16

Fuzzy Linguistic Data Summaries as a Human Consistent, User Adaptable Solution to Data Mining

J. Kacprzyk[1,2] and S. Zadrożny[1]

[1] Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01–447 Warsaw, Poland
[2] Warsaw School of Information Technology (WSISiZ)
ul. Newelska 6, 01–447 Warsaw, Poland
<kacprzyk,zadrozny>@ibspan.waw.pl

In this chapter fuzzy linguistic summaries of data (databases) in the sense of Yager (cf. Yager [1], Kacprzyk and Yager [3], and Kacprzyk, Yager and Zadrożny [4]) are presented as a flexible, user adaptable solution to data mining problem. The essence of this approach is that, for instance, if we have a (large) database on employees, then in case that we are interested in, say, age and qualifications, then the contents of the database in this respect may be summarized by, say, "most young employees are well qualified". We present the problem of deriving such linguistic summaries in the context of Zadeh's (cf. Zadeh and Kacprzyk [6]) computing with words and perceptions paradigm, and consider his recent idea of a protoform (cf. Zadeh [7]) that provides means to define and handle more general forms of summaries. We illustrate the approach on an a system developed for a small to medium computer retailer, and show how data from the Internet can qualitatively enhance the results obtained. We show that the approach presented may be viewed as an example of an inexpensive, human consistent, human friendly technology that is easily adaptable to changing interests and needs of users.

16.1 Introduction

In this chapter we address the problem that may be exemplified as follows. There is a small (or a small-to-medium, SME for short) company that – as all other companies and organizations – faces the problem of dealing with too large sets of data that are not comprehensible by the human user. They know that they need some data mining but they are fully aware of their limitations. Mainly, in comparison with larger and richer companies and organization, they need a simple and possibly inexpensive solution that is also as much human consistent as possible. They are aware that most of their employees are

J. Kacprzyk and S. Zadrożny: *Fuzzy Linguistic Data Summaries as a Human Consistent, User Adaptable Solution to Data Mining*, StudFuzz **173**, 321–340 (2005)
www.springerlink.com

not qualified computer specialists, as they cannot afford to hire such people, and hence solutions adopted should be possibly human consistent and intuitive, basically as heavily as possible based upon the use of natural language. Such solutions have to offer at least a basic adaptability with respect to the interpretation of linguistic terms that are used to express data values and relations between data. Another dimension of the adaptability may be considered from the perspective of data sources taken into account. The primary data source for such data mining tasks is, of course, a database of the user. However, in order to discover some interesting phenomena in data it may be worthwhile to acquire some other data as well as no company operates in a vacuum, separated from the outside world. The Internet seems to be such a source of choice. Nowadays, it may be still difficult to get interesting, relevant data from the Internet without a careful planning and execution. However, as soon as promises of the Semantic Web become the reality, it should be fairly easy to arrange for automatic acquisition of data that is relevant for our problem but does not have to be identified in advance. In many cases such data may be easily integrated with our own data and provide the user with interesting results. For example, coupling the data on sales per day with weather information related to a given time period (that is not usually stored in sales databases) may show some dependencies important for running the business.

Generally, data summarization is still an unsolved 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 the fact that the use of natural language is limited. 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 chapter we discuss an approach to solve this problem. It is based on the concept of a *linguistic data (base) summary* and has been originally proposed by Yager [1, 2] and further developed by many authors (see, for instance, Kacprzyk and Yager [3], and Kacprzyk, Yager and Zadrożny [4]). The essence of such linguistic data summaries is that a set of data, say, concerning employees, with (numeric) data on their age, sex, salaries, seniority, etc., can be summarized linguistically with respect to a selected attribute or attributes, say age and salaries, by linguistically quantified propositions, say "almost all employees are well qualified", "most young employees are well paid", etc. Notice that such simple, extremely human consistent and intuitive statements do summarize in a concise yet very informative form what we may be interested in.

We present the essence of Yager's [1, 2] approach to such summaries, with its further extensions (cf. Kacprzyk and Yager [3], Kacprzyk, Yager and Zadrożny [4, 5]) from the perspective of Zadeh's computing with words and perception paradigm (cf. Zadeh and Kacprzyk [6]) that can provide a general theoretical framework which is implementable, as shown in works mentioned above. In particular, we indicate the use of Zadeh's concept of a protoform of a fuzzy linguistic summary (cf. Zadeh [7], Kacprzyk and Zadrożny [8]) that can provide a "portability" and "scalability" as meant above, and also some "adaptivity" to different situations and needs by providing universal means for representing quite general forms of summaries.

As an example we will show an implementation of the data summarization system proposed for the derivation of linguistic data summaries in a sales database of a computer retailer.

The basic philosophy of the approach and its algorithmic engine makes use of the computing with words and perception paradigm introduced by Zadeh in the mid-1990s, and best and most comprehensively presented in Zadeh and Kacprzyk's [6] books. It may be viewed as a new paradigm, or "technology" in the representation, processing and solving of various real life problems when a human being is a crucial element. Such problems are omnipresent. The basic idea and rationale of computing with words and perceptions is that since for a human being natural language is the only fully natural way of communication, then maybe it could be expedient to try to "directly" use (elements of) natural language in the formulation, processing and solution of problems considered to maintain a higher human consistence, hence a higher implementability. Notice that the philosophy and justification of the computing with words and perception paradigm are in line with the requirements and specifics of problems considered, and solution concepts adopted in this paper.

