Linguistic Summarization of Time Series Data using Genetic Algorithms*

Rita Castillo-Ortega[1] Nicolás Marín[2] Daniel Sánchez[3] Andrea G.B. Tettamanzi[4]

[1,2]University of Granada, Granada, Spain, Email: {rita,nicm}@decsai.ugr.es
[3]European Centre for Soft Computing, Mieres, Asturias, Spain, Email: daniel.sanchezf@softcomputing.es
[4]Università degli Studi di Milano, Crema (CR), Italy, Email: andrea.tettamanzi@unimi.it

Abstract

In this paper, the use of an evolutionary approach when obtaining linguistic summaries from time series data is proposed. We assume the availability of a hierarchical partition of the time dimension in the time series. The use of natural language allows the human users to understand the resulting summaries in an easy way. The number of possible final summaries and the different ways of measuring their quality has taken us to adopt the use of a multi objective evolutionary algorithm. We compare the results of the new approach with our previous greedy algorithms.

Keywords: Linguistic Summarization, Multi Objective Evolutionary Algorithms, Time Series, Dimensional Data Model, Fuzzy Logic

1. Introduction

Big business companies and organizations consume and produce extensive quantities of data. The correct interpretation of these data is a valuable capacity which affects the process of decision making and selection of strategies, hence it is crucial for their future. That is, data are relevant but what it is even more important it is being able to obtain information from those data. This new information can be easily used in order to ease tasks like decision analysis, prediction or forecasting [1]. Such pieces of new information are expected to have a visible effect in the performance of the companies, since those companies taking decisions on the basis of this knowledge have far more opportunities to develop successful strategies and to be more competitive.

Due to the main role that time plays in general, the major part of the "to analyze data" is related with the time dimension. Well known examples of time series include stock exchange trends, the evolution of the sells of a given product along time, the inflow of patients to a medical center, the variation of prices of a given product during a year (as an example, crude oil, gasoline or tomatoes), etc... Many authors have focussed their researches on the so called time series data mining [2].

The most common way of expressing time series data is by means of graphical representations, leaving the description to the user, with several disadvantages. First, when many different time series are to be studied, too many time and even experts may be needed, making this approach unfeasible. As an example, consider the case of a data cube with a time dimension; then, a time series is obtained for every possible combination of values of the other dimensions at all the levels. In addition, for time series with a lot of data, it may be difficult to obtain a description because a global view is difficult, but also because some information may be hidden because of the granularity level employed in the graphical depiction. Finally, expert knowledge is frequently needed in order to provide a linguistic description of data, hence the graphical depiction may not be a good solution for a non-expert user, since it is not easy to introduce the necessary background knowledge into the graphical representation.

These facts motivate the development of techniques for performing automatic linguistic summarization of time series data, also known as *time series summarization techniques* in the literature. The use of natural language seems to be reasonable since it is easy to understand for humans, being the natural way to communicate and to describe features of time series data. The use of natural language in summarization allow us to express the results in a more understandable way so experts are not needed to interpret them. The final results are easier to comprehend even when they refer to different features or several time series. In the same way, the interpretation is less time consuming.

In our work, apart from the importance of *understandability*, we claim that a good final summary must be *accurate*, as *brief* as possible and properly *cover the whole time dimension*. Those are contradictory objectives that the final solution have to accomplish. In general, the more accurate a summary is, the less brief it is. In the same way, a summary covering the whole time domain is normally larger that those ones that do not cover the time com-

*Part of the research was supported by the Andalusian Government (Junta de Andalucía, Consejería de Innovación, Ciencia y Empresa) under project P07-TIC-03175 *Representación y Manipulación de Objetos Imperfectos en Problemas de Integración de Datos: Una Aplicación a los Almacenes de Objetos de Aprendizaje*. Part of the research was supported by the Spanish Government (Science and Innovation Department) under project TIN2009-08296.

pletely. Depending on the importance of each of these objectives to each user a different summary could be obtained. In order to face these problems we have adopted the use of a multi objective evolutionary algorithm, also known as MOEA.

Up to this moment the strategies followed by us were Greedy strategies, which select the best possibility at each step obtaining an optimal solution. We have adopted a new vision based on Evolutionary Computation in order to better explore the solution space reaching a compromise between the several objectives. Here, we have face the upgrade phase focussing our attention on the strategy used to search the solution given to the user but for the time being centering ourselves on a single characteristic of the time series.

