

# Fuzzy Querying and Random Data

Przemysław Grzegorzewski  
Systems Research Institute,  
Polish Academy of Sciences  
and University of Information  
Technology and Management,  
Newelska 6, 01-447 Warsaw, Poland  
pgrzeg@ibspan.waw.pl

Edyta Mrówka  
Systems Research Institute,  
Polish Academy of Sciences  
and University of Information  
Technology and Management,  
Newelska 6, 01-447 Warsaw, Poland  
mrowka@wsisiz.edu.pl

## Abstract

Users are often interested in answers to vaguely defined questions that are natural for human but inconvenient for the machine. The traditional query languages manage binary queries only. In this paper we suggest how to construct fuzzy queries that apply to random data. We reach our goal by combining fuzzy query with a suitable statistical test for fuzzy hypotheses.

**Keywords:** Database querying, fuzzy querying, fuzzy logic, hypotheses testing, fuzzy hypotheses.

## 1 Introduction

Querying databases is to extract information that is of interest to the user. Traditionally, users cannot express vagueness in their requests when using a formal query language such as the Boolean one. This fact is a serious limitation since a typical user tends to formulate requirements in a natural language which abounds with imprecise expressions and vague terms. For this reason several approaches have been proposed to relax the rigidity of the conventional queries and to make possible the use of queries that allow for a more intelligent and human consistent information retrieval. Employing fuzzy logic we may construct queries which include various kind of vague statements expressed in a natural language. Using such fuzzy queries we deal no longer with binary outputs – whether a record fulfil given requirement or

not – but we get an information about the degree the record complies with requirements. The FQUERY for Access by Kacprzyk and Zadrozny is an example of a computer program that enables to create fuzzy queries. For more details on the theoretical and practical results we refer the reader e.g. to [1–7].

In the present paper we consider database with crisp data and we use fuzzy logic to handle them. However, restriction to crisp framework of the database do not eliminate other difficulties. Quite often we cannot describe an object or phenomenon thoroughly. Moreover, our measurements are subjected to inevitable errors. Thus the outcomes are uncertain. More precisely, their nature is random. Statistics provide powerful tools for dealing with such data. Therefore it seems natural to adopt some statistical methods in querying. In this paper we suggest how to construct fuzzy queries that apply to random data. We reach our goal by combining fuzzy query with a suitable statistical test. The traditional theory of hypotheses testing cannot be used there. Instead of classical statistical tests we propose to apply a test for fuzzy hypothesis. This test is described in Sec. 3. Contrary to the classical test, it does not lead to the binary decision but to a fuzzy one showing a grade of acceptance of given hypothesis. In Sec. 4 we show an example of fuzzy query and statistical test for fuzzy hypothesis that work together efficiently in a real-life problem.

## 2 The concept of fuzzy query

A typical query expressed in SQL is written in a following form: SELECT <list of attributes>

FROM <list of tables> WHERE <condition>. Its role is to select records (rows) that satisfies given condition. Each record from the table either satisfies or does not satisfy the condition and as a result we obtain a crisp set of matching database records.

A syntax of a fuzzy query that enables to use fuzzy terms is then SELECT <list of attributes> FROM <list of tables> WHERE <fuzzy condition>, where < fuzzy condition > may involve such linguistic terms as: numerical fuzzy values, categorical fuzzy values, fuzzy relations and linguistic quantifiers.

As soon as we accept vague terms in queries we also have to modify our meaning of *matching* between the query and a record (row) of database. Namely, it would be unreasonable to require the answer for a fuzzy query to be completely precise, adhering to the classical yes-no logic. Now we expect the system to produce a list of rows matching a query to a degree higher than a specified threshold. Thus, instead of listing the crisp set of rows that fulfill given condition, a fuzzy query yields a fuzzy set of such rows.

The crucial point is to determine these matching degrees. In the case of a simply query a matching degree  $S_i$  for the  $i$ -th record is characterized by a membership function of a fuzzy term  $T$ . Hence the matching degree is defined as follows

$$S_i = \mu_T(A_i), \quad (1)$$

where  $A_i$  represents value of the attribute  $A$  for the  $i$ -th record.

