correct values for some entries of the form. Fig. 1 shows two simple filled-in admission forms. Each form contains information about a student i.e. Registration No., Name and Degree Completed. In both forms we are uncertain about the degree of students as shown in Fig. 1. In the above form, either the terminal degree is B.Sc. (Hons) CS or B.Sc. (Hons) CE whereas in lower form the student has not even marked the terminal degree. Thus it can be any one of the given four options for degree. This figure illustrates the nature of uncertain data to be stored in a school or college database. Due to space limitations we do not explain other applications of uncertain data.

The rest of the paper is organized as follows. In Section 2, we give a comprehensive literature survey of the related work. In Section 3, we propose a data model known as UDM-relations that handles both kinds of uncertainty i.e. ALU and TLU. We also illustrate our proposed data model with the help of a running example and show that our data model is complete as it can also capture the results of any given query.

2. RELATED WORK

In this section, we discuss some key concepts used in uncertain databases and explain different types of



Registration No. : 2006-MS-CS-21

Name: IRfan

Degree Completed

(1) Bsc (Hons) CS	☒	(2) Msc CS (16 years)	□
(3) Bsc (Hons) CE	☒	(4) Others	□

Registration No. : 2005-MS-CS-21

Name: M. AKRam

Degree Completed

(1) Bsc (Hons) CS	□	(2) Msc CS (16 years)	□
(3) Bsc (Hons) CE	□	(4) Others	□

FIG. 1. TWO SIMPLE FILLED IN FORMS

uncertain data and their sources of generation, and also give details of some of the existing data models that are being used to model uncertain data.

2.1 Uncertainty and Uncertain Data

The context of uncertain data that we generally use is the term uncertainty. Uncertainty is the quantitative measure of error present in data and thus we also call uncertain data as inexact data [22]. In other words, uncertainty exists in a system that has an environment, which has potential to generate uncertain data due to known and unknown factors, or the method or equipment used to acquire the data from the system that has potential to acquire uncertain data. The term, uncertainty, describes the attributes of a real world entity, which are difficult to state with complete confidence [23]. In this paper we term imprecise data as the uncertain data. To be more specific, imprecision means that the data is not too precise, for example, the temperature outside is between 25 and 28 centigrade [24]. After elaborating the terms uncertainty and uncertain data, now we discuss various sources of uncertain data.

2.2 Sources of Uncertain Data

Following are some of the causes and sources of uncertain data:

- Uncertain data may occur due the shortcomings of data collection instruments. First, these shortcomings may occur as a result of noise present in sensor input. Second, data is collected and transmitted from one node to the next and during this transmission process some noise may get added which results in errors [17,24-25].
- Census data is collected for the purpose of making some calculation with regard to the population of country, such data may be incomplete or imprecise [25].
- Biological data is also uncertain having unpredictable behavior [26].
- Survey forms are frequently used method of collecting data for various purposes. The data collected through survey forms may also be imprecise and incomplete [23].
- In some mobile applications, for example, spatiotemporal applications, extrapolation methods are used in order to approximate the future behavior of objects as the trajectory of objects in these applications may not be known. Uncertainty in data is directly proportional to the use of extrapolation methods [25].
- In recent years, with the development of new applications that involve uncertain data, a great deal of research is being made in this field. The application domains which have potential to generate uncertain data include economic decision making, stock market prediction, and management of moving vehicles etc. [15,24].

2.3 Types of Uncertainty and Possible Worlds Model

There are two types of uncertainty used in uncertain databases; one is TLU and the other is ALU. In TLU, we are not sure whether the tuple belongs to a database or not. In other words we are not sure about the existence of a tuple in a relation; it may or may not be part of database. This kind of tuple is also known as maybe tuple [20]. Whereas in ALU, multiple or a range of values may be assigned to a field against any tuple. For example, a sensor may indicate the temperature outside as 34 or 35 or an incorrectly filled form may show entry of a student id as 21 or 27. Table 1 gives a snapshot of both kinds of uncertainty. Due to these kinds of uncertainty, an uncertain database having multiple possibilities can be translated and then processed as a set of possible database instances. This phenomenon is discussed in the Possible Worlds Model [27]. The possible worlds model states that an uncertain database can be described as a set of possible database instances [25,28]. Each possible database instance is termed as possible world and it is obtained by assigning a possible value from the domain of possible values of attributes and those of tuples. In case of uncertainty at tuple level, the existence of each uncertain tuple will be a Boolean; it means that tuple may be either present in or absent from a database instance. At attribute level uncertainty, a possible database instance can occur by taking one value every time from the domain of each attribute. Now we discuss some important data models for uncertain data as studied in literature.

