Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Sciences 173 (2005) 281–304
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools

Janusz Kacprzyk [a,b,*], Sławomir Zadrożny [b]

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

Received 24 September 2003; received in revised form 3 January 2004; accepted 6 March 2004

Abstract

We consider linguistic data(base) summaries in the sense of Yager [Information Sciences 28 (1982) 69–86], exemplified by "most employees are young and well paid" (with some degree of truth added), for a personnel database, as an intuitive, human consistent and natural language based knowledge discovery tool. We present first an extension of the classic Yager's approach to involve more sophisticated criteria of goodness, search methods, etc. We advocate the use of the concept of a protoform (prototypical form), that is recently vividly advocated by Zadeh [A prototype-centered approach to adding deduction capabilities to search engines—the concept of a protoform. BISC Seminar, University of California, Berkeley, 2002], as a general form of a linguistic data summary. We present an extension of our interactive approach, based on fuzzy logic and fuzzy database queries, which makes it possible to implement such linguistic data summaries. We show how fuzzy queries are related to linguistic summaries, and show that one can introduce a hierarchy of protoforms, or abstract summaries in the sense of latest

* Corresponding author at: Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6, 01-447 Warsaw, Poland. Tel.: +48 22 836 44 14 01 447; fax: +48 22 837 27 72.
E-mail addresses: kacprzyk@ibspan.waw.pl (J. Kacprzyk), zadrozny@ibspan.waw.pl (S. Zadrożny).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2005.03.002

Zadeh's [A prototype-centered approach to adding deduction capabilities to search engines—the concept of a protoform. BISC Seminar, University of California, Berkeley, 2002] ideas meant mainly for increasing deduction capabilities of search engines. For illustration we show an implementation for a sales database in a computer retailer, employing some type of a protoform of a linguistic summary.

Keywords: Fuzzy logic; Computing with words and perceptions; Protoform; Data mining; Linguistic summarization

1. Introduction

The recent growth of Information Technology (IT) has implied, among others, the availability of a huge amount of data (from diverse, often remote sources, notably databases). Unfortunately, the raw data alone are often not useful and do not provide "knowledge". More important are relevant, non-trivial dependencies which are encoded in those data. Unfortunately, they are usually hidden, and their discovery is not a trivial act that requires some intelligence.

Data summarization is one of basic capabilities that is now needed by any "intelligent" system that is meant to operate in real life. Basically, due to the availability of relatively cheap and efficient hardware and software tools, we usually face an abundance of data that is beyond human cognitive and comprehension skills.

Since for a human being the only fully natural means of communication is natural language, a linguistic (for instance, by a sentence or a small number of sentences in a natural language) summarization of a set of data would be very desirable and human consistent. For instance, having a data set on employees, a statement (linguistic summary) like "almost all younger and well qualified employees are well paid" would be useful and human consistent.

This may clearly be an instance of a paradigm shift that is advocated in recent time the essence of which is to use natural language to represent values, relations, etc. characterizing some system or situation. This promises human consistency and simplicity though may be at the expense of precision. This trend has become more and more pronounced recently, and a most prominent example is the so-called "computing with words (and perceptions) paradigm" introduced by Zadeh in the mid-1990s, and extensively presented in Zadeh and Kacprzyk's [55] books.

Unfortunately, data summarization is still in general unsolved a problem in spite of vast research efforts. Very many techniques are available but they are not "intelligent enough", and not human consistent, partly due to a little use of natural language. This concerns, e.g., summarizing statistics, exemplified by the

average, median, minimum, maximum, α-percentile, etc. which—in spite of recent efforts to soften them—are still far from being able to reflect a real human perception of their essence.

In this paper we will show the use of linguistic database summaries introduced by Yager [44,46–48], and then considerably advanced by Kacprzyk [12], Kacprzyk and Yager [15], and Kacprzyk et al. [16,17], Zadrożny and Kacprzyk [56], and implemented in Kacprzyk and Zadrożny [21,23–33]. We will derive linguistic data summaries as linguistically quantified propositions as, e.g., "most of the employees are young and well paid", with a degree of validity (truth, ...), in case of a personnel database.

We follow Yager's [44,46] idea, and present its implementation, mainly using Kacprzyk and Yager [15], and Kacprzyk et al. [16,17] extensions. Basically, we advocate that the degree of truth (or, more generally, validity) originally proposed by Yager [44] is not sufficient, and other validity (quality) indicators should be added. We mention George and Srikanth's [9] solution in which a compromise between the specificity and generality of a summary is sought, and then present some extension of Kacprzyk and Strykowski's [14] approach in which a weighted sum of five quality indicators is employed that was proposed by Kacprzyk and Yager [15], and Kacprzyk et al. [16,17].

It should be noted that we do not consider in this paper some other approaches to the linguistic summarization of databases (data sets) that are based on a different philosophy, exemplified by works by Bosc et al. [4], Dubois and Prade [8], Raschia and Mouaddib [39] or Rassmussen and Yager [40–43]. Nor we consider some other related techniques exemplified by the mining of fuzzy association rules (cf. Chen et al. [5], Chen and Wei [6], Chen et al. [7], Hu et al. [11], Lee and Lee-Kwang [36]), even in the context of linguistic summaries (cf. Kacprzyk and Zadrożny [30,33]). Of course, our approach is specific and is just one of possible approaches to data mining and knowledge discovery. The interested reader can find much information on this topic in numerous papers and books, and a notable, very rich and comprehensive source is Kluwer's journal "Data Mining and Knowledge Discovery" (cf. www.kluweronline.com).

We employ Kacprzyk and Zadrożny's [21,23–26,29] idea of an interactive approach to linguistic summaries in which the determination of a class of summaries of interest is done via Kacprzyk and Zadrożny's [18–20,28] FQUERY for access, a fuzzy querying add-on to access, extended to the querying over the Internet in Kacprzyk and Zadrożny [24]. The line of reasoning is that since a fully automatic generation of linguistic summaries is not feasible at present, an interaction with the user is assumed for the determination of a class of summaries of interest, and this is done via the above Kacprzyk and Zadrożny's fuzzy querying add-on to Microsoft Access.

It should be noted that we use here our approach to fuzzy querying though there are other powerful and efficient approaches that can easily be found in the literature (cf. Kacprzyk et al. [13] for a review).

Extending our suggestion given in Kacprzyk and Zadrożny [32], we show that by relating various types of linguistic summaries to fuzzy queries, with various known and sought elements, we can arrive at a hierarchy of prototypical forms, or—in Zadeh's [53] terminology—protoforms, of linguistic data summaries. This seems to be a very powerful conceptual idea in particular in view of recent Zadeh's numerous remarks that protoforms are crucial for the formalization of human consistent reasoning, deduction capabilities of search engines, etc.

We present an implementation of the proposed approach to the derivation of linguistic summaries for a sales database of a computer retailer. We show that the linguistic summaries obtained may be very useful for supporting decision making by the owner related to some relevant aspects of business functioning and running. On the other hand, this implementation may be viewed as a step towards the implementation of protoforms of linguistic summaries. We extend the results by Kacprzyk and Strykowski [14], Kacprzyk et al. [17], etc. by including data from outside sources fetched via the Internet.

2. Linguistic summaries using fuzzy logic with linguistic quantifiers

Here we will briefly present the basic Yager's [44] approach to the linguistic summarization of sets of data (databases). This will provide a point of departure for our further analysis of more complicated and realistic summaries.