Here, we have focused our attention on the strategy used to search the solution given to the user. We illustrate it centering ourselves on a single characteristic of the time series.

2. Related work

Soft computing as well as fuzzy set theory and fuzzy logic have played an essential role when trying to transform data into words obtaining linguistic descriptions understandable by humans (see [3], [4], and [5]). One of the first works in this area was made by R. R. Yager in [6] where the author uses quantified sentences in the sense of L. Zadeh [7] and later his own OWA operators (OWA stands for Ordered Weighted Averaging) in [8] and [9].

In [10] Garrido, Marín and Pons express their interest regarding the imprecision present in the time dimension. The source of imprecision could be due to the use of natural language or to the nature of the information source. Whatever it comes, the use of fuzzy logic in fuzzy intervals helps the authors to deal with it.

Highly related to our work A. Laurent has worked on the concept of fuzzy summaries obtained from multidimensional databases in [11]. Also regarding linguistic summarization we can find works as [12] and [13], from D. Pilarski and L. Zhang, Z.Pei, and H. Chen, respectively. The first of them presents an automatic tool to generate summaries named Quatirius while the second one uses degree theory and FCA. From our point of view, it is also attractive the work of R. R. Yager and F. E. Petry in which ontologies of terms have been used in their multicriteria approach to data summarization [14].

J. Kacprzyk [15], J. Kacprzyk and R. R. Yager [16], J. Kacprzyk and S. Zadrozny [17] and [18] and J. Kacprzyk, R. R. Yager and S. Zadrozny, and S. Zadrozny [19] have also worked in linguistic summarization using fuzzy quantified sentences, as we do too, and protoforms, this way they obtain different summary profiles.

When dealing with the linguistic summarization of time series data in particular we come across with several interesting works as for example, the works of I. Kobayashi [20], D. Chiang [21] or I. Z. Batyrshin and L. B. Sheremetov [2] in which they attempt to mine and verbalize time series data.

Two greedy algorithms to obtain linguistic summaries of rough time series data have been presented in [22, 23] and studied in detail in [24].

3. The linguistic framework

This section is devoted to describe the linguistic background that will allow us to set a proper context in which to obtain linguistic summaries coming from numeric data.

Consider that the time domain is described in its finest grained level of granularity by members $T = \{t_1, ..., t_m\}$. Then, a given time series defined on this time domain will have the following form: $TS = \{< t_1, v_1 >, ..., < t_m, v_m >\}$, where every v_i is a value of the basic domain D_V of a variable V.

In order to linguistically describe the information of this time series, we have considered using fuzziness as follows:

- The basic domain of variable V under study is partitioned by a set of linguistic labels $E=\{E_1, ..., E_s\}$.
- The time dimension is hierarchically organized in n levels, namely, $L=L_1, ..., L_n$. Each level L_i has associated a partition $\{D_{i,1}, ..., D_{i,p_i}\}$ of the basic time domain.

There is no restriction concerning the form of the membership function of a label apart from that it must be normalized. In our approach, we will use trapezoidal functions. When necessary, labels $D_{i,j}$ in time dimension can be the union of a set of trapezoidal functions.

In this work, a set of labels $\{X_1, ..., X_r\}$ is a partition on X iff:

1. $\forall x \in X, \exists X_i, i \in \{1..r\} | \mu_{X_i}(x) > 0$.
2. $\forall i, j \in \{1..r\}, i \neq j, core(X_i) \cap core(X_j) = \emptyset$.

Additionally, considering the hierarchy of the time dimension, we add the following constraints:

1. $\forall i, j \in \{1..n\}, i < j, p_i > p_j$ (i.e, as we move upward in the hierarchy, the number of labels of the partition decreases).
2. $\forall i \in \{2..n\}, \forall j \in \{1..p_i\}, \forall k \in \{1..p_{i-1}\} | (D_{i,j} \subseteq D_{i-1,k}) \rightarrow (D_{i,j} = D_{i-1,k})$ (i.e., labels cannot generalize another label of an upper level).

Figure 1 illustrates the above described context for the time series.

4. Linguistic Summarization of the Series

Once we have described the form of the domains where the time series is defined, the next step is to introduce the new approach to linguistically summarize the information related to the data series.

