Linguistic summaries for periodicity detection based on mathematical morphology

Gilles Moyse, Marie-Jeanne Lesot, Bernadette Bouchon-Meunier

Linguistic summaries for periodicity detection based on mathematical morphology

Gilles Moyse, Marie-Jeanne Lesot and Bernadette Bouchon-Meunier, *Fellow, IEEE*

***Abstract*—The paper presents a methodology to evaluate the periodicity of a temporal data series, neither relying on assumption about the series form nor requiring expert knowledge to set parameters. It exploits tools from mathematical morphology to compute a periodicity degree and a candidate period, as well as the fuzzy set theory to generate a natural language sentence, improving the result interpretability. Experiments on both artificial and real data illustrate the relevance of the proposed approach.**

***Index Terms*—Linguistic summaries, Mathematical morphology, Temporal data mining, Temporal quantifier, Periodicity computing, Natural language generation**

I. Introduction

LINGUISTIC summaries aim at building human understandable representations of data sets, thanks to natural language sentences and can take different forms representing different kind on patterns [16]. This paper considers this task in the case of time series for which regularity is looked for, more precisely summaries of the form "Regularly, the data take high values". If the data are membership degrees to a fuzzy modality A, the sentence can be interpreted as "regularly, the data are A". Moreover, if the sentence indeed holds, a candidate period is looked for, and an appropriate linguistic formulation is defined, based on the choice of a relevant time unit, approximation and adverb. The final sentence can for instance be "Approximately every 20 hours, the data take high values".

This issue lies at the cross-roads of three main domains, namely linguistic summaries [29], [30], temporal data mining [9], [18], and signal processing [3], as detailed in Section II.

The method proposed in this paper, called Detection of Periodic Events (DPE), does not require assumption about the type of the time series (e.g. a specific generator function) nor expert knowledge to set parameters. It exploits tools from mathematical morphology to compute a periodicity degree and a candidate period, as well as the fuzzy set theory to generate a natural language sentence, improving the result interpretability.

The paper is organized as follows: Section II gives a brief overview of related works. Section III gives an overall view of the proposed methodology for detecting periodic event, and Sections IV, V and VI detail its successive steps, namely data grouping, periodicity computing and linguistic rendering. Lastly, Section VII presents experimental results on both artificial and real data.

Gilles Moyse, Marie-Jeanne Lesot and Bernadette Bouchon-Meunier are with the LIP6 laboratory, UPMC Univ. Paris 06, CNRS UMR 7606, LIP6 4 place Jussieu 75252 Paris cedex 05, France (e-mail: gilles.moyse@lip6.fr).

II. Related Works

This section briefly describes the principles of Linguistic Summaries and Temporal Data Mining, at the crossroads of which the DPE lies. Besides, since DPE is designed to compute periodicities, the main corresponding Signal Processing methods are presented as well.

To the best of our knowledge, DPE is the first approach combining these fields and periodic mining more specifically.

A. Linguistic Summaries

Linguistic summaries aim at building compact representations of given data sets, in the form of natural language sentences describing the main characteristics of the data. They are produced either using fuzzy logic, in which case they are called Fuzzy Linguistic Summaries, or based on Natural Language Generation techniques. Only the former is discussed in this paper; the reader is refered to [5] for a comparison between these two areas.

Fuzzy linguistic summaries, introduced in the seminal papers [29], [30], [15], are built on sentences called "protoforms", such as "QX are A" where Q is a quantifier (e.g. "most", "less than half", or "around 10"), A a linguistic modality associated with one of the attributes (e.g. "young" for the attribute "age" or "tall" for the attribute "height") and X the data to summarize. The relevance of a candidate protoform, measured by the truth degree of its instantiation for the considered data, is the Σ-count of the data set according to the chosen linguistic modality.

In the seminal papers, "QAX are B" is also considered, and its truth degree is computed as the quotient of the Σ-count according to A and B with the Σ-count according to B. Other protoforms can be evaluated [16].

Other extensions define and evaluate protoforms considering the temporal nature of data sets [13], [14]. Trend attributes are computed from the original data set in a pre-processing step, and then included in the summaries, leading for example to "Most slowly decreasing trends are of a very low variability" or "Trends that took most of the time are constant" [14]. Fuzzy Temporal Propositions [6] allow to represent knowledge and rules which are true at certain times, as exemplified by the sentences "Temperature was high in the last 30 minutes" or "Pressure was high a little before temperature was low at some point during the last half an hour". This method also relies on fuzzy logic and allows the computation of a truth degree [6].

B. Temporal Data Mining

Temporal Data Mining groups various issues related to data mining when considering the temporal aspect of the data

(see [9], [18] for exhaustive states of the art). Among the various tasks addressed by this field, we focus here on pattern detection.

Its objective is to discover patterns of interest in the framework of unsupervised learning, i.e. with unlabeled data. To this aim, an extension of the Apriori algorithm for temporal data has been proposed [1]. While these approaches work with databases over different sequences, frequent episodes mining allows to find patterns in one long sequence [22]. In the same context, other algorithms offer the possibility to mine sequential patterns within a given time-window, or having a given duration [19], [23].

On the other hand, rules which are verified on a fixed periodic basis, known as cyclic association rules, are introduced in [25]: the time axis is split into constant length segments, valid association rules are searched on each segment. This approach looking for exactly periodic patterns has been extended to search for partial periodicities, i.e. partially matching patterns, like "A?C", instead of "ABC" [11], [12]. In the latter though, the user has to provide a pattern to the algorithm. Therefore a further extension identifies a list of candidate time periods and patterns to discover, using a Fast Fourier Transform [4]. Yet this method requires the periodic event to be present throughout the data set. Lastly, the p-pattern concept, introduced in [20], extends the partial periodicity to not only accept incomplete patterns but to also allow irregular periodicities, through the use of a Chi-squared test.

