

A Probabilistic Data Model and Its Semantics

Shichao Zhang and Chengqi Zhang

Faculty of Information Technology
University of Technology, Sydney
PO Box 123, Broadway, Sydney, NSW 2007, Australia
{zhangsc, chengqi}@it.uts.edu.au

As database systems are increasingly being used in advanced applications, it is becoming common that data in these applications contain some elements of uncertainty. These arise from many factors, such as measurement errors and cognitive errors. As such, many researchers have focused on defining comprehensive uncertainty data models of uncertainty database systems. However, existing uncertainty data models do not adequately support some applications. Moreover, very few works address uncertainty tuple calculus. In this paper we advocate a probabilistic data model for representing uncertain information. In particular, we establish a probabilistic tuple calculus language and its semantics to meet the corresponding probabilistic relational algebra.

1. INTRODUCTION

Today's database systems must handle uncertainties in the data they store. Such uncertainties arise from different sources such as measurement errors, cognitive errors, approximation errors, calculation errors, the dynamic nature of real world, and partially unknown environment. For example, in an image retrieval system, an image processing algorithm may fetch images that are similar to a given sample image, and feed the results into a relational database (Lakshmanan, 1997). The results are generally uncertain. In a sensor application, depending on the reliability of the sensor, the data from the sensor would be associated with a probability. In temporal database, temporal information may be fuzzy due to the uncertainty of the dynamic nature of the real world. Finally, uncertainty arises in information retrieval, where the research community has long used probabilistic techniques for retrieval of document data based on "concepts".

In order to perform anything useful, these data (with uncertainties) must be efficiently modelled and stored in databases so that they can be subsequently accessed and used. To handle data with uncertainties, many fundamental issues and challenges have to be re-examined. Some of these include defining comprehensive uncertainty data models, semantic modelling of uncertainty data, choosing a better theory on uncertainty, integrity and consistence constrains, safety under uncertainty, establishing appropriate uncertainty relational calculus and algebra, integrating uncertainty models, and implementation issues. Among these issues, defining comprehensive uncertainty data models has attracted much research activity.

In the literature, there are two approaches to deal with uncertainty: probabilistic data models (Cavallo, 1987; Dey, 1996; Fuhr, 1997; Pittarelli, 1994) and fuzzy data models (Buckles, 1983;

Copyright© 2003, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: 19 September 2000
Communicating Editor: Bernhard Thalheim

Buckles, 1984; Prade, 1984; Zemankova, 1985). The more direct method of dealing with uncertainty would be to specifically use a probabilistic data model because of the availability of a very rich probability theory. So, our work in this paper concentrates on probabilistic data models.

From the systems point of view, model and management of uncertainty data must have considerably more functionality and capability than the conventional database management systems, due to the complicated structure of the data. In order to capture the general semantics of uncertainty data, we propose a probabilistic data model for representing such uncertain information in this paper. The model is based on PRM (Dey, 1996), and extends several operators to handle uncertainty data using axiom systems in probability theory. Furthermore, we establish a probabilistic tuple calculus to meet the corresponding probabilistic relational algebra. Before describing our probabilistic data model in detail, we simply recall the classification of the uncertainty of data and the related work in the current literature.

We begin in Section 2 with simple recall previous work on probabilistic databases. In Section 3, we will define some basic concepts of probability databases, which will be used throughout the subsequent sections. In Section 4, we establish a probability tuple calculus language (*IPTRC*). In Section 5, we present a probability relational algebra. In Section 6, we will illustrate how to take probabilistic relational algebra as a query language by some simple examples. Finally, in the last section, we compare our model with the PRM model, and summarise the contribution of this paper.

2. RELATED WORK

A number of attempts have been made to develop fuzzy and probability data models for representing uncertainty data (Dey, 1996; Fuhr, 1997; Pittarelli, 1994). Most of them are of only limited success (a general review is in Dey, 1996). Notable work includes the probabilistic relational model (PRM) by Dey and Sarkar (1996), ProbView: a flexible probabilistic database system by Lakshmanan, Leone, Ross and Subrahmanian (1997), probabilistic deductive databases by Lakshmanan and Sadri (1994), incomplete information in relational databases by Imielinski and Lipski (1984), stable semantics for probabilistic deductive databases by Ng and Subrahmanian (1995), combining deduction by uncertainty with the power of magic by Schmidt, Kiessling, Guntzer and Bayer (1987), the management of probabilistic data (MPD) by Barbara, Garcia-Molina and Porter (1992), information-theoretical characterisation of fuzzy relational databases (FRD) by Buckles and Petry (1983), extending the fuzzy database with fuzzy numbers (EFD) by Buckles and Petry (1984), the theory of probabilistic database (PD) by Cavallo and Pittarelli (1987), a probabilistic relational algebra for the integration of information retrieval and database Systems (PRA) by Fuhr and Rolleke (1997), incomplete information costs and database design (IIC) by Mendelson and Saharia (1986), an algebra for probabilistic databases (APD) by Pittarelli (1994), generalising database relational algebra for the treatment of incomplete or uncertain information and vague queries by Prade and Testemale (1984), fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems by Raju and Majumdar (1988), answering heterogeneous database queries with degrees of uncertainty by Tseng, Chen and Yang (1993), a statistical approach to incomplete information in database systems (IIDS) by Wong (1982), and implementing imprecision in information systems by Zemankova and Kandel (1985). This current research has explored the representation of incomplete and uncertain data in a suitable structure that is powerful enough to develop higher-level semantics and uncertainty event abstractions. An excellent summary of the research issues of uncertainty databases is described in the paper by Dey and Sarkar (1996). They fall into four different categories:

- The first category deals with extending the relational algebra to handle “nul” values and provides the semantics of “null” values. These representations using “null” values assume that

data values are either known with certainty or they are unknown. This assumption, however, is too restrictive to model most real applications.

- The second category deals with the uncertainty involved in the retrieval of incomplete data and the cost of data incompleteness (Mendelson, 1986; Wong 1982). These models are useful tools for deciding on desired levels of data storage, but are not adequate for the representation of data uncertainty.
- The third category models data uncertainty using fuzzy set theory (Buckles, 1983; Buckles, 1984; Prade, 1984; Zemankova, 1985). The typical assumption in these models is that some attributes (or data items) do not have precise values; rather, they take on “fuzzy” values. These models can be used to model the fuzzy type of uncertainty data.
- The fourth category of work uses the well-known probability calculus (Cavallo, 1987; Dey, 1996; Fuhr, 1997; Pittarelli, 1994). These investigations extend the relational model to represent uncertainty due to ambiguity using the well-known probability calculus. They provide a framework for handling the probabilistic type of uncertainty data.

As we have seen, the current researches on uncertainty databases are based on relational model due to the successful application of this model and the solid set theory. They are mainly focused on defining the uncertainty data model and uncertainty relational algebra. However, only a little work is on uncertainty query languages. One of our main contributions in this paper is to suggest a probability tuple calculus language and its semantics.

There are various classes of models for representing uncertainty data in the current uncertainty databases, but for the purposes of this discussion we classify these representations into two broad categories: attribute uncertainty and tuple uncertainty.

The attribute uncertainty means that some attribute-values of an object are associated with uncertainty. These models need usually to adapt a non-1NF (or N1NF) view of uncertainty relations (Lakshmanan, 1997). Data in N1NF models has better intuition than in 1NF models. N1NF models can provide a framework for describing the nature of uncertainty data. However, since N1NF models use set-valued attributes, their model poses the usual implementation problem associated with all N1NF relations (Dey, 1996).