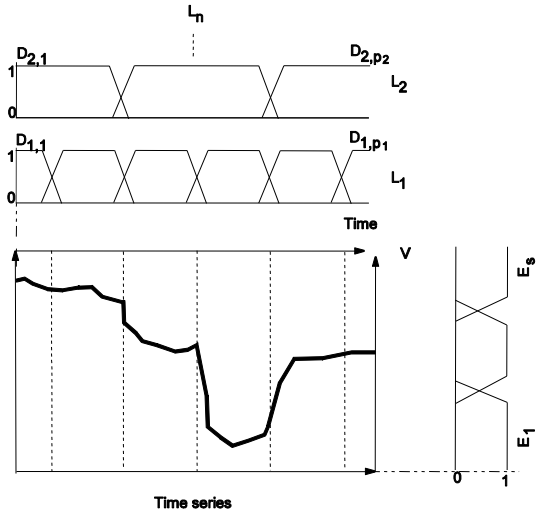

Figure 1: General context for the summarization of a time series.

4.1. Linguistically quantified sentences

Fuzzy quantification extends classical quantification by considering (fuzzy) linguistic quantifiers, a generalization of the ordinary quantifiers $\exists$ and $\forall$ of first order logic [7]. A large number of applications can be found in the literature in areas like quantifier-guided aggregation, linguistic summarization, computing with words, and quantification in fuzzy description logics, among many others.

The most usual quantified sentences considered in the literature are of the form "Q of X are A" or "Q of D are A", where Q is a linguistic quantifier, X is a (finite) crisp set, and A, D are fuzzy subsets of X. These sentences are called type I and type II sentences, respectively. Linguistic quantifiers are normal, convex fuzzy subsets of $\mathbb{Z}$ (absolute quantifiers) or $[0,1]$ (relative quantifiers). Particularly, in this paper, linguistic quantifiers are represented by trapezoidal functions.

There are many different approaches for evaluating quantified sentences; we shall use the method called *GD* introduced in [25] as follows: the evaluation of "Q of D are A" by means of GD is

$$GD_Q(A/D) = \sum_{\alpha_i \in \triangle(A/D)} (\alpha_i - \alpha_{i+1}) Q\left(\frac{|(A \cap D)_{\alpha_i}|}{|D_{\alpha_i}|}\right) \quad (1)$$

where $(A \cap D)(x) = \mathrm{Min}(A(x), D(x))$, $\triangle(A/D) = \Lambda(A \cap D) \cup \Lambda(D)$, $\Lambda(D)$ being the level set of D, and $\triangle(A/D) = \{\alpha_1, ..., \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ for every $i \in \{1, ..., p\}$, we consider $\alpha_1 = 1$ and $\alpha_{p+1} = 0$ (although α_{p+1} is not in the level set we consider it in the formula). The set D is assumed to be normalized. If not, D is normalized and the same normalization factor is applied to $A \cap D$ (D and $A \cap D$ will be divided by the greater value in D) [1].

[1] As the data set X is assume to be finite, D is considered to be finite, and hence the number of relevant α-cuts is also finite.

The GD method has been used due to its efficiency and non-strict character. The method also fulfills some interesting properties related to relative quantifiers (defined in [25]). Another point is that it is easy to implement. Anyway, let us remark that the strategy presented in this paper is not dependent on the evaluation method.

We consider that the user is interested in linguistic summaries which take the form of a collection of quantified sentences that describe the behavior of a series of data. We assume that the basic elements of these summaries are the linguistic labels described in Section 3. That is, our approach will deliver a collection of sentences of the form "Q of $D^S_{i,j}$ are A^S" where:

- $D_{i,j}$ is a label member of a certain level i of the hierarchy associated to the time dimension and
$$D^S_{i,j}(< t, v >) = D_{i,j}(t). \quad (2)$$
- A is a label or the union of a subset of labels of the partition of the variable V under study, and
$$A^S(< t, v >) = A(v). \quad (3)$$

With this kind of sentences, the approach will be able to produce sentences like "Most (Q) days of the cold season ($D^S_{i,j}$), patient inflow was high (A^S)" or "Most (Q) days of the hot season ($D^S_{i,j}$), patient inflow was low or very low (A^S)".

The user must provide a collection of quantifiers defining the kind of fuzzy quantities and percentages she/he is interested in. This can be defined by choosing among a collection of predefined quantifiers. In this work, we consider that the user provides a *totally ordered* subset $\{Q_1, ..., Q_{qmax}\}$ of a coherent family of quantifiers $\mathcal{Q}$ [26] to be used in the summarization process.

