ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

An Overview of Fuzzy Query Processing

Garima Singh*, Awadhesh Kumar Sharma
Department of Computer Science & Engineering
Madan Mohan Malaviya Engineering College
Gorakhpur-273010, U. P., INDIA
garimasingh2307@gmail.com
akscse@rediffmail.com

Abstract: In this paper the benefits of fuzzy query over crisp query are presented. We have taken a sample classical database and applied both crisp and fuzzy queries on this database to show that the fuzzy query may be preferred for some situations as they represent more efficiently how humans conceptualize the reality through their understanding and language. This paper gives the basic knowledge about fuzzy queries. This paper also discusses the work done by many researchers in the area of fuzzy query processing and pointed towards the possibility of further research in this area.

***Keywords*:** Conventional database, α cut, fuzzy predicate, membership function, fuzzy database.

I. INTRODUCTION

Crisp (precise) data refers to data that is precise and is without vagueness. Crisp queries are based on classic logic and set theory, limiting expression of user preferences. Crisp queries are meant to retrieve elements which qualify for a given crisp condition. This means using classic logic, it is possible to deal only with information that is completely true or completely false, it is not possible to deal with information that is imprecise or vague, but this type of information contains data, which would provide a better solution to the problem. In classic logic the membership of an element to a set is represented by 0 if it does not belong and 1 if it does having the set {0, 1}. This is not appropriate for handling new applications and needs.

On the other hand, in fuzzy logic the membership of an element may be any value of interval [0, 1]. Fuzzy logic is an extension of the classic systems. Fuzzy logic is the logic behind approximate reasoning instead of exact reasoning. Its importance lies in the fact that many types of human reasoning, particularly the reasoning based on common sense, are by nature approximate. Fuzzy Logic was proposed by L.A. Zadeh in 1965. The basic idea behind fuzzy logic is sets or classes whose boundaries are vague. For example: old people, good students etc.

Queries having imprecise or vague terms are called fuzzy queries. For example: "find the student with *very good* marks". Fuzzy queries are based on Fuzzy Logic i.e. a powerful tool for dealing with preferences. Fuzzy query allows users to express requirements involving preferences. With this tool, the user may express their query using linguistic terms and sentences. As the application of database technology moves outside the area of a crisp mathematical world to the area of the real world, the need to handle imprecise information becomes of great significance.

II. MOTIVATION BEHIND FUZZY QUERIES

a. Information in real-world applications is often imprecise, vague and uncertain. Ignoring this may cause of deformation of human perception of real-world and loss of important information.
b. For "humanizing" the access to information and creating query specification mechanisms closer to natural language expression and human being thinking.
c. Recently new applications have come in to existence that require database management systems with uncertainty capabilities, for example decision support systems.

III. PROBLEM OF RIGIDITY IN CRISP QUERIES

Crisp queries are suffered from a lack of flexibility. Here by using a simple example we will show that in our daily life some situations may arrive when fuzzy queries provide more user friendly than crisp queries.

Suppose there is a relation HOTEL and user want to find the hotel *cheap* and *near* to station. In the case of crisp querying system user reformulate it as not more than 5 km away and price less than 500.

Table 1: relation HOTEL

Name	Distance(Km.)	Price(Rs.)
A	4.5	500
B	5	500
C	1	520
D	5.1	350
E	6	650
F	7	600

So in SQL we will write it as-

SELECT * FROM HOTELS WHERE price $\leq$ 500 AND distance $\leq$ 5

The result of above query will be-

Table: 2

Name	Distance(Km.)	Price(Rs.)
A	4.5	500
B	5	500

Here we can see that although the other two options i.e. C, D are much better for the user but the system doesn't help him in finding the "best" answer. The problem of rigidity arrives from Boolean Logic. It could be solved allowing the specification of fuzzy queries.

Now we will solve this problem using fuzzy logic.

First we will define the membership function for fuzzy predicates: cheap and near respectively:

$$\mu_{cheap}(x) = \begin{cases} 1 & x \leq 400 \\ \frac{b-x}{b-a} & 400 < x < 600 \\ 0 & x \geq 600 \end{cases}$$

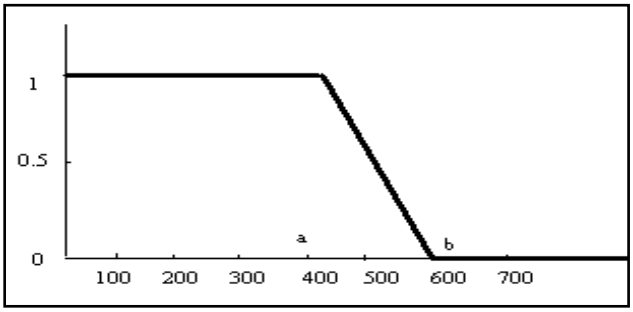

Figure 1: membership function for fuzzy predicate "cheap".