### 3 Testing fuzzy hypotheses

If a phenomenon under study is described by a probabilistic model then methods of mathematical statistics are used. Moreover, if our goal is to confirm or falsify given statement using random data then we have to apply some statistical tests. Assume that the investigated phenomenon is described by a probability distribution  $P_\theta$  which belongs to a family of distributions  $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ . We consider the null hypothesis  $H : \theta \in \Theta_H$  concerning the parameter  $\theta$ , with the alternative hypothesis  $K : \theta \in \Theta_K$ , where  $\Theta_H$  and  $\Theta_K$  are subsets of  $\Theta$ .

In the hypothesis testing problem we observe a random sample  $X_1, \dots, X_n$  and this observation can lead to one of two possible decisions: either to reject  $H$  (and to accept  $K$ ), or to not reject  $H$  (usually identified with accepting  $H$ ). Traditionally, an acceptance of  $H$  is denoted by zero and rejection of  $H$  by one. Hence, a decision rule, called a *statistical test*, can be defined as a function  $\phi : \mathcal{X} \rightarrow \{0, 1\}$ , where  $\mathcal{X}$  is a sample space.

Querying can be also regarded as a hypotheses testing problem, because we verify whether a record fulfill given requirement. In soft querying we deal with such fuzzy requirements like "about 10", "much more than 100", rather less than 5%", "quite big", etc. We may treat these fuzzy statements as fuzzy hypotheses. Thus it seems natural to consider fuzzy hypotheses testing problem, i.e. situation with fuzzy hypotheses but with crisp data and crisp test parameters.

Suppose that we consider a fuzzy null hypothesis  $H : \theta \in \Theta_H$  against fuzzy alternative hypothesis  $K : \theta \in \Theta_K = \neg\Theta_H$ , where  $\Theta_H$  and  $\Theta_K$  are fuzzy subsets of the parameter space  $\Theta$ , with membership functions  $\mu_H : \Theta \rightarrow [0, 1]$  and  $\mu_K(x) = 1 - \mu_H(x)$ , respectively. We'll transform such fuzzily formulated problem into a family of crisp testing problems. We distinguish three basic situations:

- if  $\text{supp}\mu_H$  is bounded then the problem of testing  $H : \theta \in \Theta_H$  against  $K : \theta \in \Theta_K = \neg\Theta_H$  can be transformed to a following family of crisp testing problems

$$\{H_{\theta_0} : \theta = \theta_0 \text{ vs. } K_{\theta_0} : \theta \neq \theta_0, \text{ where } \theta_0 \in \text{supp}\mu_H\}; \quad (2)$$

- if  $\text{supp}\mu_H$  is bounded only from the right then the problem of testing  $H : \theta \in \Theta_H$  against  $K : \theta \in \Theta_K = \neg\Theta_H$  can be transformed to a following family of crisp testing problems

$$\{H_{\theta_0} : \theta = \theta_0 \text{ vs. } K_{\theta_0} : \theta > \theta_0, \text{ where } \theta_0 \in \text{supp}\mu_H\}; \quad (3)$$

- if  $\text{supp}\mu_H$  is bounded only from the left then the problem of testing  $H : \theta \in \Theta_H$  against

$K : \theta \in \Theta_K = \neg\Theta_H$  can be transformed to a following family of crisp testing problems

$$\{H_{\theta_0} : \theta = \theta_0 \text{ vs. } K_{\theta_0} : \theta < \theta_0, \text{ where } \theta_0 \in \text{supp}\mu_H\}. \quad (4)$$

First situation corresponds to statements containing such fuzzy expressions like "about ...", "more or less ...", "approximately ...", "more or less between ... and ...", etc. Next two situations correspond to expressions of the type: "rather greater than ...", "rather smaller than ...", "much more than ...", "quite big", "rather high", "very small", etc.