Definition 1 (Coherent family of quantifiers) *Let $\mathcal{Q}$=$\{Q_1, ..., Q_l\}$ be a linguistic quantifier set, we shall say it is coherent if it verifies that:*

- *The membership functions of elements in $\mathcal{Q}$ are non-decreasing functions.*
- *A partial order relation $\succeq$ is defined in $\mathcal{Q}$. It has as its maximal element $Q_1 = \exists$ and as its minimal one $Q_l = \forall$. Furthermore $\forall Q_i, Q_j \in \mathcal{Q}$, $Q_i \subseteq Q_j \Rightarrow Q_j \succeq Q_i$.*
- *The membership function of the quantifier $\exists$ is given by $Q_1(x) = 1$ if $x \neq 0$ and $Q_1(0) = 0$, whereas the membership functions of $\forall$ will be $Q_l(x) = 0$ if $x \neq 1$ and $Q_l(1) = 1$.*

In addition, the user will provide a threshold τ for the minimum accomplishment degree she/he wishes for the quantified sentences comprising the summaries.

4.2. A summary of quality

Our final objective is to obtain a collection of quantified sentences using the elements defined by the user as described previously. The requirements for this collection of quantified sentences, according to the intuitive idea of summary, are the following:

- The accomplishment degree of every sentence must be greater than or equal to τ, i.e., the information provided by every sentence must hold in the data to a high (τ) degree (*accuracy*).
- The set of quantified sentences must be as small as possible (*brevity*). In particular, there is at most one quantified sentence involving a time period $D_{i,j}$.
- The union of the supports of all the time periods $D_{i,j}$ in the sentences of the summary must be T (*coverage*).

5. Multi-Objective Evolutionary Algorithm

Evolutionary algorithms (EAs) [27, 28] are a broad class of stochastic optimization algorithms, inspired by biology and in particular by those biological processes that allow populations of organisms to adapt to their surrounding environment: genetic inheritance and survival of the fittest. Each individual of the population represents a point in the space of the potential solutions for the considered problem.

The evolution is obtained by iteratively applying a (usually quite small) set of stochastic operators, known as *mutation*, *recombination*, and *selection*. Mutation randomly perturbs a candidate solution; recombination decomposes two distinct solutions and then randomly mixes their parts to form novel solutions; selection replicates the most successful solutions found in a population at a rate proportional to their relative quality.

The initial population may be either a random sample of the solution space or may be seeded with solutions found by simple local search procedures, if these are available. The resulting process tends to find, given enough time, globally optimal solutions to the problem much in the same way as in nature populations of organisms tend to adapt to their surrounding environment.

Real-world problems often involve multiple objectives, which, ideally, should be optimized simultaneously. In practice, however, this is not always possible, as some of the objectives may be conflicting. To address this type of problems, several multi-objective evolutionary algorithms (MOEAs) have been proposed, using a variety of techniques [29].

The problem of finding a good linguistic summary of some data at hand may be naturally formulated as a multi-criterion optimization problem, where three conflicting criteria, namely accuracy, brevity, and coverage, must be maximized. This means that it is not possible, in general, to come up with the "best" possible linguistic summary for some data. However, the user should be presented with a gamut of non-dominated solutions, featuring different but all in a sense optimal combinations of values for the three criteria.

A popular and effective MOEA well-suited to solving this problem is NSGA-II [30]. NSGA-II works by sorting a population of candidate solution into Pareto fronts, so that non-dominated solutions are in the first front, and applies a niching technique and elitism to improve the population along the entire Pareto front.

We have adopted this algorithm and have adapted it to handle some specificities of linguistic summarization. Besides defining an appropriate encoding to represent linguistic summaries, such adaptations have mainly consisted of the definition of problem-specific genetic operators.

5.1. Representation

The first task we have to face when dealing with genetic algorithms is the selection of the solutions representation. The use of memory in this representation is essential to obtain a correct performance in terms of both, memory and time. The following steps will be strongly influenced by the selected representations. To correctly design the genetic operators it is important to have a good knowledge of the selected representation. In the same way, the initialization of the individual as the evaluation of objectives and constraints are highly related to the representation of the individuals int the population.

A linguistic summary is represented by means of a variable-size chromosome, logically divided in genes that encode a single quantified sentences of the form "Q of D are A" where Q is a quantifier and D, A are fuzzy sets, as illustrated in Figure 2.