$$\mu_{near}(x) = \begin{cases} 1 & x \leq 4 \\ \frac{b-x}{b-a} & 4 < x < 6 \\ 0 & x \geq 6 \end{cases}$$

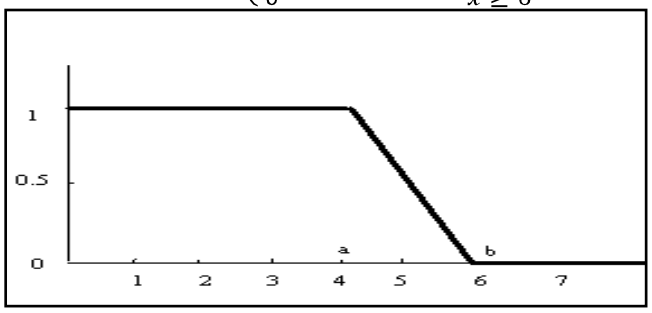

Figure 1: membership function for fuzzy predicate "near".

After calculating the membership value of each element to the fuzzy predicate near and cheap, for finding the solution we will apply the arithmetic mean [1] i.e. a type of operator allowing for compromises between the predicates used as parameters. Arithmetic mean:

$$AM(P1, \ldots, Pn)(x) = (PI(x) + \ldots + Pn(x))/n$$

Where P is a fuzzy predicate

Here

$$AM = (\mu_{near} + \mu_{cheap})/2$$

Result of the above calculation shown in the table given below:

Table: 3

Name	Distance (Km.)	Price (Rs.)	μ_{near}	μ_{cheap}	AM
A	4.5	500	0.75	0.5	0.625
B	5	500	0.5	0.5	0.500
C	1	520	1.0	1.0	0.700
D	5.1	350	0.45	.40	0.725
E	6	650	0.0	0.0	0.0
F	7	600	0.0	0.0	0.0

Now suppose the value of threshold (α) provided by user is 0.650 then the result of query will be-

Table: 4

Name	Distance(Km.)	Price(Rs.)
C	1	520
D	5.1	350

As we can see that fuzzy query provide results that are much suitable to the user.

IV. COMPARATIVE STUDY

Table: 5

	Crisp queries	Fuzzy Queries
Nature	Rigid	Flexible
Basic Idea	Classic Logic and Set Theory	Fuzzy Logic and Fuzzy Sets
Processing Cost	Low	High
Linguistic Variables	Not Allowed	Allowed
Underlying Database	Conventional Database	Conventional or Fuzzy Database

V. LITERATURE REVIEW

Human, when interacting with the database, want to make complex queries that have a lot of vagueness present in it. In real world applications we often need to test the queries based on fuzzy data. For example, some one can specify as find employee whose age is around 25 years old, find young person, find employee with good salary etc. A query is flexible if the databases contain imprecise and uncertain information or the query condition is imprecise and uncertain.

In [2] the application of fuzzy set theory in DBMS can be classified into two main classes.

Class 1 deals with the study of fuzzy query processing in conventional (non-fuzzy) DBMS

Class 2 deals with DBMS which, having the ability to store and manipulate fuzzy data directly and also supports fuzzy query.

A. Conventional DBMS with Fuzzy Queries:

Early DBMS's with uncertainty handling are developed within the framework of non-fuzzy DBMS. Generally, these systems deal with the construction and evaluation of fuzzy query against a crisp Database, and ignore the problem of direct representation of fuzzy data in DBMS.

Consequently, this model is less powerful, as it is less suitable in handling fuzzy query involving fuzzy data which are vague and ambiguous.

B. Fuzzy Databases with Fuzzy Queries:

Recent DBMS's with uncertainty handling are more advanced when compared to the earlier ones. They address the problem of direct representation of fuzzy data in the DBMS as well as the construction and evaluations of fuzzy query.

Various fuzzy database models have been proposed over last 35 years. We are mainly interested in query processing on relational data model and its extensions because at present relational database model is world wide used tool for building database system. Buckles [3] propose one of the earliest versions of Fuzzy Relational Database System (FRDBS) by merging the theory of fuzzy set and Relational Database System (RDBS).