Now let  $\{\phi_{\theta_0} : \mathcal{X} \rightarrow \{0, 1\}\}$ , where  $\theta_0 \in \text{supp}\mu_H$  denote a family of tests on significance level  $\alpha$  for verifying crisp hypotheses (2), (3) or (4). Then we get a following definition:

**Definition 1** A function  $\psi : \mathcal{X} \rightarrow \mathcal{F}(\{0, 1\})$  such that

$$\mu_\psi(0; x) = \begin{cases} \sup_{\theta_0 \in \Xi} \mu_H(\theta_0) & \text{if } \Xi \neq \emptyset \\ 0 & \text{if } \Xi = \emptyset, \end{cases} \quad (5)$$

$$\mu_\psi(1; x) = 1 - \mu_\psi(0; x) \quad (6)$$

where  $\Xi = \{\theta_0 \in \text{supp}\mu_H : \phi_{\theta_0}(x) = 0\}$  and  $\phi_{\theta_0}$  is a crisp test on a significance level  $\alpha$  for the testing problem (2), (3) or (4), is called a fuzzy test for verifying fuzzy hypothesis  $H : \theta \in \Theta_H$  against fuzzy alternative  $K : \theta \in \Theta_K = \neg\Theta_H$  on significance level  $\alpha$ .  $\square$

It is seen that our fuzzy test for fuzzy hypotheses does not always lead to binary decisions – to accept or to reject the null hypothesis – but to a fuzzy decision: we may get  $\psi = 1/0 + 0/1$  which indicates that we should accept  $H$ , or  $\psi = 0/0 + 1/1$  which means that  $H$  should be rejected, but we may also get  $\psi = \xi/0 + (1 - \xi)/1$ , where  $\xi \in (0, 1)$ , which can be interpreted as a degree of conviction that we should accept ( $\xi$ ) or reject ( $1 - \xi$ ) the hypothesis  $H$ . Therefore our fuzzy test for fuzzy hypotheses seems to be appropriate in constructing effective fuzzy querying in databases. It should also be stressed that our fuzzy test for fuzzy hypotheses is well defined because if the hypotheses are not fuzzy but crisp then fuzzy test  $\psi$  reduces to the usual statistical test.

## 4 Fuzzy query and random data

In the present section we show how to apply a fuzzy test for fuzzy hypotheses, described above, in a fuzzy query construction. Let  $A$  denote the attribute under study and let  $A_i$  denote its value for the  $i$ -th record. If our database contains random data, corresponding to measurements  $X_{i1}, \dots, X_{in}$ , then  $A_i$  is a function of these measurements, i.e.

$$A_i = h(X_{i1}, \dots, X_{in}). \quad (7)$$

In order to select records under particular fuzzy condition  $T$  described by fuzzy null hypothesis  $H$  against fuzzy alternative hypothesis  $K$ , we have to apply fuzzy test given by Definition 1. Let us denote the output of that test for the  $i$ -th attribute by  $\psi(A_i) = \xi/0 + (1 - \xi)/1$ , where  $\xi = \mu_\psi(0; A_i)$ . According to (1) we propose to compute the matching degree  $S_i$  in a following way

$$S_i = \mu_T(A_i) = f(\mu_\psi(0; A_i)), \quad (8)$$

where  $f : [0, 1] \rightarrow [0, 1]$ . In particular, we may use such natural functions like identity or the complement that lead to following definitions of matching degrees

$$S_i = \mu_\psi(0; A_i) \quad (9)$$

or

$$S_i = \mu_\psi(1; A_i) = 1 - \mu_\psi(0; A_i). \quad (10)$$

### Example

Let us consider a following simple query:

```
SELECT lake
FROM database of lakes
WHERE lake is not too deep.
```

Suppose that the statement "not too deep" is modelled by a fuzzy set with a following membership function

$$\mu_H(\theta) = \begin{cases} \frac{\theta-20}{5} & \text{if } 20 \leq \theta < 25 \\ \frac{30-\theta}{5} & \text{if } 25 \leq \theta \leq 30 \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

Now, because of the fuzzy query, we have to apply a test for fuzzy hypotheses. More precisely, we have to consider a fuzzy null hypothesis  $H : \theta \in \Theta_H$ , where  $\Theta_H$  is a fuzzy set corresponding to

the statement "not too deep" described by the membership function  $\mu_H$  given by (11), against fuzzy alternative hypothesis  $K : \theta \in \Theta_K = \neg\Theta_H$ , described by the membership function  $\mu_K(\theta) = 1 - \mu_H(\theta)$  for all  $\theta$ . Then, according to (2) we transform our fuzzy hypotheses testing problem to a following family of the crisp testing problems

$$\{H_{\theta_0} : \theta = \theta_0 \text{ vs. } K_{\theta_0} : \theta \neq \theta_0, \quad (12)$$