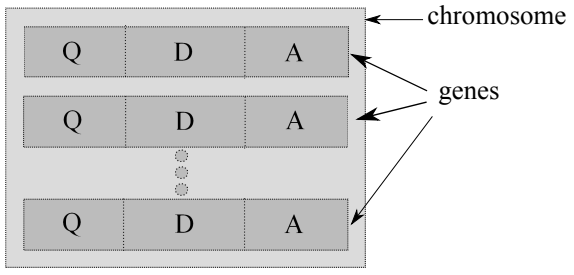

Figure 2: Structure of a chromosome.

To avoid repetition of information, each of the components of a gene contains just a reference to the place where the relevant object is stored (see Figure 3).

Each of the components depicted in Figure 3 (Q, D, and A) is stored in memory as an array of real numbers ranging in the interval [0,1]. This way the chromosomes represent the solutions without a great amount of repetition and in an easy way, obtaining a representation with a great level of abstraction.

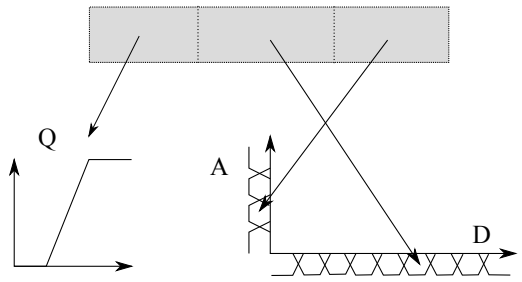

Figure 3: Information in a gene.

Considering the storage of A, some differences exist with respect to Q and D. In our algorithm it is allowed to use groups of labels A. Therefore, A may represent either a single label or a set of labels (a group).

5.2. Objectives

Once we have the representation we have to focus our attention on the desired objectives. As it was presented before, the measures of quality of a given solution are: brevity, accuracy and time coverage.

Brevity of a linguistic summary is computed as the number of quantified sentences that make up the summary. Sentences with group labels count for as many sentences as there are labels in the group.

Accuracy is computed for an individual by averaging the accuracies of the sentences that compose it. The accuracy of a single sentence is computed based on the GD method introduced in Section 4.1. However, accuracy is a compound measure that does not depends on the GD accomplishment degree alone. The precision of the quantifier Q is also important for us. We have introduced a new parameter λ that expresses the importance of the precision of the quantifier for the user. The formula is the following, for each gene of the individual:

$$\text{accuracy}(\text{gene}_{xi}) = \frac{\lambda * q_{xi} + a_{xi}}{1 + \lambda}, \qquad (4)$$

where $q, a \in [0, 1]$; $q = \frac{1}{Q}$ if Q is permitted by Q_{boundi} and $q = 0$ if it is not, and a is the value obtained by applying GD to the gene.

For the entire individual,

$$\text{Accuracy}(X) = \frac{\sum_{i=0}^{\text{length}-1} \text{accuracy}(\text{gene}_{xi})}{\text{length}}. \qquad (5)$$

The introduction of this new parameter λ allows us to model the importance of using the stricter quantifiers within the coherent family. Having several similar sentences with similar degree of accomplishment, the one with the strictest quantifier is preferred because it will be able to express the most precise information. Finally, using the average of all the individuals we take into account the importance of each quantified sentence compounding the individual.

Coverage is computed by counting the number of time points that are covered by at least one sentence in the summary. Labels representing periods of time are stored as arrays containing values ranging in $[0, 1]$. For each time period, a vector of the same length as the total time considered is maintained. Each cell contains 0 if the time point is not included in the period, 1 if it is completely included, and values between 0 and 1 depending on its inclusion degree.

For each gene in the chromosome, the corresponding vectors are aggregated by using maximum. Coverage is then obtained as the sum of the cells of the aggregated vector.

Since NSGA-II works under the assumption that the objectives are to be minimized, whereas accuracy and coverage are to be maximized, the sign of the latter two criteria is changed to obtain the corresponding objectives for NSGA-II.

5.3. Constraints

Apart from the features that we want to obtain, objectives, we also have behaviuors that we want to avoid, that is, constraints.

The problem comprises a number of constraints on the linguistic summaries produced:

- *inclusion*—the same time period should not be described by more than one sentence in a summary;
- *threshold*—the accuracy of a summary must be above a user-provided threshold;
- *Q-bound*—least strict quantifier allowed in a sentence;
- *G-bound*—maximum label group size allowed in a sentence.

