Linguistic summaries of locally periodic time series

Gilles Moyse, Marie-Jeanne Lesot

Linguistic summaries of locally periodic time series

Gilles Moyse[a,b,*], Marie-Jeanne Lesot[a,b]

[a] *Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France*
[b] *CNRS, UMR 7606, LIP6, F-75005, Paris, France*

Abstract

This paper proposes a method to linguistically summarise the local periodic components of a time series: it identifies subparts of the data which are periodic, together with their periodicity degree and period, and provides a linguistic description thereof. The generated sentences can be illustrated by the example "Approximately from March to June, the series is highly periodic with a period of exactly 2 weeks". The method proposed to identify local periodic zones relies on the determination of relevant auto-adaptive windows, based on an analytical expression of the probability distribution of the considered periodicity criterion. The linguistic description generation, in the protoform approach framework, expresses three core aspects of the identified periodic intervals, namely their time context or localisation in time, their periodicity and their period. Intensive experiments performed on both artificial and real data validate the proposed method.

Keywords: Local periodicity, Contextualisation, Linguistic summary, Time series

1. Introduction

Linguistic summaries offer a compact and user-friendly way of representing large amounts of data [1, 2, 3]. In the case where the considered data are time series, the linguistic summarisation task brings specific questions due to their temporal nature. Furthermore, the properties of such series are often changing over time, which implies that the knowledge extracted from a series at a certain time may be wrong at the next one, and hence needs to be contextualised.

Among the various information conveyed in a time series, periodicity is an important one, frequently used in fields as diverse as astronomy [4], physics [5], energy production [6], speech analysis [7], zoology [8] or biology [9]. Moreover, the question of local periodicity occurs in many applicative contexts [10, 11, 12, 13].

This paper proposes a method allowing to simultaneously address the two issues of contextualisation and periodicity: it aims at identifying local periodicity, defined as the occurrence in the time series of periodic patterns located in a specific temporal zone.

*Corresponding author
Email address: name.surname@lip6.fr (Gilles Moyse)

The method proposed in this paper is called Local Detection of Periodic Events (LDPE). It detects and characterises the periodic zones in terms of periodicity and period, locates them in time and renders this local periodic information into human readable sentences. It is based on the Detection of Periodic Events (DPE) method [14], and brings two important improvements: an auto-adaptive window, based on a probability distribution allowing to automatically adapt a periodicity computation window to the data, and a set of new linguistic rendering features to generate sentences such as "Approximately from March to June, the series is highly periodic with a period of exactly 2 weeks".

In Section 2, existing works related to both the issues of linguistic summarisation and periodicity detection are presented. The DPE method is detailed in Section 3. The different steps of periodicity contextualisation of LDPE, defined as a generalisation of DPE, are presented in Section 4. Section 5 is dedicated to the linguistic rendering issue raised by such local periodic events and presents formulations to enrich existing models and, in particular, to add new time localisation features. The experiments both on artificial and real data and the obtained results that validate our approach are presented in Section 6. Finally, a conclusion to this work and future works are given in Section 7.

2. Related works

The question of detecting periodic parts of a times series lies at the crossroad of several domains, namely linguistic data summarisation and periodicity detection. Some of the existing works in each of these two domains are successively presented in the next subsections.

2.1. Linguistic summarisation

Two sides of linguistic summarisation are presented in this subsection: first, the general summaries and second, the temporal ones.

General summaries. Linguistic summaries aim at building compact representations of datasets, in the form of natural language sentences describing their main characteristics. Two principal approaches exist to this aim [15, 16], one using fuzzy protoforms and one based on natural language generation (NLG).

Fuzzy Linguistic Summaries (FLS), introduced in the seminal papers [1, 2, 17], are built on "protoforms" whose basic form is defined as "QX are A" where Q is a quantifier (e.g. "most" or "around 10"), A a linguistic modality associated with one of the attributes (e.g. "young" for the attribute "age") and X the data to summarise. Numerous criteria to evaluate the relevance of a candidate protoform instantiation have been proposed. The first and maybe most important one is the truth degree, proposed in the seminal paper [1]: it measures the extent to which the data coincides with the considered summary, based on the Σ-count of the dataset according to the chosen fuzzy modality.