The tuple uncertainty means a tuple is associated with a degree of member belong to a relation. These models adopt a 1NF view of uncertainty relations. The operators for handling uncertainty data can be naturally extended on top of traditional relational database models. In other words, these models are implemented easier than N1NF models. However, the information of an event is usually represented in several tuples in 1NF relations. That is, data in these models do not have good intuition. For implementation, our work in this paper concentrates on tuple uncertainty.

2.1 Concepts

Let $N = \{1, 2, \dots, n\}$ be an arbitrary set of integers. A relation scheme R is a set of attribute names $\{A_1, A_2, \dots, A_n\}$, one of which may be a probability stamp pS . Corresponding to each attribute name A_i , $i \in N$, is a set D_i called the domain of A_i . If $A_i = pS$, then $D_i = (0, 1)$. The multiset $D = \{D_1, D_2, \dots, D_n\}$ is called the domain of R . A tuple x over R is a function from R to D ($x: R \rightarrow D$), such that $x(A_i) \in D_i$, $i \in N$. In other words, a tuple x is a tuple on scheme R . Restriction of a tuple x over S , $S \subset R$, written $x(S)$, is the sub-tuple containing values for attribute names in S only, i.e., $x(S) = \{ \langle A, v \rangle \in x \mid A \in S \}$.

A formal interpretation of a tuple is: a tuple x over R represents our belief about attributes (in R) of a real world object. If $pS \in R$, then we assign a probability of $x(pS) > 0$ to the fact that an object has the values $x(R - \{pS\})$ for the corresponding attributes. Symbolically,

$$x(pS) = \Pr[R - \{pS\} = x(R - \{pS\})]$$

If $pS \notin R$; i.e., if the relation scheme R is deterministic, then every tuple on R is assigned a probability of one, and is not explicitly written. However, if x is a tuple on the scheme R , and $pS \notin R$, it will be implicitly assumed that $x(pS) = 1$.

Two tuples x and y on relation scheme R are value-equivalent (written $x \equiv y$) if and only if, for all $A \in R$, $(A \neq pS) \Rightarrow (y(A) = x(A))$. Value-equivalent tuples are not allowed in a relation; they must be coalesced. Two types of coalescence operations on value-equivalent tuples are defined as follows:

1. The coalescence-PLUS operation is used in the definition of the projection operation. Coalescence-PLUS (denoted by ‘ \oplus ’) on two value-equivalent tuples x and y is defined as:

$$z = x \oplus y \Leftrightarrow (x \equiv y) \wedge (z \equiv x) \wedge (z(pS) = \text{Min}\{1, x(pS) + y(pS)\})$$

2. The coalescence-MAX operation is used in the definition of the union operation. Coalescence-MAX (denoted by ‘ \otimes ’) on two value-equivalent tuples x and y is defined as:

$$z = x \otimes y \Leftrightarrow (x \equiv y) \wedge (z \equiv x) \wedge (z(pS) = \text{Max}\{x(pS), y(pS)\})$$

Intra-Relation Integrity Constrains: Let r be any relation on scheme R with primary key K . The following intra-relational constrains are imposed on r :

1. The total probability associated with a primary key value must be no more than one. In other words, for all $x \in r$,

$$\sum_{y \in r, y(K)=x(K)} y(pS) \leq 1$$

Since it has been implicitly assumed that the probability value for a tuple on a deterministic scheme is always unity (the deterministic assumption), the above constraint reduces to key uniqueness for each tuple when deterministic relations are considered.

2. For all $x \in r$, no part of $x(K)$ can be null.
3. For all $x \in r$, if $pS \in R$, then $x(pS) \in (0, 1)$ and $x(pS)$ is not null.

Referential Integrity Constrains: Let r and s be two relations on schemes R and S respectively. Let K_R and K_S be the primary keys of R and S , and let $r:F \rightarrow s.K_S$ for some $F \supset R$. The following referential constraints are imposed on r and s :

1. For all $x \in r$, if there exists an attribute $A \in F$ such that $x(A)$ is null, then for all other attributes $B \in F$, $x(B)$ is also null. This ensures that the foreign key value of a tuple is not partially null.
2. For all $x \in r$, either $x(F)$ is null (fully), or there exists $y \in s$ such that

$$\sum_{z \in r, z(K_R F)=x(K_R F)} z(pS) \leq \sum_{z \in s, z(K_S F)=x(K_S F)} z(pS)$$

where $K_R F$ is a shorthand for $K_R \cup F$. This ensures that the probability assigned for a set of attributes must be consistent with the probability of the object that these attributes refer to.

In Dey (1996), a new operation called conditionalisation is used to derive conditional joint distribution of different attributes given other attributes.

2.2 Limitations in Previous Models

While previous models, such as the PRM model, capture uncertainty, they have some definite limitations. First, there is no corresponding probabilistic tuple calculus presented. Moreover, the

uncertainty semantics of some operators of the probabilistic relational algebra, such as Coalescence-PLUS, fuzzy-equal, and relational operations, are neither captured at their definitions, nor in a fully satisfactory fashion. We shall illustrate the limitations using some examples as follows.

First, one of the Intra-Relation Integrity Constrains is not reflected in the extended operations in PRM model. Let r be any relation on scheme R with primary key K . The total probability associated with a primary key value must be no more than one. This constraint is not considered in the extended relational operators in PRM. Now we illustrate this by the following examples.

Relation: EMP

EMP#	DEPT	pS
3025	SHOE	0.6
3025	TOY	0.3
6637	TOY	0.8
6637	AUTO	0.2

Relation: EMP'

EMP#	DEPT	pS
3025	SHOE	0.8
3025	TOY	0.1
6637	TOY	0.3
6637	SHOE	0.5

The union of relations EMP and EMP' obtained according to the union operation in Dey (1996) is as follows.

Relation: $EMP \cup EMP'$

3025	SHOE	0.8
3025	TOY	
6637	TOY	0.8
6637	AUTO	0.2
6637	SHOE	0.5

Certainly, relation $EMP \cup EMP'$ does not enforce the integrity constrains on probabilistic relations. In our opinion, it is important to enforce this constraint in the extended relational operators for probability databases. In our probability relational operators, we will give a consideration to the constraint.

Second, the definition of coalescence-PLUS operation is inadequate to match the properties of probability. The last limitation is that traditional equal relationship ($=$) used in PRM model cannot fit the equal relationship under uncertainty. Indeed, because the user may not know exactly what the probabilities of the tuples are, the queries to probabilities of the tuples in a probabilistic database should be based on approximate matching rather than exact matching. In other words, traditional equal relationship cannot fit these applications. We can easily illustrate them by examples as similar as that used in the first limitation.

3. THE PROBABILISTIC DATA MODEL

In this section, we introduced the probabilistic data model that we used. We use upper case letters such as A, B, \dots to represent attributes, and $\text{dom}(A), \text{dom}(B), \dots$ represent the domains of A, B, \dots , respectively. Lower case letters such as a, b, \dots denote the values of each domain. A special attribute pS will denote the probability attribute and its domain $\text{dom}(pS) = [0, 1]$. $\mu, \nu, \mu_1, \nu_1, \dots$ will represent the tuples of a probabilistic relation.

Definition 1. Let $A \neq pS$ be an attribute. $\text{dom}(A)$ the domain of A is a finite set of atomic values, for which the predicates $=$ and \neq are defined. $\text{dom}(A)$ is an ordered domain if there is one relation of $<$, \leq , \geq and $>$ over $\text{dom}(A)$.

Definition 2. For $\text{dom}(pS) = [0, 1]$ and $\forall a, b \in \text{dom}(pS)$, a ε -equal to b if $|a - b| < \varepsilon$, where $\varepsilon (> 0)$ given by experts is small enough. a ε -equal to b will be denoted as $a =_{\varepsilon} b$.