All the above constraints are handled by adding penalty terms to the relevant objective in case of violation and are enforced by the specialized mutation operators.

5.4. Initialization

The initial population is seeded with individuals with a random number, extracted from an exponential distribution, of sentences whose Q, D, and A are randomly extracted from a uniform distribution.

The highly variable (HV) portions of the time series are "masked", so to speak, before starting the EA. Therefore, from EA's viewpoint, it is as if the HV portions of the time series did not exist.

When dealing with evolutionary approaches, the use of an initial population obtained using randomly created individuals is not unusual. The goal is to maintain a heterogeneous sample of possible solutions. Sometimes good quality solutions can be added in this first step with the objective of assessing the existence of good solutions able to be evolved.

5.5. Genetic Operators

Regarding the genetic operators we have worked with one kind of recombination and several kind of mutations.

Recombination takes two summaries from the parent population and produces two offspring summaries by uniform crossover: each sentence in a parent individual is copied to either offspring individual with probability $\frac{1}{2}$.

Four mutation operators are actually used: one classical mutation, which randomly perturbs the genotype simulating transcription errors, and three specialized *intelligent* mutation operators, which perform meaningful manipulations on the sentences that compose a linguistic summary. These latter have been called *cover*, *split*, and *merge*.

Classical mutation increases or decreases by one Q, D, and A for all genes with a small probability p_m. If any of those changes ends up violating a constraint, the change is undone.

Cover mutation looks for non-covered time periods and tries to find suitable labels in the temporal hierarchy to cover them. These new labels will be added as the D part in new genes (sentences). As for the rest of genetic material, it is selected by taking the gene with the best accomplishment degree once all possible combinations (Q_{bound_i} and G_{bound_i}) have been tried.

Split mutation looks for a sentence that can be replaced by more than one new sentences using lower-level labels and it splits it accordingly.

Merge mutation does the opposite: it looks for sentences describing adjacent time periods that could be replaced by a single sentence using a higher-level label, and merges them accordingly.

6. Experimentation

In this section we will see the performance of the previously described genetic algorithm using a toy set of data representing the patient inflow along a given year to a certain medical centre.

6.1. Medical centre: patient inflow

Figure 4 represents the patient inflow along a given year to a certain medical centre (365 measures). As we can see, the time dimension is hierarchically organized thanks to three fuzzy partitions of the time domain, namely: one based on approximate months (in order to avoid a strong dependence of the obtained summaries with respect to the crisp boundaries of conventional months) and two others based on a meteorological criteria with two levels of granularity. Fuzziness is specially useful in these two last partitions because transitions between periods are clearly fuzzy. A fuzzy partition of the inflow basic domain with five labels completes the example.

Regarding the linguistic parameters. The quantifiers are *Most of* = {0, 0.8, 0.9, 1}, *At least 80%* = {0, 0.7, 0.8, 1}, *At least 70%* = {0 0.6 0.7 1}, and *At least 60%* = {0 0.5 0.6 1}. The accuracy threshold has been set to $\tau = 0.8$, and the boundaries $Qbound_i = 3$, $Gbound_i = 2$ in all the levels i of the time dimension. As for the evolutionary parameters: the population size and the number of generations are set to 200, $p_c = 0.5$, $p_m = 0.05$, $p_{mi} = 1$ and $\lambda = 0.7$.

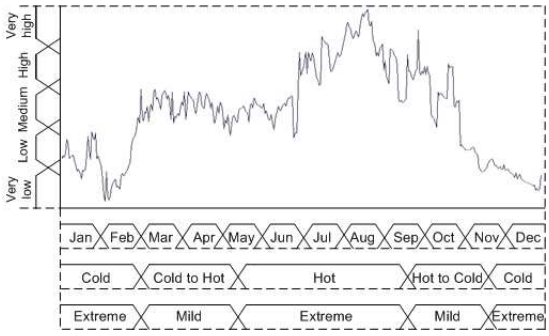

Figure 4: Patient inflow data series.

The selected solution from those in the first Pareto's front is:

- At least 70% of the days with mild weather, the patient inflow is medium or low
- Most of the days in September, the patient inflow is high or medium
- Most of the days with cold weather, the patient inflow is low or very low
- Most of the days in May, the patient inflow is very high or medium
- Most of the days in June, the patient inflow is high or medium
- Most of the days in July, the patient inflow is high or medium
- Most of the days in August, the patient inflow is very high or high

With respect to the objectives representing the quality features, we have that:

Brevity: 14
Accuracy: -0.914807
Coverage: -362

Finally, no constraints have been violated.

