Database Querying in the Presence of Suspect Values

Olivier Pivert[1(✉)] and Henri Prade[2]

[1] University of Rennes 1 – Irisa, Lannion, France
pivert@enssat.fr
[2] IRIT – CNRS/University of Toulouse 3, Toulouse, France
prade@irit.fr

Abstract. In this paper, we consider the situation where a database may contain suspect values, i.e. precise values whose validity is not certain. We propose a database model based on the notion of possibilistic certainty to deal with such values. The operators of relational algebra are extended in this framework. A very interesting aspect is that queries have the same data complexity as in a classical database context.

1 Introduction

In many application contexts, databases appear to involve suspect values (i.e., values whose validity is dubious), for various reasons: i) some attribute values may have been produced by means of a prediction process, for instance using a technique aimed to estimate null values (in the sense of unknown but applicable), see e.g. [3,4], or ii) the database may result from the integration of multiple (more or less reliable, potentially conflicting) data sources [5], or iii) the database may have gone through an automated cleaning process [11] aimed to remove inconsistencies (and in general there are several ways of restoring consistency, even in simple cases, which is a source of potential errors).

It is of course important to deal with such suspect values with the required cautiousness, in particular when answering queries. A variety of uncertain database models have been proposed to represent and handle uncertain values. In these models, an ill-known attribute value is generally represented by a probability distribution (see, e.g. [7,12]) or a possibility distribution [1], i.e. a set of weighted candidate values. However, in many situations, it may be very problematic to quantify the level of uncertainty attached to the different candidate values. One may not even know the set of (probable/possible) alternative candidates. Then, using a probabilistic model in a rigorous manner appears quite difficult, not to say impossible. In this work, we assume that all one knows is that a given precise value is suspect, i.e. not totally certain, and we show that a database model based on the notion of possibilistic certainty is a suitable tool for representing and handling suspect data. The remainder of the paper is structured as follows. Section 2 briefly presents the three-valued fragment of possibility theory that will be used in our model. Section 3 presents the uncertain database model that we advocate for representing tuples that may involve

T. Morzy et al. (Eds): ADBIS 2015, CCIS 539, pp. 44–51, 2015.
DOI: 10.1007/978-3-319-23201-0_6

suspect attribute values. Section 4 gives the definitions of the algebraic operators in this framework. In Section 5, we discuss a way to make selection queries more flexible, which makes it possible to discriminate the uncertain answers to a query. Finally, Section 6 recalls the main contributions and outlines perspectives for future work.

2 A Fragment of Possibility Theory with Three Certainty Levels

In possibility theory [6,13], each event E — defined as a subset of a universe Ω — is associated with two measures, its possibility $\Pi(E)$ and its necessity $N(E)$. Π and N are two dual measures, in the sense that $N(E) = 1 - \Pi(\overline{E})$ (where the overbar denotes complementation). This clearly departs from the probabilistic situation where $Prob(E) = 1 - Prob(\overline{E})$. So in the probabilistic case, as soon as you are not certain about E ($Prob(E)$ is small), you become rather certain about $\overline{E}$ ($Prob(\overline{E})$ is large). This is not at all the situation in possibility theory, where complete ignorance about E ($E \neq \emptyset$, $E \neq \Omega$) is allowed: This is represented by $\Pi(E) = \Pi(\overline{E}) = 1$, and thus $N(E) = N(\overline{E}) = 0$. In possibility theory, being somewhat certain about E ($N(E)$ has a high value) forces you to have $\overline{E}$ rather impossible ($1 - \Pi$ is impossibility), but it is allowed to have no certainty neither about E nor about $\overline{E}$. Generally speaking, possibility theory is oriented towards the representation of epistemic states of information, while probabilities are deeply linked to the ideas of randomness, and of betting in case of subjective probability, which both lead to an additive model such that $Prob(E) = 1 - Prob(\overline{E})$.

In the following, we assume that the certainty degree associated with the uncertain events considered (that concern the actual value of an attribute in a tuple, for instance) is unknown. Thus, we use a fragment of possibility theory where three values only are used to represent certainty : 1 (completely certain), α (somewhat certain but not totally), 0 (not at all certain). The fact that one uses α for every somewhat certain event does not imply that the certainty degree associated with these events is the same; α is just a conventional symbol that means "a certainty degree in the open interval (0, 1)". Notice that this corresponds to using three symbols for representing possibility degrees as well: 0, β ($= 1 - \alpha$), and 1 (but we are not interested in qualifying possibility).