Generally, the domain of an attribute has to be finite. However, $(0, 1)$ is an infinite set of numbers. So, it is inadequate to describe fuzzy-equal by using “ $=$ ”. Intuitively, we can divide $\text{dom}(pS) = [0, 1]$ into finite or countable sub-intervals using relation $=_{\varepsilon}$ in Definition 2, and each sub-interval is taken as a atomic value. In this way, we can model the approximating and infinite semantics of uncertainty data using relational data model.

For example, let $\varepsilon = 0.0001$, $a_1 = 0.5$, $a_2 = 0.50001$, $a_3 = 0.34564$, $a_4 = 0.34563$. We have

$$|a_1 - a_2| = 0.00001 < \varepsilon \text{ and } |a_3 - a_4| = 0.00001 < \varepsilon$$

That is, $a_1 =_{\varepsilon} a_2$ and $a_3 =_{\varepsilon} a_4$. Note that “ $=_{\varepsilon}$ ” relation does not satisfy transmission law.

Definition 3. Let A be an attribute and $\text{dom}(A)$ the domain of A . A probabilistic assignment P of A is a function from $\text{dom}(A)$ to $[0, 1]$ so that for every $x \in \text{dom}(A)$, $P(A = x) \in [0, 1]$. We use $(x, P(A = x))$ to denote a probabilistic assignment of A . $A = x$ is a certainty if $P(A = x) = 1$ and $P(A = a) = 0$ for $\forall a \in \text{dom}(A)$, $a \neq x$.

Definition 4. Let A be an attribute, $P_1(A = x)$ and $P_2(A = x)$ be two observations of $A = x$. $(x, P_1(A = x))$ is ε -equal to $(x, P_2(A = x))$ if

$$|P_1(A = x) - P_2(A = x)| < \varepsilon$$

where $\varepsilon > 0$ is small enough. $(x, P_1(A = x))$ ε -equal to $(x, P_2(A = x))$ is written as $(x, P_1(A = x)) =_{\varepsilon} (x, P_2(A = x))$.

For example, let A be an attribute, $\varepsilon = 0.0001$, $(x, P_1(A = x)) = (x, 0.8)$, $(x, P_2(A = x)) = (x, 0.79999)$, $(x, P_3(A = x)) = (x, 0.34564)$, $(x, P_4(A = x)) = (x, 0.34565)$. We have

$$|P_1(A = x) - P_2(A = x)| = 0.00001 < \varepsilon \text{ and } |P_3(A = x) - P_4(A = x)| = 0.00001 < \varepsilon$$

That is, $(x, P_1(A = x)) =_{\varepsilon} (x, P_2(A = x))$ and $(x, P_3(A = x)) =_{\varepsilon} (x, P_4(A = x))$.

Definition 5. Let $U = \{A_1, A_2, \dots, A_n\}$ be a set of attributes. The attribute domain $\text{dom}(U)$ of U is as $\text{dom}(U) = \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$. A tuple over U is a mapping $\mu: U \rightarrow \text{dom}(U)$ such that

$$\forall X \in U (\mu(X) \in \text{dom}(X))$$

Let μ be a tuple over U and $Y = \{Y_1, Y_2, \dots, Y_m\}$ a subset of U . We will denote the projection of μ onto Y as $\mu(Y) = (\mu(Y_1), \mu(Y_2), \dots, \mu(Y_m))$.

Definition 6. Let $U = \{A_1, A_2, \dots, A_n\}$ be a set of attributes, D a subset of U . A relation over relational scheme D is a set of tuples over D . A database over database scheme U is a set of relations.

Furthermore, we can define probabilistic relation scheme, probabilistic relation, probabilistic database, and the ε -equal relationship between tuples as below.

Definition 7. A probabilistic relation scheme R is a set of attributes as $\{A_1, A_2, \dots, A_n\}$, one of which may be a probability stamp pS . A tuple over R is a mapping $\gamma: R \rightarrow \text{dom}(R)$ such that

$$\forall X \in R (\gamma(X) \in \text{dom}(X) \wedge (\gamma(pS) \in (0, 1)))$$

A relation over the probabilistic relation scheme R is a set of tuples over D . This relation is called probabilistic relation.

Let r be a relation over the probabilistic relation scheme R and $v \in r$. v is a certainty if $v(pS) = 1$. And r is a certainty if μ is a certainty for $\forall \mu \in r$. Apparently, we can view conventional relations as the specific instances of probabilistic relations.

Definition 8. Let r be a relation over the probabilistic relation scheme R , $\mu, v \in r$ two tuples. μ and v are value-equivalent (written $\mu \equiv v$) if and only if, for all $A \in R$, $(A \neq pS) \Rightarrow (v(A) = \mu(A))$.

Value-equivalent tuples are not allowed in a relation; they must be coalesced.

Definition 9. Let r be a relation over the probabilistic relation scheme R , $\mu, v \in r$ two tuples. μ is ε -equal to v if $\mu \equiv v$, and $|\mu(pS) - v(pS)| < \varepsilon$, where $\varepsilon > 0$ is small enough. μ ε -equal to v is written as $\mu =_{\varepsilon} v$.

The ε -neighbor of μ is defined as: $\{v \mid v \in r \wedge v =_{\varepsilon} \mu\}$, denoted as μ_{ε} . For the sake of convenience, we suppose $\mu_{\varepsilon} = \{\mu\}$ in this article.

Definition 10. A probabilistic database is a set of the probabilistic relations.

Now we can strictly define the semantics of probabilistic relation, λ -relation as following.

Definition 11. Let R be a probabilistic relation scheme. Given $K \subseteq R$ is a set of primary keys of R . The probabilistic relation r over R is a finite set of non-null tuples over R such that:

1. for each tuple μ in r and $A \in R$, $\mu(A)$ is single-valued;
2. for $\forall \mu, v \in r$, if μ and v are two different tuples in r , then μ not $\equiv v$.

Usually, the relation of satisfying the above two conditions is in probabilistic **First Normal Form** (1NF). The semantics of a probabilistic relation r over the probabilistic relation scheme R is as the same in PRM model.

Definition 12. Let R be a probabilistic relation scheme, r_1 and r_2 two probabilistic relations over R . r_1 ε -equal to r_2 if

1. for $\forall \mu \in r_1, \exists v \in r_2$ such that $\mu =_{\varepsilon} v$; and
 2. for $\forall \mu \in r_2, \exists v \in r_1$ such that $\mu =_{\varepsilon} v$.
- r_1 ε -equal to r_2 is written as $r_1 =_{\varepsilon} r_2$.

Definition 13. Let R be a probabilistic relation scheme, r a probabilistic relation over R . A λ -cut $\lambda(r)$ of r is defined as

1. $\lambda(r)$ is a relation over scheme $R - \{pS\}$; and
2. for $\forall \mu \in r_{\lambda}, \exists v \in r$ such that $\mu = v(R - \{pS\}) \wedge (v(pS) \geq \lambda)$. Note that if $\exists v_1, v_2, \dots, v_m \in r$ and, $v_1(pS) \geq \lambda, v_2(pS) \geq \lambda, \dots, v_m(pS) \geq \lambda$, then we take only the tuple $\gamma(R - \{pS\})$ as a tuple of $\lambda(r)$, where $\gamma(pS)$ is the largest one among $v_1(pS), v_2(pS), \dots, v_m(pS)$.

4. PROBABILISTIC TUPLE CALCULUS LANGUAGE

Most of the current work has focused largely on uncertainty relational algebra. To enrich the current work on uncertainty databases, we propose a probabilistic tuple relation calculus language, called *IPTRC* in this section. First, we will give the well-formed formulae of the language, followed by their interpretations and examples.