Other approaches have been developed, not to mine patterns, but to help working with these methods. For instance, the approach proposed in [2] automatically determines upper and lower thresholds for the algorithm of frequent episodes search [22]. In order to compare the different methods, a benchmark is designed based on the general Bayes error rate, which is known to be the lowest achievable average error rate [7].

C. Signal period measurement

Signal processing is a wide area covering numerous topics, period measurement among others. Different methods have been designed in order to compute the periodicity of a signal [3].

Among them, one consists in approximating the original signal as a sine curve with optimal parameters, best matching the original data [17], [21]. Another relies on the principle of zero-crossing to estimate the period as the gap between two successive zero-crossings with same sign of slope [3].

Other methods more robust to noise have been designed, as those based on the analysis in frequency domain. Some of them are based on signal analysis after a Fast Fourier Transform [26]. In [8], the periodicity is estimated with a combination of spectral analysis and resampling techniques. Lastly, wavelets can also be used to compute the signal's periodicity, as in [24].

The method presented in the next section is simpler these latter two last and proposes a new approach of calculating periodicity with Mathematical Morphology.

III. Detection of Periodic Events (DPE)

The aim of the proposed method is to determine frequent periodic events, based on the computation of a relevance degree for sentences of the form "M every *p unit*, the data take high values", called periodicity degree. It is based on the principle that can be stated as follows: "if a measurement is repeatedly high for an approximately constant duration, and the gaps between these high values have approximately the same size, then the statement "M every *p unit*, the data take high values" holds".

The sentence can be seen as "Q data are high" in a linguistic protoform context where Q quantifies "M every p unit".

This section gives a global overview of the Detection of Periodic Events (DPE) methodology, making its input and output explicit and describing its global architecture. Its components are then detailed in Sections IV to VI.

A. Input

The input is a temporal data set denoted X containing normalized values (x_i), i.e.:

$$X = \{x_i, i = 1, ..., N\} \text{ such that } \forall i \, x_i \in [0, 1]$$

The data are considered to be sampled at regularly spaced dates, i.e. x_i is obtained at date $t_i = t_1 + (i-1) \times \Delta t$ where t_1 is the initial measurement time and Δt is the sampling rate.

The input data can in particular be membership degrees, e.g. resulting from the fuzzification of collected data through a linguistic variable. In this case, the proposed summaries can be interpreted in the protoform framework [13], [14] (see Section II).

B. Output

The proposed DPE method outputs a periodicity degree π, a period p and a describing sentence "M every *p unit*, the data take high values". The latter allows to illustrate the figures, and can be used in an automatic text generation system for instance.

The *periodicity degree* π is a numerical value in the interval [0,1]. It is not a membership to a fuzzy set, so it is not a truth degree as in the fuzzy linguistic summary paradigm. It must be interpreted as a quality measure.

In the sentence, p is the estimated dataset period in a numerical form, M is an adverb like "exactly", "approximately", "roughly", and *unit* is a time unit like "hour", "week", "second".

C. Architecture

The global architecture of the proposed DPE method consists in 3 steps: a first module, described in Section IV, performs data grouping so as to identify groups of consecutive high values and low values. Then, as described in Section V, the groups are processed in a "periodicity computing" step based on mathematical morphology that yields both a periodicity degree and a candidate period. Finally, the result is transformed into a sentence during the linguistic rendering phase detailed in Section VI.

IV. High and Low Value Detection

The first DPE step aims at identifying groups of respectively consecutive high and low values, so as to later estimate the regularity of their sizes. In this section, after briefly presenting a baseline naive method that requires user set parameters, we describe the proposed grouping method based on functional mathematical morphology.

A. Baseline Grouping

The baseline method requires the user to set a threshold t_{value} to define what is a high value. Groups are defined as consecutive values higher than t_{value} with a maximality constraint. More formally, a subset $G_{fl} = \{x_i, i \in [f, l]\}$ is a high value group iff $\forall i \in [f, l]$ $x_i \geq t_{value}$, $x_{f-1} < t_{value}$ and $x_{l+1} < t_{value}$.

In order to make the method more robust, we propose to merge such groups if they are separated by only a few low values: indeed, groups could be split because of noisy low values. Thus the baseline method also requires a second parameter t_{merge} so that groups separated by less than t_{merge} low values are merged.

More formally, if G_{fl} and $G_{f'l'}$ are such that $|f' - l| < t_{merge}$ they are merged with all values between them, leading to the new group $G_{fl'}$.

B. Mathematical Morphology Grouping

The proposed grouping method based on functional mathematical morphology operators does not depend on user-set parameters and automatically adapts to the considered data. We first briefly recall the principles of functional mathematical morphology and then describe its application to high value grouping.

1) Functional Mathematical Morphology Reminder: Mathematical Morphology (hereafter MM) [28] defines a set of tools for the analysis of spatial structures as the shape and size of objects. It has been extensively used for image processing, where these structures are defined as homogeneous regions of the image (e.g. connected sets of pixels with the same color).

Mathematical morphology is based on two basic operators, erosion and dilatation, combined in various ways to define more complex composed operators. In functional MM, given a function $f : X \to Y$ and a *structuring element* B defined as a subset of X of a known shape, e.g. an interval centered at the origin, the *erosion* is the function $\epsilon_B(f) : X \to Y$ defined as:

$$[\epsilon_B(f)](x) = \inf_{b \in B} f(x + b)$$

Dilatation is defined in a similar way, using a sup operator. These two basic operations can be used repeatedly or alternatively, leading to different type of composed operators, such as opening, closing or alternated filters. The interested reader is referred to [28] for a more detailed presentation.

2) Exploitation for High Value Grouping: In the considered application that aims at identifying sets of consecutive high values, f is the function that associates each time stamp with the observed value at this date, i.e. $f(t_i) = x_i$.

