Fuzzy Database Modeling

Z.M. Ma
Northeastern University, China

INTRODUCTION

A major goal for database research has been the incorporation of additional semantics into the data model. Classical data models often suffer from their incapability of representing and manipulating imprecise and uncertain information that may occur in many real-world applications. Since the early 1980s, Zadeh’s fuzzy logic (Zadeh, 1965) has been used to extend various data models. The purpose of introducing fuzzy logic in database modeling was to enhance the classical models such that uncertain and imprecise information can be represented and manipulated. This resulted in numerous contributions, mainly with respect to the popular relational databases (Ma & Mili, 2002b):

Where uncertainty, many current researches have concentrated on fuzzy object-oriented database models to deal with complex objects and uncertain data together. Also, rapid advances in computing power have brought opportunities for databases in emerging applications in CAD/CAM (Computer-Aided Design/Computer-Aided Manufacturing), multimedia, and geographic information systems (GIS). These applications characterizedly require the modeling and manipulation of complex objects and semantic relationships. It proved that the object-oriented paradigm lends itself extremely well to the requirements. Because the classical relational database model and its extension of fuzziness do not satisfy the need of modeling complex objects with imprecision and uncertainty, many current researches have concentrated on fuzzy object-oriented database models to deal with complex objects and uncertain data together.

BACKGROUND

Database modeling can be carried out at two different levels: conceptual data modeling and database modeling. Therefore, we have conceptual data models (e.g., ER/EER—Entity-Relationship/Extended Entity-Relationship and UML) and logical database models (relational databases and object-oriented databases). Logical database models are often created through mapping conceptual data models into logical database models. This conversion is called conceptual design of databases.

In order to deal with imprecise and uncertain information in database modeling, fuzzy set has been applied. Let $U$ be a universe of discourse, then a fuzzy value on $U$ is characterized by a fuzzy set $F$ in $U$. A membership function $\mu_F: U \rightarrow [0, 1]$ is defined for the fuzzy set $F$, where $\mu_F(u)$, for each $u \in U$, denotes the degree of membership of $u$ in the fuzzy set $F$. Thus the fuzzy set $F$ is described as follows.

$$F = \{\mu (u_1)/u_1, \mu (u_2)/u_2, ..., \mu (u_n)/u_n\}$$

When $\mu_F(u)$ is viewed to be a measure of the possibility that a variable $X$ has the value $u$ in this approach, where $X$ takes values in $U$, a fuzzy value is described by a possibility distribution $\pi_x$ (Zadeh, 1978).

$$\pi_x = \{\pi_x(u_1)/u_1, \pi_x(u_2)/u_2, ..., \pi_x(u_n)/u_n\}$$

Here, $\pi_x(u_i), u_i \in U$, denotes the possibility that $u_i$ is true. Let $\pi_x$ and $F$ be the possibility distribution representation and the fuzzy set representation for a fuzzy value, respectively. It is apparent that $\pi_x = F$ is true (Raju & Majumdar, 1988).

Also, fuzzy data can be represented by similarity relations in domain elements (Buckles & Petry, 1982), in which the fuzziness comes from the similarity relations between two values in a universe of discourse, not from the status of an object itself. Similarity relations are thus used to describe the degree similarity of two values from the same universe of discourse. A similarity relation $Sim$ on the universe of discourse $U$ is a mapping: $U \times U \rightarrow [0, 1]$ such that:

a. for $\forall x \in U, Sim(x,x) = 1$, (reflexivity)
b. for $\forall x, y \in U, Sim(x,y) = Sim(y,x)$, and (symmetry)
c. for $\forall x, y, z \in U, Sim(x,z) \geq \max (\min (Sim(x,y), Sim(y,z)))$. (transitivity)

MAJOR ISSUES AND SOLUTIONS

Fuzzy Relational Databases

Fuzzy information has been extensively investigated in the context of the relational databases. The followings are some major issues in current studies of fuzzy relational databases (Ma & Mili, 2002b):

Copyright © 2006, Idea Group Inc., distributing in print or electronic forms without written permission of IGI is prohibited.
Fuzzy Relational Database Models