In Yager's [44] approach, we have (we use here the source terminology):

- V is a quality (attribute) of interest, e.g. salary in a database of workers,
- $Y = \{y_1, \ldots, y_n\}$ is a set of objects (records) that manifest quality V, e.g. the set of workers; hence $V(y_i)$ are values of quality V for object y_i,
- $D = \{V(y_1), \ldots, V(y_n)\}$ is a set of data (the "database" on question).

A *linguistic summary* of a data set consists of:

- a summarizer S (e.g. young),
- a quantity in agreement Q (e.g. most),
- truth T—e.g. 0.7,

as, e.g., "T (*most* of employees are *young*) = 0.7". The truth T may be meant in a more general sense, e.g. as validity of, even more generally, as some quality or goodness of a linguistic summary.

Basically, given a set of data D, we can hypothetize any appropriate summarizer S and any quantity in agreement Q, and the assumed measure of truth will indicate the truth of the statement that Q data items satisfy the statement (summarizer) S.

Notice that we consider here some specific, basic form of a linguistic summary. We do not consider other forms of summaries exemplified by "over 70% of employees are under 35 years of age" that may be viewed to provide similar information as "most of employees are young" because the latter are clearly outside of the class of linguistic summaries considered. Notice also that we discuss here the linguistic summarization of sets of numeric values only. One can clearly imagine the linguistic summarization of symbolic attributes but this relevant problem is outside of the scope of this paper.

First, we should comment on the form of the basic elements of the summary, i.e. the summarizer, quantity in agreement, and how to calculate the degree of truth.

2.1. On the form of the summarizer

Since the only fully natural and human consistent means of communication for the humans is natural language, then we assume that the summarizer S is a linguistic expression semantically represented by a fuzzy set. For instance, in our example a summarizer like "young" would be represented as a fuzzy set in the universe of discourse as, e.g., $\{1, 2, \ldots, 90\}$, i.e., containing possible values of the human age, and "young" could be given as, e.g., a fuzzy set with a non-increasing membership function in that universe such that, in a simple case of a piece-wise linear membership function, the age up to 35 years is for sure "young", i.e. the grade of membership is equal to 1, the age over 50 years is for sure "not young", i.e. the grade of membership is equal to 0, and for the ages between 35 and 50 years the grades of membership are between 1 and 0, the higher the age the lower its corresponding grade of membership. Clearly, the meaning of the summarizer, i.e. its corresponding fuzzy set is in practice subjective, and may be either predefined or elicited from the user (we will comment on this later on while describing the system implemented).

Such a simple one-attribute-related summarizer exemplified by "young" does well serve the purpose of introducing the concept of a linguistic summary, hence it was assumed by Yager [44]. However, it is of a lesser practical relevance. It can be extended for some confluence of attribute values as, e.g, "*young* and *well paid*", and then to more complicated combinations.

Clearly, when we try to linguistically summarize data, the most interesting are non-trivial, *human consistent* summarizers (concepts) as, e.g.:

- *productive* workers,
- *stimulating* work environment,
- *difficult* orders, etc.

involving complicated *combinations of attributes*, e.g., a hierarchy (not all attributes are of the same importance), the attribute values are ANDed and/or ORed, k out of n, *most*, etc. of them should be accounted for, etc.

The generation and processing of such non-trivial summarizers needs some specific tools and techniques to be discussed later.

2.2. On the form of the quantity in agreement

The quantity in agreement, Q, is a proposed indication of the extent to which the data satisfy the summary. Once again, a precise indication is not human consistent, and a linguistic term represented by a fuzzy set is employed.

Basically, two types of such a linguistic quantity in agreement can be used:

- absolute as, e.g., "about 5", "more or less 100", "several", and
- relative as, e.g., "a few", "more or less a half", "most", "almost all", etc.

Notice that the above linguistic expressions are the so-called fuzzy linguistic quantifiers (cf. Zadeh [51,52]) that can be handled by fuzzy logic.

As for the fuzzy summarizer, also in case of a fuzzy quantity in agreement, its form is subjective, and can be either predefined or elicited from the user.

2.3. Calculation of the truth of a linguistic summary

Basically, the calculation of the truth (or, more generally, validity or even quality; but, we will assume the case of truth, for simplicity) of the basic type of a linguistic summary considered in this section is equivalent to the calculation of the truth value (from the unit interval) of a linguistically quantified statement (e.g., "*most* of the employees are *young*"). This may be done by two most relevant techniques using either Zadeh's [51] calculus of linguistically quantified statements (cf. Zadeh and Kacprzyk [55]) or Yager's [45] OWA operators (cf. Yager and Kacprzyk [49]); for a survey, see also Liu and Kerre [37].

A linguistically quantified proposition, exemplified by "most experts are convinced", is written as "Qy's are F" where Q is a linguistic quantifier (e.g., most), $Y = \{y\}$ is a set of objects (e.g., experts), and F is a property (e.g., convinced). Importance B may be added yielding "QBy's are F", e.g., "most (Q) of the important (B) experts (y's) are convinced (F)". The problem is to find truth (Qy's are F) or truth (QBy's are F), respectively, knowing truth (y is F), $\forall y \in Y$ which is done here using Zadeh's [x] fuzzy-logic-based calculus of linguistically quantified propositions.

Property F and importance B are fuzzy sets in Y, and a (proportional, non-decreasing) linguistic quantifier Q is assumed to be a fuzzy set in $[0,1]$ as, e.g.

$$\mu_Q(x) = \begin{cases} 1 & \text{for } x \geqslant 0.8 \\ 2x - 0.6 & \text{for } 0.3 < x < 0.8 \\ 0 & \text{for } x \leqslant 0.3 \end{cases} \tag{1}$$

Then, due to Zadeh [51]

$$\text{truth}(Qy\text{'s are } F) = \mu_Q\left[\frac{1}{n}\sum_{i=1}^{n}\mu_F(y_i)\right] \tag{2}$$

$$\text{truth}(QBy\text{'s are } F) = \mu_Q\left[\sum_{i=1}^{n}(\mu_B(y_i)\wedge\mu_F(y_i))\Big/\sum_{i=1}^{n}\mu_B(y_i)\right] \tag{3}$$

An OWA operator (Yager [45]; Yager and Kacprzyk [49]) of dimension p is a mapping $F:[0,1]^p \to [0,1]$ if associated with F is a weighting vector $W = [w_1,\ldots,w_p]^{\mathrm{T}}$, $w_i \in [0,1], w_1 + \cdots + w_p = 1$, and

$$F(x_1,\ldots,x_p) = w_1b_1 + \cdots w_pb_p = W^{\mathrm{T}}B \tag{4}$$

where b_i is the ith largest element among $x_1,\ldots,x_p$, $B = [b_1,\ldots,b_p]$.

The OWA weights may be found from the membership function of Q due to (cf. Yager [45]):

$$w_i = \begin{cases} \mu_Q(i) - \mu_Q(i-1) & \text{for } i = 1,\ldots,p \\ \mu_Q(0) & \text{for } i = 0 \end{cases} \tag{5}$$

The OWA operators can model a wide array of aggregation operators (including linguistic quantifiers), from $w_1 = \cdots = w_{p-1} = 0$ and $w_p = 1$ which corresponds to "all", to $w_1 = 1$ and $w_2 = \cdots = w_p = 0$ which corresponds to "at least one", through all intermediate situations.