A prerequisite for computing with words is to have some way to formally represent elements of natural language used. Zadeh proposed to use here the PNL (precisiated natural language). Basically, in PNL, statements about values, relations, etc. between variables are represented by constraints. In the conventional case, a statement is, e.g., that the value of variable x belongs to a set X. In PNL, statements – generally written as "x is Z" – may be different, and correspond to numeric values, intervals, possibility distributions, verity distributions, probability distributions, usuality qualified statements, rough sets representations, fuzzy relations, etc. For our purposes, the usuality qualified representation will be of a special relevance. Basically, it says "x is usually Z" that is meant as "in most cases, x is Z". PNL may play various roles among which crucial are: the description of perceptions, the definition of sophisticated concepts, a language for perception based reasoning, etc.

Recently, Zadeh [7] introduced the concept of a protoform. For our purposes, one should notice that most perceptions are summaries. For instance, a perception like "most Swedes are tall" is some sort of a summary. It can be represented in Zadeh's notation as "most As are Bs". This can be employed for reasoning under various assumptions. For instance, if we know that "x is A", we can deduce that, e.g."it is likely that x is B". We can also ask about an average height of a Swede, etc. One can go a step further, and define a protoform as an abstracted summary. In our case, this would be "QAs are Bs". Notice that we now have a more general, deinstantiated form of our point of departure "most Swedes are tall", and also of "most As are Bs". Needless to say that much of human reasoning is protoform based, and the availability of such a more general representation is vary valuable, and provides tools that can be used in many cases. From the point of view of the problem class considered in this chapter, the use of protoforms may be viewed to contribute to the portability, scalability and adaptivity in the sense mentioned above.

We discuss a number of approaches to mining of linguistic summaries. First, those based on Kacprzyk and Zadrożny's [9, 10] 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 [11, 12] FQUERY for Access, a fuzzy querying add-on to Microsoft Access©. It is shown that by relating a range of types of linguistic summaries to fuzzy queries, with various known and sought elements, we can arrive at a hierarchy of protoforms of linguistic data summaries. Basically, there is a trade off between the specificity in respect to the summaries sought and the complexity of a corresponding mining process. In the simplest case, data mining boils down directly to a flexible querying process. In the opposite case, the concept of a linguistic association rule along with well known efficient mining algorithms may be employed. Also other approaches to linguistic summaries mining are briefly discussed in Sect. 16.3.

The line of reasoning adopted here should convince the reader that the use of a broadly perceived paradigm of computing with words and perceptions, equipped with a newly introduced concept of a protoform, may be a proper tool for dealing with situations when we have to develop and implement a system that should perform "intelligent" tasks, be human consistent and human friendly, and some other relevant requirements should also be fulfilled as, e.g., to be inexpensive, easy to calibrate, portable, scalable, being able to somehow adapt to changing conditions and requirements, etc.

16.2 Linguistic Data Summaries via Fuzzy Logic with Linguistic Quantifiers

The linguistic summary is meant as a natural language like sentence that subsumes the very essence (from a certain point of view) of a set of data. This set is assumed to be numeric and is usually large, not comprehensible in its original form by the human being. In Yager's approach (cf. Yager [1], Kacprzyk and Yager [3], and Kacprzyk, Yager and Zadrożny [4]) the following context for linguistic summaries mining is assumed:

- $Y = \{y_1, \ldots, y_n\}$ is a set of objects (records) in a database, e.g., the set of workers;
- $A = \{A_1, \ldots, A_m\}$ is a set of attributes characterizing objects from Y, e.g., salary, age, etc. in a database of workers, and $A_j(y_i)$ denotes a value of attribute A_j for object y_i.

A linguistic summary of data set D consists of:

- a summarizer S, i.e. an attribute together with a linguistic value (fuzzy predicate) defined on the domain of attribute A_j (e.g. "low salary" for attribute "salary");
- a quantity in agreement Q, i.e. a linguistic quantifier (e.g. most);
- truth (validity) T of the summary, i.e. a number from the interval $[0,1]$ assessing the truth (validity) of the summary (e.g. 0.7); usually, only summaries with a high value of T are interesting;
- optionally, a qualifier R, i.e. another attribute together with a linguistic value (fuzzy predicate) defined on the domain of attribute A_k determining a (fuzzy subset) of Y (e.g. "young" for attribute "age").

Thus, the linguistic summary may be exemplified by

$$T(\text{most of employees earn low salary}) = 0.7 \tag{16.1}$$

A richer form of the summary may include a qualifier as in, e.g.,

$$T(\text{most of young employees earn low salary}) = 0.7 \tag{16.2}$$

Thus, basically, the core of a linguistic summary is a *linguistically quantified proposition* in the sense of Zadeh [13]. A linguistically quantified proposition, corresponding to (16.1) may be written as

$$Qy\text{'s are } S \tag{16.3}$$

and the one corresponding to (16.2) may be written as

$$QRy\text{'s are } S \tag{16.4}$$

Then, the component of a linguistic summary, T, i.e., its truth (validity), directly corresponds to the truth value of (16.3) or (16.4). This may be calculated by using either original Zadeh's calculus of linguistically quantified statements (cf. [13]), or other interpretations of linguistic quantifiers (cf. Liu and Kerre [14]), including Yager's OWA operators [15] and Dubois et al. OWmin operators [16]. The component of a linguistic summary that is a quantifier Q can also be interpreted from a more general perspective of the concept of a *generalized quantifier*, cf. Hájek and Holeňa [17] or Glöckner [18].

