DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

ONDERZOEKSRAPPORT NR 9519

Representing Fuzzy Decision Tables in a Fuzzy Relational Database Environment

by

Guoqing CHEN

Jan VANTHIENEN

Geert WETS

Katholieke Universiteit Leuven

Naamsestraat 69, B - 3000 Leuven

ONDERZOEKSRAPPORT NR 9519

Representing Fuzzy Decision Tables in a Fuzzy Relational Database Environment

by

Guoqing CHEN

Jan VANTHIENEN

Geert WETS

D/1995/2376/20

Representing Fuzzy Decision Tables in a Fuzzy Relational Database Environment[1]

Guoqing CHEN[2], Jan VANTHIENEN, Geert WETS
Katholieke Universiteit Leuven
Department of Applied Economic Sciences
Naamsestraat 69, B-3000 Leuven (Belgium)
E-mail: jan.vanthienen@econ.kuleuven.ac.be
geert.wets@econ.kuleuven.ac.be

Keywords: fuzzy relational databases, fuzzy decision tables, fuzzy decision making

Abstract *In this paper the representation of decision tables in a relational database environment is discussed. First, crisp decision tables are defined. Afterwards a technique to represent decision tables in a relational database system is presented. Next, fuzzy extensions are made to crisp decision tables in order to deal with imprecision and uncertainty. As a result, with crisp decision tables as special cases fuzzy decision tables are defined which include fuzziness in the conditions as well as in the actions. Analogous to the crisp case, it is demonstrated how fuzzy decision tables can be stored in a fuzzy relational database environment. Furthermore, consultation of these tables is discussed using fuzzy queries.*

1. Introduction

Fuzzy relational databases (FRDBs) are the generalizations of the classical relational data model in order to represent and manipulate imprecise and uncertain information. Fuzzy decision tables (FDTs) are the extensions of the classical decision table (DT) formalism in order to deal with imprecise and uncertain decision situations. Currently, DTs are used to represent complex decision situations in a simple manner easy to check for completeness, exclusivity and correctness [12].
This paper examines how the relational approach can be used to represent FDT knowledge, which may further allow fuzzy decision making with extended SQL facilities.

The paper is organized as follows. In section 2 the decision table concept is introduced. Next, the relationship between DTs and RDBs is explained. In Section 4 and section 5 FDTs and FRDBs are defined. Section 6 illustrates how FDTs can be implemented and consulted using FRDBs. Finally, some concluding remarks and directions for future work are given.

[1] To be presented at the Third European Congress on Intelligent Techniques and Soft Computing (EUFIT'95), Aachen, August 28-31, 1995.

[2] Currently at MIS Department, School of Economics, Tsinghua University, Beijing.

2. The Decision Table Concept

Many variations of the decision table concept exist which look similar at first sight [14]. In practice one has to distinguish between some major kinds of tables, with the decision grid chart at one end of the spectrum and the real decision table at the other end.

The most important criterion when distinguishing tables, is the question whether all columns are mutually exclusive (*single hit* versus *multiple hit*). In a single hit table each possible combination of conditions can be found in exactly one (one and only one) column. This makes an unambiguous use of the table possible.

If the columns are not exclusive, some combination of conditions is present in more than one column, which may lead to ambiguity or inconsistency. When consulting the table, the first hit rule will often be used. This rule states that the *first hit* (from left to right) will determine which set of actions has to be executed, thus preventing contradictions.

Another possibility is that *all hits* are used to determine the set of actions to be executed. In this case, each hit from left to right can add actions (not mentioned by previous columns) or delete actions (overwriting previous columns) to the set. An interesting concept of this latter form is the so called decision grid chart (a tabular (action by action) representation of a set of decision rules.

In both multiple hit cases (first hit versus all hits) the same combination of conditions can occur in different columns. As a result the overview over the columns is lost, and with it, the simplicity of inspection. For these reasons we do not consider these tables to be real decision tables.

