

About Cardinality-based Possibilistic Queries Against Possibilistic databases

Patrick Bosc

Nadia Liétard

Olivier Pivert

IRISA/ENSSAT

Technopole Anticipa BP 80518

22305 Lannion Cedex France

e-mail: {bosc, lietard, Pivert} @enssat.fr

Abstract

This paper deals with the querying of possibilistic relational databases, by means of possibilistic queries whose general form is: "to what extent is it possible that the answer to Q satisfies property P". Here, we consider cardinality-based queries (in this case, property P is about cardinality) for which a processing technique is proposed, which avoids computing all the worlds attached to the possibilistic database.

Keywords: Ill-known values, possibilistic databases, world-based interpretation.

1 Introduction

In this paper, we consider relational databases where some attribute values are imprecisely known. Different formalisms can be used to represent imprecise information (see for instance [5, 6]), and the possibilistic setting is assumed in the following.

A key question is to define a sound semantics for queries addressed to imprecise databases. Since imprecise data are represented as (possibly infinite) sets of acceptable candidates, an imprecise database can be seen as a set of regular databases, called worlds, associated with a choice for each attribute value. This approach provides a rational starting point for the definition of a query in the sense that its result is a compact representation of the results obtained in each world. Unfortunately, this approach is intractable due to the huge (possibly infinite) number of worlds. This observation leads to consider only specific queries which can be processed directly against the possibilistic database, while delivering a result equivalent to the one

defined in terms of worlds. A compact calculus valid for a subset of the relational algebra has been devised (see [2, 3] for details). In this context, the result of a query is a possibilistic relation whose interpretations correspond to more or less possible results, equivalent to those which would have been obtained with a calculus applied to worlds. This achievement is interesting from a methodological point of view, but the use of this type of result by a final user can be somewhat delicate. So, it becomes convenient to define queries which are more specialized and fit user needs. To meet this goal, possibilistic queries (concept initially introduced by Abiteboul [1] in the framework of null values) have been studied. Their generalized form is: "to what extent is it possible that the answer to Q satisfies property P". We have studied in [4] the case where P is: "contains a given (specified) tuple t". In this paper, we are interested in cardinality-based possibilistic queries where property P is: "contains at least (at most, . . .) q distinct elements" and we tackle the algorithmic aspects of their evaluation. To the best of our knowledge, there does not exist any previous research work about this issue.

The structure of the paper is the following. In section 2, the notion of a possibilistic relational database is introduced. Then, the data model requested for a valid compact processing of algebraic queries is described in section 3. Section 4 is devoted to a brief presentation of the algebraic operators which can be processed in a compact way. Section 3 and 4 are required for the comprehension of section 5. In section 5, we propose a "try and error" algorithm to process cardinality-based queries. Finally, the conclusion summarizes the contributions of the paper and draws some lines for future works.

2 Possibilistic databases and worlds

A possibilistic relational database D may have some attributes which take imprecise values. In such a case, a possibility distribution is used to represent all the more or less acceptable candidates. In this paper, only finite possibility distributions are considered.

From a semantic point of view, a possibilistic database D can be interpreted as a set of usual databases (worlds), denoted by rep(D), each of which being more or less possible (one of them is supposed to correspond to the actual state of the universe modeled). This view establishes a semantic connection between possibilistic and regular databases. It is particularly interesting since it offers a canonical approach to the definition of queries addressed to possibilistic databases as will be seen later (section 5). Any world W_i is obtained by choosing a candidate value in each possibility distribution appearing in D and its degree of possibility is the minimum of those of the candidates taken (according to the axioms of possibility theory).

Example 1. Let us consider the possibilistic database D involving two relations: im and pl whose respective schemas are IM(#i, ap, date, place) and PL(ap, lg, msp). Relation im describes satellite images of airplanes and each image, identified by a number (#i), taken on a certain location (place) a given day (date) is supposed to include a single (possibly ill known) airplane (ap). Relation pl gives the length (lg) and maximal speed (msp) of each airplane and is a regular relation. With the extension of im:

im	#i	ap	date	place
	i1	a ₁	{1/d ₁ + 0.7/d ₃ }	c ₁
	i3	{1/a ₃ + 0.3/a ₄ }	d ₁	c ₂

four worlds can be drawn, since there are two candidates for date (resp. ap) in the first (resp. second) tuple of im. Each of these worlds involves relation pl which has only precise values and one of the four regular relations issued from the possibilistic relation im.♦

3 An extended possibilistic data model

3.1 Objective

As mentioned before, a calculus based on the processing of the query Q against worlds is

intractable and a compact approach to the calculus of the answer to Q must be found out. It is then necessary to be provided with both a data model and operations which have "good" properties: i) the data model must be closed for the considered operations, and ii) any query (applying to the possibilistic database D) must be processed in a compact way. In addition, its result must be a compact representation of the results of this query if it were applied to all the interpretations (worlds) drawn from D, i.e.:

$$rep(Qc(D)) = Q(rep(D)),$$

where rep(D) denotes the set of worlds associated with D and Qc stands for the query obtained by replacing the operators of Q by their compact versions. This property characterizes data models called strong representation systems (see [6]).

