TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
AUTOMATIC CONTROL | AUTOMATYKA
2-AC/2013

CARLOS A. DONIS-DIAZ*, RAFAEL BELLO*, JANUSZ KACPRZYK**

LINGUISTIC DATA SUMMARIZATION USING AN ENHANCED GENETIC ALGORITHM

LINGWISTYCZNE PODSUMOWANIA DANYCH Z UŻYCIEM ULEPSZONEGO ALGORYTMU GENETYCZNEGO

Abstract

This paper presents work is presented an enhanced Genetic Algorithm (GA) specifically designed for the production of linguistic data summaries. The model is able to obtain not a set of 'good linguistic summaries' but a 'good set' of summaries. The model incorporates an operator and fitness function specially designed to fulfil this aim. Experiments show how the enhanced model is able to improve results obtained with the classical model of GA and to guarantee a summary with high diversity and good values for the quality measures in individual summaries

Keywords: *Linguistic Data Summarization, Data mining, Fuzzy Logic, Genetic Algorithms*

Streszczenie

Przedmiotem niniejszego artykułu jest ulepszony algorytm genetyczny (GA) zaprojektowany z zamiarem użycia głównie do tworzenia lingwistycznych podsumowań danych. Zaproponowany model pozwala na uzyskanie nie tyle zbioru „dobrych podsumowań lingwistycznych", co „dobrego zbioru" tych podsumowań. Cel ten uzyskano przez zastosowanie w modelu odpowiedniego operatora i funkcji przystosowania. Przedstawione w artykule eksperymenty obliczeniowe potwierdzają, że autorski model wpływa na poprawę wyników otrzymanych dla klasycznego modelu opartego na algorytmie genetycznym oraz gwarantuje podsumowania charakteryzujące się dużą różnorodnością oraz dobrymi wartościami miar jakości poszczególnych podsumowań.

Słowa kluczowe: *lingwistyczne podsumowania danych, eksploracja danych, logika rozmyta, algorytmy genetyczne*

* Carlos A. Donis-Diaz, Rafael Bello, e-mail: cadonis@uclv.edu.cu, Computer Science Department, Universidad Central Marta Abreu de Las Villas, Santa Clara.

** Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences, Warsaw; Department, of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

1. Introduction

Linguistic data summarization has for a long time been a subject of intensive research, and various tools and techniques from computational linguistics, natural language generation etc. have been proposed. The traditional approaches do not however provide means for the representation and processing of imprecision and vagueness that is inherent in natural language. Fuzzy logic with linguistic quantifiers is one of the most conceptually simple, developed and used approaches for the linguistic summarization of numerical data (LDS), and – what is crucial for our application, provides simple and natural means it dealing with imprecision. The concept of a linguistic data summary, using fuzzy logic with linguistic quantifiers, which will be employed in this paper, was introduced by Yager in [1–3], considerably advanced in [4–7] and presented in an implementable way in [8].

The process of generating linguistic data summaries for a given set of numerical data, usually a relational numerical database, can conveniently be represented as an optimization problem in which the best summaries from a large set of candidates are selected, and the basic objective function is assumed to be the truth degree of a linguistic summary that is equated with a degree of truth of a linguistically quantified proposition that is conceptually equivalent to a linguistic summary in question. Several works to deal with this problem have been developed [9–13]. Much of them use a genetic algorithm, in principle in its basic version.

In the present work, an enhanced genetic algorithm specifically designed for finding not a set of 'good ('best') linguistic summaries', is proposed as it was the case in virtually all works done so far, but a 'good ('best') set' of linguistic summaries. A specific genetic operator and fitness function are proposed to deal with some problems observed in searching the solution space in the specific context of linguistic summarization.

The experiments were carried out on *creep* data. The creep rupture stress (*creep*) is one of the most important mechanical properties considered in the design of new steels used in industries like aeronautical, energy and petrochemical. Basically, (sometimes called cold flow too) is a tendency of a solid material to slowly deform permanently under mechanical stresses, and is increased due to high temperature, notably close to the melting point of the material in question, i.e. the *creep* measures the stress level in which a steel structure fails when exposed to quite aggressive conditions (like high steam temperatures) over periods of time as long as 30 years.