Other criteria include the degree of focus [18], which measures the support of a given attribute in the database, and the degree of appropriateness [19] which quantifies the extent to which a summary is "surprising". Compound measures combine several criteria such as fulfilment, relevance, length, coverage, specificity, compatibility and non ambiguity of a summary [20] or its coverage, brevity, specificity and accuracy [21].

In the NLG framework, several approaches have been developed as well, more focused on sentences generation than on data extraction. Indeed, such systems, as EasyText performing polls analysis [22], are based on a user-defined set of rules used against a database to generate the result sentences. Since the method proposed in this paper is more oriented toward data analysis, NLG techniques are not further investigated.

Temporal summaries. FLS coping with the temporal dimension of time series have also been considered, as for instance specific protoforms expressing duration or time localisation of specific events: they can e.g. be based on fuzzy temporal protoform grammar [23] or on a specific hierarchical time scale [24].

Additionally, the extraction of temporal features from time series, such as trends, in order to summarise them later using standard FLS, has also been proposed [25]. Another kind of extracted temporal feature such as the exceptional character of an event in time compared to a reference value is detected and included in a FLS in [26].

The DPE method [14] and its variants [27] focus on the issue of periodicity, not taken into account in the previous approaches. They propose linguistic summaries expressing this periodicity as well as the period of a time series, in a human friendly way. The DPE method, on which the approach proposed in this paper relies, is presented in further details in Section 3.

2.2. Period detection

Period detection is a well-known problem in signal processing and numerous methods have been proposed to address it. They can be classified into four categories, depending on the series representation, namely distinguishing time, frequency, time-frequency and symbolic representations.

Time representation. The time representation is the most straightforward one: it considers the signal in its original form, i.e. as the successive values x_i associated with their timestamps, t_i, i.e. $\{(t_i, x_i), i \in [1, n]\}$, where n denotes the total length of the series.

The most common approach for periodicity detection in time domain is autocorrelation [28], which yields good results mostly with sine and stationary signals. Another approach relies on the analysis of zero-crossing of the data [29]. However, it has been proved very sensitive to noise [30].

Other approaches with time representation have been proposed [6]. Among them, the evaluation of the signal as a sine wave based on 3 consecutive points [31] or the direct computation of the frequency based on the second derivative of the signal [32] are proposed. They nonetheless are also sensitive to noise and designed to work in specific contexts.

A quasi-periodic detection method based on a Fuzzy Finite State Machine is proposed in [33]. Due to the specific configuration of the state machine in order to model the human gait studied in the paper, this solution is rather specialised, whereas the scheme proposed in this paper aims at being as general as possible.

Frequency representation. Another way to study the signal is to represent it against its constituent frequencies. This is most commonly achieved with the Fourier transform which allows to convert a signal into a periodogram, yielding the power associated to each frequency in the original signal and allowing to identify the main frequency as the most powerful one [34].

However, in the realm of discrete signals used in the digital context, the power values are associated to some frequencies only and the main frequency is usually not among them. Several methods circumventing this limitation have been proposed in order to estimate the real frequency [35, 36, 37, 38] by combining and weighting the most powerful frequency with the ones in its neighbourhood.

Nonetheless, these approximations rely on the supposed stationarity of the signal, i.e. the fact that each data point is instantiated from the same random variable throughout the measurement process, thus excluding data with evolving or different periods at different times.

Moreover, since the Fourier transform is a decomposition of the original signal on trigonometric basis functions, it is efficient only with signals made of sines. Otherwise, the decomposition yields bad results, as the Gibbs effect shows.

The Cepstrum analysis [7], based on the Fourier transform, suffers from the same biases.