Using Zadeh's [13] fuzzy logic based calculus of linguistically quantified propositions, a (proportional, nondecreasing) linguistic quantifier Q is assumed to be a fuzzy set in the interval $[0,1]$ as, e.g.

$$\mu_Q(x) = \begin{cases} 1 & \text{for } x \geq 0.8 \\ 2x - 0.6 & \text{for } 0.3 < x < 0.8 \\ 0 & \text{for } x \leq 0.3 \end{cases} \tag{16.5}$$

Then, the truth values (from $[0,1]$) of (16.3) and (16.4) are calculated, respectively, as

$$\text{truth}(Qy\text{'s are } S) = \mu_Q \left[\frac{1}{n} \sum_{i=1}^{n} \mu_S(y_i) \right] \tag{16.6}$$

$$\text{truth}(QRy\text{'s are } S) = \mu_Q \left[\frac{\sum_{i=1}^{n}(\mu_R(y_i) \wedge \mu_S(y_i))}{\sum_{i=1}^{n} \mu_R(y_i)} \right] \tag{16.7}$$

Both the fuzzy predicates S and R are assumed above to be of a rather simplified, atomic form referring to just one attribute. They can be extended to cover more sophisticated summaries involving some confluence of various attribute values as, e.g, "young and well paid". Clearly, when we try to linguistically summarize data, the most interesting are non-trivial, human-consistent summarizers (concepts) as, e.g.:

- productive workers,
- difficult orders, . . . ,

and it may easily be noticed that their proper definition may require a very complicated combination of attributes as with, for instance: a hierarchy (not all attributes are of the same importance for a concept in question), the attribute values are ANDed and/or ORed, k out of n, most, . . . of them should be accounted for, etc.

Recently, Zadeh [7] introduced the concept of a *protoform* that is highly relevant in this context. Basically, a protoform is defined as a more or less abstract prototype (template) of a linguistically quantified proposition. The most abstract protoforms correspond to (16.3) and (16.4), while (16.1) and (16.2) are examples of fully instantiated protoforms. Thus, evidently, protoforms form a hierarchy, where higher/lower levels correspond to more/less abstract protoforms. Going down this hierarchy one has to instantiate particular components of (16.3) and (16.4), i.e., quantifier Q and fuzzy predicates S and R. The instantiation of the former one consists in the selection of a quantifier. The instantiation of fuzzy predicates requires the choice of attributes together with linguistic values (atomic predicates) and a structure they form when combined using logical connectives. This leads to a theoretically infinite number of potential protoforms. However, for the purposes of mining of linguistic summaries, there are obviously some limits on a reasonable size of a set of summaries that should be taken into account. These results from a limited capability of the user in the interpretion of summaries as well as from the computational point of view.

The concept of a protoform may be taken as a guiding paradigm for the design of a user interface supporting the mining of linguistic summaries. It may be assumed that the user specifies a protoform of linguistic summaries sought. Basically, the more abstract protoform the less should be assumed about summaries sought, i.e., the wider range of summaries is expected by the user. There are two limit cases, where:

Table 16.1. Classification of protoforms/linguistic summaries

Type	Protoform	Given	Sought
0	QRy's are S	All	validity T
1	Qy's are S	S	Q
2	QRy's are S	S and R	Q
3	Qy's are S	Q and structure of S	linguistic values in S
4	QRy's are S	Q, R and structure of S	linguistic values in S
5	QRy's are S	Nothing	S, R and Q

- a totally abstract protoform is specified, i.e., (16.4)
- all elements of a protoform are specified on the lowest level of abstraction as specific linguistic terms.

In the first case the system has to construct all possible summaries (with all possible linguistic components and their combinations) for the context of a given database (table) and present to the user those verifying the validity to a degree higher than some threshold. In the second case, the whole summary is specified by the user and the system has only to verify its validity. Thus, the former case is usually more interesting from the point of view of the user but at the same time more complex from the computational point of view. There is a number of intermediate cases that may be more practical. In Table 16.1 basic types of protoforms/linguistic summaries are shown, corresponding to protoforms of a more and more abstract form.

Basically, each of fuzzy predicates S and R may be defined by listing its atomic fuzzy predicates (i.e., pairs of "attribute/linguistic value") and structure, i.e., how these atomic predicates are combined. In Table 16.1 S (or R) corresponds to the full description of both the atomic fuzzy predicates (referred to as linguistic values, for short) as well as the structure. For example:

$$Q \textit{ young} \text{ employees earn a } \textit{high salary} \tag{16.8}$$

is a protoform of Type 2, while:

$$\text{Most employees earn a } \textit{“?” salary} \tag{16.9}$$

is a protoform of Type 3.

In case of (16.8) the system has to select a linguistic quantifier (usually from a predefined dictionary) that when put in place of Q in (16.8) makes the resulting linguistically quantified proposition valid to the highest degree. In case of (16.9), the linguistic quantifier as well as the *structure* of summarizer S are given. The system has to choose a linguistic value to replace the question mark ("?") yielding a linguistically quantified proposition as valid as possible. Note that this may be interpreted as the search for a *typical* salary in the company.

Thus, the use of protoforms makes it possible to devise a uniform procedure to handle a wide class of linguistic data summaries so that the system can be easily adaptable to a variety of situations, users' interests and preferences, scales of the project, etc.

