Imprecise Data and Knowledge Based OLAP

Ermir Rogova[1], Panagiotis Chountas[1], Krassimir Atanassov[2]

[1]Harrow School of Computer Science, Data & Knowledge Management Group
University of Westminster
Watford Road, Northwick Park, HA1 3TP
London, UK

[2]CLBME – Bulgarian Academy of Sciences, B1. 105, Sofia-1113, Bulgaria

rogovae@wmin.ac.uk

Abstract. In this paper we present our approach for extending the OLAP model to include treatment of value uncertainty as part of a multidimensional model inhabited by flexible data and non-rigid hierarchical structures of organisation. A new multidimensional-cubic model named as the IF-Cube is introduced which is able to operate over data with imprecision either in the facts or in the dimensional hierarchies.

1. Introduction

In this paper we introduce the semantics of the Intuitionistic Fuzzy cubic representation in contrast to the basic multidimensional-cubic structures. The basic cubic operators are extended and enhanced with the aid of [1], [2] Intuitionistic Fuzzy Logic.

Since the emergence of the OLAP technology [3] different proposals have been made to give support to different types of data and application purposes. One of this is to extend the relational model (ROLAP) to support the structures and operations typical of OLAP. Further approaches [4], [5] are based on extended relational systems to represent data-cubes and operate over them. The other approach is to develop new models using a multidimensional view of the data [6].

Nowadays, information and knowledge-based systems need to manage imprecision in the data and more flexible structures are needed to represent the analysis domain. New models have appeared to manage incomplete datacube [7], imprecision in the facts and the definition of fact using different levels in the dimensions [8].

2. Semantics of the IF-Cube in contrast to Crisp Cube

Each element of an Intuitionistic fuzzy [1], [2] set has degrees of membership or truth (μ) and non-membership or falsity (ν), which don't sum up to 1.0 thus leaving a degree of hesitation margin (π).

Y. Ioannidis, B. Novikov and B. Rachev (Eds.): Local Proceedings of ADBIS 2007, pp. 50-55

As opposed to the classical definition of a fuzzy set given by $A' = \{< x, \mu_{A'}(x) > | x \in X\}$ where $\mu_A(x) \in [0, 1]$ is the membership function of the fuzzy set A', an intuitionistic fuzzy set A is given by:

$$A = \{< x, \mu_A(x), \nu_A(x) > | x \in X\}$$

where: $\mu_A : X \rightarrow [0, 1]$ and $\nu_A : X \rightarrow [0, 1]$ such that $0 \leq \mu_A(x) + \nu_A(x) \leq 1$ and $\mu_A(x)$ $\nu_A(x) \in [0, 1]$ denote a degree of membership and a degree of non-membership of $x \in A$, respectively.

Obviously, each fuzzy set may be represented by the following Intuitionistic fuzzy set

$$A=\{<x, \mu_{A'}(x), (x), 1-\mu_{A'}(x)>|x \in X\}$$

For each intuitionistic fuzzy set in X, we will call $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ an intuitionistic fuzzy index (or a hesitation margin) of $x \in A$ which expresses a lack of knowledge of whether x belongs to A or not. For each $x \in A$ $0<\pi_A(x)<1$.

The **IF-Cube** is an abstract structure that serves as the foundation for the multidimensional data cube model. Cube C is defined as a five-tuple *(D, l, F, O, H)* where:

- *D* is a set of dimensions
- *l* is a set of levels $l_1, \ldots, l_n$,
- A dimension $d_i = (l \leq O, l_\perp, l_\top)$ $dom(d_i)$ where $l = l_i$ $i=1...n$.
  l_i is a set of values and $l_i \cap l_j = \{\}$,
  $\leq O$ is a partial order between the elements of *l*.
  To identify the level *l* of a dimension, as part of a hierarchy we use *dl.*
  $l_\perp$: base level $l_\top$: top level
  for each pair of levels l_i and l_j we have the relation
  $\mu_{ij} : l_i \times l_j \rightarrow [0,1]$ $\nu_{ij} : l_i \times l_j \rightarrow [0,1]$ $0 < \mu_{ij} + \nu_{ij} < 1$
- *F* is a set of fact instances with schema $F = \{<x, \mu_F(x), \nu_F(x)>| x \in X\}$, where $x=<att_1, \ldots, att_n>$ is an ordered tuple belonging to a given universe *X*, $\mu_F(x)$ and $\nu_F(x)$ are the degree of membership and non-membership of *x* in the fact table *F* respectively.
- *H* is an object type history that corresponds to a cubic structure(l, F, O, H') which allows us to trace back the evolution of a cubic structure after performing a set of operators i.e. aggregation.