One can find several kinds of fuzzy relational database models. One of the fuzzy relational data models is based on similarity relations (Buckles & Petry, 1982), or proximity relation (Shenoi & Melton, 1989), or resemblance (Rundensteiner, Hawkes, & Bandler, 1989). The other one is based on possibility distribution (Prade & Testemale, 1984), which can further be classified into two categories: tuples associated with possibilities and attribute values- represented possibility distributions (Raju & Majumdar, 1988). The form of an n-tuple in each of the previously mentioned fuzzy relational model can be expressed, respectively, as:

\[ t = <p_1, p_2, \ldots, p_i, \ldots, p_n>, t' = <a_1, a_2, \ldots, a_i, \ldots, a_n, d> \]

where \( p_i \subseteq D_i \) with \( D_i \) being the domain of attribute \( A_i \), \( a_i \in D_i \), \( d \in (0, 1) \), \( \pi_{A_i} \) is the possibility distribution of attribute \( A_i \) on its domain \( D_i \), and \( \pi_{A_i}(x) \), \( x \in D_i \), denotes the possibility that \( x \) is the actual value of \( t[A_i] \).

It is clear that one can combine two kinds of fuzziness in possibility-based fuzzy relational databases, where attribute values may be possibility distributions and tuples are connected with membership degrees. Such fuzzy relational databases are called possibility-distribution-fuzzy relational models (Umano & Fukami, 1994). Another possible extension is to combine possibility distribution and similarity (proximity or resemblance) relation, and the extended possibility-based fuzzy relational databases are hereby proposed (G. Q. Chen, Vandebulcke, & Kerre, 1992; G. Q. Chen, Kerre, & Vandebulcke, 1994, 1996; Ma & Mili, 2002; Ma, Zhang, & Ma, 2000; Ma, Zhang, & Mili, 2002), where possibility distribution and resemblance relation arise in a relational databases simultaneously.

Fuzzy Data Integrity Constraints and Formalizations

Fuzzy data dependencies, mainly including fuzzy functional dependency (FFD) and fuzzy multivalued dependency (FMVD), have extensively been studied in the context of fuzzy relational databases. There are some papers that focus only on FMVD (Bhattacharjee & Mazumdar, 1998; Jyothi & Babu, 1997; Tripathy & Sakena, 1990). And some papers focus only on FFD, where we can classify two kinds of papers: The first one focuses on the axiomatization of FFD (G. Q. Chen, Kerre, & Vandebulcke, 1994, 1995; Cubero & Vila, 1994; Liao, Wang, & Liu, 1999; Liu, 1992, 1993a, 1993b; Saxena & Tyagi, 1995) and the second focuses on the lossless join and decomposition (Bhuniya & Niyogi, 1993; Bosc & Pivert, 2003; Raju & Majumdar, 1988). The later is the basis on which to implement the normalization of fuzzy relational databases (G. Q. Chen, Kerre, & Vandebulcke, 1996). Also, there are some papers that focus both on FFD and FMVD and present the axiomatization of FFD and FMVD (Liu, 1997; Ma, Zhang, Ma, & Mili, 2002; Sözat & Yazici, 2001; Yazici & Sözat, 1998).

In addition, fuzzy data dependencies can be applied in data handling. In Bosc, Dubois, and Prade (1998), FFD was used for redundancy elimination. In Intan and Mukaidono (2000), FFD was used for approximate data querying. In Chang and Chen (1998), Liao, Wang, and Liu (1999), and Ma, Zhang, and Mili (2002), FFD was used for fuzzy data compression.

Query and Data Processing

Classical relational databases suffer from a lack of flexibility in query. The given selection condition and the contents of the relation are all crisp. A query is flexible if the following conditions can be satisfied (Bosc & Pivert, 1992):

- A qualitative distinction between the selected tuples is allowed.
- Imprecise conditions inside queries are introduced when the user cannot define his or her needs in a definite way, or when a prespecified number of responses are desired and therefore a margin is allowed to interpret the query.

Here, typically, the former case occurs when the queried relational databases contain incomplete information and the query conditions are crisp, and the later case occurs when the query conditions are imprecise, even if the queried relational databases do not contain incomplete information.