2.4 Characteristics of a Data Model for Uncertain Data

In this section, we discuss various characteristics of a data model for uncertain data to which each data model studied in literature owns to a certain degree. These characteristics are:

(1) ***Expressive Power:*** The expressive power of a data model is described as the completeness of its formalism in terms of capturing the results of any query. A data model is considered a complete model if its formalism has the potential to capture the results of a given query. It means that the data model should be closed under relational algebra operations in order to query data and represent its results [14,29].

(2) ***Easy to Understand and Implement:*** The data model for uncertain data should be easier to understand and implement both for researchers and developers. It should be simple enough that researchers and developers could easily use it for the defined purpose [20,27].

(3) ***Support for Relational Paradigm:*** It is better for a data model aimed at targeting uncertain data to extend its support for relational database technology so that it can also be implemented above an application that handles uncertain data at application layer and leverages relational database technology for the purpose of storage, manipulation and retrieval [11].

(4) ***Conciseness:*** The data model should be capable of representing huge volumes of alternative worlds while utilizing much lesser space to compute and process these worlds [29].

(5) ***Granularity of Uncertainty:*** It is very important feature of a data model to determine whether it handles ALU or TLU or both [4-5,10].

(6) ***Competency for Query Evaluation:*** Possibly, data models evaluate queries at the expense of conciseness and vice vorce [29]. Thus, a data model should be competent enough to evaluate interesting queries without compromising the conciseness of its representation formalism [29].

Now we discuss various important data models with respect to above characteristics.

TABLE 1. REPRESENTATION USING OR–SET RELATION

Id	P_Identity	P_Uniform	P_Arm
P1	Guard	Security	Gun
P2	terrorist || guard	Security	knife || stick
P3	employee || terrorist	Dress	phone || pistol
P4	terrorist || com._man	Dress	phone || knife ?

2.5 Existing Data Models for Uncertain Data

In this section we give a brief description of data models for uncertain data as already studied in literature. These data models partially follow the characteristics stated above. Such models either focus on expressiveness at the cost of intuitiveness while others attempt to be simple and intuitive but incomplete in terms of expressiveness. Most of the data models only deal with ALU and they do not consider TLU. A very few imitate the concepts from relational database technology and thus are suitable to be implemented above such applications. Some of them tend to be expressive at the expense of conciseness.

V-tables [3] is a data model designed for the purpose of modeling uncertain data. It targets ALU and handles only finite sets of possible worlds. In this data model, tuples are allowed to take entries in the form of constants as well as variables and each combination of assignment of possible values to the variables gives a possible world [10]. Its advantage is that it is cost optimized and intuitive data model which makes it easier to implement. This model is not a complete data model for finite sets of possible worlds as it cannot represent the results of any relational algebra query with its formalism [10]. This model focuses on simplicity and intuitiveness at the cost of expressiveness [10]. Another disadvantage of this data model is that it does not provide concise representation of data. It follows relational database management system technology but its major drawback is of not being so expressive as to capture the results of any query using its formalism. This means that v-tables data model is not a complete model. The focal point is that one of the very important data models known as c-tables [3] inherits the idea from this model.

Another contribution in this domain is the work known as or-set relations [10]. This model also necessitates the use of variables for the sake of representing incomplete information. Like v-tables, it targets ALU and it does not handle TLU. It deals with finite sets of possible worlds only. In this data model, variables are permissible to appear in relations in order to represent a set of values in that relation but unlike v-tables, this model permits one variable to appear at a single position only. Thus the assignment of values to these variables is from a fixed finite set of values associated with each variable [10]. The similarity between v-tables and or-set relations is that, like v-tables, or-set relations are also cost optimized and too simple to understand and implement. This highly intuitiveness is achieved at the expense of expressiveness. It means that or-set relations are a simple but not a complete data model as these relations cannot capture and represent the results of a relational algebra query [10].