3 The Database Model

In the database model introduced in [2] and detailed in [10], a certainty level is attached to each ill-known attribute value (by default, an attribute value has certainty 1). For instance, the tuple $\langle 037, \mathit{John}, (40, 0.7), (\mathit{Engineer}, 0.6)\rangle$ denotes the existence of a person named *John* for sure, whose age is 40 with certainty 0.7 (which means that the possibility that his age differs from 40 is upper bounded by $1 - 0.7 = 0.3$ without further information on the respective possibility degrees

of other possible values), and whose job is *Engineer* with certainty 0.6. In the database model we introduce hereafter, the basic idea is also to represent the fact that an attribute value may not be totally certain, but we do not assume available any knowledge about the certainty level attached to a suspect value.

Let us consider a database containing suspect values. In the following, a suspect value will be denoted using a star, as in 17*. A value a^* means that it is somewhat certain (thus completely possible) that a is the actual value of the considered attribute for the considered tuple, but not totally certain (otherwise we would use the notation a instead of a^*).

In the model we propose, we restrict ourselves to the computation of the somewhat certain answers, since dealing with the answers that are only somewhat possible raises important difficulties.

The database model we propose relies on the fragment of possibility theory introduced in Section 2, where three values only are used to quantify certainty: 1 (completely certain), α (somewhat certain but not totally), 0 (not at all certain). The tuples or values that are not at all certain are discarded and do not appear in the database.

Uncertain tuples are denoted by α/t where α has the same meaning as above. α/t means that the existence of the tuple in the considered relation is only somewhat certain (thus, it is also possible to some extent that it does not exist). It is mandatory to have a way to represent such uncertain tuples since some operations of relational algebra (selection, in particular) may generate them. The tuples whose existence is completely certain are denoted by $1/t$. A relation of the model will thus involve an extra column denoted by N, representing the certainty attached to the tuples.

4 Algebraic Operators

In this section, we give the definition of the three main operators (projection, selection, join) of relational algebra in the certainty-based model defined above. We leave the set-oriented operators aside due to space limitation.

4.1 Selection

In the following, we denote by $c(t.A)$ the certainty degree associated with the value of attribute A in tuple t: $c(t.A)$ equals 1 if $t.A$ is a nonsuspect value, and it takes the (conventional) value α otherwise (with the convention $\alpha < 1$). It is the same thing for the certainty degree N associated with a tuple (the notation is then N/t).

Case of a condition of the form $A\,\theta\,q$ where A is an attribute, θ is a comparison operator, and q is a constant:

$$\sigma_{A\,\theta\,q}(r) = \{N'/t \mid N/t \in r \text{ and } t.A\,\theta\,q \text{ and } N' = \min(N,\, c(t.A))\} \qquad (1)$$

Table 1. Relation *Emp* (left), result of the selection query (right)

#id	*name*	*city*	*job*	*N*
37	John	Newton*	Engineer*	1
53	Mary	Quincy*	Clerk*	1
71	Bill	Boston	Engineer	1

#id	*name*	*city*	*job*	*N*
37	John	Newton*	Engineer*	α
71	Bill	Boston	Engineer	1

Example 1. Let us consider the relation *Emp* represented in Table 1 (left) and the selection query $\sigma_{job='Engineer'}(Emp)$. Its result is represented in Table 1 (right). ◇

Case of a condition of the form $A_1\ \theta\ A_2$ where A_1 and A_2 are two attributes and θ is a comparison operator:

$$\sigma_{A_1\ \theta\ A_2}(r) = \{N'/t \mid N/t \in r \text{ and } t.A_1\ \theta\ t.A_2 \text{ and } N' = \min(N,\ c(t.A_1),\ c(t.A_2))\}. \tag{2}$$