4.1 Symbols

- Predicates. Predicates are the same as that in well-known logics. We will use P, Q, P_1, Q_1, \dots to represent predicates, which are also used for a probabilistic relation instance in a probabilistic database. Let $PSET$ be the set of all predicates.
- Variables. Let $\mu, \nu, \mu_1, \nu_1, \dots$ represent tuple variables. A variable has the same scheme and degree (arity) as the probabilistic relation scheme it is associated with. Variables may be indexed. If μ is a variable, then $\mu(i)$ is an indexed variable where i is between 1 and the arity of μ . $\mu(i)$ must be an atom. Let $VSET$ be the set of all variables.
- Constants. We will use a, b, c, \dots to represent the constant symbols. Each constant has a scheme, an atom. Let $CSET$ be the set of all constants.

4.2 Well-Formed Formulae

We define an expression of 1NF probabilistic tuple relation calculus (*IPTRC*) as $\{\mu \mid \varphi(\mu) \wedge \kappa(\mu(pS))\}$, where μ is a tuple variable; φ is a formula, which is similar to that used in traditional tuple calculus language; $\kappa(\mu(pS))$ is a standardising-probability operator defined in Subsection 5.2. We define well-formed formula φ in the following after we define atomic formulae.

Definition 14. An atomic formula is one of the following.

1. $P(\nu)$ is an atomic formula, where predicate P is a probabilistic relation name. $P(\nu)$ means $\nu \in P$, or ν is a tuple of P , or ν is an element of P .
2. $\nu = \mu(i)$ is an atomic formula, where ν is a attribute value of the projection $\mu(i)$ of tuple μ over the i th attribute and this attribute is not pS .
3. $\nu =_{\varepsilon} \mu(pS)$ is an atomic formula, where ν is the probability of tuple μ .
4. $a \theta \nu(i), \nu(i) \theta a$, or $\nu(i) \theta \mu(i)$ are atomic formulae, where a is a constant, and θ is an arithmetic comparator ($=, >$), and the i th attribute is not pS .
5. $a \theta \nu(pS), \nu(pS) \theta a$, or $\nu(pS) \theta \mu(pS)$ are atomic formulae, where $a \in [0, 1]$, and θ is an relational comparator ($=_{\varepsilon}, >$).
6. $\nu(i) = \{\nu \mid \varphi'(\nu)\}$ is an atomic formulae, φ' is a formula with a variable ν .

Definition 15. We define a well-formed formulae φ recursively as follows:

1. each atomic formula is a formula;
2. if φ_1 and φ_2 are formulae, then $\varphi_1 \wedge \varphi_2, \varphi_1 \vee \varphi_2$, and $\neg \varphi_1$ are formulae;
3. if φ is a formula, then $\forall \nu(\varphi), \exists \nu(\varphi)$ are formulae.

Note that *IPTRC* is allowed to use $\{\nu \mid \neg(\nu \in r)\}$. Therefore, *IPTRC* must satisfy the safe constraint conditions. The above definition of φ is the same as that in conventional tuple calculus languages except that $=_{\varepsilon}$ is substituted for $=$. In the approximating sense, the role of $=_{\varepsilon}$ is the same

as =. So, the safe constraint conditions of *IPTRC* are the same as ones of conventional tuple calculus languages. And only difference is that $=_{\varepsilon}$ is substituted for = when the object is pS .

4.3 Semantics of *IPTRC*

Let $W = 2^U$ be the power set of U , $\text{dom}(W) = \{a \mid a \in \text{dom}(A) \wedge A \in W\}$.

Definition 16. An *interpretation* of *IPTRC* is a 4-tuple as $I = \langle U, \text{dom}(W), rR, MM \rangle$, where,

1. U is the universe of *IPTRC*;
2. $\text{dom}(W)$ is the domain of the semantics of *IPTRC*;
3. rR is the set of probabilistic relations over U ;
4. MM is a binary tuple as $MM = \langle CM, PM \rangle$, where,

$CM: CSET \mapsto \text{dom}(W)$;

$PM: PSET \mapsto rR$.

where CM is a function that assigns a constant in $\text{dom}(W)$ to each constant in $CSET$, and PM is a function which assigns a probabilistic relation over U for each predicate of *IPTRC*.

For simplicity, the domain of interpretation for a calculus object ϕ is defined relative to the set $\text{dom}(W)$, universe of atoms, and is denoted by $\text{dom}_{\phi}(U)$. Atoms take their values from $\text{dom}(W)$.

An interpretation of a constant $a \in CSET$, denoted as I_a , is a member of $\text{dom}_a(U)$, where $\text{dom}_a(U) = \text{dom}(W)$. An interpretation of a predicate $P \in PSET$, denoted as I_p , is a relation instance, and $I_p \in \text{dom}_p(U)$. A variable μ is interpreted as a tuple instance, and $I_{\mu} \in \text{dom}_{\mu}(U)$, where $\text{dom}_{\mu}(U) = L_1 \times L_2 \times \dots \times L_n$ and n is the degree of μ . $I_{\mu}(i)$ denotes the i th component of the tuple that is the interpretation of variable μ . For well-formed formulae in *IPTRC*, they are interpreted as true or false by assigning interpretations to their constants, predicate symbols and free variables.

The following are the rules for the interpretation of formulae in *IPTRC*.

1. $P(v)$ is true if $I_v \in I_p$.
2. $v = \mu(i)$ is true if $I_v = I_{\mu}(i)$.
3. $v =_{\varepsilon} \mu(pS)$ is true if $|I_v - I_{\mu}(pS)| < \varepsilon$.
4. $a \theta v(i)$ is true if $I_a \theta I_v(i)$; $v(i) \theta a$ is true if $I_v(i) \theta I_a$; or $v(i) \theta \mu(i)$ is true if $I_v(i) \theta I_{\mu}(i)$;
where $(\theta \in \{=, >\})$.
5. $a \theta v(pS)$ is true if $|I_a - I_v(pS)| < \varepsilon$; $v(pS) \theta a$ is true if $|I_v(pS) - I_a| < \varepsilon$; or $v(pS) \theta \mu(pS)$ is true if $|I_v(pS) - I_{\mu}(pS)| < \varepsilon$; Where $(\theta \in \{=_{\varepsilon}, >\})$.
6. $\phi_1 \wedge \phi_2$ is true if ϕ_1 and ϕ_2 are true; $\phi_1 \vee \phi_2$ is true if ϕ_1 or ϕ_2 is true; $\neg \phi_1$ is true if ϕ_1 is false.
7. $\exists v(\phi)$ is true if there is at least one assignment to v which makes $\phi(v)$ true, i.e., $\phi(v)$ is true for at least one value of I_v . $\forall v(\phi)$ is true if $\phi(v)$ is true for any assignment to v .
8. $v(i) = \{\mu \mid \phi'(\mu)\}$ is satisfied (made true) by the interpretations I_{μ} , I_v of its free variable if the following condition is met: $I_v(i)$ equals the set of assignments I_{μ} satisfying $\phi'(\mu)$ for the interpretation I_{μ} . If there are no such tuples I_{μ} , and $I_v(i)$ is empty, then this formula evaluates to false. In other words, the set constructor formula does not create an empty set.

9. $\kappa(\mu(pS))$ is satisfied by the interpretations I_μ , if

$$\sum_{v \in r, v(K) = \mu(K)} I_v(pS) \leq 1$$

holds, where K is primary key.