Comparison with the greedy results

In this part of the current section we aimed to show the better performance of the genetic algorithm with respect to the greedy one.

The following summary has been obtained using one the algorithm based on the greedy approach. The parameters have been set the same that before except for the specific ones added during the development of the genetic algorithm that do not appear here. This way: the quantifiers are *Most of* = {0, 0.8, 0.9, 1}, *At least 80%* = {0, 0.7, 0.8, 1}, *At least 70%* = {0 0.6 0.7 1}, and *At least 60%* = {0 0.5 0.6 1}, the accuracy threshold has been set to $\tau = 0.8$,

and the boundaries $Qbound_i = 3$, $Gbound_i = 2$ in all the levels i of the time dimension.

- At least 80% of the days with mild weather, the patient inflow is high or medium (1)
- At least 80% of the days with cold weather, the patient inflow is low or very low (1)
- At least 70% of the days with hot weather, the patient inflow is high or medium (0.940756)

In order to compare the summaries properly, we have calculated the values corresponding to the different genetic objectives of the greedy solution (see 5.2). We can see that in Table 1:

Objectives	Genetic	Greedy
Brevity	14	**6**
Accuracy	**-0.914807**	-0.759624
Coverage	**-362**	**-363**

Table 1: Results comparison.

As we can appreciate the brevity is better using the greedy algorithm as well as the coverage. With respect to coverage we have to remark that it is complete in both cases, but we cannot forget that the intersection between periods counts less than 1 to obtain the final value, and the greedy solution have less intersections.

Regarding the accuracy the best value is achieved by the genetic approach. This is because, apart from the degree of accomplishment of the sentences, now we take into account the strictness of the quantifier. Clearly the greedy solution has less strict quantifiers that the genetic one. Aside from that, we can see that the final accuracy value obtained with the greedy algorithm is lower than the established threshold ($0.76 < 0.8$), and this fact violates one of the constraints, meaning that the best solution obtained by the greedy algorithm would have been discarded by the genetic one.

To sum up, we can say that the best solution is that obtained using the genetic algorithm because of the complete coverage, the acceptable brevity and the high accuracy of its sentences, and, of course, because it does not violated any constraints.

7. Conclusion and future work

Linguistic summarization of data is a very interesting data mining task providing highly understandable knowledge. In this work we have proposed an enhancement of a previous proposal [22, 23] which allowed users to linguistically summarized time series data with hierarchical structure of time. We have evolved it from a greedy to an evolutionary approach with the aim of a better exploration of the solution space. We have worked with an adaptation of the multi objective NSGA-II algorithm. Using a multi objective algorithm has permit us to assure the brevity, accuracy and coverage of the final summary. As it has been showed before that the obtained result using the evolutionary approach has a better performance that the one obtained using the greedy approach. We have to take into account that this is a toy example. Better accomplishment of objectives is expected when working with more complex set of data.

Concerning further work, we will continue our research about the semantics of the final summary, and we shall consider time spans of variable length. A very important work will be to make experiments using bigger sets of data and more complex sets, as well as to introduce other characteristics to be summarized. An interesting branch of this research is our plan of using this linguistic summarization technique to describe another data with similar organization, particularly, hierarchically segmented images.

References

[1] I. Z. Batyrshin and T. Sudkamp. Perception based data mining and decision support systems. *Int. J. Approx. Reasoning*, 48(1):1–3, 2008.

[2] I. Z. Batyrshin and L. Sheremetov. Perception-based approach to time series data mining. *Appl. Soft Comput.*, 8(3):1211–1221, 2008.

[3] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8, 1965.

[4] S. Mitra, Senior Member, Fellow, S. K. Pal, and P. Mitra. Data mining in soft computing framework: A survey. *IEEE Transactions on Neural Networks*, 13:3–14, 2001.

[5] G. Chen, Q. Wei, and E. E. Kerre. *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques, Massive Computing Series*, chapter 14 Fuzzy Logic in Discovering Association Rules: An Overview, pages 459–493. Massive Computing Series. Springer, Heidelberg, Germany, 2006.

[6] R. R. Yager. A new approach to the summarization of data. *Information Sciences*, (28):69–86, 1982.

[7] L. A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computing and Mathematics with Applications*, 9(1):149–184, 1983.

[8] R. R. Yager. Toward a language for specifying summarizing statistics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33(2):177–187, 2003.