Case of a conjunctive condition $\psi = \psi_1 \wedge \ldots \wedge \psi_m$:

$$\sigma_{\psi_1 \wedge \ldots \wedge \psi_m}(r) = \{N'/t \mid N/t \in r \text{ and } \psi_1(t.A_1) \text{ and } \ldots \text{ and } \psi_m(t.A_m) \text{ and } N' = \min(N,\ c(t.A_i),\ \ldots,\ c(t.A_m))\}. \tag{3}$$

Case of a disjunctive condition $\psi = \psi_1 \vee \ldots \vee \psi_m$:

$$\sigma_{\psi_1 \vee \ldots \vee \psi_m}(r) = \{N'/t \mid N/t \in r \text{ and } (\psi_1(t.A_1) \text{ or } \ldots \text{ or } \psi_m(t.A_m)) \text{ and } N' = \min(N,\ \max_{i \text{ such that } \psi_i(t.A_i)}(c(t.A_i)))\}. \tag{4}$$

4.2 Projection

Let r be a relation of schema (X, Y). The projection operation is straightforwardly defined as follows:

$$\pi_X(r) = \{N/t.X \mid N/t \in r \text{ and } \nexists\, N'/t' \in r \text{ such that } sbs(N'/t'.X,\ N/t.X)\}.$$

The only difference w.r.t. the definition of the projection in a classical database context concerns duplicate elimination, which is here based on the concept of "possibilistic subsumption". Let $X = \{A_1, \ldots, A_n\}$. The predicate *sbs*, which expresses subsumption, is defined as follows:

$$\begin{aligned} sbs((N'/t'.X, N/t.X) \equiv\ & \forall i \in \{1, \ldots, n\},\ t.A_i = t'.A_i \text{ and} \\ & c(t.A_i) \leq c(t'.A_i) \text{ and } N \leq N' \text{and} \\ & ((\exists i \in \{1, \ldots, n\},\ c(t.A_i) < c(t'.A_i)) \text{ or } N < N'). \end{aligned} \tag{5}$$

Example 2. Let us consider relation *Emp* represented in Table 2 (left) and the projection query $\pi_{\{city,\ job\}}(Emp)$. Its result is represented in Table 2 (right). ◇

Table 2. Relation *Emp* (left), result of the projection query (right)

#id	*name*	*city*	*job*	*N*
35	Phil	Newton	Engineer*	1
52	Lisa	Quincy*	Clerk*	α
71	Bill	Newton	Engineer	α
73	Bob	Newton*	Engineer*	α
84	Jack	Quincy*	Clerk	α

city	*job*	*N*
Newton	Engineer*	1
Newton	Engineer	α
Quincy*	Clerk	α

4.3 Join

The definition of the join in the context of the model considered is:

$$r_1 \bowtie_{A=B} r_2 = \{ \min(N_1,\ N_2,\ c(t_1.A),\ c(t_2.B))/t_1 \oplus t_2 \mid \exists N_1/t_1 \in r_1,\ \exists N_2/t_2 \in r_2 \text{ such that } t_1.A = t_2.B \tag{6}$$

where $\oplus$ denotes concatenation.

Example 3. Consider the relations from Table 3 (top) and the query:

$$PersLab = Person \bowtie_{Pcity=Lcity} Lab$$

which looks for the pairs (p, l) such that p (somewhat certainly) lives in a city where a research center l is located. Its result appears in Table 3 (bottom). $\diamond$

Table 3. Relations *Person* (left), *Lab* (right), result of the join query (bottom)

#Pid	*Pname*	*Pcity*	*N*
11	John	Boston*	1
12	Mary	Boston	α
17	Phil	Weston*	α
19	Jane	Weston	1

#Lid	*Lname*	*Lcity*	*N*
21	BERC	Boston*	α
22	IFR	Weston	1
23	AZ	Boston	1

#Pid	*Pname*	*Pcity*	*#Lid*	*Lname*	*Lcity*	*N*
11	John	Boston*	21	BERC	Boston*	α
11	John	Boston*	23	AZ	Boston	α
12	Mary	Boston	21	BERC	Boston*	α
12	Mary	Boston	23	AZ	Boston	α
17	Phil	Weston*	22	IFR	Weston	α
19	Jane	Weston	22	IFR	Weston	1

In the case of a natural join (i.e., an equijoin on all of the attributes common to the two relations), one keeps only one copy of each join attribute in the resulting table. Here, this "merging" keeps the most uncertain value for each join attribute. This behavior is illustrated in Table 4.