3. Cubic operators

Selection (Σ): The selection operator selects a set of fact-instances from a cubic structure that satisfy a predicate (θ). A predicate (θ) involves a set of atomic predicates $(\theta_1, \ldots, \theta_n)$ associated with the aid of logical operators *p (* i.e. $\wedge$, $\vee$, etc.) . The set of possible facts (cubic instances) that satisfy the θ should carry a degree of membership μ and non-membership *v* expressed as

$F = \{<x, min(\mu_F(x), \mu(\theta(x))), max(\nu_F(x), \nu(\theta(x)))> | x \in X\}$

Input: $C_i =$ *(D, l, F, O, H)* and the predicate θ

Output: $C_o =$ *(D, l, F_o, O, H)* where $F_o \subseteq F$ and $F_o=\{f | (f \in F) \wedge (f$ satisfies θ)

Mathematical notation: $\sum_\theta (C_i) = C_o$

Cubic Product ($\otimes$): This is a binary operator $C_{i1} \otimes C_{i2}$. It is used to relate two cubes C_{i1} and C_{i2} assuming that $D_1 \subseteq D_2$ and O_1 , O_2 are reconcilable partial orders. Thus, l_1, l_2 could lead to l_o being a ragged hierarchy.

Input: $C_{i1} = (D_1, l_1, F_1, O_1, H_1)$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$

Output: $C_o = (D_o, l_o, F_o, O_o, H_o)$ where

$D_o = D_1 \cup D_2$, $l_o = l_1 \cup l_2$, $O_o = O_1 \cup O_2$ $H_o = H_1 \cup H_2$, $F_o = F_1 \text{ X } F_2$

$F_o = \{<<x, y>, \min(\mu_{f1}(x), \mu_{f2}(y)), \max(\nu_{f1}(x), \nu_{f2}(y),)> | <x, y> \in X \times Y\}$

Mathematical notation: $C_{i1} \otimes C_{i2} = C_o$

Join (Θ): It can be expressed using Cubic Product operator. $C_{i1} = (D_1, l_1, F_1, O_1, H_1))$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$ are candidates to join if $D_1 \cap D_2 \neq 0$,

Input: $C_{i1} = (D_1, l_1, F_1, O_1, H_1)$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$

Output: $C_o = (D_o, l_o, F_o, O_o, H_o)$

Mathematical notation: $C_{i1} \Theta C_{i2} = \sigma_p(C_{i1} \otimes C_{i2})$

Union ($\cup$): The union operator is a binary operator that finds the union of two cubes. C_{i1} and C_{i2} have to be union compatible. The operator also coalesces the value-equivalent facts using the minimum membership and maximum non-membership.

Input: $C_{i1} = (D_1, l_1, F_1, O_1, H_1)$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$

Output: $C_o = (D_o, l_o, F_o, O_o, H_o)$ where $D_o = D_1 = D_2$, $l_o = l_1 = l_2$, $O_o = O_1 = O_2$, $H_o = H_1 = H_2$, $F_o = F_1 \cup F_2 = \{ < x, max(\mu F_1(x), \mu F_2(x)), min(\nu F_1(x), \nu F_2(x)) > | \ x \in X \}$

Mathematical notation: $C_{i1} \cup C_{i2} = C_o$

Difference (-):. The difference operator removes the portion of the cube C_{i1} that is common to both cubes. C_{i1} and C_{i2} have to be union compatible

Input: $C_{i1} = (D_1, l_1, F_1, O_1, H_1)$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$

Output: $C_o = (D_o, l_o, F_o, O_o, H_o)$ where $D_o = D_1 = D_2$, $l_o = l_1 = l_2$, $O_o = O_1 = O_2$, $H_o = H_1 = H_2$,

$F_o = F_1 \cap F_2 = \{ < x, \min(\mu F_1(x), \mu F_2(x)), \max(\nu F_1(x), \nu F_2(x)) > | \ x \in X \}$

Mathematical notation: $C_{i1} - C_{i2} = C_o$

Aggregation (A): An aggregation operator A is a function $A(G)$ where $G = \{<x, \mu_F(x) , \nu_F(x)>| \ x \in X \}$ where $x = <att_1, ..., att_n>$ is an ordered tuple belonging to a given universe X, $\{att_1, ..., att_n\}$ is the set of attributes of the elements of X, $\mu_F(x)$ and $\nu_F(x)$ are the degree of membership and non-membership of x. The result is a bag of the type $\{<x', \mu_F(x') , \nu_F(x')>| \ x' \in X \}$. To this extent, the bag is a group of elements that can be duplicated and each one has a degree of μ and ν.

Input: $C_i = (D, l, F, O, H)$ and the function $A(G)$

Output: $C_o = (D, l_o, F_o, O_o, H_o)$

The definition of the extended group operators allows us to define the extended group operators ***Roll up (Δ), and Roll Down (Ω).***

Roll up (Δ): The result of applying Roll up over dimension d_i at level dl_r using the aggregation operator A over a datacube $C_i = (D_i, l_i, F_i, O, H_i)$ is another datacube $C_o = (D_o, l_o, F_o, O, H_o)$.