An important issue is related with the OWA operators for importance qualified data. Suppose that we have $A = [a_1,\ldots,a_p]$, and a vector of importances $V = [v_1,\ldots,v_p]$ such that $v_i \in [0,1]$ is the importance of a_i, $i = 1,\ldots,p$, $v_1 + \cdots + v_p = 1$. The case of an *ordered weighted averaging operator with importance qualification*, denoted OWA_Q is not trivial. In Yager's [45] approach to be used here, which seems to be highly plausible (though is sometimes criticized), some redefinition of the OWA's weights w_i's into $\bar{w}_i$'s is performed, and (4) becomes

$$F_I(x_1,\ldots,x_p) = \bar{w}_1b_1 + \cdots + \bar{w}_pb_p = \overline{W}^{\mathrm{T}}B \tag{6}$$

where

$$w_j = \mu_Q\left(\frac{\sum_{k=1}^{j}u_k}{\sum_{k=1}^{p}u_k}\right) - \mu_Q\left(\frac{\sum_{k=1}^{j-1}u_k}{\sum_{k=1}^{p}u_k}\right) \tag{7}$$

where u_k is the importance of b_k, i.e. the k-largest element of A (i.e. the corresponding v_k).

The OWA operators offer a wide array of means for aggregation based on various quantifiers, both crisp and fuzzy, though they may lead to somewhat

more complicated calculation formulas which can be of relevance in case of summarizing large databases. In the implementation presented later the user may choose between the OWA based calculations or the classic Zadeh's calculus.

3. Some other validity criteria

The basic validity criterion, i.e. the truth of a linguistically quantified statement given by (2) and (3), is certainly the most important in the general framework assumed. However, it does not grasp all aspects of a linguistic summary. Some attempts at devising other quality (validity) criteria will be briefly surveyed following Kacprzyk and Yager [15], and extensions given in Kacprzyk, et al. [16].

First, Yager [44] proposed a measure of informativeness that may be summarized as follows. Suppose that we have a data set, whose elements are from a measurement space X. One can say that the data set itself is its own most informative description, and any other summary implies a loss of information. So, a natural question is whether a particular summary is informative, and to what extent.

It turns out that the degree of truth used so far is not a good measure of informativeness (cf. Yager [44,46]). Let the summary be characterized by the triple (S, Q, T), and let a related summary be characterized by the triple (S^c, Q^c, T) such that S^c is the negation of S, i.e. $\mu_S^c(\cdot) = 1 - \mu_S(\cdot)$, and similarly $\mu_Q^c(\cdot) = 1 - \mu_Q(\cdot)$.

Then, Yager [44,46] proposed the following measure of informativeness of a summary

$$I = [T \cdot SP(Q) \cdot SP(S)] \vee [(1 - T) \cdot Sp(Q^c)Sp(S^c)] \tag{8}$$

where $SP(Q)$ is the specificity of Q given as

$$SP(Q) = \int_0^1 \frac{1}{\operatorname{card} Q_\alpha} d_\alpha \tag{9}$$

where Q_α is the α-cut of Q and card$(\cdot)$ is the cardinality of the respective set; and similarly for Q^c, S, S^c.

This measure of informativeness results from a very plausible reasoning which can be found, e.g., in Yager [44,46]. Notice that this reasoning differs from, e.g., that in Chen et al. [5].

Unfortunately, though the above measure of informativeness is plausible and a considerable step forward, it is by no means a definite solution. First, let us briefly mention George and Srikanth's [9] proposal. Suppose that a linguistic summary of interest involves more than one attribute (e.g., "age", "salary" and "seniority" in the case of employees). Basically, for the same set of data, two summaries are generated:

- a *constraint descriptor* which is the most specific description (summary) that fits the largest number of tuples in the relation (database) involving the attributes in question,
- a *constituent descriptor* which is the description (summary) that fits the largest subset of tuples with the condition that each tuple attribute value takes on at least a threshold value of membership.

George and Srikanth [9] use these two summaries to derive a fitness function (goodness of a summary) that is later used for deriving a solution (a best summary) via a genetic algorithm they employ. This fitness function represents a compromise between the most specific summary (corresponding to the constraint descriptor) and the most general summary (corresponding to the constituent descriptor).

Then, Kacprzyk [12], and Kacprzyk and Strykowski [14] proposed some additional measures that have been further developed by Kacprzyk and Yager [15] and Kacprzyk et al. [16].

For convenience of the reader, let us briefly repeat some basic notation. We have a data set (database) D that concerns some objects (e.g. employees) $Y = \{y_1, \ldots, y_n\}$ described by some attribute V (e.g. age) taking on values in a set $X = \{x_1, x_2, \ldots\}$. exemplified by $\{20, 21, \ldots, 100\}$ or {very young, young,..., old, very old}. Let $d_i = V(y_i)$ denote the value of attribute V for object y_i. Therefore, the data set to be summarized is given as a table

$$D = [d_1, \ldots, d_n] = [V(y_1), V(y_2), \ldots, V(y_n)] \tag{10}$$

In a more realistic case the data set is described by more than one attribute, and let $V = \{V_1, V_2, \ldots, V_m\}$ be a set of such attributes taking values in X_i $i = 1, \ldots, m$; $V_j(y_i)$ denotes the value of attribute V_j for object y_i and attribute V_j takes on its values from a set X_j.

The data set to be summarized is therefore:

$$\begin{aligned} D = \{&[V_1(y_1), V_2(y_1), \ldots, V_m(y_1)], [V_1(y_2), V_2(y_2), \ldots, V_m(y_2)], \ldots, \\ &[V_1(y_n), V_2(y_n), \ldots, V_m(y_n)]\} \end{aligned} \tag{11}$$

In this case of multiple attributes the description (summarizer) S is a family of fuzzy sets $S = \{S_1, S_2, \ldots S_m\}$ where $S_i \in S$ is a fuzzy set in X_i, $i = 1, \ldots, m$. Then, $\mu_S(y_i)$, $i = 1, 2, \ldots, n$, may be defined as:

$$\mu_S(y_i) = \min_{j \in \{1,2,\ldots,m\}} [\mu_{S_j}(V_j(y_i))] \tag{12}$$

So, having S, we can calculate the truth value T of a summary for any quantity in agreement. However, to find a best summary, we should calculate T for each possible summarizer, and for each record in the database in question. This computational prohibitive for virtually all non-trivial databases and number of attributes.

A natural line of reasoning would be to either to limit the number of attributes of interest or to limit the class of possible summaries by setting a more specific description by predefining a "narrower" description (e.g. very young, young and well paid, etc. employees). This will limit the search space.

We will deal here with the second option. The user can limit the scope of a linguistic summary to, e.g., those for which the attribute "age" takes on the value "young (employees)" only, i.e., to fix the summarizer related to that attribute. That is, this will correspond to the searching of the database using the query w_g equated with the fuzzy set in X_g corresponding to "young" related to attribute V_g (i.e. age), i.e. characterized by $\mu_{w_g}(\cdot)$. In such a case, $\mu_S(y_i)$ given by (12) becomes

$$\mu_S(y_i) = \min_{j_\in\{1,2,\ldots,m\}}[\mu_{Sj}(V_j(y_i)) \wedge \mu_{w_g}(V_g(y_i))], \quad i = 1,\ldots,n \tag{13}$$

where "$\wedge$" is the minimum (or, more generally, a t-norm), and then

$$r = \frac{\sum_{i=1}^{n}\mu_S(y_i)}{\sum_{i=1}^{n}\mu_{w_g}(V_g(y_i))} \tag{14}$$

and $T = \mu_Q(r)$.

Now, we will present the five new quality measures of quality of linguistic database summaries introduced, and further developed, in Kacprzyk [12], Kacprzyk and Strykowski [14], Kacprzyk and Yager [15], and Kacprzyk et al. [16]:

- a truth value [which basically corresponds to the degree of truth of a linguistically quantified proposition representing the summary given by, e.g., (2) or (3)],
- a degree of imprecision,
- a degree of covering,
- a degree of appropriateness,
- a length of a summary,

and these degrees will now be formally defined.