A suitable data model has been defined in [2] and is briefly described here.

3.2 Representing possibly missing tuples

Because of some operations (e.g. selection), there is a need at the compact level for expressing that some tuples can have no representative in some worlds. A simple solution is to introduce a new attribute, denoted by N (valued in [0, 1]), which states whether or not it is legal to build worlds where no representative of the corresponding tuple is present, and, if so, the influence of this choice in terms of degree of possibility. The value of N associated with a tuple t expresses the certainty of the presence of a representative of t in any world. A tuple is denoted by a pair N/t where N equals 1 for tuples of initial possibilistic relations as well as when no candidate has been discarded.

Example 2. Let us consider the following extension of the possibilistic relation im:

im	#i	ap	date	place
	i ₁	B-727	d ₁	c ₁
	i ₂	ATR-72	d ₁	c ₂
	i ₃	{1/B-727 + 0.7/ATR-42}	d ₂	c ₄
	i ₄	{1/B-727 + 1/B-747}	d ₂	c ₂

The selection based on the condition "ap = B-727" discards the candidates which are different from this desired value. Thanks to the introduction of attribute N, the result of the selection is:

res	#i	ap	date	place	N
	i ₁	B-727	d ₁	c ₁	1
	i ₃	B-727	d ₂	c ₄	0.3
	i ₄	B-727	d ₂	c ₂	0

From res, it is possible to derive the interpretation made of the single tuple $\langle i_1, B-727, d_1, c_1 \rangle$ whose possibility degree is: $\min(1, 1 - 0.3, 1 - 0) = 0.7$. ♦

3.3 Multiple attribute possibility distributions

Another aspect of the model is related to the fact that it is sometimes necessary to express dependencies between candidate values coming from different attributes in a same tuple.

For instance, let us consider a given tuple t where the two attributes A and B take the imprecise values $t.A = \{a_1, a_2\}$ and $t.B = \{b_1, b_2, b_3\}$. If an operation retains only the pairs (a_1, b_1) and (a_2, b_3) , it is impossible to represent this situation with a Cartesian product of subsets of $t.A$ on the one hand and $t.B$ on the other hand, and the correct associations must be explicitly represented. This requires that the model incorporates attribute values defined as possibility distributions over several domains. This is feasible in the relational framework thanks to the concept of a nested relation. In such relations, exclusive candidates are represented as weighted tuples. Therefore, level-one relations keep their conjunctive meaning, whereas nested relations have a disjunctive interpretation.

Example 3. Let us consider the following intermediate relation int-r involving the nested attribute $X(\text{date}, \text{place})$:

int-r	#i	ap	X		N
			date	place	
	i ₁	B-727	$\{1/\langle d_1, c_1 \rangle + 0.7/\langle d_1, c_2 \rangle + 0.4/\langle d_3, c_2 \rangle\}$		1
	i ₃	B-727	$\langle d_1, c_2 \rangle$		0.3
	i ₄	$\{0.4/B-737\}$	$\{0.3/\langle d_3, c_2 \rangle\}$		0

This relation is associated with 12 worlds since the first tuple admits 3 interpretations, the second and third ones have two interpretations among which \emptyset ♦

4 Compact version of the operators

In order to meet the objective of a compact processing of algebraic queries, the operators must be adapted so as to accept compact relations both as inputs and outputs. It turns out that only the selection, projection, fk-join and union admit a compact version. Due to space limitations, we limit ourselves to a brief introduction and their behavior is then illustrated by an example. See [2, 3] for more details about the definition of these operators.

4.1 Unary operations

The three roles of the selection are: the removal of unsatisfactory candidate values, the computation of the degree of certainty attached to each output tuple and the introduction of appropriate nested relations in the output relation if needed.

The role of the projection in the regular case is to remove undesired attributes. Here, the projection must: 1) keep the duplicates in level-one relations, 2) suppress nested relations if necessary, 3) update the possibility degrees.