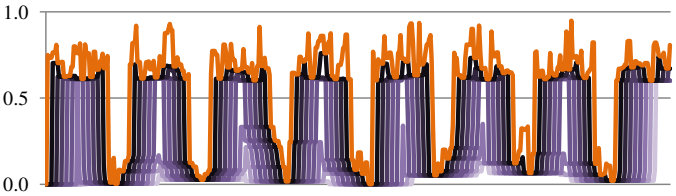

Figure 1. Successive erosions. The original dataset is the outer line in orange, and the successive erosions are the purple lines turning inward.

The chosen operators are successive erosions with a structuring element defined as $B = (-1, 0, 1)$, so as to take into account the previous, current and next time stamp, i.e.:

$$\begin{aligned} [\epsilon_B(f)](t_i) &= \min(f(t_{i-1}), f(t_i), f(t_{i+1})) \\ x_i^1 &= \min(x_{i-1}, x_i, x_{i+1}) \end{aligned} \quad (1)$$

denoting x_i^1 the value obtained after one erosion step. On the edge of the data set, we set $x_1^1 = 0$ and $x_N^1 = 0$. The result of k successive erosions is denoted by x_i^k and x_i^0 is the i^{th} data point from the original dataset.

Since erosion allows to slim the data progressively, we propose to perform a full erosion, that is successive erosions for all $k = 1..z$ until all values equal 0, i.e. $x_i^z = 0$ for all i. Figure. 1 illustrates a data set being eroded.

We exploit the erosion history, computing for each data point the *erosion score*:

$$es_i = \sum_{k=0}^{z} x_i^k \quad (2)$$

As illustrated on Fig. 2, these scores can be used to define a robust method to detect high values, even for highly noisy data: indeed, high values in homogeneous high regions are the last ones to be set to 0, so the ones with high es erosion scores. So it makes it possible to identify high values with an automatic adaptation to the data level, without requiring a user-set threshold as t_{value} in the baseline method. Moreover, it automatically ignores small areas of low values within high value groups, so the user does not have to set a merging threshold, as $t_{merging}$ in the baseline method. Isolated low values, which are considered as noise, are partially ignored since they become zero after a few erosions.

We propose to symmetrically compute erosion scores from the complement of the data, i.e. $\overline{x_i} = 1 - x_i$ for all $i = 1..n$, leading to the score $\overline{es_i}$.

Lastly, groups of high values are automatically defined as sets of consecutive values for which $es_i \geq \overline{es_i}$ with a maximality constraint, and groups of low values conversely.

The proposed mathematical morphology grouping is illustrated in Fig. 2 which shows the considered data set, the erosion scores es and $\overline{es}$ and the limits of the groups.

In the following, we respectively denote G_j^H, $j = 1..n^H$ and G_j^L, $j = 1..n^L$ the identified high and low value groups.

V. Periodicity Computing

Once groups of high and low values have been identified, the periodicity computing step estimates if they alternate in a regular manner, i.e. if a regularity in sizes can be observed.

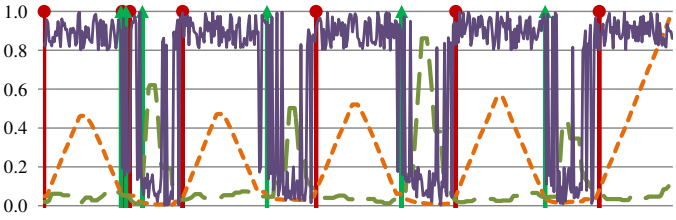

Figure 2. Considered data set X (solid purple line), the erosion scores es (small dash orange line) and $\overline{es}$ (large dash green line) and the group marks (red vertical line topped with a dot representing the beginning of a group and green line topped with a triangle indicating its end).

Indeed, in such a case, it means that the data alternatively and regularly contain high and then low values, thus presenting the periodicity property defined earlier.

In this section we describe the definition of the computed periodicity degree and candidate period.

A. Size Computation

Each group, containing high or low values, is characterized by its size. This size is computed either in a crisp or in a fuzzy way, i.e. using a crisp or a fuzzy cardinality, leading to:

	Crisp size	Fuzzy size
G_j^H	$s_j^H = \sum_{x \in G_j^H} 1$	$\widetilde{s}_j^H = \sum_{x \in G_j^H} x$
G_j^L	$s_j^L = \sum_{x \in G_j^L} 1$	$\widetilde{s}_j^L = \sum_{x \in G_j^L} 1 - x$

These values computed for all groups, respectively for $j = 1..n^H$ and $j = 1..n^L$, define the sets $S^H, \widetilde{S}^H, S^L$ and $\widetilde{S}^L$.

B. Periodicity Degree Computation

As described in the two following subsections, regularity is then computed separately for each type of group (high/low values); the two regularity scores are then merged into the periodicity degree.

1) Regularity Score: For each type of group, the regularity ρ is defined as the opposite of a statistical dispersion measure.

For the dispersion measure, we propose to consider the coefficient of variation CV, defined as the quotient of the mean absolute deviation and the average. Indeed, as a deviation measure, the mean absolute deviation is more robust to noise than the more usual standard deviation (for a detailed discussion, see [10]). Moreover, comparing this mean absolute deviation to the average defines a relative deviation, which makes it possible to adapt to the value level: a deviation of 1 for instance must be considered small when the average is 1,000 but important if the average value is 0.1.

More formally, for a set of sizes s_i, $i = 1..n$, where n can be n^H for high values groups and n^L for low values groups, we define the regularity score as:

$$\rho = 1 - \min(CV, 1) \tag{3}$$

$$CV = \frac{d}{\mu} \qquad \mu = \frac{1}{p}\sum_{j=1}^{n} s_j \qquad d = \frac{1}{p}\sum_{j=1}^{n} |s_j - \mu| \tag{4}$$

The $\min$ in the expression of ρ ensures that the result is in $[0, 1]$. Indeed CV can be greater than 1 when the deviation is important. The theoretical upper bound of CV increases with the number of data and may take very high values, leading to a possibly drastic reduction if used for normalisation. Now CV values higher than 1 mean that the deviation is high, i.e. that no periodicity holds. So using the $\min$ to "cut" CV when it becomes too large is relevant.