Input: $C_i = (D_i, l_i, F_i, O, H_i)$

Output: $C_o = (D_o, l_o, F_o, O, H_o)$

An object of type history is a recursive structure $H = \begin{cases} \omega \text{ is the initial state of the cube} \\ (l, D, A, H') \text{ is the state of the cube after performing an operation on the cube} \end{cases}$

The structured history of the datacube allows us to keep all the information when applying *Roll up* and get it all back when *Roll Down* is performed. To be able to apply the operation of *Roll Up* we need to make use of the IF_{SUM} aggregation operator.

Roll Down (Ω): This operator performs the opposite function of the *Roll Up* operator. It is used to roll down from the higher levels of the hierarchy with a greater degree of generalization, to the leaves with the greater degree of precision. The result of applying *Roll Down* over a datacube $C_i = (D, l, F, O, H)$ having $H=(l', D', A', H')$ is another datacube $C_o= (D', l', F', O, H')$.

Input: $C_i=(D, l, F, O, H)$

Output: $C_o=(D',l', F', O, H')$ where $F' \rightarrow$ *set of fact instances defined by operator A.*

To this extent, the *Roll Down* operative makes use of the recursive history structure previously created after performing the *Roll Up* operator.

The definition of aggregation operator points to the need of defining the IF extensions for traditional group operators [9], [10], [11] such as *SUM, AVG, MIN and MAX*. Based on the standard group operators, we provide their IF extensions and meaning.

IF_{SUM} : The IF_{sum} aggregate, like its standard counterpart, is only defined for numeric domains. Given a fact F defined on the schema X $(att_1, ...,att_n)$, let att_{n-1} defined on the domain $U=\{u_1 , ..., u_n)$. The fact F consists of fact instances F_i *with* $1 \leq i \leq m$. The fact instances F_i are assumed to take Intuitionistic Fuzzy values for the attribute att_{n-1} for $i = 1$ *to* m we have $F_i[att_{n-1}] = \{<\mu_i(u_{ki}), v_i(u_{ki})>/ u_{ki} \mid 1 \leq k_i \leq n \}$. The IF_{sum} of the attribute att_{n-1} of the fact table F is defined by:

$$IF_{SUM}((att_{n-1})(F)) =$$

$$\{<u>/ y \mid ((u= \min_{i=1}^{m} (\mu_i(u_{ki}), v_i(u_{ki})) \wedge (y = \sum_{ki=k1}^{km} u_{ki}) (\forall_{k1, ...km} : 1 \leq k1, ...km \leq n))\}$$

IF_{AVG} : The IF_{AVG} aggregate, like its standard counterpart, is only defined for numeric domains. This aggregate makes use of the IF_{SUM} that was discussed previously and the standard *COUNT*. The IF_{AVG} can be defined as:

$$IF_{AVG}((att_{n-1})(F) = IF_{SUM}((att_{n-1})(F)) / COUNT((att_{n-1})(F))$$

IF_{MAX} : The IF_{MAX} aggregate, like its standard counterpart, is only defined for numeric domains. The IF_{sum} of the attribute att_{n-1} of the fact table F is defined by:

$$IF_{MAX}((att_{n-1})(F)) =$$

$$\{<u>y|((u= \min_{i=1}^{m} (\mu_i(u_{ki}),v_i(u_{ki}))\wedge(y= \max_{i=1}^{m} (\mu_i(u_{ki}),v_i(u_{ki})))(\forall_{k1, ...km} :1\leq k1, ...km\leq n))\}$$

IF_{MIN} : The IF_{MIN} aggregate, like its standard counterpart, is only defined for numeric domains. Given a fact F defined on the schema X $(att_1, ...,att_n)$, let att_{n-1} defined on the domain $U=\{u_1 , ..., u_n)$. The fact F consists of fact instances f_i *with* $1 \leq i \leq m$. The fact instances f_i are assumed to take Intuitionistic Fuzzy values for the attribute att_{n-1} for $i = 1$ *to* m we have $f_i[att_{n-1}] = \{<\mu_i(u_{ki}), v_i(u_{ki})>/ u_{ki} \mid 1 \leq k_i \leq n \}$. The IF_{sum} of the attribute att_{n-1} of the fact table F is defined by:

$$IF_{MIN}((att_{n-1})(F)) =$$

$$\{<u>/ y|((u= \min_{i=1}^{m} (\mu_i(u_{ki}),v_i(u_{ki}))\wedge(y= \min_{i=1}^{m} (\mu_i(u_{ki}),v_i(u_{ki})))(\forall_{k1, ...km} :1\leq k1, ...km\leq n))\}$$

We can observe that the IF_{MIN} is extended in the same manner as IF_{MAX} aggregate except for replacing the symbol **max** in the IF_{MAX} definition with **min**.