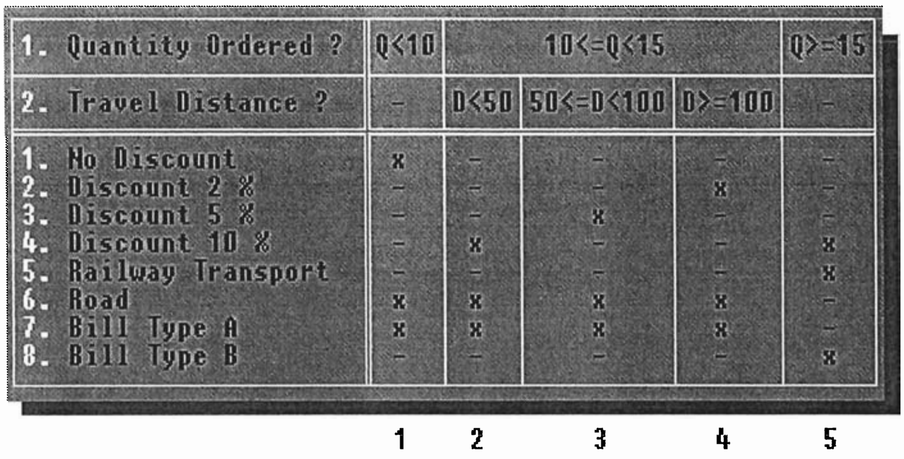
A decision table consists of four parts:

1. The condition subjects are the criteria which are relevant to the decision making process. They represent the items about which information is needed to take the right decision. Condition subjects are found in the upper left part of the table.
2. The condition states are logical expressions determining the relevant sets of values for a given condition. Every condition has its set of condition states. Condition states are found at the right hand side of the table.

3. The action subjects describe the results of the decision making process. They are found in the lower left part of the table.

4. The action values are the possible values a given action can take. They are found at the right hand side of the table.

In figure 1 an example of a decision table is shown.



	1	2	3	4	5
1. Quantity Ordered ?	Q<10	10<=Q<15			Q>=15
2. Travel Distance ?	-	D<50	50<=D<100	D>=100	-
1. No Discount	x	-	-	-	-
2. Discount 2 %	-	-	-	x	-
3. Discount 5 %	-	-	x	-	-
4. Discount 10 %	-	x	-	-	x
5. Railway Transport	-	-	-	-	x
6. Road	x	x	x	x	-
7. Bill Type A	x	x	x	x	-
8. Bill Type B	-	-	-	-	x

Figure 1 A sample decision table

Each table column represents a decision rule of the form:

IF CS_1 is S_{1k} AND CS_2 is S_{2m} AND ...

THEN action AS_j AND ...

If each column only contains simple states (no contractions or irrelevant conditions), the table is called an *expanded* decision table *(canonical form)*, in the other case the table is called a *contracted* decision table *(consolidated form)*. The translation from one form to the other is defined as *expansion (rule expansion)* and *contraction (consolidation)* respectively (CODASYL [8]).

The condition subjects and action subjects can refer to other tables (*subtables*). The replacement of these references by the tables themselves, the *junction* of tables, is called (*table*) *expansion*. The reverse process, the *division* into subtables, is defined as *factoring*. Two types of subtables are possible: the action subtable, i.e. a further specification of a certain action, and the condition subtable, determining the value of a condition. All subtables are of the closed type, this means that after ending a subtable, the calling table regains control.

Some combinations of conditions may be *impossible*, in other words, they *cannot* occur. Such combinations may be deleted from the table. Keep in mind that only real impossibilities are to be deleted, combinations that *should not* occur must stay in the table, since they will occur at some point in time (according to Murphy's Law).