For notational simplicity, let us rewrite (13) and (14) as:

$$\mu_S(y_i) = \min_{j\in\{1,2,\ldots,m\}}[(\mu_{Sj}(V_j(y_i)))], \quad i = 1,\ldots,n \tag{15}$$

and

$$r = \frac{\sum_{i=1}^{n}[\mu_S(y_i) \wedge \mu_{w_g}(V_g(y_i))]}{\sum_{i=1}^{n}\mu_{w_g}(V_g(y_i))} \tag{16}$$

where, clearly, (15) and (16) are equivalent to (12) and (13) though rewritten in the form more suitable for our present discussion.

The *degree of truth*, T_1 is the basic validity criterion introduced in the source Yager's [44,46] works and commonly employed. It is clearly equal to

$$T_1 = \mu_Q(r) \tag{17}$$

and (17) results clearly from Zadeh's [51] calculus of linguistically quantified propositions.

The *degree of imprecision* is an obvious and important validity criterion. Basically, a very imprecise linguistic summary (e.g. on almost all winter days the temperature is rather cold) has a very high degree of truth yet it is not useful.

Suppose that description (summarizer) S is given as a family of fuzzy sets $S = \{s_1, s_2, \ldots, s_m,\}$. For a fuzzy set s_j, $j = 1, \ldots, m$, we can define its degree of fuzziness as, e.g.:

$$\text{in}(s_j) = \frac{\text{card}\{x \in X_j : \mu_{s_j}(x) > 0\}}{\text{card}X_j} \tag{18}$$

where card denotes the cardinality of the corresponding (non-fuzzy) set. That is, the "flatter" the fuzzy set s_j the higher the value of in (s_j). The degree of imprecision (fuzziness), T_2 of the summary—or, in fact, of S—is then defined as:

$$T_2 = 1 - \sqrt[m]{\prod_{j=1,\ldots,m} \text{in}(s_j)} \tag{19}$$

Notice that the degree of imprecision T_2 depends on the form of the summary only and not on the database, that is its calculation does not require the searching of the database (all its records) which is very important.

The *degree of covering*, T_3 is defined as

$$T_3 \frac{\sum_{i=1}^{n} t_i}{\sum_{i=1}^{n} h_i} \tag{20}$$

where

$$t_i = \begin{cases} 1 & \text{if } \mu_S(y_i) > 0 \quad \text{and} \quad \mu_{w_g}(V_g(y_i)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$h_i = \begin{cases} 1 & \text{if } \mu_{w_g}(V_g(y_i)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

The degree of covering says how many objects in the data set corresponding to the query w_g are "covered" by the particular summary, i.e. to the particular description S. Its interpretation is simple as, e.g., if it is equal to 0.15, then this means that 15% of the objects are consistent with the summary in question. The value of this degree depends clearly on the contents of the database.

The *degree of appropriateness* is the most relevant degree of validity. To present its idea, suppose that the summary containing the description (fuzzy sets) $S = (S_1, S_2, \ldots, S_m)$ is partitioned into m partial summaries each of which encompasses the particular attributes $S_1, S_2, \ldots, S_m$, such that each partial summary corresponds to one fuzzy value only, then if we denote:

$$S_j(y_i) = \mu_{S_j}(V_j(y_i)) \tag{21}$$

then

$$r_j = \frac{\sum_{i=1}^{n} h_i}{n}, \quad j = 1, \ldots, n \tag{22}$$

where

$$h_i = \begin{cases} 1 & \text{if } S_j(y_i) > 0 \\ 0 & \text{otherwise} \end{cases}$$

and the degree of appropriateness, T_4 is defined as:

$$T_4 = \operatorname{abs}\left(\prod_{j=1,\ldots,m} r_j - T_3 \right) \tag{23}$$

The degree of appropriateness means that, for a database concerning the employees, if 50% of them are less than 25 years old and 50% are highly qualified, then we may expect that 25% of the employees would be less than 25 years old and highly qualified; this would correspond to a typical, fully expected situation. However, if the degree of appropriateness is, e.g., 0.39 (i.e. 39% are less than 25 years old and highly qualified), then the summary found reflects an interesting, not fully expected relation in our data. This degree describes therefore how characteristic for the particular database the summary found is. T_4 is very important because, for instance, a trivial summary like "100 % of sale is of any articles" has full validity (truth) if we use the traditional degree of truth but its degree of appropriateness is equal 0 which is correct.

The *length* of a summary is relevant because a long summary is not easily comprehensible by the human user. This length, T_5 may be defined in various way, and the below form has proven to be useful:

$$T_5 = 2(0.5^{\operatorname{card} S}) \tag{24}$$

where card S is the number of elements in S.

Now, the (total) degree of validity, T, of a particular linguistic summary is defined as the weighted average of the above 5 degrees of validity, i.e.,

$$T = T(T_1, T_2, T_3, T_4, T_5; w_1, w_2, w_3, w_4, w_5) = \sum_{i=1,2,\ldots,5} w_i T_i \tag{25}$$

And the problem is to find an optimal summary, $S^* \in S$, such that

$$S^* = \arg\max_S \sum_{i=1,2,\ldots,5} w_i T_i \tag{26}$$

where, $w_1, \ldots, w_5$ are weights assigned to the particular degrees of validity, with values from the unit interval, the higher, the more important such that $\sum_{i=1,2,\ldots,5} w_i = 1$.

The definition of weights, $w_1, \ldots, w_5$ is a problem in itself, and will not be dealt with in more detail. The weights can be predefined or elicited from the user. In the case study presented later on the weights are determined by using the well-known Saaty's AHP (analytical hierarchy process) approach that works well in the problem considered.

4. Fuzzy querying, linguistic summaries, and their protoforms

4.1. Linguistic summaries via fuzzy queries. Protoforms of linguistic summaries

In Kacprzyk and Zadrożny's [21,29] approach, *interactivity*, i.e. *user assistance*, is in the definition of summarizers (indication of attributes and their combinations). This proceeds via a user interface of a fuzzy querying add-on. Basically, the queries (referring to summarizers) allowed are:

- *simple* as, e.g., "salary is *high*"
- *compound* as, e.g., "salary is *low* AND age is *old*"
- *compound* (*with quantifier*), as, e.g., "*most* of {salary is *high*, age is *young*, ..., training is *well above average*}.

We will also use "natural" linguistic terms, i.e. (7 ± 2!) exemplified by: *very low, low, medium, high, very high*, and also "comprehensible" fuzzy linguistic quantifiers as: *most*, *almost all*, ..., etc.

In Kacprzyk and Zadrożny [18–20,28], a conventional DBMS is used, and a fuzzy querying tool is developed to allow for queries with fuzzy (linguistic) elements of the "simple", "compound" and "compound with quantifier" types. The main issues are: (1) how to extend the syntax and semantics of the query, and (2) how to provide an easy way of eliciting and manipulating those terms by the user.

FQUERY for Access is embedded in the native Access's environment as an add-on. All the code and data is put into a database file, a *library*, installed by the user. Definitions of attributes, fuzzy values, etc. are maintained in a dictionary (a set of regular tables), and a mechanism for putting them into the query-by-example (QBE) sheet (grid) is provided. Linguistic terms are represented

within a query as parameters, and a query transformation is performed to provide for their proper interpretation during the query execution.

