Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 281 (2015) 300–307

FUZZY sets and systems

www.elsevier.com/locate/fss

Fuzziness in database management systems: Half a century of developments and future prospects

Janusz Kacprzyk [a,*], Sławomir Zadrożny [a], Guy De Tré [b]

[a] *Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6, 01-447 Warsaw, Poland*

[b] *Department of Telecommunications and Information Processing, Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium*

Received 30 December 2014; received in revised form 20 May 2015; accepted 15 June 2015

Available online 19 June 2015

Abstract

This comprehensive, bird's view research note combines the state of the art, a brief presentation of the history and some original solutions, and position like views of some prospective future developments of one of the most relevant and interesting areas related to the use of fuzzy logic in database management systems, notably in its querying component, and – to some extent – in a broader issue of data and information management. We briefly summarize the roots of those new applications of fuzzy logic, more relevant proposals and development in the context of fuzzification of the basic relational database model, and then some of its further generalizations. We particularly focus on fuzzy querying as a human consistent and friendly way of retrieving information due to real human intentions and preferences expressed in natural language represented via fuzzy logic and possibility theory. We mention some extensions, notably fuzzy queries with linguistic quantifiers, and point their close relation to linguistic summaries. As for newer, prospective developments, we mainly focus on bipolar queries that can accomodate the users' intentions and preferences involving some sort of a required and desired, mandatory and optional, etc. conditions. We show various ways of handling such queries. We conclude with some brief position statements of our view on relevant and promising directions, and challenges.

Keywords: Database management systems; DBMS; Database querying; Fuzzy logic; Possibility theory; Bipolar queries

1. Context and introductory remarks

The glut of data of various forms and huge quantities, that we have experienced over the last years or decades, calls for a new generation of *database systems* incorporating new, more effective and efficient solutions and techniques to fully take advantage, for solving real problems, of the availability and richness of vast data resources. Data representation, handling, querying, etc., together with related problems of data mining and knowledge discovery, constitute a comprehensive solution along the line of the so called end-to-end data processing [1].

* Corresponding author.

E-mail addresses: Janusz.Kacprzyk@ibspan.waw.pl (J. Kacprzyk), Slawomir.Zadrozny@ibspan.waw.pl (S. Zadrożny), Guy.DeTre@UGent.be (G. De Tré).

http://dx.doi.org/10.1016/j.fss.2015.06.011

0165-0114/

The database community [1,3] has for years advocated a paradigm shift the idea of which is that the tools and techniques, and resulting systems, should be tailored to average users who may be at best domain experts but not necessarily database professionals. An immediate consequence is that since for the humans the only fully natural means of communication and articulation is natural language, then human interaction with database and information systems should mainly proceed using natural language. However, natural language is inherently imprecise and vague, which is difficult to deal with using conventional tools and techniques. More generally, information to be processed is very often *imperfect*, i.e., imprecise, uncertain, incomplete or ambiguous, and has to be properly handled. An efficient storing and handling of imperfect information is viewed as one of the main challenges for information management in this century [74]. Fuzzy logic can provide powerful means to account for many aspects of imperfect information, in particular related to imprecision, in the database context, and will be advocated here.

An overwhelming majority of contributions of the fuzzy logic in the database related context have been proposed with respect to Codd's *relational data model* [43]. It is worth mentioning that Zadeh was a member of Codd's group at IBM Almaden in l968, on a sabbatical leave from the UC Berkeley.

Briefly, the basic structure of a relational database is a *relation* R, representing information on a set of objects of a given type. A relation comprises *tuples* representing information on individual elements of a given object type. Each tuple t is composed of the values of *attributes* comprising a *schema* of a given relation. All possible values of an attribute A form its *domain* dom_A. A number of operations on the set of relations are available to process data, forming a *relational algebra*. The relational data model is naturally amenable to be extended using fuzzy logic due to its firm logical foundations, and the basic concept of algebra of *relations* finds its immediate counterpart in fuzzy relation etc. We will provide a brief account of these results.

It should be added that object oriented and semi-structured data models have been extended using fuzzy logic, too; cf., e.g., [44,45,115,10,80,64,61,108,111,112], but this important development will not be dealt with here due to a lack of space and more specific a character. Moreover, the case of more complex forms of data to be queried, e.g., graphs, will not be dealt with and details can be found in, e.g., Castelltort and Laurent [40,41] or Pivert et al. [89,67,93].

2. Tools and techniques

Among the two main lines of research on fuzziness in data(base) management the first concerns the standard data(base) and allows for the fuzziness only in *queries*. The second assumes the data themselves to be fuzzy, and concerns the concept of a *fuzzy database*.

The first line of research may be seen as related to intelligent, natural language based HCIs (human – computer interfaces) focusing on how to enrich classical query languages with elements of natural language, modeled by using fuzzy logic to handle imprecision. Such elements/terms as "cheap", "large" etc. are used by human beings when they are looking, e.g., for some goods, hotel accommodations, etc. Another boost of interest in fuzzy querying is related to data warehousing and data mining, exemplified by a combination of fuzzy querying and data mining interfaces (e.g., [70,71]) or fuzzy logic and the OLAP (Online Analytical Processing) technology (e.g., [76]).

A query against a database may be identified with a condition which expresses *preferences* (or *intentions*) of the user. Such a condition usually comprises a number of *atomic conditions* of varying *importance*, linked with some logical connectives.

It is clear that a simple use of even a complicated combination of logical connectives cannot adequately reflect preferences and intentions of the human user, and that is why a multitude of more sophisticated query types have been proposed exemplified, just to name a few, by: the skyline queries, Chomicki's [42] winnow operator, top-k queries, queries based on possibilistic logic, etc. For example, we refer the interested reader to the following works which are not the source papers on the above types of queries but consider them in a fuzzy/possibilistic setting: [11,13,14,27,31, 35,34,45,47,52,66,67,75,86,87,91,96].

As the preferences are often not clear-cut, they may be accounted for by using fuzzy predicates inside a condition, and this is our main concern here. Thus, the *matching of a tuple against a query* is *gradual* rather than binary.

One of the first works on this topic is that of Tahani [98], a student of Zadeh who proposed fuzzy modeling of linguistic terms in queries. Later on, the use of non-standard, fuzzy logic based, logical connectives in the query conditions has been proposed to better aggregate matching degrees of atomic conditions – cf., e.g., Kacprzyk and Ziółkowski [73], and Kacprzyk, Zadrożny and Ziółkowski [72], Bosc and Pivert [21,22], to name a few.