Applying the regularity score to each of the 4 possible size types, $S^H, \widetilde{S}^H, S^L$ and $\widetilde{S}^L$, yields 4 regularity measures, respectively denoted ρ^H, $\widetilde{\rho}^H$, ρ^L and $\widetilde{\rho}^L$.

2) Aggregation to the Periodicity Degree: These measures are then aggregated to define the periodicity degree π. The used aggregation operator is the average since it leads to results robust to noise. Moreover, it is close to the principle of periodicity developed in this paper which combines the size regularity scores of the high and low value groups.

Since two ways of calculating the size are used, two periodicity degrees are computed:

$$\pi = \frac{\rho^H + \rho^L}{2} \quad \text{and} \quad \widetilde{\pi} = \frac{\widetilde{\rho}^H + \widetilde{\rho}^L}{2} \tag{5}$$

where μ is defined by (4) for each of the 4 sets $S^H, \widetilde{S}^H, S^L$ and $\widetilde{S}^L$.

C. Candidate Period Computation

Lastly a candidate period can be computed from the average sizes. Indeed, for a perfectly regular phenomenon, the period is defined as the time elapsed between two occurrences of an event. In this paper, the event is "high value".

The period is thus approximated as the sum of the average size of high value groups and the average size of low value groups. As for the calculation of π, since two ways of calculating sizes are proposed, two candidate periods are derived:

$$p = \mu^H + \mu^L \quad \text{and} \quad \widetilde{p} = \widetilde{\mu}^H + \widetilde{\mu}^L \tag{6}$$

It must be underlined that the candidate period is relevant only if the periodicity degree π is high enough.

VI. Linguistic Rendering

To enrich the periodicity degree π and candidate period p, the final linguistic rendering step builds an interpretable sentence, relevant only if π is high enough, as for the candidate period.

A. Considered Issues

The linguistic rendering stage is based on the way human usually express time. In this paper, three interconnected aspects of the time formulation are taken into account: the choice of a relevant time unit so as to avoid periods expressed as very large or very small numbers, the selection of an approximation to favor integers rather than decimals and among these, multiples of 5, and the enrichment with an appropriate adverb to quantify the precision of the approximation, as illustrated below.

Indeed, based on general observations, it seems that speakers prefer using small numbers and thus adapt the used unit. For instance, the statement "I meet her every week" seems

preferable to "I meet her every 168 hours". The usually desired approximation can be illustrated by the fact that one would rather say "This happens every 45 minutes" than "This happens every 44.2 minutes", even though this appreciation is highly dependent on the context in which it is produced. It can happen that a precise time has to be used, e.g. for sportive events or auctions. However, since the present method is not bound to any specific environment, the previous postulate is retained here.

Lastly, the third feature of adverb selection makes it possible to define the level of accuracy of statements. In a context of time expression, it can be a linguistic expression as "exactly", "approximately", "nearly", "roughly", "around". For instance, one would not say "the game lasted 1 hour and 7 minutes", but rather "the game lasted approximately 1 hour", adding the adverb to indicate that this period is not exact.

The aim of the linguistic rendering step is to generate a sentence taking these constraints into account. The format of the resulting formulation is "*Adverb* every *approximatePeriod unit*, the data take high values". It is generated in three steps: unit determination, period approximation and adverb selection.

B. Unit Determination

In order to find the most appropriate representation, a set of considered units is defined, for example {*seconds*, *minutes*, *hours*, *days*, *weeks*}. The equivalence between them is also specified ($60\,\mathrm{min} = 1\,\mathrm{h}$, $24\,\mathrm{h} = 1\,\mathrm{day}$...). Lastly, the interval in which the value is to be expressed is fixed. Setting it to [1,60] for instance imposes the period to be represented as a number between 1 and 60.

The candidate period computed in Section V-C is then expressed in all possible units. The unit leading to a number in the desired interval is retained. In the case where two units may apply, then the smallest value is selected.

It must be remarked that the interval must be chosen so that all period values have an expression. For instance, if [1,10] is picked, then $45\,\mathrm{min}$ can not be formulated, since it is too large to be expressed in *minutes*, and too small to be expressed in *hours*. This is why the lower bound should be set to 1, and the upper bound to the largest conversion factor, here 60.

For instance, if the computed candidate period is $3{,}708\,\mathrm{s}$, then the conversions are $3{,}708\,\mathrm{s} = 61.80\,\mathrm{m} = 1.03\,\mathrm{h} = 0.04\,\mathrm{d} = 0.006\,\mathrm{w}$. As 1.03 is the only value in [1,60], the selected unit is *h*.

C. Period approximation

The obtained expression of the candidate period is then approximated to a value that must be a compromise between a natural, user-friendly value, and a relevant approximation.

As illustrated with the previous examples, it is assumed that a user-friendly representation of time is usually an integer, multiple of 5 if possible. The relevance of the approximation is determined from a user-set linguistic threshold t_{ling} defining the maximal acceptable difference between the initial and the approximated values.

Thus to find the best suitable representation, the value is first rounded to the nearest multiple of 5. If the difference with the initial value is greater than t_{ling}, the value is rounded to its integer value. If the difference is still greater than t_{ling}, then the value is rounded to one or more decimals. Formally, the approximated period p_{ling} used for the generation of the sentence is computed as

$$p_{ling} = \begin{cases} RM\left(p,5\right) \text{ if } \left|RM\left(p,5\right)-p\right|/p < t_{ling} \\ \min\limits_{d\in\{0,\ldots,d_{max}\}} \left|R\left(p,d\right)-p\right|/p < t_{ling} \text{ otherwise} \end{cases}$$

where $RM(p,5)$ is the rounding of p to the nearest multiple of 5, $R(p,d)$ the rounding to d decimals, and $R(p,0)$ the rounding to the nearest integer. The objective is thus to find the smallest decimal rounding verifying the acceptability condition. d can be up to d_{max} which is the maximum allowed number of decimals. It directly depends on t_{ling} and can be defined as $d_{max} = \lceil |\log\left(t_{ling}\right)| \rceil$.