After a few attempts in the form of incomplete data models, the most fundamental work in this research domain appears as c-tables [10]. The c-tables (or conditional tables) data model is the extension of v-tables. In this model, there appear some conditions which are used in relations. Of these conditions, one is called the local condition which is local to that relation only and the other one is global condition. The scope of global condition is up to all relations of a database. A different local condition is applied against every tuple of a relation and it has its effect only to that tuple. However, the scope of global condition extends to all relations of database [10]. Basically, these conditions are Boolean formulas and their purpose is to constrain possible values for each variable [10]. The fundamental importance of this model is that this is the first data model considered to be a complete data model. It can capture and represent the results of a query using its formalism. This means that this data model has a higher degree of expressiveness. But the major drawback of this model is that it provides expressiveness at the cost of intuitiveness. Also, it deals with ALU only. This model is not fit for applications in which tuples themselves are uncertain. It is too complex to understand and implement by both researchers and developers. Thus, it has failed to find its application in practice because of its non-intuitiveness [10].

WSD (World-Set Decompositions) [10] is another work in this domain. This data model gives the concept of a world-set relation. This model is based on vertical decompositions of a world-set relation. A world-set relation is a table that stores a given set of possible worlds and each tuple of this relation represents one possible world [10]. In WSD, a world-set relation is decomposed vertically in such a way that the cartesian product of these decomposed relations is again a world-set relation. Each decomposed relation of a world-set relation is called a component. Decomposition of a world-set relation takes place on the assumption that the attributes of a relation are independent of each other. However, in cases where the attributes of a relation depend on each other, such attributes can be combined in one component. This model is considered a complete data model; that is, it is a mature model in terms of expressiveness. It is also simple and understandable in terms of its implementation. It also follows the concepts of relational database technology that means it is easier to implement this data model over such applications which utilize relational database concepts. The major drawback of this model is that it only supports ALU and it has nothing to do with TLU.

U-relations [23] is also a data model that exploits the concept of vertical partitioning [10] of a world-set relation as used in [30], in which a world-set relation is decomposed vertically resulting in multiple uncertain relation each representing one tuple with alternatives at attribute-level. It actually combines some good features of two existing representations systems [10,31]. The concept of vertical partitioning of a world-set relation come from WSD [10] and the concept of how to translate relational algebra queries originates from ULDB [11]. For the purpose of query translation, it exploits the operations of conventional relational algebra with the augmentation of an operation 'possible'. The selection, projection and join operations in u-relations data model are carried out in the same manner as those of used in state-of-the-art relational data model. An additional operation 'possible' is the core of this model which is carried out through projection operation. In u-relations query evaluation is carried out the same way as is done in RDBMS.

A comparative analysis of various data models for uncertain data has been shown in Table 2 from the aspects of completeness, expressiveness, scalability, succinctness, cost optimization, and intuitiveness.

TABLE 2. COMPARISON OF VARIOUS DATA MODELS

	v-tables	or-set	c-tables	WSD	U-relations
Completeness	✗	✗	✓✓	✓	✓
Expressiveness	✓	✓	✓✓	✗	✗
Scalability	✗	✗	✓	✓✓	✓✓
Succinctness	✗	✗	✓	✓	✓✓
Cost optimized	✓	✓	✗	✓	✓✓
Intuitiveness	✓	✓	✓	✗	✗
ALU	✓	✓	✓	✓	✓
TLU	✗	✗	✗	✗	✗

* For stronger models a double check (✓✓) is used and single check (✓) is for weaker models

3. PROPOSED DATA MODEL

In this section, we propose UDM-relations, a data model for uncertain data that is intuitive as well as a complete data model. Our model uses vertical partitioning approach [31] in order to translate an uncertain relation into UDM-relations. The fundamental characteristic of our model is that it deals with ALU as well as TLU. Most of the data models for uncertain data as studied in literature target only ALU but our model not only handles ALU but it also deals with TLU. A prominent feature that makes our data model unique in this field is that it can differentiate between uncertain data and certain data in query results in the same way as this data is present in base (uncertain) relation. We can conclude from our query results as to which data is certain in base (uncertain) relation. To the best of our knowledge, the data models studied in literature so far are not able to make difference about the certainty of data in query results i.e. whether the data in query results appear certain or uncertain in base (uncertain) relation. In other words, these models only tell the way to process uncertain data and their query results do not differentiate between certain data and uncertain data as it appears in base relation. While UDM-relations are capable of showing the uncertainty in query results as it is present in base (uncertain) relations. Our model is aimed at modeling finite sets of possible worlds. Now, we introduce our model with the following example.