2. Linguistic data summaries for *creep* data: the use of a genetic algorithm

In this section are presented the main theoretical aspects needed to introduce the proposed enhanced model.

2.1. Linguistic data summarization

In this paper, the linguistic data summarization approach that uses the fuzzy logic with linguistic quantifiers proposed in [4] is considered. A basic description is presented here.

Having: $Y = \{y_1, ..., y_n\}$ a set of objects (records) in a database, e.g., the set of workers, and $A = \{A_1, ..., A_m\}$ a set of attributes (fuzzy variables) characterizing objects from Y, e.g., salary, age, etc. in a database D of workers, and $A_j(y_i)$ denotes the value of attribute A_j for object y_i. A linguistic summary from 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 quantifier Q (a linguistic quantifier), i.e. a fuzzy set with universe of discourse in the interval [0, 1] expressing a quantity in agreement, e.g. *most*;
- a truth degree T (validity) of the summary, i.e. a number from the interval [0, 1] assessing the truth of the summary (e.g. 0.7); usually, only summaries with a high value of T are of interest;
- optionally, a qualifier R, i.e. a fuzzy filter determining a fuzzy subset of Y; can be composed for one or a set of triplets <fuzzy variable, operator, linguistic value> (e.g. 'young' for attribute 'age').

Thus, linguistic summaries may be exemplified by:

$$T \text{ (most employees earn low salary)} = 0.7 \tag{1}$$

$$T \text{ (most young employees earn low salary)} = 0.7 \tag{2}$$

and their foundation is Zadeh's [14] linguistically quantified proposition corresponding to either, for (1) and (2):

$$Qy's \text{ are } S \tag{3}$$

$$QRy's \text{ are } S \tag{4}$$

The T, i.e., the truth value of (3) or (4) may be calculated by using either Zadeh's original calculus of linguistically quantified statements [14], or other interpretations of linguistic quantifiers. In this work is used the first where a (proportional, nondecreasing) linguistic quantifier Q is assumed to be a fuzzy set in [0, 1] and the values of T are calculated as:

$$T(Qy's \text{ are } S) = \mu_Q\left[\frac{1}{n}\sum_{i=1}^{n}\mu_S(y_i)\right]$$

and defining:

$$r = \frac{\sum_{i=1}^{n}\mu_R(y_i) \wedge \mu_S(y_i)}{\sum_{i=1}^{n}\mu_R(y_i)}$$

we have:

$$T(QRy's) \text{ are } S = \mu_Q(r)$$

Besides using T (truth), other validity measures have been proposed to determine the quality of summaries. In [5] can be found: the truth degree ($T1$) that corresponds with the mentioned T; the degree of imprecision ($T2$) that only depends on the form of the summarizer and expresses the fuzziness of the summary due to its description S; the degree of covering ($T3$) that says how many objects in the database corresponding with the qualifier R, are covered by the summary, i.e. by the particular description S; the degree of appropriateness ($T4$) that describes how characteristic for the particular database the summary found is; finally the length of the summary ($T5$) that is defined using the number of terms of the summarizer S.

As mentioned, several approaches have been used to mine the best summaries from a large set. The definition of *protoforms*, as a set of templates used to specify the form of summaries to be mined reducing the search space, is one of them. In [15–17] the concept of a *protoform* as a more or less abstract prototype of a linguistically quantified proposition is presented. The most abstract protoform corresponds to (3) and (4), while (1) and (2) are examples of fully instantiated protoforms. Thus, 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 (3) and (4), i.e., Q, S and R.

In the present work, the term *proposition* is used, to be more specific a linguistically quantified proposition, to refer a linguistic summary and the term (linguistic) *summary* refers to a set of propositions. This is basically consistent with [4, 8], and in particular the modern natural language generation (NLG) based approach [18].

