On Tuning OWA Operators in a Flexible Querying Interface

Sławomir Zadrożny[1] and Janusz Kacprzyk[2]

[1] Warsaw School of Information Technology,
ul. Newelska 6, 01-447 Warsaw, Poland
[2] Systems Research Institute PAS,
ul. Newelska 6, 01-447 Warsaw, Poland

Abstract. The use of the Yager's OWA operators within a flexible querying interface is discussed. The key issue is the adaptation of an OWA operator to the specifics of a user's query. Some well-known approaches to the manipulation of the weights vector are reconsidered and a new one is proposed that is simple and efficient.

1 Introduction

We consider a flexible querying interface supporting an extended version of SQL such as those proposed by Kacprzyk and Zadrożny [1, 2] and Bosc et al. [3]. Basically an extension of a traditional querying language is meant here to support *linguistic terms* in queries exemplified by fuzzy values such as "young" and fuzzy relations (fuzzy comparison operators) such as "much greater than" in the following SQL query:

```
SELECT *
FROM   employees
WHERE  (age IS young) AND
       (salary IS MUCH GREATER THAN 50000 USD)
```
(1)

Another class of relevant linguistic terms are *linguistic quantifiers* such as "most", "almost all" etc. In the extended query language advocated here they play the role of flexible aggregation operators. Kacprzyk and Ziółkowski [4], and then Kacprzyk, Ziółkowski and Zadrożny [5] proposed to use them to aggregate conditions in the WHERE clause of the SQL SELECT statement as, e.g. in

"*Most* of conditions among 'age IS *young*, salary IS *high*,. . . ' are to be satisfied"

Bosc et al. (cf. e.g. [6]) proposed to use linguistic quantifiers with subqueries or against groups of rows as, e.g., in

```
SELECT    deptno
FROM      employees
GROUP BY  deptno
HAVING    most_of (young are well-paid)
```
(2)

H. Legind Larsen et al. (Eds.): FQAS 2006, LNAI 4027, pp. 97–108, 2006.

Whatever the role a linguistic quantifier, it has to be somehow modelled, and a user has to be provided with some means for its definition and manipulation. Here we assume that the linguistic quantifiers are originally defined and interpreted in the sense of Zadeh. Then, during a query execution they are automatically re-interpreted in terms of Yager's OWA operators due to their high operability and intuitive appeal. Before the actual query execution the user may modify the OWA operators present in the query so as to better adjust them to his or her needs. We focus here on the guidelines that should be presented to the user in order to help him or her in an appropriate definition and manipulation of the OWA operator.

We will now briefly discuss linguistic quantifiers and Yager's OWA operators, well-known approaches to their tuning, show how they are used and manipulated in queries, and finally present an algorithm implemented in our FQUERY for Access package [1, 2].

2 Linguistic Quantifiers and the OWA Operators

Our starting point is Zadeh's calculus of linguistically quantified propositions [7] used to. It is a framework meant to model such expressions of natural language like

$$\text{“}\textit{Most}\text{ Swedes are }\textit{tall}\text{”} \tag{3}$$

where "Most" is an example of a linguistic quantifier. Other examples include "almost all", "much more than 50%" etc. We are here interested only in *relative* quantifiers such that: - their semantics refers to the proportion of elements possessing a certain property (in Example (3) it is the set of tall Swedes) among all the elements of the universe of discourse (in Example (3) it is the set of all Swedes);
and *nondecreasing* such that: - the larger such a proportion the higher the truth value of a proposition containing such a linguistic quantifier.

A linguistically quantified proposition exemplified by (3) might be formally written in a general form as

$$QxP(x) \tag{4}$$

where Q denotes a linguistic quantifier (e.g., *most*), $X = \{x\}$ is a universe of discourse (e.g., a set of Swedes), and $P(x)$ is a predicate corresponding to a certain property (e.g., of being *tall*).

The truth value of (4) is computed as follows. The relative quantifier Q is equated with a fuzzy set defined in $[0, 1]$. In particular, for a regular nondecreasing quantifier its μ_Q is assumed to be nondecreasing and normal, i.e.,

$$x \leq y \Rightarrow \mu_Q(x) \leq \mu_Q(y); \quad \mu_Q(0) = 0; \quad \mu_Q(1) = 1 \tag{5}$$

The particular $y \in [0, 1]$ correspond to proportions of elements possessing property P and $\mu_Q(y)$ assesses the degree to which a given proportion matches