An *IPTRC* expression $\{\mu \mid \varphi(\mu)\}$ where μ is a free variable with arity k and $\varphi(\mu)$ is a well-formed formula. An interpretation of this expression is the set of instances of μ that satisfy the formula $\varphi(\mu)$, i.e., an element of $\text{dom}_\mu(U)$.

Tuple relation calculus language is mainly used for describing properties of query language. We now illustrate the use of *IPTRC* by examples.

Example 1. For example, let $R = \{\text{Number, Name, Ring, } pS\}$, and *SHOOT* is a relation over scheme R as follows.

SHOOT: The scores of shooter team

Number	Name	Ring	pS
2001	John	10	0.4
2001	John	9	0.5
2001	John	8	0.08
2001	John	7	0.02
2002	Allen	10	0.6
2002	Allen	9	0.3
2002	Allen	8	0.1
2003	Li	10	0.1
2003	Li	9	0.3
2003	Li	8	0.5
2003	Li	7	0.1
2004	Tom	9	0.1
2004	Tom	8	0.3
2004	Tom	7	0.6

Q1: What are the names, rings and probabilities of those shooters in the shooter team that the probability of the ten-point ring is great than 0.3?

The *IPTRC* expression of Q1 is as follows

$$\{\mu \mid (\exists v) (\text{SHOOT}(v) \wedge (v(\text{Ring}) = 10 \wedge v(pS) > 0.3) \wedge (\mu(1) = v(\text{Name}) \wedge \mu(2) = v(\text{Ring}) \wedge \mu(3) = v(pS)))\}$$

The answer to Q1 is as follows.

Name	Ring	pS
John	10	0.4
Allen	10	0.6

Q2: What is the information of Allen in the score table?

The *IPTRC* expression of Q2 is as follows

$$\{\mu \mid (\exists v) (\text{SHOOT}(v) \wedge (v(\text{Name}) = \text{'Allen'}) \wedge (\mu = v))\}$$

The answer to Q2 is as follows.

Number	Name	Ring	pS
2002	Allen	10	0.6
2002	Allen	9	0.3
2002	Allen	8	0.1

Q3: What are the numbers and names of those shooters in the shooter team whose the probability of the nine-point ring is better than Tom's?

The *IPTRC* expression of Q3 is as follows

$$\{\mu \mid (\exists v) (\text{SHOOT}(v) \wedge (\exists v_1) (\text{SHOOT}(v_1) \wedge v_1(\text{Name}) = \text{'Tom'} \wedge (v(\text{Ring}) = 9 \wedge v(pS) > v_1(pS)) \wedge (\mu(1) = v(\text{Number}) \wedge \mu(2) = v(\text{Name}))))\}$$

The answer to Q3 is as follows.

Number	Name
2001	John
2002	Allen
2003	Li

5. PROBABILISTIC RELATIONAL ALGEBRA

There is a lot of research on probabilistic relational algebras in the current literature. In order to enrich the work, we define some considerable operators for handling data with uncertainty in probabilistic relational algebras using axiom systems in probability theory in this section. First, we will give the definitions of these operators, and then we will give a probabilistic relational algebra and its semantics that consider these operators.

5.1 Several Operators

Let r be a probabilistic relation on scheme R with primary key K . Value-equivalent tuples are not allowed in a probabilistic relation (Dey, 1996). They must be coalesced. We define two types of coalescence operations on value-equivalent tuples.

1. The coalescence-PLUS operation is used in the definition of the projection operation. Coalescence-PLUS (denoted by \oplus) on two value-equivalent tuples μ_1 and μ_2 is defined as:

$$v = \mu_1 \oplus \mu_2 \Leftrightarrow (\mu_1 \cong \mu_2) \dot{\vee} (v \cong \mu_1) \wedge (v(pS) = \mu_1(pS) + \mu_2(pS) - \mu_1(pS) * \mu_2(pS))$$

This definition can meet the probabilistic significance levels between events. For example, to solve the projection operation on SHOOT over $A = \{\text{Ring}, pS\} \subset R$ according to the above coalescence-PLUS, $\text{SHOOT}_1 = \prod_A(r)$ is as follows.

SHOOT1: The statistical results of shooter team

Ring	pS
10	0.784
9	0.7795
8	0.68409
7	0.6472

By the requirements of Intra-Relation Integrity Constrains, we have

SHOOT2: The statistical results of shooter team

Ring	pS
10	0.27083139
9	0.26927687
8	0.23631766
7	0.22357408

- The coalescence-MAX operation is used in the definition of the union operation. Coalescence-MAX (denoted by \otimes) on two value-equivalent tuples μ_1 and μ_2 is defined as:

$$v = \mu_1 \otimes \mu_2 \Leftrightarrow (\mu_1 \cong \mu_2) \wedge (v \cong \mu_1) \wedge (v(pS) = \text{Max}\{\mu_1(pS), \mu_2(pS)\})$$

This operator is the same as that in PRM model. In order to keep probabilistic significance under our relational operations, it also needs another operator as follows.

- The total probability associated with a primary key value must be no more than one (Dey, 1996). In order to meet this constraint in our extended relational algebra, we define an operation, called as *standardising-probability* (denoted by κ) on a tuple $\mu(pS)$. Let

$$\Delta_K = \sum_{v \in r, v(K)=\mu(K)} v(pS)$$

We define the standardising-probability κ as:

$$\kappa(\mu(pS)) = \mu(pS), \text{ if } \Delta_K \leq 1; \mu(pS) / \Delta_K, \text{ otherwise}$$

This operation is used for the extended relational algebraic operators to satisfy the Intra-Relation Integrity Constrains. We will define the probabilistic relational operators using the above new operations in next subsection.

5.2 The Extended Relational Operations

We now extend the traditional relational algebraic operators in order to provide facilities for handling probabilistic data in real world applications.

Let r and s be relations on the same probabilistic scheme R with primary key K . Then the union, difference and intersection on the two relations are defined as follows.

1. Extended Union (\cup^P)

$$r \cup^P s = \{ \mu \mid (((\mu \in r) \wedge (\forall v \in s(v \text{ not} \cong \mu))) \vee ((\mu \in s) \wedge (\forall v \in r(v \text{ not} \cong \mu)))) \vee (\exists v \in r \exists \gamma \in s(\mu = v \otimes \gamma)) \wedge \kappa(\mu(pS)) \}$$

2. Extended Difference (\cup^P)

$$r \cup^P s = \{ \mu \mid ((\mu \in r) \wedge (\forall v \in s(v \text{ not} \cong \mu))) \vee ((\exists v \in r, \exists \gamma \in s((\mu \cong v \cong \gamma \wedge v \neq_\epsilon \gamma) \wedge (v(pS) > \gamma(pS)) \wedge (\mu(pS) = v(pS) - \gamma(pS)))))) \}$$

3. Extended Intersection (\cap^P)

$$r \cap^P s = \{ \mu \mid \exists v \in r, \exists \gamma \in s((\mu \cong v \cong \gamma) \wedge (\mu(pS) = \text{Min}\{ v(pS), \gamma(pS) \})) \}$$

Example 2. Let EMPA and EMPB be two relations below.

Relation: EMPA

EMP#	DEPT	PS
3025	shoe	0.80001
3025	toy	0.100001
6637	toy	0.30001
6637	auto	0.2

Relation: EMPB

EMP#	DEPT	PS
3025	shoe	0.8
3025	toy	0.1
6637	toy	0.3
6637	shoe	0.5

Let $\epsilon = 0.001$. Then the results of the above operations on EMPA and EMPB are as follows.