3. The decision table as database relation

The decision table can be seen as a set of ordered n-tuples ($ct_1, ..., ct_{cnum}, av_1, ..., av_{anum}$), with $ct_i \in CT_i$ and $av_j \in AV_j$, that can be represented as a relational table. This relational table, as representation of a decision table, has the following characteristics:

1. each row represents a column of the decision table;
2. the rows do not have any particular order (but some orderings are more useful);
3. all rows are distinct (exclusivity);

The order of the columns (conditions and actions) is not important to the description of the problem at the logical level (unless a certain order has to be respected at execution time because of side effects, for instance an ordering of the actions);

1. the meaning of each column is explained through a named domain as heading (condition or action subject);
2. on each row position of the table, an attribute value (a condition state or an action value, possibly "nil") is found, and not a set of values.

It is clear that such relational table is identical to the transposed expanded decision table, so that the rows correspond with the columns of the decision table and vice versa. The identity is formal and does not refer to the utility of both representation methods.

Since every condition combination occurs precisely once in the relation, the condition attribute values uniquely identify the n-tuples in the relation (candidate key). It is, indeed, the intention of the decision table to indicate which actions should be executed for a given combination of conditions. So the set of condition attributes is defined as primary key. The action attributes can then be indicated as the non-key attributes.

A combination of non-key attributes (actions), that is part of the primary key of another table and thereby refers to that table (foreign key), corresponds with a condition assignment as action in a condition subtable (called condition reference).

By analogy with dependencies in relational tables, the relationship between conditions and actions (or possibly the interrelationship between conditions or actions) in decision tables can be expressed as a cause-effect relation. Such logical if...then...-relation corresponds with the "implication statement" in propositional logic, which is equivalent to the "dependency statement" for functional dependencies. The decision table, being a set of implications, can therefore be described in terms of functional dependencies. In the first instance, especially the dependency between conditions and actions is important, so that functional dependency can be defined as:

Given a decision table DT with conditions C_i (i=1..cnum) and actions A_j (j=1..anum) and X, Y subsets of resp. the condition set and action set of DT: Y is functional dependent on X if with every combination of X-values in DT corresponds one and only one configuration of Y-values.

Since every combination of condition states occurs at most once, each action is functional dependent on the complete condition set (primary key). The formal correspondence between the decision table and the relational table is given in fig. 1.

Decision table	**Relational table**
condition (row)	key attribute (column)
condition states	key domain
action	non-key attribute
action value	non-key domain
stub	heading
number of rows	degree
entry	attribute value
column	n-tuple
number of columns	cardinality

figure 1: Terminology of the decision table and the relational table

4. Fuzzy decision tables

Fuzzy extensions to DTs are aimed to facilitate decision making with imprecision and uncertainty which are necessary in many cases. Fuzzy set theory [15] is a rigorous mathematical framework. It aims at quantifying and reasoning with the fuzziness that is found in the real world. Fuzziness reflects a type of uncertainty and imprecision due to vagueness in concept. This usually refers to the problem of boundary determination. Zadeh proposed a solution to this problem by introducing a gradual transition from non-membership of an element satisfying a given property to full-membership of the element satisfying the given property. This gradual transition allows partial degrees of membership. More formally, we define a fuzzy set as follows:

Let U be the universe of discourse. A fuzzy set F on U is characterized by a membership function μ_F: $U \rightarrow [0,1]$, which associates with each element u of U a number $\mu_F(u)$ representing the grade of membership of u in F. $\mu_F(u) = 0$ means non-membership, $\mu_F(u) = 1$ means full membership, and $\mu_F(u)$ with $0 < \mu_F(u) < 1$ means partial membership. Symbolically,

$F = \{ \mu_F(u)/u \mid u \in U \text{ and } \mu_F(u) \in [0,1] \}$.