Many researchers develop many techniques for the processing of fuzzy queries. Here we will discuss some of the main techniques to process the fuzzy query.

a. Janusz Kacprzyk and Andrzej Ziolkowski [4] proposed a new approach for database querying system for handling imprecise queries using fuzzy-logic-based calculus of linguistically quantified propositions. An algorithm was proposed to deal with queries involving linguistic quantifiers and importance of the attributes. The meaning of imprecise term is elicited from user and then record of specified term are fetched and matched with query. The formula for finding the matching degree of different types of query was defined in the algorithm.
b. Leonid José and Tineo Rodríguez [5] provide the extension of relational database management systems which support fuzzy quantified queries so that users may address queries involving preferences. An interpretation of fuzzy quantifiers was presented that is based on a simple linguistic transformation principle. With this interpretation, it is possible to derive Boolean queries, which return a α-cut of the initial fuzzy query. Then, the fuzzy query can be processed on this set thus avoiding the exhaustive scan of the entire database.
c. Tien-Chin Wang et al. [6] proposed a fuzzy query language (FQL) which can be applied to query imprecise data. Fuzzy language architecture based on SQL and fuzzy sets for the evaluation of records was developed that significantly improve the robustness of database query operations. In this architecture first data is transform into some clusters based on membership degree then possibility value is calculated for comparing it with α threshold. There also exists WEIGHT clause in the query.
d. V.Balamurugan and K.Senthamarai Kannan [7] proposed a framework to handle the complexities by using fuzzy set theory. The framework is divided into two stages- preprocessing and query implementation stage. The preprocessing stage involves collecting information; identify fuzziness associated with attributes and form different fuzzy sets. In the query implementation stage presence of linguistic variables and hedges are checked and then membership values are calculated and manipulated based on hedges. Finally defuzzification is carried out by finding the α-cut. Main aim of this model is to make use of the standard facilities available in the conventional DBMS.
e. Jaydev Mishra [8] defined an algorithm to find the membership value for each tuple of the relation based on the fuzzy attributes on which fuzzy query is made. In this algorithm first list of fuzzy attributes are identified from given fuzzy query and then according to the formula given in algorithm membership value of each tuple is calculated.

VI. CONCLUSION

Imprecise, uncertain or vague information is a common phenomenon in real world. Many authors have proposed framework for processing the fuzzy query on the conventional database system but these system do not allow storing the fuzzy data in the database. The available technology of conventional database system for efficient data access and query solving may not be used directly in fuzzy querying. So the Fuzzy databases are required in order to store fuzzy data and process the fuzzy queries efficiently. Based on the previous study we can say that processing of fuzzy query over fuzzy database and development of an efficient query evaluation mechanism is an open area of further research.

VII. REFERENCES

[1]. Patrick Bosc and Olivier Pivert, “Sqlf a relational database language for fuzzy querying,” IEEE Transactions on Fuzzy Systems, Vol. 3, No. 1, pp. 1-17, February 1995.

[2]. K. K. Phang, Mashkuri Hj. Yaacob, T. C. Ling “Development of fuzzy database systems,” Malaysian Journal of Computer Science, Vol. 10 No. 1, pp. 42-46, June 1997.

[3]. B.P. Buckles, F.E. Petry, and H.S. Sachar. “A domain calculus for fuzzy relational databases”. Fuzzy Sets and Systems, 29:327–340, 1989.

[4]. Janusz Kacprzyk and Andrzej Ziolkowski “Database queries with fuzzy linguistic quantifiers,” IEEE Transactions on Systems, Man, and Cybernetics, Vol.Smc-16, No. 3, May/June 1986.

[5]. Leonid Jose “Extending RDBMS for allowing fuzzy quantified queries,” Lecture Notes in Computer Science, Vol. 1873, Mohamed Ibrahim-Josef Küng-Norman Revell (Eds.) Springer Verlag, (2000), Pp. 407-416.

[6]. Tien-Chin Wang, Hsien-Da Lee and Chun-Ming Chen, ”Intelligent queries based on fuzzy set theory and SQL,” unpublished.

[7]. V.Balamurugan and K.Senthamarai Kannan, ”A framework for computing linguistic hedges in fuzzy queries,” international journal of database management system vol 2 no.1 february 2010.

[8]. Jaydev Mishra, ”Fuzzy query processing,” International Journal of Research and Reviews in Next Generation Networks Vol. 1, No. 1, March 2011.

[9]. Galindo J., Urrutia A., Piattini M., “Fuzzy databases: modeling, design and implementation,” published by Idea Group Publishing Hershey, USA, 2005.