Time frequency representation. Instead of studying the data either in the time or in the frequency domain, and in order to overcome some of the limitations associated with these approaches, the time-frequency representation has been proposed. Its main idea is to perform a frequency decomposition over small parts of the data in time and to associate the obtained results with the time interval of the analysed data.

The first proposed time frequency representation is the Short Fourier Transform [39, 40], whose principle is to compute the Fourier transforms of the convolution of the signal with a window moving from the beginning to the end of the data and smaller than the data. However, this transform locally suffers from the same problems as the Fourier one over the analysed parts of the signal. Moreover, a specific window must be chosen, mainly in terms of shape and size.

A more sophisticated approach relies on wavelets [41, 42] where the basis functions are not trigonometric as in the Fourier transform but specific ones called wavelets which are stretched and translated in order to study the signal at different times and on different scales. Even though the wavelet approach does not need a window to be defined, it still requires a wavelet to be chosen.

More recently, the Hilbert-Huang Transform has been proposed [43]. It is based on the Empirical Mode Decomposition, which is an algorithm allowing the decomposition of the signal into functions, as in the Fourier or the Wavelet transform. However, it does not clearly outperforms previous approaches as wavelets [44] and is still empirical [45] but suggests interesting developments as shown by its numerous applications [46, 47].

Hybrid approaches using several of the techniques listed above have been developed for period estimation. In [48], a Fourier transform is used to extract the most powerful frequencies, which are then validated or discarded with an autocorrelation analysis. However,

the method suffers from the same limitations as the techniques it is based on.

In [49] a wavelet decomposition followed by a SVM-based classification technique is used in order to keep base wavelet components, reject noisy ones and estimate the signal period, yielding however a complex method.

Symbolic representation. Symbolic representations are obtained form the transformation of the initial time series into a limited number of symbols. Several of these methods also perform a dimensionality reduction, as Piecewise Component Analysis [50] and its Adaptive version APCA [51] or using Chebyshev polynomials [52].

Symbolic representations allow to use specific tools e.g. from the association rules domain, as the extraction of cyclic association rules satisfied on a fixed periodic basis [53], or methods based on frequent pattern discovery [54, 55, 56]. However, these methods either need to be run with a given candidate period or seek specific patterns, which is not relevant in the context of determining arbitrary periodicity in a time series without an a priori model.

To overcome the issue of given candidate periods as input of the method, a Fourier transform can be used beforehand [57]. However, this approach conveys similar problems as the ones using the Fourier transform mentioned above.

Another approach computing periodicity based on events given as a symbolic time series is proposed in [58]. It uses a χ^2 test on the inter-event distance and returns a periodic behaviour if these distances are not significantly different. Nevertheless, it needs several thresholds to be set up previously, as well as an indexing of "interesting" and "non interesting" zones.

The Detection of Periodic Events (DPE) [14], described in the next section, also uses a symbolic representation: it decomposes the series as the alternation of high and low value groups. It has the advantages of not requiring any parameter as well as being oriented toward linguistic rendering from its very definition. Moreover, several optimisations have been proposed so as to enable its fast computation [59].

However, the DPE method only applies to time series considered as a whole. In this paper, we propose to contextualise it to cases where only subsets of the series are periodic.

3. DPE: Detection of Periodic Events

The aim of the DPE method is to provide linguistic summaries of time series focused on their periodicity, as illustrated in Section 5. It follows the principle according to which *it considers a time series as periodic if it alternates, in a regular manner, groups of respectively high and low values, where regularity is defined in terms of the group sizes* [14]. Its main computational steps are recalled in the following paragraphs.