A FDT consists of a condition part and an action part, each part allowing fuzziness to be represented [7]. More formally,

FDT (form 1): Let CS_i be a condition subject with domain CD_i (i = 1, ..., cnum), CT_i be a set of condition states S_{ik} (k = 1, ..., ni , i = 1, ..., cnum) with S_{ik} being a fuzzy logic expression, AS_j be an action subject incorporated with linguistic terms and fuzzy sets, and AV_j = {true (x), false (-), nil (.)} be an action value set (j = 1, ... , anum), then a fuzzy decision table (FDT) is a function from $CT_1 \times CT_2 \times \ldots \times CT_{cnum}$ to $AV_1 \times AV_2 \times \ldots \times AV_{anum}$ such that each possible condition combination is mapped into one action configuration. ■

Moreover, in a FDT, when all the decision rules involving fuzziness are of the form: "If X is A then Y is B", the FDT can now be (equivalently) expressed in a form where Y is an action subject and B is one of the action subject values. In this way, a value of AS_j (j = 1,2,...,anum) will be not only true(x) or false(-), but also a fuzzy set or a linguistic term. Thus we have another form of FDTs:

FDT (form 2): Let CS_i be a condition subject with domain CD_i (i = 1, ..., cnum), CT_i be a set of condition states S_{ik} (k = 1, ..., ni , i = 1, ..., cnum) with S_{ik} being a fuzzy logic expression, AS_j be an action subject, and $AV_j = \{av \mid av \text{ is a fuzzy set of } AS_j\}$ be an action value set (j = 1, ... , anum), then a fuzzy decision table (FDT) is a function from $CT_1 \times CT_2 \times \ldots \times CT_{cnum}$ to

$AV_1 \times AV_2 \times \ldots \times AV_{anum}$ such that each possible condition combination is mapped into one action configuration. ■

Both FDT forms guarantee the property of completeness because each possible condition combination will lead to a decision. The matching of fuzzy conditions can be made based on closeness measures of fuzzy sets. When consulting a FDT, multiple alternatives (action configurations) with nonzero matching degrees may be possible for a given condition combination, which is desirable in certain cases, and can be restricted by given thresholds. A lower threshold means a higher degree of tolerance for imprecision and uncertainty.

Example 2. FDTs with fuzziness in condition and action parts.

A FDT in form 1:

1. Type of Book (CS1)	hard cover				normal		
2. Wholesaler (CS2)	yes			no	-		
3. Quantity (CS3)	L	H	VH	-	L	H	VH
1. Discount small (AS1)	x	-	-	x	-	x	x
2. Discount big (AS2)	-	x	x	-	-	-	-
3. Free delivery (AS3)	-	x	x	-	-	-	x
4. Charged delivery (AS4)	-	-	-	-	x	x	-

A FDT in form 2:

1. Type of Book (CS1)	hard cover				normal		
2. Wholesaler (CS2)	yes			no	-		
3. Quantity (CS3)	L	H	VH	-	L	H	VH
1. Discount (AS1)	small	big	big	small	-	small	small
2. Delivery (AS2)	-	free	free	-	charged	charged	free

In the FDT of form 1, fuzzy sets or linguistic terms (low(L), high(H), very high(VH), small, big) appear with condition states and action subjects, while in the FDT of form 2, fuzzy sets or linguistic terms appear with condition states and action subject values. Note that the symbol "-" appearing in the condition part denotes the irrelevance of the condition state.

5. Fuzzy relational databases

A FRDB represents imprecise attribute values and close domain elements with possibility distributions and closeness relations respectively [3]. With the relational scheme $\mathbf{R}(A_1, A_2, ..., A_n)$, any n-tuple of a relation is of the form: $(p_{A1}, p_{A2}, ..., p_{An})$ where p_{Ai} is a (excluding) possibility distribution of attribute A_i on its domain D_i, and a closeness relation (reflexive and symmetric) is associated with each D_i. Based on this framework of fuzzy data representation, a number of related issues have been discussed, such as data closeness and redundancy, fuzzy functional dependency (FFD), extended relational algebra, q-keys and fuzzy normal forms [2], [4-6], [10].