$$\text{where } \theta_0 \in [20, 30]\}.$$

By Definition 1 and assuming our characteristic is normally distributed one may get a following membership function of our fuzzy test

$$\mu_\psi(0; \bar{X}_i) = \begin{cases} 0 & \text{if } \bar{X}_i < 20 - \gamma \\ \eta_1 & \text{if } 20 - \gamma \leq \bar{X}_i < 25 - \gamma \\ 1 & \text{if } 25 - \gamma \leq \bar{X}_i \leq 25 + \gamma \\ \eta_2 & \text{if } 25 + \gamma < \bar{X}_i \leq 30 + \gamma \\ 0 & \text{if } \bar{X}_i > 30 + \gamma \end{cases} \quad (13)$$

$$\mu_\psi(1; \bar{X}_i) = 1 - \mu_\psi(0; \bar{X}_i),$$

where  $\bar{X}_i = \sum_{j=1}^n X_{ij}$  is a sample average,  $\gamma = t_{1-\frac{\alpha}{2}}^{[n-1]} \frac{S_i}{\sqrt{n}}$ ,  $S_i = \sqrt{\frac{1}{n-1} \sum_{j=1}^n (X_{ij} - \bar{X}_i)^2}$  denotes a sample standard deviation and  $t_{1-\frac{\alpha}{2}}^{[n-1]}$  is a quantile of order  $1 - \frac{\alpha}{2}$  from the  $t$ -Student distribution with  $n - 1$  degrees of freedom,  $\eta_1 = \frac{\bar{X}_i - 20 + \gamma}{5}$  and  $\eta_2 = \frac{30 + \gamma - \bar{X}_i}{5}$ .

Let  $A$  denote the attribute "lake's depth". Its value  $A_i$  for the  $i$ -th record (particular lake) is given by

$$A_i = \bar{X}_i, \quad (14)$$

where  $\bar{X}$  is a sample average of measurements of the depth obtained in  $n$  places of the considered lake. Now suppose that  $T$  denotes fuzzy statement: "typical depth of the lake is not too deep". It seems natural to use method (9) for computing matching degrees for of our simple query. Hence we get

$$S_i = \mu_T(A_i) = \mu_\psi(0; \bar{X}_i), \quad (15)$$

where  $\mu_\psi(0; \bar{X}_i)$  is given by (13).

## 5 Conclusions

In the present paper we have shown how to construct fuzzy queries that apply to random data. It could be done by combining fuzzy query with

a suitable statistical test for fuzzy hypotheses. The idea was presented by the example of a simple query and normally distributed random data. However, it can be also applied for more complex queries and for more general requirements on the data.

## References

- [1] M.Anvari, G.F. Rose (1987). Fuzzy Relational Databases. In: *The Fuzzy Analysis of Fuzzy Information*, J. Bezdek (ed.), CRC Press, Boca Raton, FL, USA.
- [2] P. Bosc, O. Pivert (1992). Fuzzy Querying in Conventional Databases. In: *Fuzzy Logic for Management of Uncertainty* L.A. Zadeh, J. Kacprzyk (eds.), Wiley, New York, pp. 645-671.
- [3] B.P. Buckles, F.E. Petry (1982). A Fuzzy Representation of Data for Relational Databases. *Fuzzy Sets and Systems*, 7, 213-226.
- [4] J. Kacprzyk, S. Zadrozny (1995). FQUERY for Access: Fuzzy Querying for a Windows-Based BDMS. In: *Fuzziness in Database Management Systems*, P. Bosc, J. Kacprzyk (eds.), Physica-Verlag, Heidelberg, pp. 415-433.
- [5] J. Kacprzyk, S. Zadrozny (1997). Fuzzy Queries in Microsoft Access v. 2. In: *Fuzzy Information Engineering - A Guided Tour of Applications*, D. Dubois, H. Prade, R.R. Yager (eds.), Wiley, New York, pp. 223-232.
- [6] J. Kacprzyk, A. Ziolkowski (1986). Database Queries with Linguistic Quantifiers. *IEEE Trans. SMC*, 16, 474-479.
- [7] F.E. Petry (1996). *Fuzzy Databases: Principles and Applications*. Kluwer, Boston.