Table 4. Result of the natural join query (assuming a common attribute *City*)

#Pid	*Pname*	*City*	*#Lid*	*Lname*	*N*
11	John	Boston*	21	BERC	α
11	John	Boston*	23	AZ	α
12	Mary	Boston*	21	BERC	α
12	Mary	Boston	23	AZ	α
17	Phil	Weston*	22	IFR	α
19	Jane	Weston	22	IFR	1

A crucial point is that the join operation does not induce intertuple dependencies in the result, due to the semantics of certainty. This is not the case when a probabilistic or a full possibilistic [1] model is used, and one then has to use a variant of c-tables [8] to handle these dependencies, which implies a non-polynomial complexity. On the other hand, since none of the operators of relational algebra induces intertuple dependencies in our model, the queries have the same data complexity as in a classical database context; see [10] for a more complete discussion.

5 Making Selection Queries More Flexible

If one assumes that the relation concerned by a selection is a base relation (i.e., where all the tuples have a degree $N = 1$), a tuple in the result is uncertain iff it involves at least one suspect value concerned by the selection condition. If such a tuple involves several such suspect values, it will be no more uncertain ($N = \alpha$) than if it involves only one. However, one may find it desirable to distinguish between these situations. For instance, considering the query

$$\sigma_{job=\text{`}Engineer\text{'} \text{ and } city=\text{`}Boston\text{'} \text{ and } age=30}(Emp)$$

the tuple ⟨John, Engineer*, Boston, 30⟩ could be considered more satisfactory (less risky) than, e.g., ⟨Bill, Engineer*, Boston*, 30⟩, itself more satisfactory than ⟨Paul, Engineer*, Boston*, 30*⟩.

For a selection condition $\psi = \psi_1 \wedge \ldots \psi_m$ and a tuple t, this amounts to saying that "every attribute value (certain and suspect) of t must satisfy the condition ψ_i that concerns it, and the less there are suspect values concerned by a ψ_i in t, the more t is preferred". In other words, the condition becomes:

$\psi_1 \wedge \ldots \wedge \psi_m$ and *as many* $(t.A_1, \ldots, t.A_m)$ *as possible* are totally certain.

In a user-oriented language based on the algebra described above, one may then introduce an operator IS CERTAIN (meaning "is totally certain"), in the same way as there exists an operator IS NULL in SQL.

The fuzzy quantifier [14] *as many as possible* (*amap* for short) corresponds to a function from [0, 1] to [0, 1]. Its membership function μ_{amap} is such that: i) $\mu_{amap}(0) = 0$, ii) $\mu_{amap}(1) = 1$, iii) $\forall x, y, x > y \Rightarrow \mu_{amap}(x) > \mu_{amap}(y)$. Typically, we shall take $\mu_{amap}(x) = x$.

The selection condition as expressed above is made of two parts: a "value-based one" — that may generate uncertain answers —, and a "representation-based" one that generates gradual answers. A tuple of the result is assigned a satisfaction degree μ (seen as the complement to 1 of a suspicion degree), on top of its certainty degree N. For a conjunctive query made of m atomic conjuncts ψ_i, the degree μ associated with a tuple t is computed as follows:

$$\mu(t) = \mu_{amap}\Big(\frac{\sum_{i=1}^{m} certain(t,\ i)}{m}\Big) \tag{7}$$

where

$$certain(t,i) = \begin{cases} 1 \text{ if } \psi_i \text{ if of the form } A\,\theta\,q \text{ and } c(t.A) = 1, \\ 1 \text{ if } \psi_i \text{ if of the form } A_1\,\theta\,A_2 \text{ and } \min(c(t.A_1),\ c(t.A_2)) = 1, \\ 0 \text{ otherwise.} \end{cases}$$

In order to display the result of the query, one rank-orders the answers on N first, then on μ (in an increasing way in both cases).