Relation: EMPA \cup^P EMPB

EMP#	DEPT	pS
3025	shoe	0.80001
3025	toy	0.100001
6637	toy	0.3
6637	auto	0.2
6637	shoe	0.5

Relation: EMPA \cap^P EMPB

EMP#	DEPT	pS
6637	auto	0.2

Relation: $EMPA \cap^P EMPB$

EMP#	DEPT	pS
3025	shoe	0.8
3025	toy	0.1
6637	toy	0.3

4. Extended Projection(Π^P)

Let r be a relation on the probabilistic scheme R , and let $S \subset R$. The projection of r onto S is defined as

$$\prod_S^P(r) = \{\mu \mid (\mu = \bigoplus_{v \in r, v(S) \cong \mu} v(S)) \wedge \kappa(\mu(pS))\}.$$

5. Extended Selection (σ^P)

Let r be a relation on the probabilistic scheme R . Θ be a set of comparators over domains of attribute names in R . Let Q be a predicate (called the selection predicate) formed by attributes in R , comparators in Θ , constants in the domain of A for all $A \in R$, and logical connectives. The selection on r for P , written $\sigma_Q^P(r)$, is the set $\{a \in r \mid Q(\mu)\}$. That is,

$$r = \sigma_Q^P(r_1) = \{\mu \mid (\mu \in r) \wedge Q(\mu)\}$$

where, Q a probabilistic predicate is the same as that in Dey (1996).

6. Extended Natural Join($\triangleright\triangleleft^P$)

Let r and s be any two relations on scheme R and S respectively, and let $R' = R - \{pS\}$ and $S' = S - \{pS\}$. The natural join of r and s is defined as:

$$r \triangleright\triangleleft^P s = \{\mu \mid (\exists v \in r, \exists \gamma \in s((\mu(R') = v(R')) \wedge (\mu(S') = \gamma(S'))) \wedge (\mu(pS) = v(pS)\gamma(pS))) \wedge \kappa(\mu(pS))\}$$

5.3 The Relational Algebra and Several Theorems

After defined the above operations, the probabilistic relational algebra can be simply presented as follows.

Let U be a set of attribute names, which is called the universe of the probabilistic relational algebra (written as PRA). $W = 2^U$ be the power set of U , $\text{dom}(W) = \{a \mid a \in \text{dom}(A) \wedge A \in W\}$. $\text{dom}(W)$ is the domain of the semantics of PRA. $D = \{D_1, D_2, \dots, D_n\}$ is the set of distinct probabilistic relation schemes, where $D_i \subseteq U$, for $1 \leq i \leq n$. $RSET = \{r_1, r_2, \dots, r_n\}$ is the set of probabilistic relations, such that r_i is a relation on R_i , $1 \leq i \leq n$. $CSETR$ is the set of all constant relations over the subsets of U . rR is the set of relations over the subsets of U . Θ denotes a set of comparators over domains in $\text{dom}(W)$. The probabilistic relational algebra over U , $\text{dom}(W)$, D , rR and Θ is the 7-tuple as $IE = \langle U, \text{dom}(W), D, rR, \Theta, O, f \rangle$, where,

1. U is the universe of PRA;
2. $\text{dom}(W)$ is the domain of the semantics of PRA;
3. D is the set of distinct probabilistic relation schemes;
4. rR is the set of probabilistic relations over U ;
5. Θ is a set of comparators over domains in $\text{dom}(W)$;
6. O is a set of $\cup^P, -^P, \cap^P, \Pi^P, \sigma^P, \triangleright\triangleleft^P$, and λ -cut;

7. f is a binary tuple as $f = \langle fC, fR \rangle$, where,

$fC: CSETR \mapsto \text{dom}(W)$;

$fR: RSET \mapsto rR$;

Where, fC is a function that assigns a constant relation in $\text{dom}(W)$ to each constant in $CSETR$, and fR is a function which assigns a probabilistic relation over u for each relation in $RSET$.

An algebra expression over AE is recursively defined as

1. A constant relation $c \in CSETR$ is an algebra expression;
2. For $\forall r \in RSET$, r is an algebra expression;
3. For e is an algebra expression, $\lambda(e)$, $\Pi^P(e)$ and $\sigma^P(e)$ are two algebra expressions;
4. Let e_1, e_2 be two algebra expressions, $\theta \in \{\cup^P, -^P, \cap^P, \triangleright\triangleleft^P\}$, $e_1 \theta e_2$ is an algebra expression.

The result of an algebra expression must be a relation. We now present the semantics of algebra expressions. For each $e \in AE$ of degree k and each $I \in \{I\}$ (where, $\{I\}$ is the set of all instances over fixed schema of n relations), the value of e on I , denoted $e(I)$, is a relation of degree k . The formal definitions are as follows:

1. $c(I) = fC(c)$;
2. $r(I) = fR(r)$;
3. $\lambda(e(X))(I) = \{t(X - \{pS\}) \mid t \in e(I) \wedge t(pS) \geq \lambda\}$,
 $\Pi^P(e(X))(I) = \{t(Y) \mid t(X) \in e(I) \wedge (Y \subseteq X)\}$,
 $\sigma_F^P(e)(I) = \{t \mid t \in e(I) \wedge F(t)\}$;
4. $(e_1 \cup^P e_2)(I) = \{t \mid t \in e_1(I) \vee t \in e_2(I)\}$,
 $(e_1 -^P e_2)(I) = \{t \mid t \in e_1(I) \wedge t \notin e_2(I)\}$,
 $(e_1 \triangleright\triangleleft^P e_2)(I) = \{t_1 \circ t_2 \mid t_1 \in e_1(I) \vee t_2 \in e_2(I)\}$.

Next we will present several theorems.

Theorem 1. The First Normal Form (1NF) is closed under extended union, extended intersection, extended difference, extended natural join, extended projection, and extended selection.

Proof: It can be gained directly from the above definitions of extension operators.

Theorem 2. The probability significance is met under extended union, extended intersection, extended difference, extended natural join, extended projection, and extended selection.

Proof: The axioms or properties of probability certainly hold under these operations according the definitions of them. So, the probability significance is met under extended union, extended intersection, extended difference, extended natural join, extended projection, and extended selection.

Theorem 3. If E is an expression of 1NF probabilistic relational algebra, there is a safe expression E' of 1NF probabilistic tuple calculus which is equivalent to E .

Proof: It can be proven from the above expressions of extension operators.

Theorem 4. If E is a safe expression of 1NF probabilistic tuple calculus, then there is an expression E' of 1NF probabilistic relational algebra which is equivalent to E .

Proof: It can be constructed according to the proof of the similar theorem in conventional relational theory (Ullman, 1988).

Apparently, the above definitions of *IPTRC* and probabilistic relational algebra are as similar as the conventional tuple calculus languages and relational algebras except that $=_{\epsilon}$ is substituted for $=$. In the approximating sense, the role of $=_{\epsilon}$ is the same as one of $=$. So, the proof of this theorem can be constructed directly from the proof of the similar theorem in conventional relational theory. And only difference between the two proofs is that $=_{\epsilon}$ is substituted for $=$ when the object is pS .

6. ALGEBRAIC QUERY LANGUAGE

The probabilistic relational algebra provides operations: union (\cup^P), difference ($-^P$), and intersection (\cap^P) to *insert* and *delete* tuples on probabilistic databases. And the operations such as projection (Π^P), selection (σ^P), and join (\bowtie^P) are used to answer queries on probabilistic relational databases. These operators are as similar as that in traditional relational algebras. However, because the domain $(0, 1)$ of pS is infinite, probabilistic relational algebra taken as a query language is different from conventional query languages at probabilistic attribute pS . In this section, we now show the algorithm of probabilistic relational operations in this subsection.