4.2 Binary operations

Beyond selections and projections, two binary operations can be processed in a compact fashion: union and fk-join . The latter allows for the composition of a possibilistic relation r of schema $R(W, Z)$, where W and Z may take imprecise values, and a regular relation s whose schema is $S(W, Y)$ where the functional dependency $W \rightarrow Y$ holds. It consists in completing tuples of r by adding the image of the W -component. By definition, this leads to a resulting relation involving the nested relation $X(W, Y)$, which "connects" the pairs of candidates over W and Y .

Last, the union of two independent relations whose schemas are compatible keeps all the tuples issued from the two input relations without any duplicate removal.

Example 4. Let us consider the possibilistic database composed of the relations im1(IM) , im2(IM) and pl(PL) where IM and PL are the schemas introduced in example 1. The relations im1 and im2 are assumed to contain images of airplanes taken by two distinct satellites. Let us take the query Q : $\text{fk-join}(\text{union}(\text{select}(\text{im1}, \text{date} \notin \{d_3, d_4\}), \text{select}(\text{im2}, \text{date} \notin \{d_3, d_4\})), \text{select}(\text{pl}, \text{msp} > 900), \{\text{ap}\}, \{\text{ap}\})$.

With the extensions given hereafter:

pl	ap	lg	msp
	a ₁	20	1000
	a ₂	25	800
	a ₄	20	1200
	a ₅	20	1000

im1	#i	ap	date	place	N
	i ₁	a ₃	{1/d ₁ + 0.7/d ₃ }	c ₁	1
	i ₂	{1/a ₂ + 0.7/a ₁ }	d ₁	c ₂	1

im2	#i	ap	date	place	N
	i ₃	{1/a ₄ + 1/a ₅ }	{0.6/d ₄ + 1/d ₁ }	c ₃	1

We obtain the resulting relation **res** hereafter:

res	#i	X		date	place	N
		ap	lg	msp		
	i ₂	{0.7/<a ₁ , 20, 1000>}		d ₁	c ₂	0
i ₃	{1/<a ₄ , 20, 1200> + 1/<a ₅ , 20, 1000>}		{1/d ₁ }	c ₃	0.4	

5 Cardinality-based queries

5.1 Introducing a post processing

On the basis of the definitions of the algebraic operators evoked above, a query can be processed in a tractable way since operations are performed in a compact fashion. However, one may wonder about the usability of the result delivered by such a query, i.e., of a compact relation as such. We think that a convenient direction is to provide users with queries which are close to their needs and which call on (embedded) algebraic queries. Hereafter, we are interested in cardinality-based possibilistic queries, which are of the form: "to what extent is it possible that the answer to Q has at least (at most, exactly, ...) q distinct elements?", and for the sake of space we only deal with the case "at least". The evaluation of such queries is based on a two-step mechanism:

- 1) a compact processing of the embedded algebraic query, which builds a compact relation according to the procedure depicted in the preceding section,
- 2) a post processing producing the final answer.

5.2 Problem raised by cardinality-based queries

The post processing of the compact result of Q aims at the determination of the possibility attached to worlds involving a certain number of elements.

The problem is that tuples of the resulting relation can have representatives which are indeed duplicates. For instance, if the result of the compact processing of the algebraic query Q is:

res	A	B	N
	{1/a ₁ + 0.6/a ₂ }	b	0.3
	a ₁	b	1

the degree of possibility that the answer contains at least 2 different tuples cannot be obtained by taking the most possible representative of the two tuples of **res** because they are identical (<a₁, b>). That is the reason why the procedure attached to the post processing must rely on a "try and error" (or a "branch and bound") technique in order to identify the most satisfactory world with respect to the desired cardinality. For instance, the query:

"to what extent is it possible that there exists at least two shots with an aircraft of maximal speed over 900 km/h, taken on a date different from d₃ and d₄?"

boils down to determining the possibility of the most possible world issued from table **res** delivered in example 4, which contains at least 2 tuples. It turns out that there is such a world (indeed involving exactly two tuples), which is possible at the degree: min(0.7, 1) = 0.7. If "at least two" is replaced by "at least 3" (or more) in the query, the degree obtained becomes 0.

5.3 The algorithm

The algorithm proposed here is based on a "try and error" technique. It aims at delivering the possibility degree π of the most possible world satisfying the desired cardinality criterion. Such an algorithm calculates a series of vectors $V = (x_1, x_2, \dots, x_n)$ where each component x_i takes its values in a finite set E_i . Ultimately, it aims at finding the best solution, i.e., the best vector V.