Example 4. Let us consider the relation represented in Table 1 (top) and the selection query $\sigma_\psi(Emp)$ where ψ is the condition

job = 'Engineer' and *city* = 'Boston' and *age* > 30 and
amap (*job* IS CERTAIN, *city* IS CERTAIN, *age* IS CERTAIN)

Let us assume that the membership function associated with the fuzzy quantifier *amap* is $\mu_{amap}(x) = x$. The result of the query appears in Table 5 (bottom). ⋄

Table 5. Relation *Emp* (top), result of the selection query (bottom)

#id	*name*	*city*	*job*	*age*	*N*
38	John	Boston*	Engineer*	32	1
54	Mary	Quincy*	Engineer*	35	1
72	Bill	Boston	Engineer	40	1
81	Paul	Boston*	Engineer*	31*	1
93	Phil	Boston	Engineer	52*	1

#id	*name*	*city*	*job*	*age*	*N*	μ
72	Bill	Boston	Engineer	40	1	1
93	Phil	Boston	Engineer	52*	α	0.67
38	John	Boston*	Engineer*	32	α	0.33
81	Paul	Boston*	Engineer*	31*	α	0

This extended framework, where two degrees (N and μ) are associated with each tuple in the relations, can be easily made compositional. One just has to manage the degrees μ, in the definition of the algebraic operators, as in a gradual (fuzzy) relation context, see [9]. In base relations, it is assumed that $\mu(t) = 1\ \forall t$.

6 Conclusion

In this paper, we have proposed a database model and defined associated algebraic operators for dealing with the situation where some attribute values in a dataset are suspect, i.e., have an uncertain validity, in the absence of further information about the precise levels of uncertainty attached to such suspect values. The framework used is that of possibility theory restricted to a certainty scale made of three levels. It is likely that the idea of putting some kind of tags on suspect values/tuples/answers is as old as information systems. However, the benefit of handling such a symbolic tag in the framework of possibility theory is to provide a rigorous setting for this processing.

A very important point is that the data complexity of all of the algebraic operations is the same as in the classical database case, which makes the approach perfectly tractable. Moreover, the definitions of both the model and the operators are quite simple and do not raise any serious implementation issues.

References

1. Bosc, P., Pivert, O.: About projection-selection-join queries addressed to possibilistic relational databases. IEEE Trans. on Fuzzy Systems **13**(1), 124–139 (2005)
2. Bosc, P., Pivert, O., Prade, H.: A model based on possibilistic certainty levels for incomplete databases. In: Godo, L., Pugliese, A. (eds.) SUM 2009. LNCS, vol. 5785, pp. 80–94. Springer, Heidelberg (2009)
3. Chen, S.M., Chang, S.T.: Estimating null values in relational database systems having negative dependency relationships between attributes. Cybernetics and Systems **40**(2), 146–159 (2009)
4. Beltran, W.C., Jaudoin, H., Pivert, O.: Analogical prediction of null values: the numerical attribute case. In: Manolopoulos, Y., Trajcevski, G., Kon-Popovska, M. (eds.) ADBIS 2014. LNCS, vol. 8716, pp. 323–336. Springer, Heidelberg (2014)
5. Destercke, S., Buche, P., Charnomordic, B.: Evaluating data reliability: An evidential answer with application to a web-enabled data warehouse. IEEE Trans. Knowl. Data Eng. **25**(1), 92–105 (2013)
6. Dubois, D., Prade, H.: Possibility Theory. Plenum, New York (1988)
7. Haas, P.J., Suciu, D.: Special issue on uncertain and probabilistic databases. VLDB J. **18**(5), 987–988 (2009)
8. Imielinski, T., Lipski, W.: Incomplete information in relational databases. J. of the ACM **31**(4), 761–791 (1984)
9. Pivert, O., Bosc, P.: Fuzzy Preference Queries to Relational Databases. Imperial College Press, London (2012)
10. Pivert, O., Prade, H.: A certainty-based model for uncertain databases. IEEE Transactions on Fuzzy Systems (2015) (to appear)
11. Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Data Eng. Bull. **23**(4), 3–13 (2000)
12. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. Synthesis Lectures on Data Management. Morgan & Claypool Publishers (2011)
13. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems **1**(1), 3–28 (1978)
14. Zadeh, L.: A computational approach to fuzzy quantifiers in natural languages. Computing and Mathematics with Applications **9**, 149–183 (1983)