Considering the previous example of $1.03\,\mathrm{h}$, and using $t_{ling} = 5\%$, one gets $p_{ling} = 1\,\mathrm{h}$ since $RM(1.03,5) = 1$ and $|1-1.03|/1.03 = 0.03 < t_{ling}$.

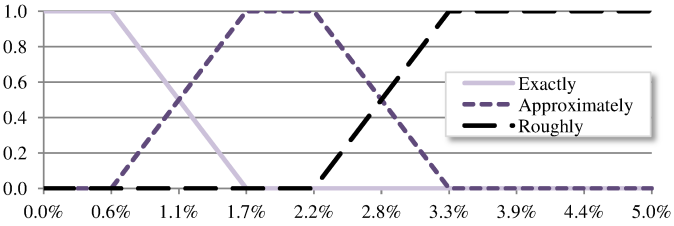

Figure 3. The linguistic variable *Precision* with $t_{ling} = 5\%$.

D. Adverb Selection

The last step of linguistic rendering aims at selecting an adverb describing the quality of the approximation to enrich the period formulation. This adverb M is a modifier based on the linguistic variable "Precision" illustrated in Fig. 3.

The error err made during the previous period approximation and computed as:

$$err = \frac{|p - p_{ling}|}{p} \tag{7}$$

The most relevant modality, denoted by m^*, is computed as the one to which err belongs the most, i.e.

$$M = \underset{m\in Precision}{\operatorname{argmax}} \ \mu_m\left(err\right) \tag{8}$$

With the previous example, $p = 1.03\,\mathrm{h}$ and $p_{Ling} = 1.00\,\mathrm{h}$, so $err = 3.0\%$. Therefore $\mu_{Exactly}\left(err\right) = 0$, $\mu_{Approximately}\left(err\right) = 0.3$ and $\mu_{Roughly} = 0.7$. As a result, $M = Roughly$ and the final sentence is "Roughly each hour the data take high values".

VII. Experimental Results

This section presents experimental results obtained both with artificial and real data. The aim is to compare the proposed methods, to study their behaviors, and to evaluate the quality of the generated sentences.

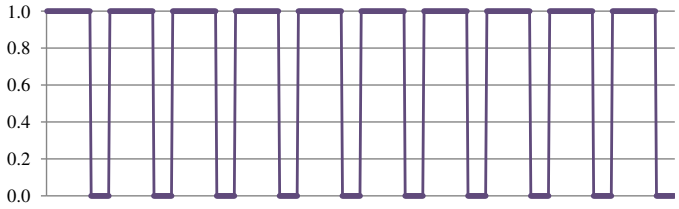

Figure 4. No noise data set, $\nu_x = 0$, $\nu_y = 0$, $p^H = 35$ and $p^L = 15$.

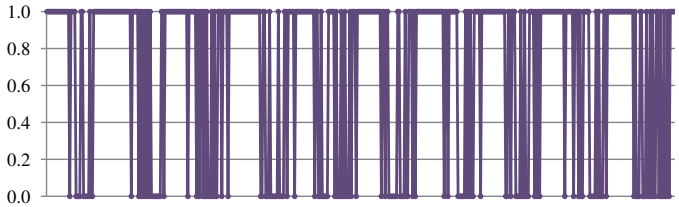

Figure 5. Data set with noise on the time axis, $\nu_x = 0.5$, $\nu_y = 0$, $p^H = 35$ and $p^L = 15$.

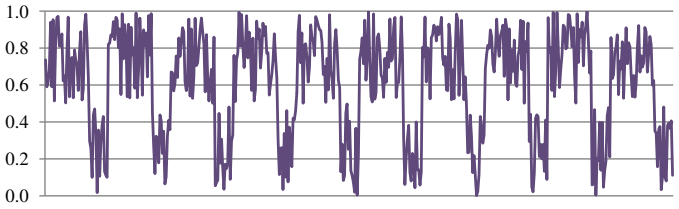

Figure 6. Data set with noise on the value axis, $\nu_x = 0$, $\nu_y = 0.5$, $p^H = 35$ and $p^L = 15$.

Table I
EXPERIMENTAL SETUP

	Test code		Parameters		
	Crisp	Fuzzy	t_{value}	t_{merge}	t_{ling}
Baseline	B	~B	0.8	2%	5%
MM	M	~M	-	-	5%

A. Artificial data

1) Data Generation: The data are generated as series of noisy periodic "rectangles", where noise applies either to the period (time axis) or to the values (value axis). The rectangles high values are p^H long, and the low ones are p^L long.

Therefore a noiseless signal has a period $p = p^H + p^L$ and can be generated as $\hat{x}_i^* = 1_{[0;p^H[}\left((i-1) \mod (p^H + p^L)\right)$.

Given two noise parameters ν_x and ν_y, a noisy series is generated as:

$$\begin{aligned} \hat{x}_i &= 1_{[0;p^H[}\left((i - 1 + \nu_x\epsilon_t) \mod (p^H + p^L)\right) \\ y_i &= \begin{cases} \nu_y\epsilon_v & \text{if } \hat{x}_i < 0.5 \\ -\nu_y\epsilon_v & \text{otherwise} \end{cases} \\ x_i &= \min\left(\max\left(\hat{x}_i + y_i, 0\right), 1\right) \end{aligned}$$

where $\epsilon_t \sim \mathcal{U}(0, p)$ and $\epsilon_v \sim \mathcal{U}(0, 1)$, denoting $\mathcal{U}$ the uniform distribution.