Example: Suppose an aerial photograph, captured just before a terrorist attack on an office building, shows four persons (P_1, P_2, P_3, and P_4) at distinct positions. It is known that some of the armed terrorists were in security uniform like that of security personnel working in office while others were in official dress. Due to low resolution of the image, it is difficult to identify the persons. But we can draw some conclusion on the basis of their positions. Assume we know that P_1 who is in security uniform is a security guard with a gun. P_2 is also in security uniform and may be either a terrorist or a security guard carrying with him a knife or stick. P_3 is in dress uniform and seems to be either an employee of the office having phone in his hand or a terrorist with a pistol. Another very blurred object (we call it P_4) present on the main road outside the office building can also be seen in image. It seems either this is a person or an object of some kind. If he is a person, he seems to be in dress and carries in his hand a small object like a knife or phone. Fig. 2 shows the approximate drawing of the given scenario.

P_1 is a security guard as he is in security uniform and standing close to the security barrier where normally security guards serve their duties. P_2 is suspicious as either a security guard or a terrorist because he is in security uniform and walking close to the security barrier as well as close to the outer wall used by terrorists to enter the building. P_3 is also suspicious as an employee of the office or a terrorist. He is present near the office which makes him eligible to be an employee of the office and there is also doubt that a terrorist may have crossed the outer wall adjacent to office building. There is an object or a person so called P_4 on the main road outside the office building. If he is a person, he may be a terrorist or a common man walking on the main road carrying in his hand a small object like knife or a phone. But if he is not a person, we are not interested to include it in our database. Because we are not sure as if he is a person or an object, we are also not certain whether this record can be part of our database or not. This indicates TLU. Due to this kind of uncertainty, we have annotated the tuple with symbol "?" in Table 1. Table 1 gives representation

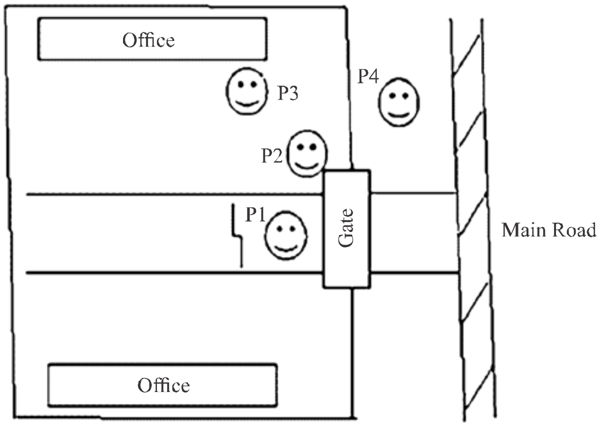


FIG. 2. MAP OF BUILDING ALONGWITH PERSONS

of the given example using or-set relation. We can compute the number of possible worlds of or-set relation i.e. 2*2*2*2*2*2 = 64 possible worlds.

UDM-relations are a succinct and intuitive data model for uncertain data. Our model is based on two steps: (i) it uses vertical partitioning approach that partitions an uncertain relation in such a way that every non-key attribute of an uncertain relation becomes an isolated relation known as UDM-relation. The key-attribute of uncertain relation is kept intact in each UDM-relation. Thus the key-attribute in an uncertain relation becomes the key-attribute in each of the UDM-relation (ii) Each distinct element (or entry) appearing in each attribute of uncertain relation now becomes the attribute of respective UDM-relation. The entries of tuples for each attribute of UDM-relation are represented with 1 or 0; where 1 represents the presence of an id in respective attribute and 0 represents its absence. For the purpose of query processing, we keep the key attribute of uncertain relation in each derived relation. The entries appearing in tuples of every attribute of derived relation are shown as either 1 or 0, where 1 indicates the presence of that tuple in respective attribute and 0 does indicate its absence. Table 3(a-d) show four UDM-relations that represent the given scenario. Our model uses an additional UDM-relation represented as tuple-or relation that deals with TLU. The given scenario depicts an uncertain database represented through four UDM-relations.