(1) Extended Union (\cup^P)

Algorithm of union

input: relations r and s on scheme R

output: relation $r_1 = r \cup^P s$ on scheme R

begin

$r_1 \leftarrow r$;

for all $v \in s$ do

for all $\mu \in r_1$ do

if $v \cong \mu$ then

$\mu \leftarrow (v \otimes \mu)$

else $r_1 \leftarrow r_1 \cup \{v\}$;

for all $\mu \in r_1$ do

$\kappa(\mu(pS))$

end.

(2) Extended Difference ($-^P$)

Algorithm of difference

input: relations r and s on scheme R

output: relation $r_1 = r -^P s$ on scheme R

begin

$r_1 \leftarrow r$;

for all $v \in s$ do

for all $\mu \in r_1$ do

if $v \cong \mu$ then

if $v(pS) < \mu(pS)$ then

$\mu(pS) \leftarrow \mu(pS) - v(pS)$

else $r_1 \leftarrow r_1 - \{\mu\}$;

end.

(3) Extended Intersection (\cap^P)

Algorithm of intersection

input: relations r and s on scheme R

output: relation $r_1 = r \cap^P s$ on scheme R
begin
 $r_1 \leftarrow r$;
for all $\mu \in r_1$ do
for all $v \in s$ do
if $v \cong \mu$ then
 $\mu(pS) \leftarrow \text{Min}\{\mu(pS), v(pS)\}$
else $r_1 \leftarrow r_1 - \{\mu\}$;
end.

(4) Extended Projection (\prod_s^P)

Algorithm of projection

input: relations r on scheme R and a subscheme $S \subset R$
output: relation $r_1 = \prod_s^P(r)$ on scheme
begin

$r_1 \leftarrow \emptyset$;
 $s \leftarrow \{\tau \mid \tau = \tau_1(S) \wedge \tau_1 \in r\}$;
for all $\mu \in s$ do
if $(\exists v \in r_1 (v \cong \mu))$ then
 $v(pS) \leftarrow v(pS) \oplus \mu(pS)$
else $r_1 \leftarrow r_1 \cup^P \{\mu(S)\}$;
for all $\mu \in r_1$ do
 $\kappa(\mu(pS))$
end.

(5) Extended Selection (σ^P)

Algorithm of selection

input: relations r on scheme R and a probabilistic predicate Q
output: relation $r_1 = \sigma^P(r)$ on scheme R
begin
 $r_1 \leftarrow \emptyset$;
for all $\mu \in r$ do
if $Q(\mu)$ then
 $r_1 \leftarrow r_1 \cup \{\mu\}$;
end.

(6) Extended Natural Join ($\triangleright \triangleleft^P$)

Algorithm of natural join

input: relations r on scheme R and s on scheme S
output: relation $r_1 = \triangleright \triangleleft^P$ on scheme $D = R \cup S$
begin
 $r_1(D) \leftarrow \emptyset$;
for all $\mu \in r$ do
for all $v \in s$ do
 $r_1 \leftarrow r_1 \cup \{(\mu(R - \{pS\}), v(S - \{pS\}), \mu(pS) * v(pS))\}$;
for all $\mu \in r_1$ do
 $\kappa(\mu(pS))$
end.

7. COMPARISON AND CONCLUSIONS

To model the probabilistic nature of data, uncertain data has to be incorporated into existing databases (typically relational databases). However, current approaches have many shortcomings and have not established an acceptable extension of the relational model. Dey and Sarkar (1996) gave a good review of these probabilistic data models. In order to overcome some shortcomings in the current models, they proposed a probabilistic relational model and algebra (PRM) that is a consistent extension of the conventional relational model.

They have defined several important operators, such as conditionalisation, coalescence-plus, and coalescence-MAX that are useful for extending conventional relational models to support uncertainty data. As an attempt, our model in this paper is based on PRM model such that some excellent techniques can be inherited. Meanwhile, our model has the advantage of proposing probabilistic tuple calculus and its semantics. For simplicity, the comparison of our model with the PRM model focuses only on the following four cases.

1. One of Intra-Relation Integrity Constrains is not reflected in the extended operations in PRM model. We applied standardising-probability operator in our extended operations such that the probability significance is met under these operations.
2. The definition of coalescence-PLUS operation in PRM model is inadequate to match the properties of probability. In our model, for two value-equivalent tuples μ_1 and μ_2 , and $\nu = \mu_1 \oplus \mu_2$, we defined $\nu(pS)$ as $\mu_1(pS) + \mu_2(pS) - \mu_1(pS) * \mu_2(pS)$. This definition satisfies the properties of probability.
3. Traditional equal relationship (=) used in PRM model cannot fit the equal relationship under uncertainty. Indeed, because the user may not know exactly what the probabilities of the tuples are, the queries to probabilities of the tuples in a probabilistic database should be based on approximate matching rather than exact matching. In order to meet these applications, we use “ ϵ -equal” and “ ϵ -neighbor” to fit the fuzzy-equal operator and the approximate matching respectively, such that the approximation and infinite semantics of uncertainty data can be modeled in our probabilistic data model.
4. PRM model focuses mainly on the probabilistic relational algebra. We not only presented a probabilistic relational algebra and its semantics, but also established an equivalent probabilistic tuple calculus language and its semantics.

We have seen, our probabilistic data model can overcome some of these limitations in PRM model. In particular, we establish a probabilistic tuple calculus to meet our probabilistic relational algebra. This work can enrich the work on uncertainty data in the current literature. The key points of this work are:

1. It defined “ ϵ -equal” to fit the fuzzy-equal in this section. With this definition, we can model the approximating and infinite semantics of uncertainty data using probabilistic relational data model.
2. It established a probability tuple calculus language (*IPTRC*). In particular, the semantics of *IPTRC* was developed, and some examples were applied to illustrate how to use *IPTRC*.
3. It defined a probability relational algebra and its semantics. This probabilistic relational algebra is demanded to satisfy the axioms or properties of probability.
4. It illustrated how to take probabilistic relational algebra as a query language.

We plan to extend our work in several directions. First, we will address optimisation issues of our model and define new data model of handling probabilistic and fuzzy information. Second, we will explore a proper method for supporting the semantics of uncertainty data.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewer for his/her detailed constructive comments on the first version of this paper.