Usually, most interesting are linguistic summaries required by a summary of Type 5. They may be interpreted as fuzzy IF-THEN rules:

$$\text{IF } R(y) \text{ THEN } S(y) \tag{16.10}$$

that should be instantiated by a system yielding, e.g., a rule

$$\text{IF } y \text{ IS } \textit{young} \text{ THEN } y \text{ EARNS } \textit{low salary} \tag{16.11}$$

with a truth degree being a function of the two components of the summary that involve the truth (validity) T and the linguistic quantifier Q. In the literature (cf., e.g., Dubois and Prade [19]) there are considered many possible interpretations for fuzzy rules. Some of them were directly discussed in the context of linguistic summaries by some authors (cf. Sect. 16.3.3 in this chapter).

Some authors consider the concept of a *fuzzy functional dependency* as a suitable candidate for the linguistic summarization. The fuzzy functional dependencies are an extension of the classical crisp functional dependencies considered in the context of relational databases. The latter play a fundamental role in the theory of normalization. A functional dependency between two sets of attributes $\{A_i\}$ and $\{B_i\}$ holds when the values of attributes $\{A_i\}$ fully determine the values of attributes $\{B_i\}$. Thus, a functional dependency is a much stronger dependency between attributes than that expressed by (16.10). The classical crisp functional dependencies are useless for data summarization (at least in case of regular relational databases) as in a properly designed database they should not appear, except the trivial ones. On the other hand, fuzzy functional dependencies are of an approximate nature and as such may be identified in a database and serve as linguistic summaries. They may be referred to as extensional functional dependencies that may appear in a given instance of a database in contrast to intentionally interpreted crisp functional dependencies that are, by design, avoided in any instance of a database. A fuzzy functional dependency may be exemplified with

$$\text{AGE determines SALARY} \tag{16.12}$$

to be interpreted in such a way that "*usually* any two employees of a *similar* age have also *similar* salaries". Such a rule may be, as previously, associated with a certain linguistic quantifier (here: usually) and a truth qualification degree. Many authors discuss various definitions of fuzzy functional dependencies, cf., e.g., Bosc, Dubois and Prade [20].

16.3 Various Approaches to the Mining of Linguistic Summaries

The basic concept of a linguistic summary seems to be fairly simple. The main issue is how to generate summaries for a given database. The full search of the solution space is practically infeasible. In the literature a number of ways to solve this problem have been proposed. In what follows we briefly overview some of them.

The process of mining of linguistic summaries may be more or less automatic. At the one extreme, the system may be responsible for both the construction and verification of summaries (which corresponds to Type 5 protoforms/summaries given in Table 16.1). At the other extreme, the user proposes a summary and the system only verifies its validity (which corresponds to Type 0 protoforms/summaries in Table 16.1). The former approach seems to be more attractive and in the spirit of data mining meant as the discovery of interesting, unknown regularities in data. On the other hand, the latter approach, obviously secures a better interpretability of the results. Thus, we will discuss now the possibility to employ a flexible querying interface for the purposes of linguistic summarization of data, and indicate the implementability of a more automatic approach.

16.3.1 A Fuzzy Querying Add-on for Formulating Linguistic Summaries

Since we consider a problem that should be solved, and put to practice, we should find a proper way to implement the algorithmic base presented in the previous section. For this purpose we need first of all appropriate user interfaces since the tools involve many entities that should be elicited from the user, calibrated, illustratively displayed, etc.

In Kacprzyk and Zadrożny's [9, 10] approach, the interactivity, i.e. a 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. In Kacprzyk and Zadrożny [11, 12, 21], a conventional database management system is used and a fuzzy querying tool, FQUERY for Access, is developed to allow for queries with fuzzy (linguistic) elements. An important component of this tool is a *dictionary* of linguistic terms to be used in queries. They include fuzzy linguistic values and relations as well as fuzzy linguistic quantifiers. There is a set of built-in linguistic terms, but the user is free to add his or her own. Thus, such a dictionary evolves in a natural way over time as the user is interacting with the system. For example, an SQL query searching for *troublesome orders* may take the following WHERE clause (we make the syntax of a query to FQUERY for Access more self-descriptive in this example; examples of linguistic terms in italic):

WHERE *Most* of the conditions are met out of
PRICE*ORDERED-AMOUNT IS *Low*
DISCOUNT IS *High*
ORDERED-AMOUNT IS *Much Greater Than* ON-STOCK

It is obvious that the condition of such a fuzzy query directly corresponds to summarizer S in a linguistic summary. Moreover, the elements of a dictionary are perfect building blocks of such a summary. Thus, the derivation of a linguistic summary of type (16.3) 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,
- the system retrieves records from the database and calculates the validity of each summary adopted, and
- a most appropriate linguistic summary is chosen.

Referring to Table 16.1, we can observe that Type 0 as well as Type 1 linguistic summaries may be easily produced by a simple extension of FQUERY for Access. Basically, the user has to construct a query, a candidate summary, and it is to be determined which fraction of rows matches that query (and which linguistic quantifier best denotes this fraction, in case of Type 1). Type 3 summaries require much more effort as their primary goal is to determine typical (exceptional) values of an attribute (combination of attributes). So, query/summarizer 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: $Q =$ "most" and $S =$ "age=?" we look for a typical value of "age". From the computational point of view Type 5 summaries represent the most general form considered: fuzzy rules describing dependencies between specific values of particular attributes.

The summaries of Type 1 and 3 have been implemented as an extension to Kacprzyk and Zadrożny's [22, 23, 24] FQUERY for Access.

16.3.2 Linguistic Summaries and Fuzzy Association Rules

The discovery of general rules as given by (16.10) (i.e. of Type 5) is essentially a difficult task. As mentioned earlier, some additional assumptions about the structure of particular fuzzy predicates and/or quantifier have usually to be done. One set of such assumptions leads to the idea of using *fuzzy association rules* as linguistic summaries.