4. QUERY PROCESSING

An uncertain relation is a set of possible relations. Query processing on an uncertain relation means that a given query Q be evaluated against every possible world of that relation. In data models for uncertain data, the given query Q is evaluated in a way to make it possible for the data model to capture the results of query using its formalism [32]. In our query processing approach, we apply following operators i.e. filter operator (σ), project operator (π), combine operator ($\propto$) and deletion operator (^). These operators are used to evaluate a given query Q on UDM-relations. For the purpose of representing the results of a given query Q using the formalism of UDM-relations, we evaluate the given query using following steps.

(i) The filter operator is applied to select the tuples of key attribute from a UDM-relation based on the predicate given in query Q.

(ii) The project operator projects the attributes of a UDM-relation as given in the query.

(iii) Based on the common key attribute data, the combine operator is used to merge two UDM-relations or the intermediate resultant relations obtained after applying filter operator or project operator. This operator reconstructs the results of query into a relation.

(iv) After the reconstruction of attributes into a relation, the deletion operator replaces all entries of 0 appearing in the resultant relation with symbol '^'. This symbol indicates that the entry essentially be considered as deleted.

Now we evaluate a query (Q_1) given in example 4.1 on UDM-relations as given in Tables 3(a-d) using these operators.

Example 4.1. Q_1: Select arms and identities of persons whose uniform is dress.

Using relation algebra operations, Q_1 can be translated as: $\sigma_{\text{identity, arm}}$ (R) where uniform = "dress". This query results in identities and arms of those persons whose uniform is dress. But to evaluate this query on UDM-relations given in Table 3, we first apply the filter operator over UDM-relation P_uniform which results in tuples where 1 appears in U_dress attribute. The result of filter operator is shown in Table 4(a). On the resultant relation, we apply project operator to project the attributes of id and U_dress. The result of project operator is depicted in

Table 4 (b). On the basis of common key attribute i.e. Id, the combine operator merges the resultant relation with UDM-relations P_Identity, P_Arm and Tuple_or and the results are shown in Table 4(c-e). The deletion operator is applied on the resultant relation shown in Table 4(e) that replaces all 0,s with ⊥ and the result is given in Table 4(f). The resultant relation shown in Table 4(f) is self explanatory. It tells that the there are two persons with ids 3 and 4 whose uniform is dress. About their identities and arms it tells that person with id 3 is either a terrorist or an employee carrying with him either a pistol or a phone? About entry with id 4, there appears 1 in attribute Tu which shows that this tuple is uncertain too. The results of our query clearly depict the uncertainty as it is present in or-set relation as shown in Table 1.

TABLE 3(a). P_ARM RELATION

Id	A_gun	A_knife	A_pistol	A_stick	A_Pshone
P1	1	0	0	0	0
P2	0	1	0	1	0
P3	0	0	1	0	1
P4	0	1	0	0	1

TABLE 3(b). P_IDENTITY RELATION

Id	P_guard	P_terrorist	P_emp	P_com_man
P1	1	0	0	0
P2	1	1	0	0
P3	0	1	1	0
P4	0	1	0	1

TABLE 3(c). P_UNIFORM RELATION

Id	U_security	U_dress
P1	1	0
P2	1	0
P3	0	1
P4	0	1

TABLE 3(d). TUPLE_OR RELATION

Id	Tu
P1	0
P2	0
P3	0
P4	1

The standard heuristics of classical relational algebra in respect of query optimization are also applicable for UDM-relations. It is known from query processing for relational model that in a tree structure representing a query plan, all operations are evaluated one after the other starting from the leave nodes. So, in a tree representing a query plan on UDM-relations, we usually push down filter and project operators (to evaluate them before combine operator) in order to reduce the cost of reading and writing intermediate (temporary) relations. Conversely, if we apply the combine operator before filter and project operators in a query execution plan, the cost of reading and writing intermediate relations can be quite high. Fig. 3 shows three possible query execution plans. Of these three plans, L_1 clearly seems to be least efficient because in this plan the combine operator is evaluated before the filter operator. Because the combine operator is pushed down in plan L_1, it only increases the cost of reading and writing all four tuples in each intermediate relation and the filter operator being the last operation in this execution plan has nothing to do with the cost of reading and writing tuples in intermediate relations. While in query execution plans L_2 and L_3, the filter and project operators are pushed down (in order to evaluate both operators before the combine operator) and the combine operator is pushed up, both of these plans reduce the cost of reading and writing tuples in intermediate relations. In these plans, the combine operator is applied only on intermediate relations with filtered tuples (two tuples instead of four) and the relations which are part of query. However, without statistics it is difficult to state which one of the two plans (L_2 and L_3) should be preferred. Now we give some algebraic properties of these operators for UDM-relations.