2.2. The use of genetic algorithms in linguistic data summarization

The basic principles of GAs were first laid down rigorously by Holland [19], and they are well described in many books, such as [20]. The basic idea is to maintain a population of chromosomes, which represents candidate solutions to the concrete problem being solved, which evolves over time through a process of competition and controlled variation. Each chromosome in the population has an associated fitness to determine which chromosomes are used to form new ones in the competition process. The new ones are created using genetic operators such as crossovers and mutations. GAs have a great measure of success in search and optimization problems.

A GA starts off with a population of randomly generated chromosomes (solutions), and advances toward better chromosomes by applying genetic operators. The population undergoes evolution in the form of natural selection. During successive iterations called generations, a new population of chromosomes is formed using a selection mechanism and specific genetic operators such as crossovers and mutations. An evaluation or fitness function must be devised for each problem to be solved. Given a particular chromosome (a possible solution), the fitness function returns a single numerical fitness, which is supposed to be proportional to the utility or adaptation of the solution represented by that chromosome.

The classic GA model has been used previously on linguistic data summarization to efficiently handle the search space [11, 12]. The classic form of the model employed in those works refers to the basic form of the GA presented in [20], i.e. using the basic operators: selection, crossover and mutation. Other works, mainly developed for time series applications, use GA or related evolutionary heuristics with specific adaptations [9, 10].

3. An enhanced genetic algorithm for discovering linguistic data summaries

The proposed model uses an approach that permits obtaing the 'best set' of propositions (i.e. the 'best summary') what is different to the approach that looks for the set of 'best propositions' (i.e. the summary of 'best propositions'). The former approach has the advantage of getting, not only a summary with high quality propositions, but also with 'desirable characteristics' for the interaction among all propositions. The model was designed following two main aims: (1) to obtain propositions with a high goodness; (2) to get a final set of propositions with a high diversity in the sense of comprising propositions with sufficient differences in their structures and components. This is in line with recent developments in the linguistic data summarization, cf. [16, 17].

3.1. The genetic representation

In the present work, the chromosome represents a whole linguistic summary (i.e. a set of linguistically quantified propositions) and each gene codifies just one such proposition. Making a parallel with approaches for rules discovery used by evolutionary models, the proposed model falls into the Pittsburgh approach [21]. In this approach, each individual corresponds to a complete set of rules and each run of the evolutionary procedure discovers one set of rules, probably the best one between all iterations. An important feature is that in order to guide the discovery process, a complete set of propositions is evaluated instead of a single proposition so the model can consider the importance of the interaction between propositions; i.e. in the present work, the model is able to control the diversity inside the set of propositions to be obtained.

The protoform used to search the propositions has the following form:

Protoform	Given	Sought
QRy's are *S*	structure of *S* (*creep is* <linguistic value>)	*R*, *Q* and linguistic values in *S*

Taking this into consideration, the genes encode the three main components of a linguistically quantified proposition: the quantifier (*Q*), the qualifier (*R*) and the summarizer (*S*).

3.2. The fitness function

To achieve the aims mentioned above, the proposed fitness function contains: (1) four measures to control the quality of the propositions: degree of Truth ($T1$), imprecision degree ($T2$), covering degree ($T3$) and appropriateness degree ($T4$); and (2), a variable called *Diversity* to control the interaction between the propositions inside the summary. The function to be maximized for a chromosome i is defined in the interval [0, 1] as $F_i = m_g G_i + m_d D_i$ where G and D represent the degree of Goodness and Diversity respectively, and m_g, m_d are the importance degrees assigned to the terms. The Goodness (G) of a chromosome j is calculated as the average value of the goodness of genes. The goodness for each gene (proposition) g_j is characterized by a weighted sum of the quality measures ($T1^{St}$, $T2$, $T3$, $T4$) where

$T1^{St}$ represents a term called *Linguistic Strength* that is calculated as: $T1^{St} = T1 \cdot \overrightarrow{St}$ been $\overrightarrow{St}$ a vector of values in the interval [0, 1]) defined as a parameter to express the preference for each linguistic label of the quantifier. In this work it was defined as: St[Most] = 1, St[Much] = 0.75, St[Half] = 0.20, St[Some] = 0.15 and St[Few] = 0.05 so *Most* and *Much* are preferred; this is in line with some indications of a special role played by the linguistic quantifier '*most*' which can be found in many contributions shown in, e.g., [22]. The Goodness of a gene j is calculated as $g_j = 0.4 \cdot T1^{St} + 0.1 \cdot T2 + 0.25 \cdot T3 + 0.25 \cdot T4$.