DPE takes as input a temporal dataset, denoted X, containing normalised values x_i: $X = \{x_i,\, i = 1...n\}$ such that $\forall i,\, x_i \in [0, 1]$ and the two bounds are attained, i.e. $\exists l$ s.t. $x_l = 0$ and $\exists m$ s.t. $x_m = 1$. It returns a periodicity degree π, a period p and a descriptive sentence instantiating the protoform "*Prec* every p *units*, the values are high". The periodicity degree π is a quality measure in $[0, 1]$ conveying the extent to which the dataset is periodic, 1 meaning perfect periodicity. The period p is the estimated period. *Prec* is a precision

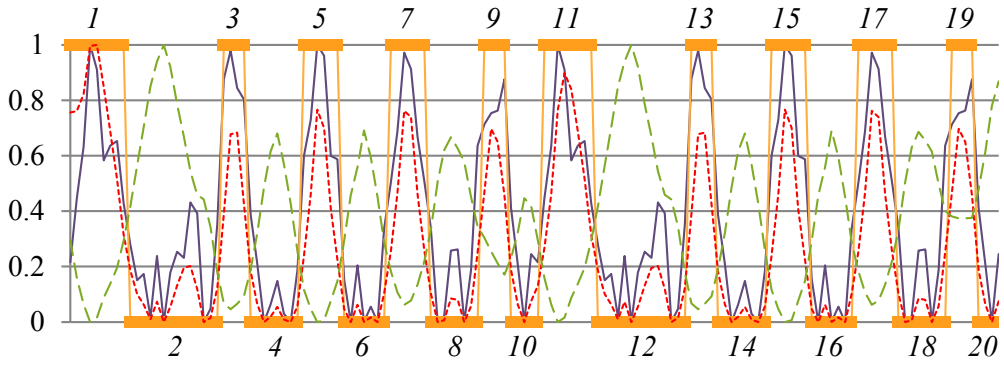

Figure 1: Group clustering based on erosion scores. The horizontal axis represents time, the solid dark line the considered time series X, the red short dashes line the erosion score and the green long dashes line the erosion score of $\overline{X}$. The vertical orange lines are group separators, the horizontal thick orange ones indicate consecutive points belonging to the same groups whose numbers are indicated above the graphic for high ones, and below for low ones.

adverb like "exactly", "approximately" or "roughly", and *unit* is a time unit like "hour", "week" or "second".

The DPE method consists of 3 modules: the first one builds a symbolic representation of the time series based on data clustering to identify groups of consecutive high and low values. The second one processes these groups to compute a periodicity degree and a candidate period. The last one performs the linguistic rendering. The first two modules are detailed in the next subsections, the linguistic one in Section 5.

3.1. High and low values clustering

The DPE method first proposes to identify high and low value groups based on the extraction of the time series "skeleton", obtained with mathematical morphology tools [60]. It more precisely exploits repeated erosions: denoting $x_i^0 = x_i$, the value obtained after one erosion is defined as $x_i^1 = \min(x_{i-1}, x_i, x_{i+1})$, and the result of k successive erosions is $x_i^k = \min(x_{i-k}, \ldots, x_{i+k})$. They are exploited to define the *erosion score* es_i:

$$es_i = \frac{1}{\max_j es_j} \sum_{k=0}^{z_i} x_i^k \tag{1}$$

where $z_i = \arg\min_k x_i^k = 0$. Since $\exists l$ s.t. $x_l = 0$, the existence of z_i is guaranteed.

As detailed in [14], the erosion score allows to identify high value groups and to remove isolated low values. Complement erosion scores, denoted $\overline{es}_i$, are similarly computed from the complement of the data, i.e. from $\overline{x_i} = 1 - x_i$ for all $i = 1..n$.

Groups of high values are then 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. Fig. 1 illustrates the group clustering based on erosion score.

Thus, after the erosion scores computation, the time series is decomposed into high and low value groups, stored in a sorted list $G = (G_k)_{k=1\ldots g}$.

To each group is associated a size s_j and a type $\tau \in \{H, L\}$ for High or Low. τ_j denotes the type of the group G_j, G_j^τ denotes the j^{th}group of type τ and s_j^τ its size. Moreover, g^τ denotes the number of groups of type τ. Similarly, n denotes the total number of points in X, and n^τ the number of points of type τ, where the type of a point is defined as the type of the group it is assigned to. Obviously, $g = g^H + g^L$ and since each data point belongs to a group H or L, $n = n^H + n^L$ and $\sum_j s_j = n$.