4.1 Algebraic Properties of Filter, Project and Combine Operators for UDM-Relations

Following are the algebraic properties of filter, project and combine operators. It should be noted that the process of combining relations is the inverse of vertical partitioning and we can say that combining relations is commutative as well as associative. It commutes with filter and project operators. These algebraic properties show a kind of relationship in terms of operations between three operators. To illustrate, property 1 below states that say R and S are two relations and the application of combine operator is commutative i.e., if the filter operator is applied first on relation R and then combined with relation S or the filter operator is applied first on relation S and then combined with relation R, the result of evaluation in any order remains the same. Thus it can be said that the combine operator is commutative with filter operator. Other properties can be interpreted in a similar fashion.

TABLE 4(a). σU_dress=1 (P_Uniform)

Id	U_security	U_dress
3	0	1

TABLE 4(b). $\Pi_{\text{Id, dress}}$ (RELATION IN TABLE 4(a))

Id	U_dress
3	1

TABLE 4(c). P_Identity ∞_{Id} (RELATION IN TABLE 4(b))

Id	P_guard	P_terrorist	P_emp	P_com_man	U-dress
3	0	1	1	0	1
4	0	1	0	1	1

$$combine\ (S,\ filter_{\sigma}(R))=combine\ (R,\ filter_{\sigma}(S)) \quad (1)$$

$$combine\ (S,\ T,\ filter_{\sigma}(R))=combine\ (T,\ R,\ filter_{\sigma}(S)) \quad (2)$$

$$combine\ (\pi_{R*}(S),\ \pi_{R\ -\ R*}\ (S))=S$$
$$where\ R*\subseteq R\ and\ R=schema\ (S) \quad (3)$$

$$combine\ (combine\ (S,T),\ R)=combine\ (S,\ combine\ (T,R)) \quad (4)$$

$$\pi_{R*}\ (combine\ (S,T)=combine\ (\pi_{R*)"R}(S),\ \pi_{R*)"U}(T)) \quad (5)$$

$$where\ U=schema\ (T)\ and\ R=schema\ (S)$$
$$combine\ (S,T)\ \infty\ R=combine\ ((S\ \infty\ T),\ R) \quad (6)$$

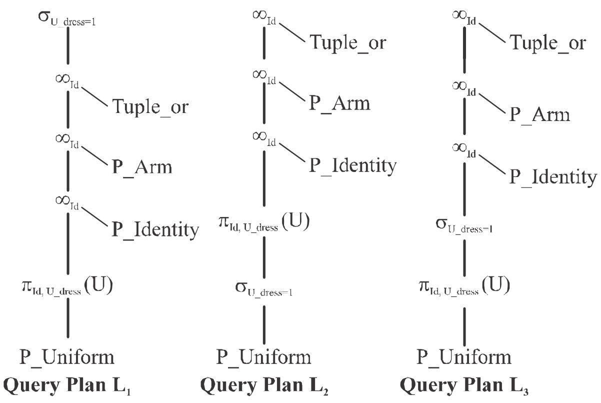


FIG. 3. THREE POSSIBLE QUERY PLANS

TABLE 4(d). RELATION IN TABLE 4(c) ∞_{Id} (P_Arm)

Id	P_guard	P-terrorist	P_emp	P_com_man	U_dress	A_gun	A_knite	A_pistol	A_stick	A_phone
3	0	1	1	0	1	0	0	1	0	1
4	0	1	0	1	1	0	1	0	0	1

TABLE 4(e). RELATION IN TABLE 4(d) ∞_{Id} (Tuple_or)

Id	P_guard	P_terrorist	P_emp	P_com_man	U_dress	A_gun	A_knite	A_pistol	A_stick	A_phone	Tu
3	0	1	1	0	1	0	0	1	0	1	0
4	0	1	0	1	1	0	1	0	0	1	1

TABLE 4(f). APPLYING DELETION OPERATOR (^) OVER RELATION SHOWN IN TABLE 4(e)

Id	P_guard	P_terrorist	P_emp	P_com_man	U_dress	A_gun	A_knite	A_pistol	A_stick	A_phone	Tu
3	^	1	1	^	1	^	^	1	^	1	^
4	^	1	^	1	1	^	1	^	^	1	1