The Diversity (D) degree is calculated taking into account the number of clusters of genes (C) existing in the chromosome: $D_i = C_i/n$ where C is obtained by a clustering process using a similarity function (L) to determine if two propositions are similar or not. L is defined as:

$$L(p1, p2) = \begin{cases} Yes & if \ \sum_{k=0}^{m} H\left(p1_k, p2_k\right) < 2 \\ No & otherwise \end{cases}$$

where $p1$ and $p2$ are vectors of size m representing the propositions to be compared. Its components refer to the linguistic values (labels) used in the propositions. There is one component for each fuzzy variable; if a specific fuzzy variable is not used in the proposition, the respective component is equal to zero.

The function $H(p1_k, p2_k)$ is defined as:

$$H(p1_k, p2_k) = \begin{cases} 1 & if \ \left|p1_k - p2_k\right| > round(20\% \ of \ V_k) \ \text{ or } \ if \ \ p1_k = 0 \text{ and } p2_k \neq 0 \ \text{ or} \\ & if \ \ p1_k \neq 0 \text{ and } p2_k = 0 \\ 0 & otherwise \end{cases}$$

where V_k is the number of labels of the fuzzy variable represented by the k-th component of the vector. This function determines if there is a difference between the labels used in $p1$ and $p2$ for a specific fuzzy variable.

3.3. The genetic operators

When using the normal operators (selection, crossover and mutation) in the classical GA the following problem arises: the crossover operator improves the degree of diversity of the summary but not the degree of goodness of individual propositions due to this operator has not direct influence over the genes (propositions) in the chromosome. On the other hand, the mutation operator does not guarantee a sufficient perturbation inside the chromosome to solve the situation.

To deal with this problem, the use of an additional operator is proposed to be added to the evolution process in the enhanced model. This operator, called the *Propositions Improver*, implements a local search based on a best first strategy [23] when looking for a better variant in the neighborhood of the proposition. It implements a greedy random procedure based on six possible transformations of the proposition. Four of these transformations occur on the qualifier, one on the summarizer and one on the quantifier. The transformations are:

– Change in R (the qualifier) a randomly selected fuzzy predicate by another randomly generated.
– Change in R the linguistic value of a randomly selected fuzzy predicate by another randomly generated.
– Add in R a new randomly generated fuzzy predicate.
– Delete in R a new randomly selected fuzzy predicate.
– Change in Q (the quantifier) its linguistic value by a 'nearby' one, i.e. by the following (backward or forward) linguistic value in the set of terms.
– Change in S (the summarizer) its linguistic value by a 'nearby' one.

The stopping criterion for the local search occurs when at least one of the following values is reached:

– the total number of new generated propositions is equal to 8,
– the number of continuously generated propositions without improvement is equal to 5 or
– the value of T (truth) is equal to or greater than 0.85

4. Experiments results and analysis

Linguistic data summarization has proven effective to describe *creep* trends regarding specific variables used in the process of designing new ferritic steels as experimented in [24]. However, its use in this work was reduced to obtain a small set of fully instantiated propositions to be compared with results obtained with an artificial neural network model. Unlike this, a complete mining of propositions is performed in the present work. The creep data and the fuzzy modeling used is the same as in [24].

Several experiments were performed to measure and compare the performance of the proposed enhanced model (Enhanced) in relation to the classical model (Classical) for obtaining a good linguistic summary from *creep* data. Ten runs of the models were used for each experiment and each run was limited to a fixed number of generated propositions: a maximum of 250.000 what represent an insignificant amount of possible propositions for this problem (the 6.91E-15 percent). In this way, can be ensured that one model does not take advantage over the other with respect to the amount of propositions considered to find the best solution.

Is important to note that for the *creep* problem, the propositions having *Most* or *Much* as quantifiers are more interesting, that is why the parameter $\overrightarrow{\mathrm{St}}$ was set preferring these values in both models.

