Dynamically modelling of fuzzy sets for flexible data retrieval

Miroslav Hudec

Abstract Flexible querying allows users to implement linguistic terms to better qualify data they wish to obtain and rules to reveal. The question is how to properly construct fuzzy sets for each linguistic term. This issue is considered from the two aspects: user's view on particular linguistic term and on the current content in database. Evidently, the user can obtain the picture about stored data before running a query. This approach can be used in situations when a non-commutative operator is required. The rules extraction by linguistic quantifiers is another task where modelling of fuzzy sets can be applied. Institutions of official statistics deal with large amount of surveyed data and potentially useful administrative data, what makes them interesting for this approach.

1 Introduction

The increasing use of computers by business and governmental agencies has created mountains of data that contain potentially valuable knowledge (Rasmussen and Yager, 1997). The same holds for agencies of official statistics. Firstly, databases could contain crisp values which are not always accurately surveyed. Secondly, data from administrative sources contain valuable information which should be examined.

Flexible querying allows users to implement linguistic terms to better qualify data they wish to obtain and rules to reveal. For example, to find municipalities *where migration is small and unemployment is high*, or to find to which extent the rule *most of companies which report to Intrastat have value of trade near exemption threshold* is true. The linguistic terms clearly suggest that there is a smooth transition between acceptable and unacceptable records.

[1] Miroslav Hudec, Institute of Informatics and Statistics, Bratislava; email: hudec@infostat.sk

Several fuzzy query implementations have been proposed e.g. (Bosc and Pivert, 2000; Hudec, 2009; Kacprzyk and Zadrożny, 1995) and fuzzy queries for data mining (Rasmussen and Yager, 1997). In all approaches, the matching degree critically depends on constructed membership functions (Hudec and Sudzina, 2012).

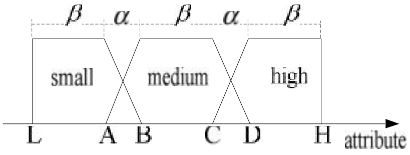
This paper examines construction of fuzzy sets for flexible queries and its usage in aggregation by fuzzy linguistic quantifiers and in situation when commutative operators are not appropriate.

2 Defining appropriate fuzzy sets for each linguistic term

Let D_{mim} and D_{max} be the lowest and the highest domain values of the attribute A (database column) i.e. Dom(A) = [D_{min}, D_{max}] and L and H be the lowest and the highest values from an current database content; that is, $[L, H] \subseteq [D_{min}, D_{max}]$. For many attributes in databases holds $[L, H] \subset [D_{min}, D_{max}]$; that is, intervals [$D_{min}$, L] and/or [H, D_{max}] are empty. This fact should be considered in data retrieval and rule extraction. Theoretically, the domain of attribute value of export is [exemption threshold value, $+\propto$]. The highest value of realized export is far from the "upper limit" of R+. In construction of term *high*, we need to consider stored real values.

Let the linguistic domain have elements {*small, medium, high*}. The linguistic domain covers the crisp sub domain of an attribute in a way illustrated in figure 1.

Figure 1: *Linguistic and crisp domain*



The first aspect allows users to freely define parameters of fuzzy sets (A, B, C, D). If the user is not familiar with the current database content, the query might easily end up with an empty answer. Moreover, the user is usually familiar with values of D_{min} and D_{max} but not with values of L and H.

The second aspect is focused on construction of membership functions (A, B, C, D) directly from current content of a database. The first method is the uniform domain covering method (Tudorie, 2008), depicted in figure 1. At the beginning, values of L and H are obtained from current database content. The length of fuzzy set core β and the slope α (Figure 1) are created in the following way (Tudorie, 2008):

$$\alpha = \frac{1}{8}(H - L) \tag{1}$$

$$\beta = \frac{1}{4}(H - L) \tag{2}$$

Consequently, it is easy to calculate required parameters A, B C and D.

The uniform domain covering method is appropriate when the distribution of attribute values in the domain is more or less uniform. If it is not the case, the uniform domain coverage could lead to a conclusion that the meaning of the linguistic term is far from real data. For these situations, the method can be improved by the statistical mean (Tudorie, 2008) or the logarithmic transformation (Hudec and Sudzina, 2012).

For the solution of data retrieval task both aspects should be taken into account. The above mentioned methods could be used to suggest parameters of fuzzy sets. In the second step, users can modify these parameters, if they are not satisfied with suggested ones, before running a query (Hudec and Sudzina, 2012).

3 Linguistic quantifiers

A special role among the aggregation operators play linguistic quantifiers such as *most* or *few*. For example, to find out whether in the Intrastat database *most of businesses have small value of intra-EU trade (are near the exemption threshold)*.