5. CONCLUSION

This paper makes an important contribution towards the management of uncertain data. The relational data models developed so far for the management of uncertain data, are capable of dealing with attribute-level uncertainty only. We have shown the nature of uncertain data through a couple of examples: one for school or college database (Section 1) and the other one is a scenario of a terrorist attack (Section 3). The prominent feature of our model is that it supports ALU as well as TLU while employing in standard relational database technology. Another distinctive characteristic of our data model is that it also shows uncertainty in query results if it is found in UDM relations; no other data model studied in literature so far own this feature. We have introduced relational algebraic operators that provide a guideline in order to efficiently evaluate a standard relational query on UDM-relations. For future directions, we aim to extend our model for probabilistic information and thus develop PUDM, a probabilistic uncertain data model that is a customized data model for probabilistic data.

ACKNOWLEDGEMENT

Authors are highly grateful to Higher Education Commission (HEC), Pakistan, for extending their financial support for this project.

REFERENCES

[1] Zhang, W., Yue, K., and Liu, W., "Learning Uncertain Knowledge from Uncertain Data", Journal of Information and Computational Science, Volume 8, No. 6, pp. 933-940, Hong Kong, June, 2011.

[2] Yuan, W., Guan, D., Huh, E., and Lee, S., "Harness Human Sensor Networks for Situational Awareness in Disaster Reliefs: A Survey", IETE Technical Review, Volume 30, No. 3, pp. 240-247, India, September, 2013.

[3] Imieliñski, T., and Lipski, Jr, W., "Incomplete Information in Relational Databases", Journal of the ACM, Volume 31, No. 4, pp. 761-791, New York, USA, October,1984.

[4] Abiteboul, S., Kanellakis, P., and Grahne, G., "On the Representation and Querying of Sets of Possible Worlds", Theoretical Computer Science, Volume 78, No. 1, pp. 159-187, 1991.

[5] Lakshmanan, L.V., Leone, N., Ross, R., and Subrahmanian, V. S., "Probview: A Flexible Probabilistic Database System", ACM Transactions on Database Systems, Volume 22, No. 3, pp. 419-469, New York, USA, September, 1997.

[6] Jampani, R., Xu, F., Wu, M., Perez, L.L., Jermaine, C., and Haas, P.J., "MCDB: A Monte Carlo Approach to Managing Uncertain Data", Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 687-700, Vancouver, Canada, June, 2008.

[7] Cheng, R., Singh, S., and Prabhakar, S., "U-DBMS: A Database System for Managing Constantly-Evolving Data", Proceedings of 31st International Conference on Very Large Data Bases, pp. 1271-1274, Italy, October, 2005.

[8] Barbará, D., Garcia-Molina, H., and Porter, D., "The Management of Probabilistic Data", IEEE Transactions on Knowledge and Data Engineering, Volume 4, No. 5, pp. 487-502, USA, 1992.

[9] Deshpande, A., Guestrin, C., Madden, S.R., Hellerstein, J.M., and Hong, W., "Model-Driven Data Acquisition in Sensor Networks", Proceedings of 30th International Conference on Very Large Data Bases, Volume 30, pp. 588-599, VLDB Endowment, August, 2004.

[10] Folino, G., Shah, A.A., and Krasnogor, N., "On the Scalability of Multi-Criteria Protein Structure Comparison in the Grid ", Mehran University Research Journal of Engineering & Technology, Volume 32, No. 1, pp. 31-38, Jamshoro, Pakistan, January, 2013.

[11] Benjelloun, O., Sarma, A.D., Halevy, A., and Widom, J., "ULDBs: Databases with Uncertainty and Lineage", Proceedings of 32nd International Conference on Very Large Databases, pp. 953-964, September, 2006.

[12] Sen, P., and Deshpande, A., "Representing and Querying Correlated Tuples in Probabilistic Databases", 23rd IEEE International Conference on Data Engineering, pp. 596-605, April, 2007.

[13] Antova, L., Koch, C., and Olteanu, D., "From Complete to Incomplete Information and Back", Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 713-724, June, 2007.

[14] Das Sarma, A., Benjelloun, O., Halevy, A., and Widom, J., "Working Models for Uncertain Data", Proceedings of 22nd IEEE International Conference on Data Engineering, pp. 7-7, April, 2006.