Originally, the association rules were defined for binary valued attributes in the following form (cf. Agraval and Srikant [25]):

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \longrightarrow A_{n+1} \qquad (16.13)$$

and note that much earlier origins of that concept are mentioned in the work by Hájek and Holeňa [17]).

Thus, such an association rule states that if in a database row all the attributes from the set $\{A_1, A_2, \ldots, A_n\}$ take on value 1, then also the attribute A_{n+1} is expected to take on value 1. The algorithms proposed in the literature for mining the association rules are based on the following concepts and definitions. A row in a database (table) is said to *support* a set of attributes $\{A_i\}_{i \in I}$ if all attributes from the set take on in this row value 1. The support of a rule (16.13) is the fraction of the number of rows supporting the set of attributes $\{A_i\}_{i \in \{1,\ldots,n+1\}}$ in a database (table). The *confidence* of a rule in a database (table) is the fraction of the number of rows supporting the set of attributes $\{A_i\}_{i \in \{1,\ldots,n+1\}}$ among all rows supporting the set of attributes $\{A_i\}_{i \in I}$. The well known algorithms (cf. Agrawal and Srikant [25] and Mannila et al. [26]) search for rules having values of the support measure above some minimal threshold and a high value of the confidence measure. Moreover, these algorithms may be easily adopted for the non-binary valued data and more sophisticated rules than one shown in (16.13).

In particular, *fuzzy association rules* may be considered:

$$A_1 \text{ IS } R_1 \wedge A_2 \text{ IS } R_2 \wedge \ldots \wedge A_n \text{ IS } R_n \longrightarrow A_{n+1} \text{ IS } S \qquad (16.14)$$

where R_i is a linguistic term defined in the domain of the attribute A_i, i.e. a qualifier fuzzy predicate in terms of linguistic summaries (cf. Sect. 16.2 of this chapter) and S is another linguistic term corresponding to the summarizer. The confidence of the rule may be interpreted in terms of linguistic quantifiers employed in the definition of a linguistic summary. Thus, a fuzzy association rule may be treated as a special case of a linguistic summary of type defined by (16.4). The structure of the fuzzy predicates R_i and S is to some extent fixed but due to that efficient algorithms for rule generation may be employed. These algorithms are easily adopted to fuzzy association rules. Usually, the first step is a preprocessing of original, crisp data. Values of all attributes considered are replaced with linguistic terms best matching them. Additionally, a degree of this matching may be optionally recorded and later taken into account. For example:

$$\text{AGE} = 45 \longrightarrow \text{AGE IS } \textit{medium} \text{ (matching degree 0.8)} \qquad (16.15)$$

Then, each combination of attribute and linguistic term may be considered as a Boolean attribute and original algorithms, such as a priori [25], may be applied. They, basically, boil down to an efficient counting of support for all conjunctions of Boolean attributes, i.e., so-called itemsets (in fact, the essence of these algorithms is to count support for as small a subset of itemsets as possible). In case of fuzzy association rules attributes may be treated strictly as Boolean attributes – they may appear or not in particular tuples – or interpreted in terms of fuzzy logic as in linguistic summaries. In the latter case they appear in a tuple to a degree, as in (16.15) and the support counting should take that into account. Basically, a scalar cardinality may be employed (in the spirit of Zadeh's calculus of linguistically quantified propositions). Finally, each *frequent* itemset (i.e., with the support higher than a selected threshold) is split (in all possible ways) into two parts treated as a conjunction of atomic predicates and corresponding to the premise (predicate R in terms of linguistic summaries) and consequence (predicate S in terms of linguistic summaries) of the rule, respectively. Such a rule is accepted if its confidence is higher than the selected threshold. Note that such an algorithm trivially covers the linguistic summaries of type (16.3), too. For them the last step is not necessary and each whole frequent itemset may be treated as a linguistic summary of this type.

Fuzzy association rules were studied by many authors including Lee and Lee-Kwang [27] and Au and Chan [28]. Hu et al. [29] simplify the form of fuzzy association rules sought by assuming a single specific attribute (class) in the consequent. This leads to the mining of fuzzy classification rules. Bosc et al. [30] argue against the use of scalar cardinalities in fuzzy association rule mining. Instead, they suggest to employ fuzzy cardinalities and propose an approach for the calculation of rules' frequencies. This is not a trivial problem as it requires to divide the fuzzy cardinalities of two fuzzy sets. Kacprzyk, Yager and Zadrożny [4, 22, 23, 24, 31] advocated the use of fuzzy association rules for mining linguistic summaries in the framework of flexible querying interface. Chen et al. [32] investigated the issue of generalized fuzzy rules where a fuzzy taxonomy of linguistic terms is taken into account. Kacprzyk and Zadrożny [33] proposed to use more flexible aggregation operators instead of conjunction, but still in context of fuzzy association rules.