It must be noted that the erosion score can be computed quickly (1 million point dataset in 1.5 second) and incrementally [59].

3.2. Periodicity computing

Based on the previously mentioned principle, the periodicity degree measures the regularity of group sizes of both H and L types. As formally detailed below, the candidate period is defined as the sum of the average sizes of H and L groups.

Periodicity degree computation. To measure group size regularity, DPE exploits the coefficient of variation defined as the quotient of a size deviation measure and the size average. The deviation measure used here is the mean absolute deviation, more robust to noise than the more usual standard deviation [61].

More formally, for any type of group $\tau \in \{H, L\}$, the average μ^τ, the mean absolute deviation d^τ, the coefficient of variation CV^τ and the regularity ρ^τ are defined as:

$$\mu^\tau = \frac{1}{g^\tau} \sum_{j=1}^{g^\tau} s_j^\tau = \frac{n^\tau}{g^\tau} \qquad d^\tau = \frac{1}{g^\tau} \sum_{j=1}^{g^\tau} \left| s_j^\tau - \mu^\tau \right| \qquad CV^\tau = \frac{d^\tau}{\mu^\tau} \tag{2}$$

$$\rho^\tau = 1 - \min\left(CV^\tau, 1\right) \tag{3}$$

The min in the expression of ρ^τ ensures that the result is in $[0, 1]$ (see [14] for more details). The regularity is computed for both H and L groups.

The periodicity degree π is then computed as the regularity average across high and low value groups:

$$\pi = \frac{\rho^H + \rho^L}{2} \tag{4}$$

Candidate period computation. The candidate period is 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, so the period is approximated here as the sum of the average sizes of H and L groups:

$$p = \mu^H + \mu^L \tag{5}$$

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

The final linguistic part of the DPE method is not detailed here since a superseding one is proposed and described in Section 5.

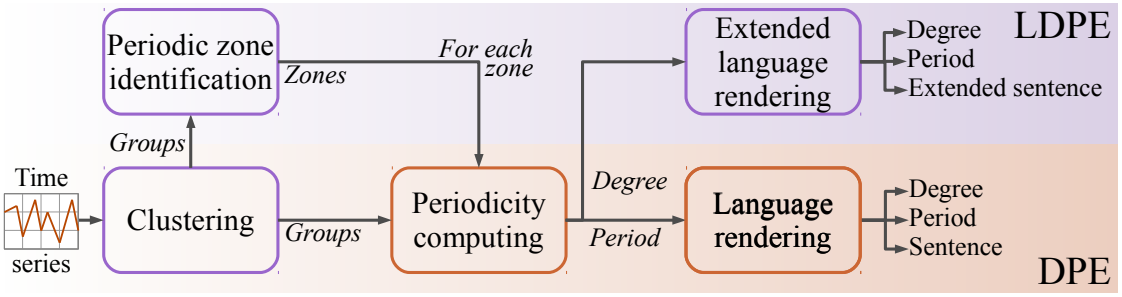

Figure 2: Schematic view of LDPE, that encompasses DPE

4. LDPE: Local Detection of Periodic Events

The Local Detection of Periodic Events method, LDPE, is a generalisation of DPE that contextualises π, p and the linguistic expression for each part of the dataset it identifies as locally periodic: if the dataset is periodic or non periodic as a whole, it returns a single zone and then yields a result equivalent to DPE otherwise it determines these quantities for each zone independantly. Its global architecture is graphically represented on Fig. 2 together with its relation to DPE.

Therefore LDPE allows to extract knowledge from data whose properties are changing over time. In a sense, it is to DPE as Short-Time Fourier Transform is to Fourier Transform, i.e. a transposition of a method on parts rather than on the whole data.