For a better interpretation and analysis, the Wilcoxon's test and Monte Carlo's technique were used to compare the results pairs to pairs and to calculate a more precise signification of the differences respectively. A value less than 0.05 in Monte Carlo's technique were considered as significant for the differences.

Table 1 shows the results obtained. In this table, the rows represent the results obtained with each model. Columns refer the parameters used to measure the quality of the obtained summary:

– Columns (1) to (4) represent the mean values obtained by models for the indicators used in the fitness function to measure the quality of the propositions composing the summary:

  - column (1) represents the linguistic strength,
  - column (2) represents the degree of imprecision,
  - column (3) represents the degree of covering,
  - column (4) represents the degree of appropriateness.
- Column (5) represents the mean goodness value of propositions that compose the summary. This value is calculated as a weighted sum of the previous indicators.
- Column (6) represents the mean value of diversity.
- Column (7) represents the mean value of fitness.
- Column (8) represents the mean number of propositions in the summary that: (8A) have the desired linguistic values for the quantifier (*most* and *much*); (8B) do not have the desired values of truth ($T < 0.85$).

Table 1

Behavior of the two variants of the GA model

GA model	Mean values of							Mean number of propositions with (8)	
	$T1^{St}$ (1)	$T2$ (2)	$T3$ (3)	$T4$ (4)	G (5)	D (6)	*Fitness* (7)	Quantifier (*most*, *much*) (A)	$T1 < 0.85$ (B)
Classical	0.1962	0.8918	0.3158	0.3805	0.3417	1.000	0.5392	12.0	15.5
Enhanced	0.5157	0.8960	0.5287	0.5343	0.5616	1.000	0.6931	16.3	1.8

When analyzing the indicators used to measure the quality of the propositions (columns 1 to 4), it can be observed that the enhanced model provides better results in all values except for the imprecision degree. This is because this indicator depends only on the linguistic terms used in the summarizer and in the present application the summarizer has only one fuzzy predicate and its linguistic terms are randomly selected with the same probability. For the rest of values, the enhanced model presents significant differences in relation to values obtained with the classical model. As a direct consequence, the value of goodness (column 5) obtained with the enhanced model is better than the value of the classical model and presents significant differences.

The column (8A) indicates how the enhanced model is able to find a better mean number of propositions with the desired quantifier. Column (8B) shows the number of propositions on the summary (from a total number of 30, the size used for the chromosomes) with a degree of truth less than a value considered good for this application. This column reflects how the classical model is unable to evolve a big number of the propositions towards better ones using the normal operators.

Those results show how the local search implemented in the enhanced model through the *Propositions Improver* operator is able to obtain individual propositions with an improved quality.

When analyzing the values of the diversity degree shown in column (6) it can be noted that both models obtain good values. This result to remark allows us that the effectiveness of the crossover operator for fitting the desired behavior in the diversity degree between the propositions composing the summary.

Finally, column (7) shows the better overall behavior of the enhanced model in comparison to the classical model. The differences in the fitness values are significant.

5. Conclusions

A model to obtain linguistic data summaries using an enhanced GA has been proposed. The use of a local search in the form of an additional operator in the classical GA has shown an improvement in results. The proposed fitness function including the term *Diversity*, the parameter St and the used quality measures guarantees a summary with high quality propositions, a good degree of diversity and many propositions with the desired quantifiers. Our future works will be further explored possibilities to use local search in GA type schemes, exemplified by memetic algorithms, as well as the use of new approaches for the derivation of linguistic data summaries based on natural language technology, notably natural language generation (NLG) [18].

References

[1] Yager R.R., *A new approach to the summarization of data*, Information Sciences, 28, 1982, 69-86.

[2] Yager R.R., *On linguistic summaries of data*, [in:] G. Piatetsky-Shapiro, W.J. Frawley (Eds.) *Knowledge Discovery in Databases*, AAAI Press/The MIT Press, Menlo Park, 1991, 347-363.

[3] Yager R.R., *Database discovery using fuzzy sets*, International Journal of Intelligent Systems, 11, 1996, 691-712.

[4] Kacprzyk J., *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, Heidelberg and New York 2000, 153-161.