Once we have defined our Intuitionistic Fuzzy multidimensional model and have defined the IF cubic-algebra, the concept of knowledge based OLAP is introduced.

4. The Case for Knowledge Based OLAP-KNOLAP

Let us consider the Intuitionistic fuzzy set M defined as: {Milk<0.8,0.1>, Whole-Milk<0.7,0.1>, Condensed-Milk<0.4,0.3>}} which is presented in "Fig.1". Then the next step is to calculate the <μ, ν> values for "Pasteurized milk", "Whole Pasteurized milk " and "Condensed whole milk."

- If the hierarchical IF structure expresses preferences in a query, the choice of the maximum values for μ and minimum value ν from the pairs of values <μ, ν> from the parent elements to the sub elements allows us not to exclude any possible answer (high possibility necessity degrees). In real cases, the lack of answers to a query generally makes this choice preferable, because it consists of widening the query answer rather than restricting it.
- If the hierarchical IF represents an ill-known concept, the choice of the maximum value for μ and minimum value ν allows us to preserve all the possible values, but it also makes the answer less specific. In a way, it also participates in enlarging the query, as a less specific datum may share more common values with the query (the possibility degree of matching can thus be higher, although the necessity degree can decrease).

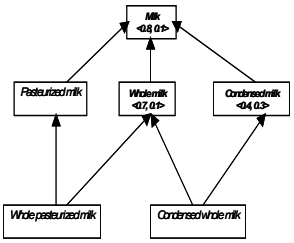

Fig. 1. "IF Hierarchy 'Milk' "

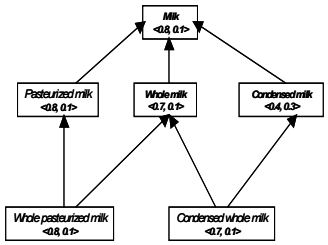

Fig. 2. "Fully weighted Hierarchy 'Milk' "

"Fig.2" is a fully weighted Hierarchy after applying the maximum values for μ and minimum value ν from the pairs of values <μ, ν> from the parent elements to the sub elements, i.e. from (whole-milk, condensed-milk) to (condensed-whole-milk), from (milk) to (pasteurized milk), and from (whole-milk, pasteurized milk) to (pasteurized-whole-milk).

The complete study of the hierarchical IF requires the formal definition of the IF hierarchical closure. We will further need to formally define the containment of an IF hierarchical set to another.

5. Conclusions

In this paper we have presented a new multidimensional-cubic model named as the IF-Cube. The main contribution of this new model is that is able to operate over data with imprecision in the facts and the summarisation hierarchies. Classical

models imposed a rigid structure that made the models present difficulties when merging information from different but still reconcilable sources. We introduce the automatic recommendation of analysis according to the context of users' explorations in order to guide the decision making with the aid of Intuitionistic fuzzy set over a universe that has a hierarchical structure and the corresponding hierarchies.

There is finally a need to formally define the closure of Intuitionistic fuzzy set over a universe that has a hierarchical structure as well the containment between different versions of these sets.

References

[1] K., Atanassov (1999). *Intuitionistic Fuzzy Sets*, Springer-Verlag, Heidelberg
[2] K., Atanassov *Intuitionistic Fuzzy Sets,* Fuzzy Sets and Systems, 20, 87–96, (1986)
[3] R. Kimball, *The Data Warehouse Toolkit.* New York: John Wiley & Sons, 1996.
[4] S., Chaudhuri U. Dayal V., Ganti. Database Technology for Decision Support Systems. In: Computer, Vol. 34, p. 48-55, 2001
[5] M., Jarke et al. Fundamentals of data warehouses. Springer, London, 2002
[6] H. Thomas & A., Datta, A Conceptual Model and Algebra for On-Line Analytical Processing in Decision Support Databases. Information Systems Research 12: 83-102, 2001
[7] C. Dyreson, Information retrieval from an incomplete data cube, VLDB, Morgan Kaufman Publishers, pp. 532-543, 1996.
[8] T. Pedersen, C. Jensen, and C. Dyreson, A foundation for capturing and querying complex multidimensional data, Information Systems, vol. 26, pp. 383-483, 2001.
[9]D., Dubois et al (1988). Handling Incomplete or Uncertain Data and Vague Queries in Database Applications, Plenum Press, 1988
[10] H., Prade (1993). Annotated bibliography on fuzzy information processing. *Readings on Fuzzy Sets in Intelligent Systems*, H. Prade, D. Dubois, and R. Yager, Eds. Morgan Kaufmann Publishers Inc., 1993
[11]E. Rundensteiner L. Bic. Aggregates in posibilistic databases, VLDB'89, pp. 287-295, 1989