This problem is depicted in way Qx(Px), where Q denotes a linguistic quantifier, X ={x} is a universe of disclosure (set of all companies) and P(x) is a predicate corresponding to a query condition. In the first step we need to construct membership function for the term *small value of trade*. The uniform domain covering method (1) and (2) is the best option, because the main goal is not to retrieve data but to reveal rules. Value of *L* (figure 1) is the exemption value. The truth value of statement is computed by the following equation (Zadrożny and Kacprzyk, 2009):

$$Truth(Qx(Px)) = \mu_Q(\frac{1}{\text{n}}\sum_{i=1}^{n}\mu_{\text{P}}(x_i)) \tag{3}$$

where *n* is the cardinality of *X* and μ_Q (the quantifier *most*) might be given as:

$$\mu_Q(y) = \begin{cases} 1, & for\ \ y > 0.85 \\ 2y-6, & for\ \ 0.5 \le y \le 0.85 \\ 0, & for\ \ y < 0.5 \end{cases}$$

4 Non-commutative aggregation operator

T-norm functions are used for the aggregation under uncertainty. From the axiom that all t-norm functions are commutative, implies that they are applicable only if the order of elementary conditions is irrelevant. There exists a class of problems where elementary conditions are not independent, that is, the second elementary condition depends on answers obtained from the first one. Obviously this requires using a non-commutative operator. The *among* operator (Tudorie, 2008) meets this requirement:

$$\mu_{\text{P}_1 AMONG P_2} = \min(\mu_{P_1/P_2}(\text{a}_1), \mu_{\text{P}_2}(\text{a}_2)) \tag{4}$$

where a_1 and a_2 are database attributes, μ_{P2} is the membership function defining fulfilment of independent elementary condition and $\mu_{P1/P2}$ is the fulfilment degree of depended elementary condition relative to the independent one.

The example of this query is: *select companies which exported small amount of goods (P_1) among companies having high value of trade (P_2).*

In the first step companies with high value of trade (vt) are selected. The membership function of linguistic term *high* is calculated by one of methods examined in section 2 for the domain $[L_{vt}, H_{vt}]$ from the current content in database. Companies selected by P_2 create sub set of all companies in database. This subset constitutes reduced sub domain $[L_{ag\text{-}red}, H_{ag\text{-}red}] \subseteq [L_{ag}, H_{ag}]$ of amount of goods (ag). The fuzzy set small amount of goods is created on sub domain $[L_{ag\text{-}red}, H_{ag\text{-}red}]$. Even if the user can define parameters for membership function μ_{p2}, without suggestion from current database content, defining the membership function for $\mu_{p1/p2}$ is beyond his capabilities.

5 Conclusion

In this paper, we suggested a flexible SQL-like query language for data retrieval and data mining. The problem of construction of membership functions for data retrieval tasks and data mining can be satisfactorily solved if we merge the user's opinion about linguistic terms with the current content in database. This approach is also a supporting tool for queries where elementary conditions are not independent and for extracting rules by linguistic quantifiers.

In addition, this approach is open for further improvements like: querying over missing values when users know functional dependencies between attributes and querying using priorities between elementary conditions.

The research reported herein was funded by the European Commission via the Seventh Framework Programme for Research (FP7/2007-2013) under Grant agreement n°244767. This work was supported by the Slovak Research and Development Agency under the contract No. DO7RP-0024-10.

References

1. Bosc, P., Pivert, O.: SQLf query functionality on top of a regular relational database management system. In: Pons, M., Vila, M.A., Kacprzyk, J. (eds.) Knowledge Management in Fuzzy Databases, pp. 171-190. Physica-Verlag, Heidelberg (2000).
2. Hudec, M.: An approach to fuzzy database querying, analysis and realisation. Comput. Sci. Inf. Syst. 6(2), 127-140 (2009).
3. Hudec, M., Sudzina, F.: Construction of fuzzy sets and applying aggregation operators for fuzzy queries. In: In: 14th International Conference on Enterprise Information Systems (ICEIS 2012), Wroclav (2012). Accepted for publication.
4. Kacprzyk, J., Zadrożny, S.: FQUERY for Access: Fuzzy querying for windows-based DBMS. In: Bosc, P., Kacprzyk, J. (eds.) Fuzziness in Database Management Systems, pp. 415-433. Physica-Verlag, Heidelberg (1995).
5. Rasmussen, D., Yager, R.R.: Summary SQL - A Fuzzy Tool for Data Mining. Intell. Data Anal. 1, 49-58 (1997)
6. Tudorie, C.: Qualifying objects in classical relational database querying. In: Galindo J. (ed.) Handbook of Research on Fuzzy Information Processing in Databases, pp. 218-245. IGI Global, London (2008).
7. Zadrożny, S., Kacprzyk, J.: Issues in the practical use of the OWA operators in fuzzy querying. J. Intell. Inf. Syst. 33, 307-325 (2009).