It is obvious that fuzzy queries directly correspond to summarizers in linguistic summaries. Thus, the derivation of a linguistic summary may proceed in an interactive (user-assisted) way as follows:

- the user formulates a set of linguistic summaries of interest (relevance) using the fuzzy querying add-on described above,
- the system retrieves records from the database and calculates the validity of each summary adopted, and
- a most appropriate linguistic summary is chosen.

The use of fuzzy querying is very relevant because we can restate the summarization in the fuzzy querying context. First, (2) may be interpreted as:

Most records match query S (27)

where S replaces F in (2) since we refer here directly to the concept of a summarizer (of course, this should properly understood because S is in fact the whole condition, e.g., price = *high*, while S is just the fuzzy value, i.e. *high* in this condition; this should not lead to confusion).

Similarly, (3) may be interpreted as:

Most records meeting conditions B match query S (28)

Thus, (27) says something only about a subset of records taken into account by (26). In database terminology, B corresponds to a *filter* and (26) claims that *most* records passing through B match query S. Moreover, since the filter may be fuzzy, a record may pass through it to a degree from $[0, 1]$.

It may be noticed that the concept of a protoform in the sense of Zadeh [53] is highly relevant in this context. First of all, a protoform is defined as an abstract prototype, that is, in our context, for the query (summary) given by (26) and (27) as follows, respectively:

Most R's are S (29)

Most BR's are S (30)

where R means "records", B is a filter, and S is a query.

Evidently, as protoforms may form a hierarchy, we can define higher level (more abstract) protoforms, for instance replacing *most* by a general linguistic quantifier Q, we obtain, respectively:

QR's are S (31)

QBR's are S (32)

Basically, the more abstract forms correspond to cases in which we assume less about summaries sought. There are two limit cases, where we: (1) assume totally abstract protoform or (2) assume all elements of a protoform are given on the lowest level of abstraction as specific linguistic terms. In case 1 data summarization will be extremely time-consuming, but may produce interesting, unexpected view on data. In case 2 the user has to guess a good candidate formula for summarization but the evaluation is fairly simple being equivalent to the answering of a (fuzzy) query. Thus, the second case refers to the summarization known as *ad hoc queries.* This may be illustratively shown in Table 1 in which five basic types of linguistic summaries are shown, corresponding to protoforms of a more and more abstracted form. In Table 1, $S^{\text{structure}}$ denotes that attributes and their connection in a summary are known, while S^{value} denotes a summarizer sought.

Type 1 may be easily produced by a simple extension of fuzzy querying as in FQUERY. Basically, the user has to construct a query—candidate summary, and has to be determined what is the fraction of rows matching this query and what linguistic quantifier best denotes this fraction. A Type 2 summary is a straightforward extension of Type 1 by adding a fuzzy filter. Type 3 summaries require much more effort. Their primary goal is to determine typical (exceptional) values of an attribute. So, query S consists of only one simple condition built of the attribute whose typical (exceptional) value is sought, the "=" relational operator and a placeholder for the value sought. For example, using the following summary in a context of personal data: Q = *"most"* and S = "age=?" (here "?" denotes a placeholder mentioned above) we look for a typical value of age. A Type 4 summary may produce typical (exceptional) values for some, possibly fuzzy, subset of rows. From the computational point of view. Type 5 summaries represent the most general form considered here: fuzzy rules describing dependencies between specific values of particular attributes. Here the use of B is essential, while previously it was optional. The summaries of Types 1 and 3 have been implemented as an extension to Kacprzyk and Zadrożny's [23–26] FQUERY for Access. Two approaches to Type 5 summaries producing has been proposed. Firstly, a subset of such summaries may be produced exploiting similarities with *association*

Table 1
Classification of linguistic summaries

Type	Given	Sought	Remarks
1	S	Q	Simple summaries through ad-hoc queries
2	$S\ B$	Q	Conditional summaries through ad-hoc queries
3	$Q\ S^{\text{structure}}$	S^{value}	Simple value oriented summaries
4	$Q\ S^{\text{structure}}$ B	S^{value}	Conditional value oriented summaries
5	Nothing	$S\ B\ Q$	General fuzzy rules

rules concept and employing their efficient algorithms. Second, genetic algorithm may be employed to search the summaries' space. The results of the latter are briefly presented in the next section.

Notice that the protoforms are a powerful conceptual tool because we can formulate many different types of linguistic summaries in a uniform way, and devise a uniform and universal way to handle different linguistic summaries. This is a confirmation of Zadeh's frequent claims that protoforms are so powerful.

4.2. A fuzzy querying add-in as a means to implement linguistic summaries through fuzzy querying

The roots of the approach adopted are our previous papers on the use of fuzzy logic in querying databases (cf. Kacprzyk and Ziółkówski [35], Kacprzyk et al. [34]) in which we argued that the formulation of a precise query is often difficult for the end user (see also Kacprzyk et al. [13]). For example, a customer of a real-estate agency looking for a house would rather express his or her criteria using imprecise descriptions as *cheap*, *large* garden, etc. Also, to specify which combination of the criteria fulfillment would be satisfactory, he or she would often use, e.g., *most* of them or *almost all*. All such vague terms may be relatively easily interpreted using fuzzy logic. This has motivated the development of the whole family of fuzzy querying interfaces, notably our FQUERY for Access package described above.

The summaries of Types 1 and 3 mentioned in the previous section have been implemented as an extension to our FQUERY for Access.

FQUERY for Access is an add-in that makes it possible to use fuzzy terms in queries. Briefly speaking, the following types of fuzzy terms are available:

- fuzzy values, exemplified by *low* in "profitability is *low*",
- fuzzy relations, exemplified by *much greater than* in "income is *much greater than* spending", and
- linguistic quantifiers, exemplified by *most* in "*most* conditions have to be met".

The elements of the first two types are elementary building blocks of fuzzy queries in FQUERY for Access. They are meaningful in the context of numerical fields only. There are also other fuzzy constructs allowed which may be used with scalar fields.

If a field is to be used in a query in connection with a fuzzy value, it has to be defined as an *attribute*. The definition of an attribute consists of two numbers: the attribute's values lower (LL) and upper (UL) limit. They set the interval which the field's values are assumed to belong to, according to the user. This interval depends on the meaning of the given field. For example, for *age* (of

a person), the reasonable interval would be, e.g., [18], in a particular context, i.e. for a specific group. Such a concept of an attribute makes it possible to universally define fuzzy values.

Fuzzy values are defined as fuzzy sets on $[-10,+10]$. Then, *the matching degree* $md(\cdot\,,\cdot)$ of a simple condition referring to attribute AT and fuzzy value FV against a record R is calculated by

$$md(AT = FV, R) = \mu_{FV}(\tau(R(AT)))$$

where $R(AT)$ is the value of attribute AT in record R, μ_{FV} is the membership function of fuzzy value FV, $\tau{:}[LL_{AT}, UL_{AT}] \rightarrow [-10,10]$ is the mapping from the interval defining AT onto $[-10,10]$ so that we may use the same fuzzy values for different fields. A meaningful interpretation is secured by τ which makes it possible to treat all fields domains as ranging over the unified interval $[-10,10]$.

For simplicity, it is assumed that the membership functions of fuzzy values are trapezoidal as in Fig. 1 and τ is assumed linear.