Example 1. A FRDB relation with imprecise attribute values.

Name	Sex	Age	Height	Hair-color
N1	M	25	185	black
N2	F	young	{.8/170, 1/175, 1/180/, .8/185}	{brown, red}

It is worth mentioning that the imprecision of attribute values in the tuple for N2 is reflected by a subset ({brown, red}), a linguistic term (young), and a possibility distribution ({.8/170, 1/175, 1/180/, .8/185}). In addition, closeness relations can be specified for domains (e.g., for the domain of Hair-color) to reflect the relationship between domain elements.

6. Representing FDT knowledge with an FRDB approach

Many efforts have been made to integrate (crisp) decision or production rules with (crisp) relational database systems in the context of decision support systems or expert database systems. In a recent study, [13] have described two such techniques. The first technique is to represent each DT of the hierarchy by a relational table where condition and action subjects are treated as attributes, and each decision rule is stored as a different tuple. This technique is easy to use and convenient for consultation in decision making. The second technique is to represent each DT by three relational tables for subjects, rules, and rule-parts respectively. It is based on the concept of entity-relationship (ER) methodology, and more flexible to decision situation changes. In this study, however, we will only concentrate on a fuzzy extension in accordance with the first technique.

Method (Representing FDT in FRDB): When viewed vertically (column by column), a FDT can be seen as a set of ordered n-tuples of the form: $(ct_1, ..., ct_{cnum}, av_1, ..., av_{anum})$ represented in a FRDB table with the relation scheme $\mathbf{R}'(CT_1, ..., CT_{cnum}, AS_1, ..., AS_{anum})$. ■

Example 3. Relational tables representing the FDTs described in example 2.

The FDT in form 1 is represented in a FRDB table (R1) as follows:

Type-of-book	Wholesaler	Quantity	Discount-small	Discount-big	Free-delivery	Charged-delivery
hard-cover	yes	L	x	-	-	-
hard-cover	yes	H	-	x	x	-
hard-cover	yes	VH	-	x	x	-
hard-cover	no	-	x	-	-	-
normal	-	L	-	-	-	x
normal	-	H	x	-	-	x
normal	-	VH	x	-	x	-

The FDT in form 2 is represented in the following FRDB table (R2) where the fuzziness involved in the FDT knowledge is represented as fuzzy attribute values:

Type-of-book	Wholesaler	Quantity	Discount	Delivery
hard-cover	yes	L	small	-
hard-cover	yes	H	big	free
hard-cover	yes	VH	big	free
hard-cover	no	-	small	-
normal	-	L	-	charged
normal	-	H	small	charged
normal	-	VH	small	free

In both cases, each row of the relational table represents a column of the fuzzy decision table. Therefore the matching of fuzzy conditions in a FDT can be measured in a FRDB based on the concept of data closeness [4]. In addition, the relationship between conditions and actions in a FDT can be expressed as a cause-effect relation, to which the concept of functional dependency may apply. Based upon the notion of identical functional dependency (IFD) introduced for the FRDB model in [5], we will have the following IFDs:

$$(CT_1, ..., CT_{cnum}) \dashrightarrow_{id} AS_j \qquad j = 1, 2, ..., anum.$$

Furthermore, in analogue to the case of crisp databases, these IFDs can result in the notion of relation keys (hereby denoted as I-keys). Apparently, $(CT_1, ..., CT_{cnum})$ forms an I-key of scheme **R'**. Moreover, FFDs may be used to express the cause-effect relation between conditions and actions:

$$(CT_1, ..., CT_{cnum}) \text{--->}_q AS_j \qquad j = 1, 2, ..., \text{anum}$$

and $(CT_1, ..., CT_{cnum})$ forms an q-key of **R'**. Importantly, these FFDs may further play roles in the FDT design.