[9] R. R. Yager. A human directed approach for data summarization. In *IEEE International Conference on Fuzzy Systems*, pages 707–712, 2006.

[10] C. Garrido, N. Marín, and O. Pons. Fuzzy intervals to represent fuzzy valid time in a temporal relational database. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS)*, 17:173–192, 2009.

[11] A. Laurent. A new approach for the generation of fuzzy summaries based on fuzzy multidimensional databases. *Intell. Data Anal.*, 7:155–177, April 2003.

[12] D. Pilarski. Linguistic summarization of databases with quantirius: a reduction algorithm for generated summaries. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 18(3):305–331, 2010.

[13] L. Zhang, Z. Pei, and H. Chen. Extracting fuzzy linguistic summaries based on including degree theory and fca. In *Proceedings of the 12th international Fuzzy Systems Association world congress on Foundations of Fuzzy Logic and Soft Computing*, IFSA '07, pages 273–283, Berlin, Heidelberg, 2007. Springer-Verlag.

[14] R R. Yager and F. E. Petry. A multicriteria approach to data summarization using concept ontologies. *IEEE T. Fuzzy Systems*, 14(6):767–780, 2006.

[15] J. Kacprzyk. Fuzzy logic for linguistic summarization of databases. In *IEEE International Fuzzy Systems Conference*, pages 813–818, 1999.

[16] J. Kacprzyk and R. R. Yager. Linguistic summaries of data using fuzzy logic. In *International Journal of General Systems*, volume 30, pages 133–154, 2001.

[17] J. Kacprzyk and S. Zadrozny. Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools. *Inf. Sci. Inf. Comput. Sci.*, 173(4):281–304, 2005.

[18] J. Kacprzyk and S. Zadrozny. Data mining via protoform based linguistic summaries: Some possible relations to natural language generation. In *CIDM*, pages 217–224, 2009.

[19] J. Kacprzyk, R. R. Yager, and S. Zadrozny. A fuzzy logic based approach to linguistic summaries in databases. *International Journal of Applied Mathematical Computer Science*, 10:813–834, 2000.

[20] I. Kobayashi and N. Okumura. Verbalizing time-series data: With an example of stock price trends. In *IFSA/EUSFLAT Conf.*, pages 234–239, 2009.

[21] D. Chiang, L. R. Chow, and Y. Wang. Mining time series data by a fuzzy linguistic summary system. *Fuzzy Sets Syst.*, 112:419–432, June 2000.

[22] R. Castillo-Ortega, N. Marín, and D. Sánchez. Fuzzy quantification-based linguistic summaries in data cubes with hierarchical fuzzy partition of time dimension. In H. Yin and E. Corchado, editors, *IDEAL'09*, volume 5788 of *LNCS*, pages 578–585. Springer, Heidelberg, 2009.

[23] R. Castillo-Ortega, N. Marín, and D. Sánchez. Linguistic summary-based query answering on data cubes with time dimension. In T. Andreasen, R. R. Yager, H. Bulskov, H. Christiansen, and H. L. Larsen, editors, *FQAS'09*, volume 5822 of *LNAI*, pages 560–571. Springer, Heidelberg, 2009.

[24] R. Castillo-Ortega, N. Marín, and D. Sánchez. A fuzzy approach to the lingusitic summarization of time series. *Journal of Multiple-Valued Logic and Soft Computing, Special Issue Soft Computing Techniques in Data Mining*, 17(2,3):157–182, 2011.

[25] M. Delgado, D. Sánchez, and M.A. Vila. Fuzzy cardinality based evaluation of quantified sentences. *International Journal of Approximate Reasoning*, 23:23–66, 2000.

[26] M. A. Vila, J. C. Cubero, J. M. Medina, and O. Pons. The generalized selection: an alternative way for the quotient operations in fuzzy relational databases. In B. Bouchon-Meunier, R. Yager, and L. Zadeh, editors, *Fuzzy Logic and Soft Computing*. World Scientific Press, 1995.

[27] Kenneth A. DeJong. *Evolutionary Computation: A unified approach*. MIT Press, Cambridge, MA, 2002.

[28] Agoston E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer-Verlag, Berlin, 2003.

[29] Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors. *Evolutionary Multi-Criterion Optimization*, volume 2632 of *LNCS*. Springer-Verlag, Berlin, 2003.

[30] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. pages 849–858. Springer, 2000.