REFERENCES

- BARBARA, D., GARCIA-MOLINA, H. and PORTER, D. (1992): The management of probabilistic data, *IEEE Trans. Knowledge and Data Eng.*, 4(5): 487–502.
- BUCKLES, B. and PETRY, F. (1983): Information-theoretical characterisation of fuzzy relational databases, *IEEE Trans. Syst. Man Cybern.*, 13(1): 74–77.
- BUCKLES, B. and PETRY, F. (1984): Extending the fuzzy database with fuzzy numbers, *Information Sciences*, 34: 145–155.
- CAVALLO, R. and PITTARELLI, M. (1987): *The theory of probabilistic database*, In: Proceedings of the 13th VLDB Conference, Brighton, England, 71–81.
- DEY, D. and SARKAR, S. (1996): A probabilistic relational model and algebra, *ACM Trans. on database systems*, 21(3): 339–369.
- FUHR, N. and ROLLEKE, T. (1997): A probabilistic relational algebra for the integration of information retrieval and Database Systems, *ACM Trans. on Information systems*, 15(1): 32–66.
- GUPTA, M. (1992): Intelligence, uncertainty and information. In: B. AYYUB, M. GUPTA and L. KANAL (Editors), *Analysis and Management of Uncertainty: Theory and Applications*, 3–11.
- IMIELINSKI, T. and LIPSKI, W. (1984): Incomplete information in relational databases, *J. ACM*, 31(4).
- LAKSHMANAN, L. and SADRI, F. (1984): Probabilistic deductive databases. In: Proceedings of the International Logic Programming Symposium, Cambridge, MA, 254–268.
- LAKSHMANAN, L., LEONE, N., ROSS, R. and SUBRAHMANIAN, V. (1997): ProbView: A flexible probabilistic database system, *ACM Trans. on database systems*, 22(3): 419–469.
- MENDELSON, H. and SAHARIA, A. (1986): Incomplete information costs and database design, *ACM Trans. on database systems*, 11(2): 159–185.
- NG, R. and SUBRAHMANIAN, V. (1995): Stable semantics for probabilistic deductive databases. *Inf. Comput.* 110(1): 42–83.
- PITTARELLI, M. (1994): An algebra for probabilistic databases, *IEEE Trans. Knowledge and Data Eng.*, 6(2): 293–302.
- PRADE, H. and TESTEMALE, C. (1984): Generalising database relational algebra for the treatment of incomplete or uncertain information and vague queries, *Information Sciences*, 34: 115–143.
- RAJU, K. and MAJUMDAR, A. (1988): Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems, *ACM Trans. on database systems*, 12(2): 129–166.
- SCHMIDT, H., KIESSLING, W., GUNTZER, U. and BAYER, R. (1987): Combining deduction by uncertainty with the power of magic. In: Proceedings of DOOD-87 (Kyoto, Japan), 205–224.
- TSENG, F., CHEN, A. and YANG, W. (1993): Answering heterogeneous database queries with degrees of uncertainty, *Distributed and Parallel Databases: An International Journal*, 1(3): 281–302.
- ULLMAN, J. (1988): Principles of database and knowledge-base systems, Computer Science Press.
- VARDI, M. (1985): Querying logical databases. In: Proceedings of the Fourth ACM SIGACT/SIGMOD Symposium on Principles of Database Systems, 57–65.
- WONG, E. (1982): A statistical approach to incomplete information in database systems, *ACM Trans. on database systems*, 7(3): 470–488.
- ZEMANKOVA, M. and KANDEL, A. (1985): Implementing imprecision in information systems, *Information Sciences*, 37: 107–141.

SELECTED PUBLICATION LIST, 2002

- ZHANG, C. and ZHANG, S. (2002): *Association rules mining: models and algorithms*, (monograph) published by Springer-Verlag Publishers in Lecture Notes in Computer Sciences, LNAI-2307, 243.
- ZHANG, S. and SHANG, C. (2002): Anytime mining for multi-user applications, *IEEE Transactions on Systems, Man and Cybernetics (Part A)*, 32(4): 515–521.
- ZHANG, S. and ZHANG, C. (2002): Encoding the propagation in belief networks, *IEEE Transactions on Systems, Man and Cybernetics (Part A)*, 32(4): 526–531.
- ZHANG, S. and ZHANG, C. (2002): Discovering causality in large databases, *Applied Artificial Intelligence*, 16(5): 333–358.
- ZHANG, C., ZHANG, S. and ZHANG, Z. (2003): Temporal constraint satisfaction in matrix method. *Applied Artificial Intelligence*, 17(2): 135–154.
- ZHANG, Z. and ZHANG, C. (2002): An improved matchmaking algorithm for middle agents. In: Proceedings of the First Autonomous Agents and Multi-Agent Systems, July.
- LUO, X. and ZHANG, C. (2002): A hybrid model for sharing information in heterogeneous fuzzy, uncertain and default reasoning environment, accepted by *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*.
- WU, X., ZHANG, C. and ZHANG, S. (2002): Mining both positive and negative association rules. In: *Proceedings of 19th International Conference on Machine Learning*, Sydney, Australia, July, 658–665.

- ZHANG, C., ZHANG, S. and WEBB, G.I. (2003): Identifying approximate itemsets of interest in large databases, *Applied Intelligence*, 18: 91–104.
- ZHANG, S. and ZHANG, C. (2002): Propagating temporal relations of intervals by matrix. *Applied Artificial Intelligence*, 19(1): 1–27.
- ZHANG, S., WU, X. and ZHANG, C. (2003): Multi-database mining. *IEEE Computational Intelligence Bulletin*, June, 2(1): 5–13.
- ZHANG, S., YANG, Q. and ZHANG, C. (2003): Data preparation for data mining. *Applied Artificial Intelligence – Special Issue on Data Cleaning and Preprocessing*, 17(5–6): 375–382.
- WU, X. and ZHANG, S. (2003): Synthesizing high-frequency rules from different data sources. *IEEE Transactions on Knowledge and Data Engineering*, March/April, 15(2): 353–367.
- ZHANG, S., ZHANG, C. and YAN, X. (2003): PostMining: Maintenance of association rules by weighting. *Information Systems*, October, 28(7): 691–707.
- ZHANG, S. and ZHANG, C. (2003): Discovering associations in very large databases by approximating. *Acta Cybernetica*, 16: 155–177.
- ZHANG, Z., ZHANG, C. and ZHANG, S. (2003): An agent-based hybrid framework for database mining. *Applied Artificial Intelligence – Special Issue on Data Cleaning and Preprocessing*, 17(5–6): 383–398.
- LI, Y., ZHANG, C. and ZHANG, S. (2003): Cooperative strategy web-based data cleaning. In: *Applied Artificial Intelligence – Special Issue on Data Cleaning and Preprocessing*, 17(5–6): 443–460.
- YAN, X., ZHANG, C. and ZHANG, S. (2003): Towards databases mining: Preprocessing collected data. *Applied Artificial Intelligence – Special Issue on Data Cleaning and Preprocessing*, 17(5–6): 545–561.
- ZHANG, S. and LIU, L. (2003): Mining dynamic databases by weighting. *Acta Cybernetica*, 16: 179–205.
- NIU, L., YAN, X., ZHANG, C. and ZHANG, S. (2003): Product hierarchy-based customer profiles for electronic commerce recommendation. *Asian Journal of Information Technology*, 2(1): 18–24.

BIOGRAPHICAL NOTES

Professor Chengqi Zhang received a PhD degree from the University of Queensland, Brisbane in Computer Science and a Doctor of Science (higher doctorate) degree from Deakin University. He is currently a research professor in Faculty of Information Technology at University of Technology, Sydney. His areas of research are Data Mining and Multi-Agent Systems. He has published more than 170 refereed papers, edited seven books, and published two monographs till today. He is a Senior Member of the IEEE Computer Society (IEEE), an Associate Editor of the editorial board for Knowledge and Information Systems: an International Journal, and a member of editorial board of International Journal of Web Intelligence and agent systems. He has served as a member of Program Committees in many international or national conferences. He is currently a Program Committee Co-Chair of PRICAI 2004 and General Co-Chair of PAKDD 2004.



Chengqi Zhang

Shichao Zhang received his PhD degree from the Deakin University, Australia. Dr Zhang is now an Assistant President of the Guangxi Normal University. He is also currently a Senior Research Fellow (SL-4) at the Faculty of Information Technology, University of Technology Sydney, Australia. He was promoted to Professor at the School of Mathematical and Computing Sciences of the Guangxi Normal University in 1994. He worked as a Research Fellow at the National University of Singapore, the University of New England and Deakin University. His recent research interests include database classification, data analysis, and data mining. He has authored three monographs (two of them by Springer), over 30 refereed international journal articles (three of them in IEEE Transactions), and over 40 refereed international conference papers. He is a member of Editorial Board of 'Asian Journal of Information Technology'. He has served as a PC member for international conferences, including the 2003 IEEE International Conference on Data Mining (ICDM03).



Shichao Zhang