16.3.3 Other Approaches

George and Srikanth [34, 35] use a genetic algorithm to mine linguistic summaries. Basically, they consider the summarizer in the form of a conjunction of atomic fuzzy predicates and a void subpopulation. Then, they search for two linguistic summaries referred to as a *constraint descriptor* ("most specific generalization") and a *constituent descriptor* ("most general specification"), respectively. The former is defined as a compromise solution having both the maximum truth (validity) and number of covered attributes (these criteria are combined by some aggregation operator). The latter is a linguistic summary having the maximum validity and covering all attributes. As in virtually all other approaches, a dictionary of linguistic quantifiers and linguistic values over domains of all attributes is assumed. This is sometimes referred to as a *domain* or *background knowledge*. Kacprzyk and Strykowski [36, 37] have also implemented the mining of linguistic summaries using genetic algorithms. In their approach, the fitting function is a combination of a wide array of indices assessing a validity/interestingness of given summary. These indices include, e.g., a degree of imprecision (fuzziness), a degree of covering, a degree of appropriateness, a length of a summary, and yields an overall degree of validity (cf.

also Kacprzyk and Yager [3]). Some examples of this approach are presented and discussed in Sect. 16.4 of this chapter.

Rasmussen and Yager [38, 39] propose an extension, SummarySQL, to the SQL language, an industry standard for querying relational databases, making it possible to cover linguistic summaries. Actually, they do not address the problem of mining linguistic summaries but merely of verifying them. The user has to conceive a summary, express it using SummarySQL, and then has it evaluated. In [39] it is shown how SummarySQL may also be used to verify a kind of fuzzy gradual rules (cf. Dubois and Prade [40]) and fuzzy functional dependencies. Again, the authors focus on a smooth integration of a formalism for such rule expression with SQL rather than on the efficiency of a verification procedure.

Raschia and Mouaddib [41] consider the problem of mining hierarchies of summaries. Their understanding of summaries is slightly different than that given by (16.4). Namely, their summary is a conjunction of atomic fuzzy predicates (each referring to just one attribute). However, these predicates are not defined by just one linguistic value but possibly by fuzzy sets of linguistic values (i.e., fuzzy sets of higher levels are considered). It is assumed that both linguistic values as well as fuzzy sets of higher levels based on them form *background knowledge* provided by experts/users. The mining of summaries (in fact what is mined is a whole hierarchy of summaries) is based on a concept formation (conceptual clustering) process. The first step is, as usually, a translation of the original tuples from database into so-called *candidate tuples*. This step consists in replacing in the original tuples values of their attributes with linguistic values best matching them which are defined over respective domains. Then, candidate tuples obtained are aggregated to form final summaries of various levels of hierarchy. This aggregation leads to possibly more complex linguistic values (represented by fuzzy sets of a higher level). More precisely, it is assumed that one candidate tuple is processed at a time. It is inserted into appropriate summaries already present in the hierarchy. Each tuple is first added to a top (root), most abstract summary, covering the whole database (table). Then, the tuple is put into offspring summaries along the selected branch in the hierarchy. In fact, a range of operations is considered that may lead to a rearrangement of the hierarchy via the formation of new node-summaries as well as splitting the old ones. The concept of a linguistic quantifier does not directly appear in this approach. However, each summary is accompanied with an index corresponding to the number of original tuples covered by this summary.

16.4 Examples of Linguistic Summaries and Possible Extensions

Finally, to show the essence and virtues of the solution proposed we will briefly present an implementation of a system for deriving linguistic database

Table 16.2. The basic structure of the database

Attribute Name	Attribute Type	Description
Date	Date	Date of sale
Time	Time	Time of sale transaction
Name	Test	Name of the product
Amount (number)	Numeric	Number of products sold in the transaction
Price	Numeric	Unit price
Commission	Numeric	Commission (in %) on sale
Value	Numeric	Value = amount (number) × price, of the product
Discount	Numeric	Discount (in %) for transaction
Group	Test	Product group to which the product belongs
Transaction value	Numeric	Value of the whole transaction
Total sale to customer	Numeric	Total value of sales to the customer in fiscal year
Purchasing frequency	Numeric	Number of purchases by customer in fiscal year
Town	Test	Town where the customer lives or is based

summaries for a computer retailer. Basically, we will deal with its sales database, and will only show some examples of linguistic summaries for some interesting (for the user!) choices of relations between attributes.

The basic structure of the database is as shown in Table 16.2.

Linguistic summaries are generated using a genetic algorithm [36, 37]. We will now give a couple of examples of resulting summaries. First, suppose that we are interested in a relation between the commission and the type of goods sold. The best linguistic summaries obtained are as shown in Table 16.3.

Table 16.3. Linguistic summaries expressing relations between the group of products and commission

Summary
About 1/3 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 1/3 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 16.4. 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

As we can see, the results can be very helpful, for instance while negotiating commissions for various products sold.

Next, suppose that we are interested in relations between the groups of products and times of sale. The best results obtained are as in Table 16.4.

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. But, again, they provide very useful information.

Finally, let us show in Table 16.5 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 product and day of sale.

Table 16.5. 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 nonregular customers

Notice that the linguistic summaries obtained do provide much of relevant and useful information, and can help the decision maker make decisions. It should be stressed that in the construction of the data mining paradigm presented we do not want to replace the decision maker but just to provide him or her with a help (support). This is clearly an example of the promising philosophy of decision support, i.e. to maintain user's autonomy and just to provide a support for decision making, and by no means to replace the user.

The system for deriving linguistic summaries developed and implemented for a computer retailer has been found useful by the user who has indicated its human friendliness, and ease of calibration and adaptation to new tasks (summaries involving new attributes of interest) and users (of a variable preparation, knowledge, flexibility, etc.). However, after some time of intensive use, the user has come to a conslusion (quite obvious!) that all summaries that could be derived by the system have been based on the own database of the company. Clearly, these data contain most relevant information on the functioning of the company. However, no company operates in a vacuum and some external data (e.g. on climate when the operation and/or results depend on climatic conditions, national and global economic indicators, etc.) can be of utmost importance and should be taken into account to derive more relevant summaries. Moreover, such external data do provide an easy and quick adaptation mechanism because they reflect what may be changing in the environment.