[15] Chen, T., Chen, L., Ozsu, M.T., and Xiao, N., "Optimizing Multi-Top-k Queries over Uncertain Data Streams", IEEE Transaction on Knowledge and Data Engineering, Volume 25, No. 8, pp. 1814-1829, August, 2013.

[16] Cao, K., Han, D., Wang. G., Hu, Y., and Yuan, Y., "An Algorithm for Outlier Detection on Uncertain Data Stream", Web Technologies and Applications: Lecture Notes in Computer Science, Volume 7808, pp. 449-460, 2013.

[17] Lv, T., He, W., and Yan, P., "A Survey of Modelling Uncertain Data", Proceedings of 3rd IEEE International Conference on Computer Science and Information Technology, pp. 172-176, July, 2010.

[18] Galhardas, H., Florescu, D., Shasha, D., and Simon, E., "AJAX: An Extensible Data Cleaning Tool", ACM Sigmod Record, Volume 29, No. 2, pp. 590-590, May, 2000.

[19] Andritsos, P., Fuxman, A., and Miller, R.J., "Clean Answers Over Dirty Databases: A Probabilistic Approach", 22nd IEEE International Conference on Data Engineering, pp. 30-30, Atlanta, USA, April, 2006.

[20] Das Sarma, A., "Managing Uncertain Data", Ph.D. Dissertation, Stanford InfoLab, 2009.

[21] Bohannon, P., Fan, W., Flaster, M., and Rastogi, R., "A Cost-Based Model and Effective Heuristic for Repairing Constraints by Value Modification", Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 143-154, Baltimore, USA, June, 2005.

[22] Carpi, A., and Egger, A.E., "Data: Uncertainty, Error, and Confidence", Vision Learning, Volume 3, 2008.

[23] Motro, A., "Modern Database Systems", Chapter-22, ACM Press Addison-Wesley Publishing Company, New York, USA, 1995.

[24] Zhang, W., Lin, X., Pei, J., and Zhang, Y., "Managing Uncertain Data: Probabilistic Approaches", 9th IEEE International Conference on Web-Age Information Management, pp. 405-412, Zhangjiajie Hunan, July, 2008.

[25] Aggarwal, C.C., and Yu, P.S., "A Survey of Uncertain Data Algorithms and Applications", IEEE Transactions on Knowledge and Data Engineering, Volume. 21, No. 5, pp. 605-623, May, 2009.

[26] Idrees, M., Khan, M.U.G., and Shah, A., "Unified Data Model for Biological Data", Mehran University Research Journal of Engineering & Technology, Volume 33, No. 3, pp. 261-277, Jamshoro, Pakistan, July, 2014.

[27] Green, T.J., and Tannen, V., "Models for Incomplete and Probabilistic Information", Current Trends in Database Technology, EDBT, pp. 278-296, 2006.

[28] Bornholt, J., Mytkowicz T., and McKinley, K.S., "Uncertain: A First-Order Type for Uncertain Data", Proceedings of 19th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 220-29, Utah, USA, March, 2014.

[29] Antova, L., Jansen, T., Koch, C., and Olteanu, D., "Fast and Simple Relational Processing of Uncertain Data", 24th IEEE International Conference on Data Engineering, pp. 983-992, Mexico, April, 2008.

[30] Antova, L., Koch, C., and Olteanu, D., "$10^{10\text{^}6}$ Worlds and Beyond: Efficient Representation and Processing of Incomplete Information", International Journal on Very Large Databases, Volume 5, No. 5, pp. 1021-1040, 2009.

[31] Stonebraker, M., Abadi, D.J., Batkin, A., Chen, X., Cherniack, M., Ferreira, M., and Zdonik, S., "C-Store: A Column-Oriented DBMS", Proceedings of 31st International Conference on Very Large Data Bases, pp. 553-564, Italy, October, 2005.

[32] Calì, A., Lembo, D., and Rosati, R., "On the Decidability and Complexity of Query Answering Over Inconsistent and Incomplete Databases", Proceedings of 22nd ACM SIGMOD-SIGACT-SIGART, PODS, pp. 260-271, California, USA, June, 2003.

[33] Yi, K., Li, F., Kollios, G., and Srivastava, D., "Efficient Processing of Top-k Queries in Uncertain Databases", 24th IEEE International Conference on Data Engineering, pp. 1406-1408, Maxico, April, 2008.