A solution is a vector V which represents a world associated with the possibilistic relation **res** resulting from the compact evaluation of Q. Its dimension is the number n of tuples in relation **res**. The components of vector V are precise tuples (the ith position of V is the representative tuple produced by the ith tuple of relation **res**). Some positions of V may be empty, which occurs when the value N in

relation **res** is different from 1 (the corresponding tuple may have no representative in a given world). The algorithm and the corresponding data structures are the following:

```

Procedure optimalSolution(i integer)
begin
  compute Ei;
  for xj in Ei do
    if satisfactory(xj) then
      memorize(xj);
      if solutionFound then
        if better then keepSolution endif
      else if stillPossible then
        optimalSolution(i + 1) endif
      endif;
    undo(xj);
  endif;
endfor;
end;

```

E_i: list of the precise tuples corresponding to the possible representatives of the ith tuple from **res** (including \emptyset if $N < 1$);

Π_i: list of the respective possibility degrees π_j of each x_j in E_i ($1 - N$ if x_j is \emptyset);

V: represents a world of **res**;

Pos: vector of the same dimension as **V** containing the possibility degrees of the tuples of **V** (the possibility degree associated with **V** is the minimum over **Pos**);

Card: cardinality of vector **V** (the number of tuples in **V** different from \emptyset);

BestΠ: possibility degree of the most possible world found until then;

satisfactory(x_j): $\pi_j > \text{Best}\Pi$;

memorize(x_j):

$V[i] \leftarrow x_j$;

$\text{Pos}[i] \leftarrow \pi_j$;

$b \leftarrow (x_j \neq \emptyset \text{ and } x_j \text{ is not already present in } V)$;

if b then $\text{Card} \leftarrow \text{Card} + 1$;

solutionFound: ($i = n$) and $\text{Card} \geq q$;

better: $\min_{k \text{ in } [1, n]} \text{Pos}[k] > \text{Best}\Pi$;

keepSolution: $\text{Best}\Pi \leftarrow \min_{k \text{ in } [1, n]} \text{Pos}[k]$;

stillPossible: $\text{Card} + (n - i) \geq q$;

undo(x_j): $\text{Pos}[i] \leftarrow 0$; if b then $\text{Card} \leftarrow \text{Card} - 1$.

Remarks. In order to reduce the number of worlds to be computed, the following improvements to the algorithm above can be introduced:

- 1) When **BestΠ** is equal to 1 the processing can be stopped.
- 2) The sets E_i can be ranked in decreasing order on

the possibility degrees. In that case, once an unsatisfactory x_j is found, the loop can be stopped (because the following x -values would be unsatisfactory too). However, this ordering does not prevent from computing the whole set of worlds in some case (when the only satisfying world is the last one built).

Another improvement consists in taking advantage of the number n of tuples in relation **res**. For instance, if the user is interested in at least 5 responses ($q = 5$) while the relation **res** contains only 3 tuples, the result is obviously 0.

As for any of the algorithms of this family, the complexity of the previous procedure is exponential. Let us consider a relation **res** containing n tuples and m imprecise attributes. Let us assume that there are p candidate values per possibility distribution, and that, for each tuple, $N=1$. The time complexity in terms of recursive calls (which also corresponds to the number of computed worlds) is in $O((p^{m \cdot n})$, but one may hope reduce the number of computed worlds thanks to the pruning conditions above. In any case, this processing remains more efficient than the one which consists in computing the worlds of the database, then evaluating the cardinality-based query on each of them, since the computation of worlds here is limited to one relation (**res**). It remains also more efficient than the strategy which would consist in computing the worlds of relation **res**, then testing the cardinality of each of them, since one would then lose the possibility of discarding the non-satisfactory worlds before their computation.

5.4 Example

Let us consider the possibilistic relation **res**, which is assumed to be the result of the compact processing of a given algebraic query Q :

res	A	B	N
	a_2	$\{1/b_3 + 0.9/b_2\}$	1
	a_2	b_3	1
	$\{1/a_2 + 0.5/a_1\}$	b_3	0.4
	$\{0.8/a_4 + 0.6/a_5\}$	$\{0.7/b_1\}$	0

and the query: "to what extent is it possible that relation **res** has at least two distinct elements?"

The corresponding lists E_i and Π_i , after ranking them in decreasing order on the possibility degrees, are:

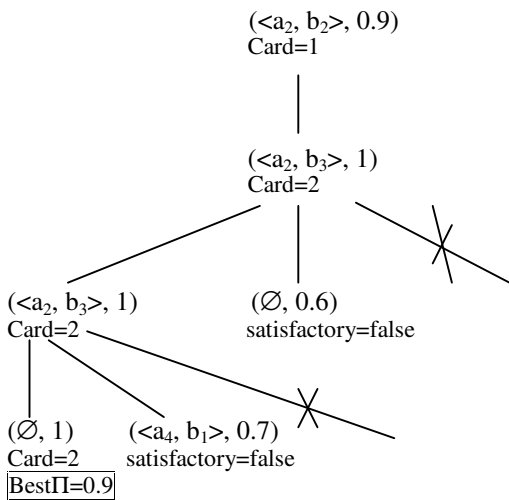
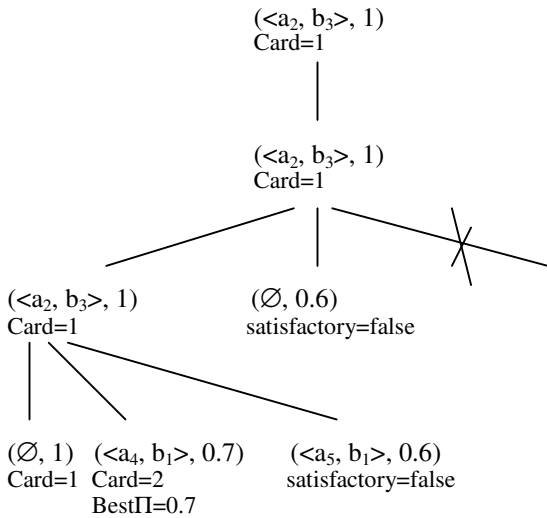
$$E_1 = (\langle a_2, b_3 \rangle, \langle a_2, b_2 \rangle) \quad \Pi_1 = (1, 0.9)$$

$$E_2 = (\langle a_2, b_3 \rangle) \quad \Pi_2 = (1)$$

$$E_3 = (\langle a_2, b_3 \rangle, \emptyset, \langle a_1, b_3 \rangle) \quad \Pi_3 = (1, 0.6, 0.5)$$

$$E_4 = (\emptyset, \langle a_4, b_1 \rangle, \langle a_5, b_1 \rangle) \quad \Pi_4 = (1, 0.7, 0.6)$$

To illustrate how the algorithm works on this example, we use a tree as a representation. The tree is split into two parts for space reasons.



In the first part of the tree, and for the node $(\emptyset, 0.6)$, $\text{satisfactory}(\emptyset)$ is false since at that point of the computations $\text{BestPI} = 0.7$. Given that the possibility degrees π_j are ranked in decreasing order, the following children of the parent of $(\emptyset, 0.6)$ are not generated. The same reasoning can be done for the other crossed edges. Thus, we compute 5 worlds instead of 18 thanks to the pruning conditions above.

6 Conclusion

This paper addresses the issue of querying relational databases where some attribute values are imprecise and represented by possibility distributions. An adapted model with a subset of the relational algebra has been briefly presented. In this context, the result of a query is a possibilistic relation, which may not be easily interpretable by a final user. This situation led us to consider a new type of queries called possibilistic queries, whose general form is: "to what extent is it possible that the answer to Q satisfies property P?". In this paper, cardinality-based possibilistic queries have been investigated. Their treatment is based on a two-step mechanism. The first step is the evaluation of the algebraic query involved, and the second one is a post processing which relies on a "try and error" technique.

This work opens different lines for future research. One of them is related to the performances obtained for cardinality-based queries. It would be interesting to assess in a more precise way the additional cost linked to the presence of imprecise data (with respect to similar queries on precise data). Also, the work presented should also be extended in order to deal with predicates of the form: $(\text{card} \leq q)$ or $(\text{card} = q)$ but this should not raise major problems.

References

- [1] S. Abiteboul, P. Kanellakis, and G. Grahne, "On the representation and querying of sets of possible worlds", *Theoret. Comput. Sci.*, vol. 78, pp. 159-187, 1991.
- [2] P. Bosc and O. Pivert, "Towards an algebraic query language for possibilistic databases", 12th Conference on Fuzzy Systems (FUZZ-IEEE'03), pp. 671-676, 2003.
- [3] P. Bosc and O. Pivert, "About projection-selection-join queries addressed to possibilistic relational databases", *IEEE Transactions on Fuzzy Systems*, vol. 31, pp. 124-139, 2005.
- [4] P. Bosc, L. Duval, and O. Pivert, "An initial approach to the evaluation of possibilistic queries addressed to possibilistic databases", *Fuzzy Sets and Systems*, vol. 140, pp. 151-166, 2003.
- [5] P. Bosc and H. Prade, "An introduction to the treatment of flexible queries and uncertain or imprecise databases", in: *Uncertainty Management in Information Systems*, A. Motro and P. Smets (Eds), Kluwer Academic Publishers, pp. 285-324, 1997.
- [6] T. Imielinski and W. Lipski, "Incomplete information in relational databases", *Journal of the ACM*, vol. 31, pp. 761-791, 1984.