Following this rationale and philosophy, we have extended the class of linguistic summaries handled by the system to include those that take into account data easily (freely) available from Internet sources. These data are, on the one hand, most up to date so that their inclusion can be viewed as an obvious factor contributing to an efficient adaptation to most recent changes. A good example for the case cosnidered was the inclusion of data on wheather that has a cosniderable impact on the operation of the computer retailer. It is quite obvious that though such data are widely available because meteorological services are popular around the world, the Internet is the best source of such data. This is particularly true in the case of a small company that has limited funds for data, and also limited human resources to fetch such data. Data from the Internet may be therefore viewed as considerably contributing to an inexpensive technology that is so relevant for any small or medium company who has limited funds.

Using the data from meteorological commercial (inexpensive) and academic (free) services available through the Internet, we have been able to extend the system of linguistic database summarization described above.

For instance, if we are interested in relations between group of products, time of sale, temperature, precipitacion, and type of customers, the best linguistic summaries (of both our "internal" data from the sales database, and "external" meteorological data from an Internet service) are as shown in Table 16.6.

Table 16.6. Linguistic summaries expressing relations between the attributes: group of products, time of sale, temperature, precipitacion, and type of customers

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

Notice that the use of external data gives a new quality to possible linguistic summaries. It can be viewed as providing a greater adaptivity to varying conditions because the use of free or inexpensive data sources from the Internet makes it possible to easily and quickly adapt the form and contents of summaries to varying needs and interests. And this all is practically at no additional price and effort.

16.5 Concluding Remarks

In this chapter we have presented an interactive, fuzzy logic based approach to the linguistic summarization of databases, and have advocated it as a means to obtain human consistent summaries of (large) sets of data. Such "raw" sets of data are incomprehensible by the human being, while they linguistic summaries are easily comprehensible. Our intention was to show it as an example of a simple inexpensive information technology that can be implementable even in small companies, and is easily adaptable to varying needs of the users, their preferences, profiles, and proficiency.

Moreover, through the use of Zadeh's computing with words and perceptions paradigm, and of protoforms we have attained the above characteristics to a higher extent and at a lower cost and effort.

References

1. R.R. Yager: A new approach to the summarization of data. Information Sciences, 28, pp. 69–86, 1982.
2. R.R. Yager R.R.: On linguistic summaries of data. In W. Frawley and G. Piatetsky-Shapiro (Eds.): Knowledge Discovery in Databases. AAAI/MIT Press, pp. 347–363, 1991.
3. J. Kacprzyk and R.R. Yager: Linguistic summaries of data using fuzzy logic. International Journal of General Systems, 30, 33–154, 2001.
4. J. Kacprzyk, R.R. Yager and S. Zadrożny. A fuzzy logic based approach to linguistic summaries of databases. International Journal of Applied Mathematics and Computer Science, 10, 813–834, 2000.
5. J. Kacprzyk, R.R. Yager and S. Zadrożny. Fuzzy linguistic summaries of databases for an efficient business data analysis and decision support. In W.

Abramowicz and J. Zurada (Eds.): Knowledge Discovery for Business Information Systems, pp. 129-152, Kluwer, Boston, 2001.

6. L.A. Zadeh and J. Kacprzyk (Eds.): Computing with Words in Information/Intelligent Systems. Part 1. Foundations. Part 2. Applications, Springer–Verlag, Heidelberg and New York, 1999.
7. 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.
8. J. Kacprzyk and 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, pp. 417–425, IOS Press, Amsterdam, 2002.
9. J. Kacprzyk and S. Zadrożny. Data Mining via Linguistic Summaries of Data: An Interactive Approach. In T. Yamakawa and G. Matsumoto (Eds.): Methodologies for the Conception, Design and Application of Soft Computing. Proc. of IIZUKA'98, pp. 668–671, Iizuka, Japan, 1998.
10. J. Kacprzyk and 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, pp. 325-345, World Scientific, Singapore, 2001.
11. J. Kacprzyk and S. Zadrożny. FQUERY for Access: fuzzy querying for a Windows-based DBMS. In P. Bosc and J. Kacprzyk (Eds.): Fuzziness in Database Management Systems, pp. 415-433, Springer-Verlag, Heidelberg, 1995.
12. J. Kacprzyk and S. Zadrożny. The paradigm of computing with words in intelligent database querying. In L.A. Zadeh and J. Kacprzyk (Eds.): Computing with Words in Information/Intelligent Systems. Part 2. Foundations, pp. 382–398, Springer–Verlag, Heidelberg and New York, 1999.
13. L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. Computers and Mathematics with Applications. 9, 149–184, 1983.
14. Y. Liu and E.E. Kerre. An overview of fuzzy quantifiers. (I). Interpretations. Fuzzy Sets and Systems, 95, 1–21, 1998.
15. R.R. Yager and J. Kacprzyk (Eds.): The Ordered Weighted Averaging Operators: Theory and Applications. Kluwer, Boston, 1997.
16. D. Dubois, H. Fargier and H. Prade. Beyond min aggregation in multicriteria decision: (ordered) weighted min, discri-min,leximin. In R.R. Yager and J. Kacprzyk (Eds.): The Ordered Weighted Averaging Operators. Theory and Applications, pp. 181–192, Kluwer, Boston, 1997.
17. P. Hájek, M. Holeňa. Formal logics of discovery and hypothesis formation by machine. Theoretical Computer Science, 292, 345–357, 2003.
18. I. Gl̀ockner. Fuzzy quantifiers, multiple variable binding, and branching quantification. In T.Bilgi c et al. IFSA 2003. LNAI 2715, pp. 135–142, Springer-Verlag, Berlin and Heidelberg, 2003.
19. D. Dubois and H. Prade. Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distributions. Fuzzy Sets and Systems, 40, 143–202, 1991.
20. P. Bosc, D. Dubois and H. Prade. Fuzzy functional dependencies – an overview and a critical discussion. Proceedings of 3rd IEEE International Conference on Fuzzy Systems, pp. 325–330, Orlando, USA, 1994.
21. J. Kacprzyk and S. Zadrożny. Computing with words in intelligent database querying: standalone and Internet-based applications. Information Sciences, 134, 71–109, 2001.