Linguistic quantifiers provide for a flexible aggregation of simple conditions. In FQUERY for Access the fuzzy linguistic quantifiers are defined in Zadeh's [51] sense, as fuzzy set on $[0,10]$ interval instead of the original $[0,1]$. They may be interpreted either using original Zadeh's [51] approach or via the OWA operators (cf. Yager [45] or Yager and Kacprzyk [49]); Zadeh's interpretation will be used here. The membership functions of fuzzy linguistic quantifiers are assumed piece-wise linear, hence two numbers from $[0,10]$ are needed. Again, a mapping from $[0,N]$, where N is the number of conditions aggregated, to $[0,10]$ is employed to calculate the matching degree of a query. More precisely, the matching degree, $md(\cdot\,,\cdot)$, for the *query* "Q of N conditions are satisfied" for record R is equal to $md(Q, \text{condition}_i, R) = \mu_Q\,[\tau(\sum_i md(\text{condition}_i,\ R))]$.

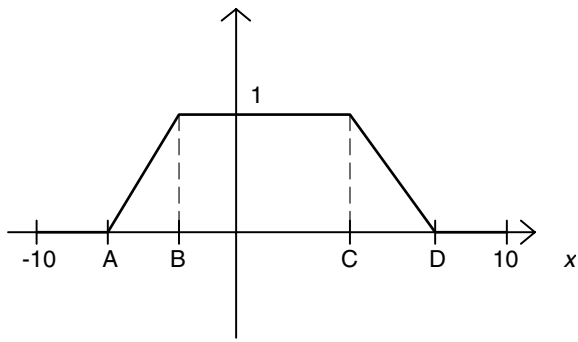

Fig. 1. An example of the membership function of a fuzzy value.

We can also assign different importance degrees for particular conditions. Then, the aggregation formula is equivalent to (3). The importance is identified with a fuzzy set on [0, 1], and then treated as property B in (3).

FQUERY for Access has been designed so that queries containing fuzzy terms are still syntactically correct Access's queries. It has been attained through the use of parameters. Basically, Access represents the queries using SQL. Parameters, expressed as strings limited with brackets, make it possible to embed references to fuzzy terms in a query. We have assumed special naming convention for parameters corresponding to particular fuzzy terms. For example, a parameter like:

[*FfA_FV fuzzy value name*] will be interpreted as a fuzzy value
[*FfA_F Q fuzzy quantifier name*] will be interpreted as a fuzzy quantifier

Before a fuzzy term may be used in a query, it has to be defined using the toolbar provided by FQUERY for Access and stored internally. This feature, i.e. maintenance of dictionaries of fuzzy terms defined by users, strongly supports our approach to data summarization to be discussed next. In fact, the package comes with a set of predefined fuzzy terms but the user may enrich the dictionary too.

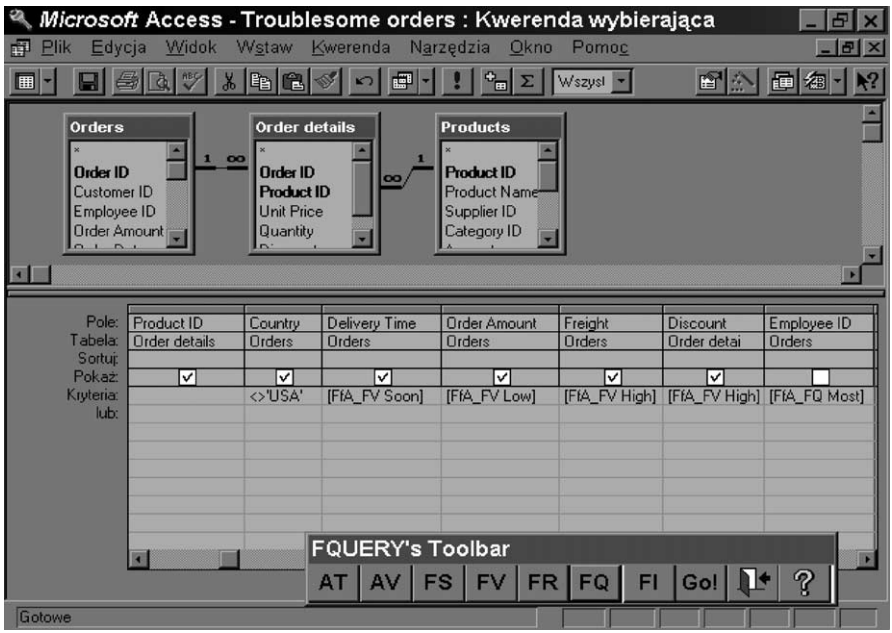

Fig. 2. Composition of a fuzzy query.

When the user initiates the execution of a query it is automatically transformed by appropriate FQUERY for Access's routines and then run as a native query of Access. The transformation consists primarily in the replacement of parameters referring to fuzzy terms by calls to functions implemented by the package which secure a proper interpretation of these fuzzy terms. Then, the query is run by Access as usually.

FQUERY for Access provides its own toolbar. There is one button for each fuzzy element, and the buttons for declaring attributes, starting the querying, closing the toolbar and for help (cf. Fig. 2).

Details can be found in Kacprzyk and Zadrożny [18–20,28] and Zadrożny and Kacprzyk [56].

5. Implementation

As a simple illustration of Type 5 summaries, an implementation is shown for a sales database of a medium size computer retailer in Southern Poland. For illustration we will only show some examples of linguistic summaries for some interesting (for the user!) choices of relations between attributes.

First, discovered interesting relations between the commission and the type of goods sold are shown in Table 2. As we can see, the results can be very helpful in, e.g., negotiating commissions for various products sold.

Next, the relations between the groups of products and times of sale are shown in Table 3. Notice that in this case the summaries are much less obvious than in the former case expressing relations between the group of product and commission. It should also be noted that the weighted average is here very low but this, by technical reasons, should not be taken literally as these values are mostly used to order the summaries obtained.

Finally, let us show in Table 4 some of the obtained linguistic summaries expressing relations between the attributes: size of customer, regularity of customer (purchasing frequency), date of sale, time of sale, commission, group of

Table 2
Linguistic summaries expressing relations between the group of products and commission

Summary
About 1/2 of sales of network elements is with a high commission
About 1/2 of sales of computers is with a medium commission
Much sales of accessories is with a high commission
Much sales of components is with a low commission
About 1/2 of sales of software is with a low commission
About 1/2 of sales of computers is with a low commission
A few sales of components is without commission
A few sales of computers is with a high commission
Very few sales of printers is with a high commission

Table 3
Linguistic summaries expressing relations between the groups of products and times of sale

Summary
About 1/3 of sales of computers is by the end of year
About 1/2 of sales in autumn is of accessories
About 1/3 of sales of network elements is in the beginning of year
Very few sales of network elements is by the end of year
Very few sales of software is in the beginning of year
About 1/2 of sales in the beginning of year is of accessories
About 1/3 of sales in the summer is of accessories
About 1/3 of sales of peripherals is in the spring period
About 1/3 of sales of software is by the end of year
About 1/3 of sales of network elements is in the spring period
About 1/3 of sales in the summer period is of components
Very few sales of network elements is in the autumn period
A few sales of software is in the summer period

Table 4
Linguistic summaries expressing relations between the attributes: size of customer, regularity of customer (purchasing frequency), date of sale, time of sale, commission, group of product and day of sale

Summary
Much sales on Saturday is about noon with a low commission
Much sales on Saturday is about noon for bigger customers
Much sales on Saturday is about noon
Much sales on Saturday is about noon for regular customers
A few sales for regular customers is with a low commission
A few sales for small customers is with a low commission
A few sales for one-time customers is with a low commission
Much sales for small customers is for non-regular customers

product and day of sale. This is an example of the most sophisticated form of linguistic summaries accommodated in the system described.