The representation of FDT knowledge in FRDB tables enables us to carry out fuzzy decision making with extended SQL facilities such as SQLf [1]. Usually, a SQL-like fuzzy query may be exemplified as follows:

```
SELECT   (l) Discount, Delivery
FROM     R2
WHERE    Type-of-book is hard-cover AND
         Wholesaler is yes AND
         Quantity is around 30
```

That is, action configurations (Discount, Delivery) may be obtained with different degrees according to the matching between query conditions (e.g., around 30 vs. the values of attribute Quantity), and are restricted by the tolerance threshold l.

7. Conclusion and future research

Many desirable properties and useful functionality of relational databases may be utilized in the storage, construction, verification and consultation of decision tables. In the context of fuzzy decision making and fuzzy data modeling, an approach has been proposed to represent FDTs in the FRDB environment. The cause-effect relations between conditions and actions of a FDT are reflected by IFDs or FFDs in fuzzy databases. The concepts of I-keys and q-keys also apply. Moreover, fuzzy decision making can be realized via fuzzy queries against FRDB tables.

Current research and further studies include the FDT modeling, building FDTs hierarchies and verification in the FRDB environment with the present approach, and the exploration of corresponding issues with other representation techniques.

References

[1] Bosc P.; and Pivert O., About equivalents in SQLf: a relational language supporting imprecise querying, Proceedings of International Fuzzy Engineering Symposium, Yokohama (Japan), 309-320, 1991.

[2] Buckles B., Petry F. and Sachar H.S., Design of similarity-based relational databases, In Fuzzy logic in Knowledge Engineering, H. Prade and C.V. Negoita eds., Verlag TUV Rheinland, 3-7, 1986.

[3] Chen G.Q., Vandenbulcke J. and Kerre E.E., A step towards the theory of fuzzy database design, Proceedings of the 4th World Congress of International Fuzzy Systems Association (IFSA'91), Brussels, 44-47, 1991.

[4] Chen G.Q., Vandenbulcke J. and Kerre E.E., A general treatment of data redundancy in a fuzzy relational data model, Journal of the American Society for Information Science, 304-311, 1992.

[5] Chen G.Q., Kerre E.E. and Vandenbulcke J., On the lossless-join decomposition in a fuzzy relational data model, Proceedings of International Symposium on Uncertainty Modeling & Analysis (ISUMA'93), IEEE Press, Maryland (USA), 440-446, 1993.

[6] Chen G.Q., Kerre E.E. and Vandenbulcke J., A computational algorithm for the FFD closure and a complete axiomatization of fuzzy functional dependency (FFD), International Journal of Intelligent Systems 9, 421-439, 1994.

[7] Chen G.Q., Vanthienen J. and Wets G., Fuzzy decision tables: extending the classical formalism to enhance intelligent decision making, International Joint Conference of 4th IEEE International Conference on Fuzzy Systems and the 2nd International Fuzzy Engineering Symposium (FUZZ-IEEE/IFES'95), 599-606, 1995.

[8] CODASYL, A Modern Appraisal of Decision Tables, Report of the Decision Table Task Group, ACM, New York, 1982.

[9] Codd E., A Relational Model of Data for Large Shared Data Banks, Comm. of the ACM, 13(6), 377-387 (1970).

[10] Raju K. and Majumdar A., The study of joins in fuzzy relational databases, Fuzzy sets Syst. 21, 19-34, 1987.

[11] Salah A., An integration of decision tables and a relational database system into a Prolog environment, Birmingham, Doctoral dissertation (1986).

[12] Vanthienen J. and Wets G., From Decision Tables to Expert System Shells, Data & Knowledge Engineering 13, 265-282, 1994.

[13] Vanthienen J. and Wets G., Integration of the decision table formalism with a relational database environment, to appear in: Information Systems, 1995.

[14] Vanthienen J. and Dries E., Decision Tables: Refining the Concept and a Proposed Standard, to appear in: Comm. of the ACM.

[15] Zadeh L. A., Fuzzy sets, Information and Control 8 (1965) 338-358.