22. J. Kacprzyk and S. Zadrożny. Computing with words: towards a new generation of linguistic querying and summarization of databases. In P. Sinčak and J. Vaščak (Eds.): Quo Vadis Computational Intelligence?, pp. 144–175, Springer-Verlag, Heidelberg and New York, 2000.
23. J. Kacprzyk and S. Zadrożny. On a fuzzy querying and data mining interface, Kybernetika, 36, 657–670, 2000.
24. J. Kacprzyk J. and S. Zadrożny. On combining intelligent querying and data mining using fuzzy logic concepts. In G. Bordogna and G. Pasi (Eds.): Recent Research Issues on the Management of Fuzziness in Databases, pp. 67–81, Springer–Verlag, Heidelberg and New York, 2000.
25. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. Proceedings of the 20th International Conference on Very Large Databases, Santiago de Chile, 1994.
26. H. Mannila, H. Toivonen and A.I. Verkamo. Efficient algorithms for discovering association rules. In U.M. Fayyad and R. Uthurusamy (Eds.): Proceedings of the AAAI Workshop on Knowledge Discovery in Databases, pp. 181–192, Seattle, USA, 1994.
27. Lee J.-H. and H. Lee-Kwang. An extension of association rules using fuzzy sets. Proceedings of the Seventh IFSA World Congress, pp. 399–402, Prague, Czech Republic, 1997.
28. W.-H. Au and K.C.C. Chan. FARM: A data mining system for discovering fuzzy association rules. Proceedings of the 8th IEEE International Conference on Fuzzy Systems, pp. 1217–1222, Seoul, Korea, 1999.
29. Y.-Ch. Hu, R.-Sh. Chen and G.-H. Tzeng. Mining fuzzy association rules for classification problems. Computers and Industrial Engineering, 43, 735–750, 2002.
30. P. Bosc, D. Dubois, O. Pivert, H. Prade and M. de Calmes. Fuzzy summarization of data using fuzzy cardinalities. Proceedings of IPMU 2002, pp. 1553–1559, Annecy, France, 2002.
31. J. Kacprzyk and S. Zadrożny. On linguistic approaches in flexible querying and mining of association rules. In H.L. Larsen, J. Kacprzyk, S. Zadrożny, T. Andreasen and H. Christiansen (Eds.): Flexible Query Answering Systems. Recent Advances, pp. 475–484, Springer-Verlag, Heidelberg and New York, 2001.
32. G. Chen, Q. Wei and E. Kerre. Fuzzy data mining: discovery of fuzzy generalized association rules. In G. Bordogna and G. Pasi (Eds.): Recent Issues on Fuzzy Databases, pp. 45–66. Springer-Verlag, Heidelberg and New York, 2000.
33. J. Kacprzyk and S. Zadrożny. Linguistic summarization of data sets using association rules. Proceedings of The IEEE International Conference on Fuzzy Systems, pp. 702–707, St. Louis, USA, 2003.
34. R. George and Srikanth R. Data summarization using genetic algorithms and fuzzy logic. In F. Herrera and J.L. Verdegay (Eds.): Genetic Algorithms and Soft Computing, pp. 599–611, Springer-Verlag, Heidelberg, 1996.
35. R. George and R. Srikanth. A soft computing approach to intensional answering in databases. Information Sciences, 92, 313–328, 1996.
36. J. Kacprzyk and P. Strykowski. Linguistic data summaries for intelligent decision support. In R. Felix (Ed.): Fuzzy Decision Analysis and Recognition Technology for Management, Planning and Optimization – Proceedings of EFDAN'99, pp. 3–12, Dortmund, Germany, 1999.
37. J. Kacprzyk and P. Strykowski. Linguitic summaries of sales data at a computer retailer: a case study. Proceedings of IFSA'99, pp. 29–33, Taipei, Taiwan R.O.C, vol. 1, 1999.

38. D. Rasmussen and R.R. Yager. Fuzzy query language for hypothesis evaluation. In Andreasen T., H. Christiansen and H. L. Larsen (Eds.): Flexible Query Answering Systems, pp. 23–43, Kluwer, Boston, 1997.
39. D. Rasmussen and R.R. Yager. Finding fuzzy and gradual functional dependencies with SummarySQL. Fuzzy Sets and Systems, 106, 131–142, 1999.
40. D. Dubois and H. Prade. Gradual rules in approximate reasoning. Information Sciences, 61, 103–122, 1992.
41. G. Raschia and N. Mouaddib. SAINTETIQ: a fuzzy set-based approach to database summarization. Fuzzy Sets and Systems, 129, 137–162, 2002.