Notice that we do not present just one linguistic summary but some set of them: not too many and not too few, and leave their interpretation, usefulness, etc. to the user. This is—in our opinion—very important as it may guarantee user's autonomy that is so relevant for all tools that are meant for decision support. For clarity, we will not give the degrees of validity of the particular summaries: these are simply the best summaries obtained.

These are the most valid summaries, and they give the user much insight into relations between the attributes chosen, moreover they are simple and human consistent.

Notice that these summaries concern data from the company's known database. However, companies operate in an environment (economic, climatic, social, etc.), and aspects of this environment may be relevant because they may

Table 5
Linguistic summaries expressing relations between group of products, time of sale, temperature, precipitation, and type of customers

Summary
Very few sales of software is in hot days to individual customers
About 1/2 of sales of accessories is in rainy days on weekends by the end of the year
About 1/3 of sales of computers is in rainy days to individual customers

greatly influence the operation, economic results, etc. of a particular company. A notable example may here be the case of climatic data that can be fetched from some sources, for instance from paid or free climatic data services. The inclusion of such data may be implemented but its description is beyond the scope of this paper. We can just mention that one can obtain for instance the linguistic summaries as in Table 5 in the case when we are interested in relations between group of products, time of sale, temperature, precipitation, and type of customers.

It is easy to see that the contents of all the linguistic summaries obtained does give much insight to the user (analyst) in what is happening in the company and its operation, and can be very useful. Moreover, the use of a protoform of linguistic summaries provide a much needed universality and can greatly simplify the conceptual and algorithmic design, and hence the implementation.

6. Concluding remarks

We considered linguistic data(base) summaries in the sense of Yager. First, we discussed an extension of the classic Yager's approach to involve more sophisticated criteria of goodness, search methods, etc. We advocate the use of the concept of a protoform, that is recently vividly advocated by Zadeh, as a general form of a linguistic data summary. We present an extension of our interactive approach, based on fuzzy logic and fuzzy database queries, which makes it possible to implement such linguistic data summaries. We show how fuzzy queries are related to linguistic summaries, and show that one can introduce a hierarchy of prototype forms (protoforms), or abstract summaries in the sense of latest Zadeh's ideas meant mainly for increasing deduction capabilities of search engines. For illustration we show an implementation for a sales database in a computer retailer, employing some type of a protoform of a linguistic summary.

References

[4] P. Bosc, D. Dubois, O. Pivert, H. Prade, M. de Calmes, Fuzzy summarization of data using fuzzy cardinalities, in: Proceedings of IPMU'2002, Annecy, France, 2002, pp. 1553–1559.

[5] G. Chen, D. Liu, J. Li, Influence and conditional influence—new interestingness measures in association rule mining, in: Proceedings of FUZZ-IEEE'2001, Vancouver, Canada, 2001, pp. 1440–1443.
[6] G. Chen, Q. Wei, Fuzzy association rules and the extended mining algorithm, Information Sciences 147 (2002) 201–228.
[7] G. Chen, Q. Wei, E.E. Kerre, Fuzzy data mining: discovery of fuzzy generalized association rules, in: G. Bordogna, G. Pasi (Eds.), Recent Research Issues on Fuzzy Databases, Springer-Verlag, Heidelberg and New York, 2000, pp. 45–66.
[8] D. Dubois, H. Prade, Gradual rules in approximate reasoning, Information Sciences 61 (1992) 103–122.
[9] R. George, R. Srikanth, Data summarization using genetic algorithms and fuzzy logic, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Heidelberg, 1996, pp. 599–611.
[11] Y.-Ch. Hu, R.-Sh. Chen, G.-H. Tzeng, Mining fuzzy association rules for classification problems, Computers and Industrial Engineering 43 (2002) 735–750.
[12] J. Kacprzyk, Intelligent data analysis via linguistic data summaries: a fuzzy logic approach, in: R. Decker, W. Gaul (Eds.), Classification and Information Processing at the Turn of the Millennium, Springer-Verlag, Berlin, Heidelberg and New York, 2000, pp. 153–161.
[13] J. Kacprzyk, G. Pasi, P. Vojtaš, S. Zadrożny, Fuzzy querying: issues and perspective, Kybernetika 36 (2000) 605–616.
[14] J. Kacprzyk, P. Strykowski, Linguistic summaries of sales data at a computer retailer: a case study, in: Proceedings of IFSA'99, vol. 1, Taipei, Taiwan, 1999, pp. 29–33.
[15] J. Kacprzyk, R.R. Yager, Linguistic summaries of data using fuzzy logic, International Journal of General Systems 30 (2001) 133–154.
[16] J. Kacprzyk, R.R. Yager, S. Zadrożny, A fuzzy logic based approach to linguistic summaries of databases, International Journal of Applied Mathematics and Computer Science 10 (2000) 813–834.
[17] J. Kacprzyk, R.R. Yager, S. Zadrożny, Fuzzy linguistic summaries of databases for an efficient business data analysis and decision support, in: W. Abramowicz, J. Zurada (Eds.), Knowledge Discovery for Business Information Systems, Kluwer, Boston, 2001, pp. 129–152.
[18] J. Kacprzyk, S. Zadrożny, Fuzzy querying for Microsoft Access, in: Proceedings of FUZZ-IEEE'94, vol. 1, Orlando, USA, 1994, pp. 167–171.
[19] J. Kacprzyk, S. Zadrożny, Fuzzy queries in Microsoft Access vol. 2, in: Proceedings of FUZZ-IEEE/IFES '95, Yokohama, Japan, Workshop on Fuzzy Database Systems and Information Retrieval, 1995, pp. 61–66.
[20] J. Kacprzyk, S. Zadrożny, FQUERY for Access: fuzzy querying for a Windows-based DBMS, in: P. Bosc, J. Kacprzyk (Eds.), Fuzziness in Database Management Systems, Physica-Verlag, Heidelberg, 1995, pp. 415–433.
[21] J. Kacprzyk, S. Zadrożny, Data mining via linguistic summaries of data: an interactive approach, in: T. Yamakawa, G. Matsumoto (Eds.), Methodologies for the conception, design and application of soft computing—Proceedings of IIZUKA'98, Iizuka, Japan, 1998, pp. 668–671.
[23] J. Kacprzyk, S. Zadrożny, On combining intelligent querying and data mining using fuzzy logic concepts, in: G. Bordogna, G. Pasi (Eds.), Recent Research Issues on the Management of Fuzziness in Databases, Physica-Verlag, Heidelberg and New York, 2000, pp. 67–81.
[24] J. Kacprzyk, S. Zadrożny, Data mining via fuzzy querying over the Internet, in: O. Pons, M.A. Vila, J. Kacprzyk (Eds.), Knowledge Management in Fuzzy Databases, Physica-Verlag, Heidelberg and New York, 2000, pp. 211–233.
[25] J. Kacprzyk, S. Zadrożny, On a fuzzy querying and data mining interface, Kybernetika 36 (2000) 657–670.