$\hat{x}_i$ is obtained by randomly shifting the noiseless signal on the time axis, thus creating high values in low values groups and vice versa as compared to the noiseless signal $\hat{x}_i^*$.

y_i adds vertical noise to the rectangle signal, downward if the signal is in its high part, and upward otherwise. The min/max transformation applied to $\hat{x}_i + y_i$ ensures final values in [0,1] for x_i.

When $\nu_x = \nu_y = 0$, then $y_i = 0$ and x_i describes a noiseless rectangle signal; when $\nu_x \neq 0$ and $\nu_y = 0$, the period is randomly changed, but $x_i \in \{0, 1\}$ holds; when $\nu_x = 0$ and $\nu_y \neq 0$, the period is not changed but $x_i \in \{0, 1\}$ does not hold, and only $x_i \in [0, 1]$ does. Fig. 4 to 6 show examples of such data with different noise levels.

2) Experimental Protocol: 40 data series are generated with an increasing noise value from 0 to 1 at a 0.05 pace (21 values) with $p^H = 35$ and $p^L = 15$. The periodicity degree π, the candidate period p_C, and the error in period evaluation Δp are computed. Δp is defined as $\Delta p = |p_C - p| / p$.

They are computed with 4 methods, according to the grouping method, that can be *baseline* or *mathematical morphology* based, and to the cardinality definition, that can be *crisp* or *fuzzy*. The notations and values of the parameters are displayed in Table I. The linguistic rendering parameters (considered units, expression interval and modifier) are described in Section VI.

The average and standard deviation of π is computed on 40 runs with increasing time noise ν_x (Fig. 7) and value noise ν_y (Fig. 8). The average and standard deviation of Δp is computed on 40 runs with increasing time noise ν_x (Fig. 9) and value noise ν_y (Fig. 10). Lastly, some examples of the generated sentences along with the computed period are presented in Table II.

3) Result Interpretation:

a) Periodicity Degree: It can be observed from Fig. 7 and Fig. 8, that in all cases when the noise parameters are 0, the periodicity degree equals 1. Moreover it globally decreases when noise increases, presenting the expected behavior.

Yet, for $\nu_x \geq 0.3$ the baseline method produces a periodicity degree equal 0: for such noise levels, illustrated e.g. on Fig. 5, the method fails to identify relevant groups and actually merges them all. This is due to the fact that the groups of low values are 15 points wide, and the t_{merge} parameter used in this grouping is 10 points wide. Thus, groups tend to be merged excessively, leading to a disruption in the groups construction. The following regularity assessment step then fails to give relevant results. This raises the issue of the parameter selection for the baseline approach, that requires expert knowledge about the data. On the contrary, the MM grouping method decreases as expected.

Regarding increasing noise in values ν_y (Fig. 8), it can be seen that the baseline grouping has less amplitude than the MM approach. More precisely, it makes no difference as long as $\nu_y < 1 - t_{value}$. Indeed, all values above $1 - t_{value} = 0.2$ are considered as high. On the contrary, the MM method automatically adapts to the data characteristics, and starts decreasing as soon as noise is detected.

It can also be observed that the group size definition, crisp or fuzzy, makes no significant difference for a given grouping method.

b) Periodicity Evaluation: It can be noted on Fig. 9 and 10 that in all cases the error in the estimation of the period increases with noise. The baseline method leads to higher

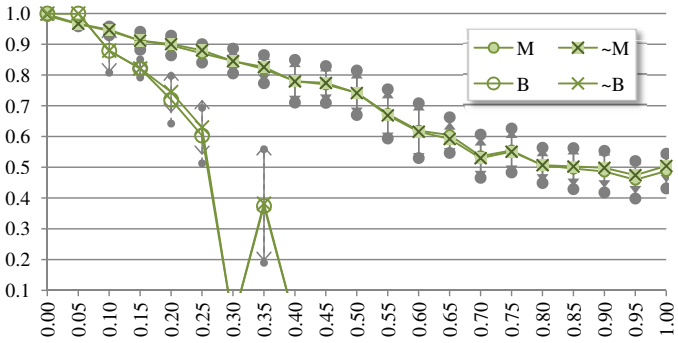

Figure 7. Periodicity degree π with increasing ν_x for $\nu_y = 0$.

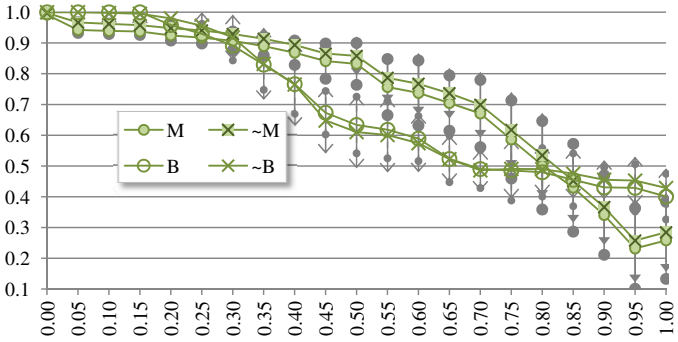

Figure 8. Periodicity degree π with increasing ν_y for $\nu_x = 0$.

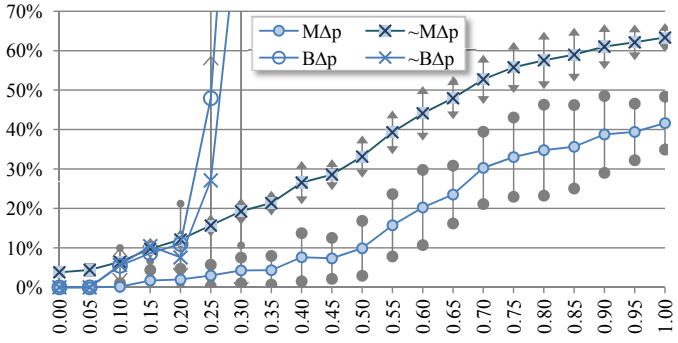

Figure 9. Error in period evaluation Δp with increasing ν_x for $\nu_y = 0$.