[5] Kacprzyk J., Yager R.R., *Linguistic summaries of data using fuzzy logic*, International Journal of General Systems, 30, 2001, 133-154.

[6] Kacprzyk J., Yager R.R., Zadrożny S., *A fuzzy logic based approach to linguistic summaries of databases*, International Journal of Applied Mathematics and Computer Science, 10, 2000, 813-834.

[7] Kacprzyk J., Yager R.R., Zadrożny S., *Fuzzy linguistic summaries of databases for an efficient business data analysis and decision support*, [in:] W. Abramowicz, J. Żurada (Eds.) *Knowledge Discovery for Business Information Systems*, Kluwer, Boston 2001, 129-152.

[8] Kacprzyk J., Zadrożny S., *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, 144-175.

[9] Castillo-Ortega R. et al., *Linguistic Summarization of Time Series Data using Genetic Algorithms*, 7th Conference of European Society for Fuzzy Logic and Technology – EUSFLAT 2011, Atlantis Press, Aix-les-Bains 2011, 416-423.

[10] Castillo-Ortega R. et al., *A Multi-Objective Memetic Algorithm for the Linguistic Summarization of Time Series*, 13th Annual Genetic and Evolutionary Computation Conference – GECCO' 2011, ACM, Dublin 2011, 171-172.

[11] George R., Srikanth R., *Data summarization using genetic algorithms and fuzzy logic*, [in:] F. Herrera, J.L. Verdegay (Eds.) *Genetic Algorithms and Soft Computing*, Physica-Verlag, Heidelberg 1996, 599-611.

[12] Kacprzyk J., Wilbik A., Zadrożny S., *Using a Genetic Algorithm to Derive a Linguistic Summary of Trends in Numerical Time Series*, International Symposium on Evolving Fuzzy Systems, Ambleside 2006, 137-142.

[13] Kacprzyk J., Wilbik A., Zadrożny S., *Linguistic summarization of time series using a fuzzy quantifier driven aggregation*, Fuzzy Sets and Systems, 159, 2008, 1485-1499.

[14] Zadeh L.A., *A computational approach to fuzzy quantifiers in natural languages*, Computers and Mathematics with Applications, 9, 1983, 149-184.

[15] Kacprzyk J., Zadrożny S., *Protoforms of linguistic data summaries: towards more general natural-language-based data mining tools*, [in:] A. Abraham, J.R.D. Solar, M. Koeppen (Eds.) *Soft Computing Systems*, IOS Press, Amsterdam 2002, 417-425.

[16] Kacprzyk J., Zadrożny S., *Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools*, Information Sciences, 173, 2005, 281-304.

[17] Kacprzyk J., Zadrożny S., *Protoforms of Linguistic Database Summaries as a Human Consistent Tool for Using Natural Language in Data Mining*, International Journal of Software Science and Computational Intelligence, 1, 2009.

[18] Kacprzyk J., Zadrożny S., *Computing with words is an implementable paradigm: fuzzy queries, linguistic data summaries and natural language generation*, IEEE Transactions on Fuzzy Systems, 18, 2010, 461-472.

[19] Holland J.H., *Adaptation in natural and artificial systems*, MIT Press, Cambridge 1992.

[20] Goldberg D.E., *Genetic algorithms in search*, Optimization and Machine Learning, Addison-Wesley, Reading 1989.

[21] Smith S., *Flexible learning of problem solving heuristics through adaptive search*, 8th International Conference on Artificial Intelligence, Morgan Kaufmann, 1983, 422-425.

[22] Zadeh L.A., Kacprzyk J., *Computing with Words in Information/Intelligent Systems*, Physica-Verlag (Springer-Verlag), Heidelberg and New York 1999.

[23] Russell S.J., Norvig P., *Artificial Intelligence: A Modern Approach*, Third ed., Prentice Hall, 2009.

[24] Díaz C.A.D., Perez R.B., Morales E.V., *Using Linguistic Data Summarization in the study of creep data for the design of new steels*, 11th International Conference on Intelligent Systems Design and Applications – ISDA 2011, Cordoba 2011, 160-165.