[26] J. Kacprzyk, S. Zadrożny, Computing with words: towards a new generation of linguistic querying and summarization of databases, in: P. Sinčak, J. Vaščak (Eds.), Quo Vadis Computational Intelligence? Physica-Verlag, Heidelberg and New York, 2000, pp. 144–175.
[27] J. Kacprzyk, S. Zadrożny, On linguistic approaches in flexible querying and mining of association rules, in: H.L. Larsen, J. Kacprzyk, S. Zadrożny, T. Andreasen, H. Christiansen (Eds.), Flexible Query Answering Systems. Recent Advances, Springer-Verlag, Heidelberg and New York, 2001, pp. 475–484.
[28] J. Kacprzyk, S. Zadrożny, Computing with words in intelligent database querying: standalone and Internet-based applications, Information Sciences 34 (2001) 71–109.
[29] J. Kacprzyk, S. Zadrożny, Data mining via linguistic summaries of databases: an interactive approach, in: L. Ding (Ed.), A New Paradigm of Knowledge Engineering by Soft Computing, World Scientific, Singapore, 2001, pp. 325–345.
[30] J. Kacprzyk, S. Zadrożny, Fuzzy linguistic summaries via association rules, in: A. Kandel, M. Last, H. Bunke (Eds.), Data Mining and Computational Intelligence, Physica-Verlag (Springer-Verlag), Heidelberg and New York, 2001, pp. 115–139.
[31] J. Kacprzyk, S. Zadrożny, Using fuzzy querying over the Internet to browse through information resources, in: B. Reusch, K.-H. Temme (Eds.), Computational Intelligence in Theory and Practice, Physica-Verlag (Springer-Verlag), Heidelberg and New York, 2001, pp. 235–262.
[32] J. Kacprzyk, S. Zadrożny, Protoforms of linguistic data summaries: towards more general natural-language-based data mining tools, in: A. Abraham, J. Ruiz del Solar, M. Koeppen (Eds.), Soft Computing Systems, IOS Press, Amsterdam, 2002, pp. 417–425.
[33] J. Kacprzyk, S. Zadrożny, Linguistic summarization of data sets using association rules, in: Proceedings of FUZZ-IEEE'03, St. Louis, USA, 2003, pp. 702–707.
[34] J. Kacprzyk, S. Zadrożny, A. Ziółkowski, FQUERY III+: a 'human consistent' database querying system based on fuzzy logic with linguistic quantifiers, Information Systems 6 (1989) 443–453.
[35] J. Kacprzyk, A. Ziółkowski, Database queries with fuzzy linguistic quantifiers, IEEE Transactions on Systems, Man and Cybernetics, SMC 16 (1986) 474–479.
[36] J.-H. Lee, H. Lee-Kwang, An extension of association rules using fuzzy sets, in: Proceedings of Seventh IFSA World Congress, Prague, Czech Republic, 1997. pp. 399–402.
[37] Y. Liu, E.E. Kerre, An overview of fuzzy quantifiers. (I) Interpretations, Fuzzy Sets and Systems 95 (1998) 1–21.
[39] G. Raschia, N. Mouaddib, SAINTETIQ: a fuzzy set-based approach to database summarization, Fuzzy Sets and Systems 129 (2002) 137–162.
[40] D. Rasmussen, R.R. Yager, Using summary SQL as a tool for finding fuzzy and gradual functional dependencies, in: Proceedings of IPMU'96, Granada, Spain, 1996, pp. 275–280.
[41] D. Rasmussen, R.R. Yager, A fuzzy SQL summary language for data discovery, in: D. Dubois, H. Prade, R.R. Yager (Eds.), Fuzzy Information Engineering: a Guided Tour of Applications, Wiley, New York, 1997, pp. 253–264.
[42] D. Rasmussen, R.R. Yager, SummarySQL—A fuzzy tool for data mining, Intelligent Data Analysis—An International Journal 1 (Electronic Publication). Available from: <http//:www.east.elsevier.com/ida/browse/96-6/ida96-6.htm> 1997.
[43] D. Rasmussen, R.R. Yager, Finding fuzzy and gradual functional dependencies with SummarySQL, Fuzzy Sets and Systems 106 (1999) 131–142.
[44] R.R. Yager, A new approach to the summarization of data, Information Sciences 28 (1982) 69–86.
[45] R.R. Yager, On ordered weighted averaging operators in multicriteria decision making, IEEE Transactions on Systems, Man and Cybern, SMC 18 (1988) 183–190.
[46] R.R. Yager, On linguistic summaries of data, in: G. Piatetsky-Shapiro, B. Frawley (Eds.), Knowledge Discovery in Databases, MIT Press, Cambridge, USA, 1991, pp. 347–363.

[47] R.R. Yager, Linguistic summaries as a tool for database discovery, in: Proceedings of FUZZ-IEEE'95/IFES'95, Workshop on Fuzzy Database Systems and Information Retrieval, Yokohama, Japan, 1995, pp. 79–82.
[48] R.R. Yager, Database discovery using fuzzy sets, International Journal of Intelligent Systems 11 (1996) 691–712.
[49] R.R. Yager, J. Kacprzyk, The Ordered Weighted Averaging Operators: Theory and Applications, Kluwer, Boston, 1997.
[51] L.A. Zadeh, A computational approach to fuzzy quantifiers in natural languages, Computers and Maths with Applications 9 (1983) 149–184.
[52] L.A. Zadeh, Syllogistic reasoning in fuzzy logic and its application to usuality and reasoning with dispositions, IEEE Transaction on Systems, Man and Cybernetics, SMC 15 (1985) 754–763.
[53] L.A. Zadeh, A prototype-centered approach to adding deduction capabilities to search engines—the concept of a protoform. BISC Seminar, 2002, University of California, Berkeley, 2002.
[55] L.A. Zadeh, J. Kacprzyk (Eds.), Computing with Words in Information/intelligent Systems, 1. Foundations, 2. Applications, Physica-Verlag, Heidelberg and New York, 1999.
[56] S. Zadrożny, J. Kacprzyk, Fuzzy querying using the 'query-by-example' option in a Windows-based DBMS, in: Proceedings of Third European Congress on Intelligent Techniques and Soft Computing—EUFIT'95, vol. 2, Aachen, Germany, 1995, pp. 733–736.

Further reading

[1] T.M. Anwar, H.W. Beck, S.B. Navathe, Knowledge mining by imprecise querying: a classification based system, in: Proceedings of international conference on data engineering, Tampa, USA, pp. 622–630.
[2] G. Bordogna, G. Pasi, Modeling linguistic qualifiers of uncertainty in a fuzzy database, International Journal of Intelligent Systems 15 (2000) 995–1014.
[3] P. Bosc, D. Dubois, H. Prade, Fuzzy functional dependencies—an overview and a critical discussion, in: Proceedings of 3rd IEEE international conference on fuzzy systems—FUZZ-IEEE'94, Orlando, USA, pp. 325–330.
[10] R. George, R. Srikanth, A soft computing approach to intensional answering in databases, Information Sciences 92 (1996) 313–328.
[22] J. Kacprzyk, S. Zadrożny, The paradigm of computing with words in intelligent database querying, in: L.A. Zadeh, J. Kacprzyk (Eds.), Computing with Words in Information/intelligent Systems, Part 2 Foundations, Springer–Verlag, Heidelberg and New York, 1999, pp. 382–398.
[38] O. Pons, M.A. Vila, J. Kacprzyk (Eds.), Knowledge Management in Fuzzy Databases, Physica-Verlag, Heidelberg and New York, 2000.
[50] R.R. Yager, J. Kacprzyk, Linguistic data summaries: a perspective, in: Proceedings of IFSA'99 Congress, vol. 1, Taipei, Taiwan, ROC, 1999, pp. 44–48.
[54] L.A. Zadeh, J. Kacprzyk (Eds.), Fuzzy Logic for the Management of Uncertainty, Wiley, New York, 1992.
[57] S. Zadrożny, J. Kacprzyk, On database summarization using a fuzzy querying interface, in: Proceedings of IFSA'99 World Congress, vol. 1, Taipei, Taiwan ROC, 1999, pp. 39–43.