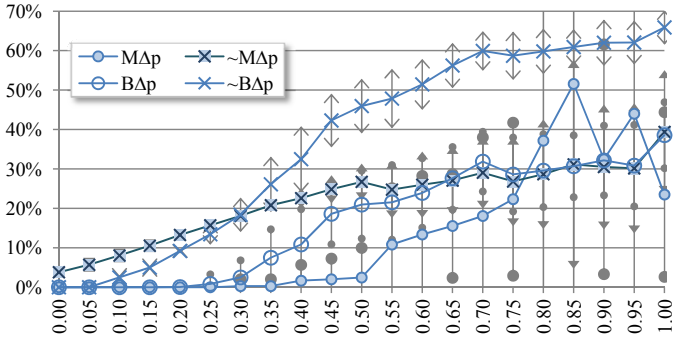

Figure 10. Error in period evaluation Δp with increasing ν_y for $\nu_x = 0$..

errors, which is due to the fact that the groups it identifies are erroneous. Whether noise applies to time or values, the MM method leads to errors lower than 5% for noise levels lower than 0.25, which is a very satisfying result. High errors for high noise levels are not an issue, as in such cases the periodicity degree is low, indicating that the candidate period is not relevant.

It can also be observed that a significant difference appears between the different cardinalities: the crisp one returns significantly lower errors than the fuzzy one. This is due to the fact that the fuzzy cardinality takes into account the values whereas the period is computed only on time axis, whatever the values. So the crisp approximation made by ignoring the actual values of the groups is relevant in this case.

B. Real Data

1) Data Presentation: The RATP is the main public transport operator company in Paris and it monitors the quality of the air underground. It released measurements from several sensors in different metro stations on an hourly basis from Jan, 1^{st} 2012 to Apr, 1^{st} 2012, available on its website [27].

For the test, we use one week of normalized amount of CO_2 in the station Châtelet, from Jan, 16^{th} 2012 to Jan, 21^{st} 2012. Fig. 11 show the dataset used for the tests.

Visual inspection indicates that the data are indeed periodic, matching the a priori knowledge of public transportation: there are two major rush hours in the tube, a smaller one around 9am and a larger one around 6pm.

2) Experimental Protocol: We apply the proposed DPE method, setting the baseline parameters as $t_{value} = 0.7$ and $t_{merge} = 10$. The linguistic rendering parameters are the ones described in Section VI. Table III lists the obtained results.

3) Result Interpretation: It can be observed that all methods compute a high periodicity degree, the highest one being obtained for the MM grouping with fuzzy cardinality. These high degrees are compatible with the expectations.

Regarding the period evaluation, the crisp MM approach returns the expected result, indicating a period of 24.2h, linguistically rendered as “exactly one day”.

These results are compatible with the observations made on artificial data, showing that a crisp evaluation is more appropriate for the period evaluation and that the baseline grouping method is less relevant than the MM approach for the periodicity degree.

VIII. Conclusions and Future Work

The proposed DPE method for the generation of linguistic summaries of the form “M every *p unit*, the data take high values” where M is a precision adverb, exploits tools from mathematical morphology to identify groups of high values in an automatically adaptive manner that does not require assumption about the data series or knowledge about its characteristics to set parameters. It also exploits the fuzzy set theory to generate a relevant linguistic formulation illustrating the numerical candidate period. Experiments performed on artificial and real data illustrate the relevance of the proposed DPE method.

Future works aim at generalizing the experimental analysis, in particular considering other types of data series, and defining a quality measure to compare the different methods, among themselves as well as to existing approaches, especially

Table III
RESULTS WITH REAL DATA WITH BEST RESULTS IN BOLD

Method	π	Period	Generated sentence
B	0.73	20.60 h	The period is approximately 20 hours.
~B	0.72	17.05 h	The period is exactly 17 hours.
M	0.82	24.20 h	**The period is exactly 1 day.**
~M	**0.86**	17.05 h	The period is exactly 17 hours.

Table II
LINGUISTIC GENERATION RESULTS

Candidate period	Generated sentence
62.50 min	The period is approximately 1 hour.
42.60 min	The period is roughly 45 minutes.
58.17 s	The period is approximately 1 minute.
55.44 s	The period is exactly 55 seconds.
49.15 h	The period is approximately 2 days.

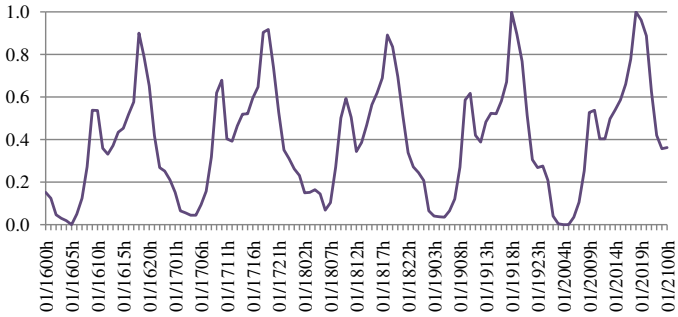

Figure 11. Quantity of CO_2 per hour from 01/16/2012 to 01/21/2012 in the station Châtelet.

those from the signal processing area. Furthermore, new fuzzy quantifiers as "from time to time", "often", "rarely" can be defined with this method. Using other mathematical morphology operators like watershed or SKIZ should also be considered. Finally the detection of period over a subset of data would be interesting as well.

REFERENCES

[1] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. of the 11th Int. Conf. on Data Engineering*. IEEE Comput. Soc. Press, 1995, pp. 3–14.

[2] M. J. Atallah, R. Gwadera, and W. Szpankowski, "Detection of Significant Sets of Episodes in Event Sequences," in *4th IEEE Int. Conf. on Data Mining (ICDM'04)*. IEEE, 2004, pp. 3–10.

[3] V. Backmutsky, J. Blaska, and M. Sedlacek, "Methods of finding actual signal period time," in *IMEKO XVI World Congress*, Austria, 2000, pp. 243–248.

[4] C. Berberidis, I. P. Vlahavas, W. G. Aref, M. J. Atallah, and A. K. Elmagarmid, "On the Discovery of Weak Periodicities in Large Time Series," in *Proc. of the 6th European Conf. on Principles of Data Mining and Knowledge Discovery PKDD '02*. Springer-Verlag London, UK, 2002, pp. 51–61.

[5] B. Bouchon-Meunier and G. Moyse, "Fuzzy Linguistic Summaries: Where Are We, Where Can We Go?" in *CIFEr 2012*, 2012, pp. 317–324.

[6] P. Cariñena, A. Bugarín, M. Mucientes, and S. Barro Ameneiro, "A language for expressing fuzzy temporal rules," *Mathware & soft computing*, vol. 7, no. 2, pp. 213–227, 2000.

[7] D. Chudova and P. Smyth, "Pattern discovery in sequences under a Markov assumption," in *Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining - KDD '02*. ACM Press, 2002, pp. 153–162.

[8] M. J. Costa, B. Finkenstädt, P. D. Gould, J. Foreman, K. J. Halliday, A. J. W. Hall, and D. A. Rand, "Estimating periodicity of oscillatory time series through resampling techniques." University of Warwick. Centre for Research in Statistical Methodology, Tech. Rep., 2011.

[9] T.-C. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[10] S. Gorard, "Revisiting a 90-year-old debate: the advantages of the mean deviation," *British Journal of Educational Studies*, vol. 53, no. 4, pp. 417–430, Dec. 2005.

[11] J. Han, G. Dong, and Y. Yin, "Efficient mining of partial periodic patterns in time series database," in *Proc. of the 15th Int. Conf. on Data Engineering*. IEEE, 1999, pp. 106–115.

[12] J. Han, W. Gong, and Y. Yin, "Mining Segment-Wise Periodic Patterns in Time-Related Databases," *Proc. Int. Conf. on Knowledge Discovery and Data Mining*, pp. 214 – 218, 1998.

[13] J. Kacprzyk, A. Wilbik, and S. Zadrozny, "Linguistic Summaries of Time Series via a Quantifier Based Aggregation Using the Sugeno Integral," in *IEEE Int. Conf. on Fuzzy Systems*. IEEE, 2006, pp. 713–719.

[14] ——, "Linguistic summarization of time series using a fuzzy quantifier driven aggregation," *Fuzzy Sets and Systems*, vol. 159, no. 12, pp. 1485–1499, Jun. 2008.

[15] J. Kacprzyk and R. R. Yager, ""Softer" optimization and control models via fuzzy linguistic quantifiers," *Information Sciences*, vol. 34, no. 2, pp. 157–178, 1984.

[16] J. Kacprzyk and S. Zadrozny, "Protoforms of Linguistic Data Summaries: Towards More General Natural-Language-Based Data Mining Tools," in *Soft computing systems*, A. Abraham, J. Ruiz-del Solar, and M. Koeppen, Eds., 2002, pp. 417–425.

[17] J. Lafler and T. D. Kinman, "The Calculation of RR Lyrae Periods by Electronic Computer," *The Astrophysical Journal Supplement Series*, vol. 11, p. 216, 1965.

[18] S. Laxman and P. S. Sastry, "A survey of temporal data mining," *Sadhana*, vol. 31, no. 2, pp. 173–198, 2006.

[19] C.-H. Lee, M.-S. Chen, and C.-R. Lin, "Progressive partition miner: An efficient algorithm for mining general temporal association rules," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 1004–1017, 2003.

[20] S. Ma and J. L. Hellerstein, "Mining partially periodic event patterns with unknown periods," in *Proceedings 17th International Conference on Data Engineering*. IEEE Comput. Soc, 2001, pp. 205–214.

[21] M. K. Mahmood, J. E. Allos, and M. A. H. Abdul-Karim, "Microprocessor Implementation of a Fast and Simultaneous Amplitude and Frequency Detector for Sinusoidal Signals," *IEEE Transactions on Instrumentation and Measurement*, vol. 34, no. 3, pp. 413–417, 1985.

[22] H. Mannila, H. Toivonen, and A. Inkeri Verkamo, "Discovery of Frequent Episodes in Event Sequences," *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 259–289, 1997.

[23] N. Méger and C. Rigotti, "Constraint-based mining of episode rules and optimal window sizes," in *Proc. of the 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases*. Springer-Verlag, 2004, pp. 313–324.

[24] P. K. G. Narayanan, A. Ortega, and S. N. Shrikanth, "Pitch period estimation using multipulse model and wavelet transform," in *INTERSPEECH'07*, 2007, pp. 2761–2764.

[25] B. Ozden, S. Ramaswamy, and A. Silberschatz, "Cyclic association rules," in *Proc. of the 14th Int. Conf. on Data Engineering*. IEEE Comput. Soc, 1998, pp. 412–421.

[26] J. D. Plautz, M. Straume, R. Stanewsky, C. F. Jamison, C. Brandes, H. B. Dowse, J. C. Hall, and S. A. Kay, "Quantitative Analysis of Drosophila period Gene Transcription in Living Animals," *Journal of Biological Rhythms*, vol. 12, no. 3, pp. 204–217, 1997.

[27] RATP, "Qualité de l'air mesurée dans nos stations (T1 2012)," Paris, 2012. [Online]. Available: http://data.ratp.fr/

[28] J. Serra, "Introduction to mathematical morphology," *Computer Vision, Graphics, and Image Processing*, vol. 35, no. 3, pp. 283–305, 1986.

[29] R. R. Yager, "A new approach to the summarization of data," *Information Sciences*, vol. 28, no. 1, pp. 69–86, 1982.

[30] L. A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages," *Computers & Mathematics with Applications*, vol. 9, no. 1, pp. 149–184, 1983.