To do so, LDPE relies on the preliminary identification of high and low value groups, as DPE, e.g. performed by the method described in Section 3.1. It then computes periodicity degrees, not on the whole dataset as DPE, but locally, on temporal windows associated to each group, yielding the so-called periodicity fronts. The windows are automatically defined, in an adaptive process. They then lead to the identification of the periodic zones. Finally, the linguistic sentences are generated. These steps are successively detailed below.

4.1. Periodicity front

LDPE considers as input the decomposition of the time series in high and low value groups detailed in Section 3.1. As in DPE, the periodicity is computed based on these groups. However, it is computed for different group subsets in LDPE , whereas all groups are used in DPE. Therefore, several periodicities are computed in LDPE, one for each subset, while only one is returned in DPE.

Hence, the periodicity π_j associated with the j^{th} group is computed with the subset of groups identified as relevant for the j^{th} group. Formally, given a sorted group list $G = (G_k)_{k=1...g}$ and denoting j^- and j^+ group indices such that $1 \leq j^- \leq j \leq j^+ \leq g$:

$$\pi_j = \pi\left(G, j^-, j^+\right) \tag{6}$$

where $\pi(G, j^-, j^+)$ is the periodicity computed using the same method as the one described in subsection 3.2 but taking into account only the groups of indices j^- to j^+. This section proposes an auto-adaptive method to define the values of these indices j^- and j^+ .When

no ambiguity arises, it is denoted $\pi(j^-, j^+)$. Using this notation, in the case of DPE, the unique periodicity computed can be written $\pi = \pi(G, 1, g)$.

Technically, $\pi(j^-, j^+)$ is computed using the expressions in Eq. (2) where the bounds are adapted such that the sums are computed from the first group of type τ whose index is greater than or equal to j^- to the last group of type τ whose index is lesser than or equal to j^+ and g^τ is adapted accordingly.

We call *periodicity front* the sequence of all local periodicity degrees π_j for $j = 1...g$.

So as for π_j to be a relevant estimation of the local periodicity for group j, the determination of j^- and j^+ faces a dual constraint: on the one hand, they must be close enough to j so as to represent the periodicity *locally* in the neighbourhood of j, and on the other hand they must be far enough from j to be significant. Indeed, bounds too close to j may yield a high periodicity "by chance", exemplified by the trivial case where $j^+ - j^- \leq 2$ where only one group of each type is taken into account, trivially yielding 1 as a periodicity computation result.

We propose to apply a methodology based on significance tests to determine whether, for given values j^- and j^+, the value π_j is obtained by chance or is significant at level α, where α is a user-set parameter in $[0, 1]$. It leads to the definition of an auto-adaptive window around j.

4.1.1. Probability distribution of the group size deviation

To determine whether a given π_j is obtained by chance, we compute the probability that π_j equals the obtained periodicity value on the groups whose indices range from j^- to j^+. If this probability is greater than a significance level α, then we can assume that π_j is obtained by chance so the bounds j^- and j^+ are rejected as not significant and another test is run using a larger interval.

For any group j, and its associated neighbourhood defined as $\bigcup_{l=j^-}^{j^+} G_l$, we denote g_j^τ, n_j^τ, μ_j^τ and d_j^τ the local values of, respectively, the number of groups, number of points, size average and deviation of this neighbourhood, for each type τ, high or low: they are computed as indicated in Eq. (2), applied to the neighbourhood $\bigcup_{l=j^-}^{j^+} G_l$ instead of G.

The probability that π_j is obtained by chance depends on the way the points are distributed in the groups of the considered neighbourhood. Furthermore, this distribution is linked to d^H and d^L only, since μ^H and μ^L are fixed for a given neighbourhood, for which the number of points and groups do not vary. Hence, due to Eq. (2) to (4), the probability that π_j is obtained by chance for a given neighbourhood only depends on d_j^H and d_j^L.

Furthermore, d_j^τ depends on n_j^τ and g_j^τ only. So the probability that π_j equals a given value can be computed as a combination of the probability that the group size deviations d_j^H and d_j^L are equal to some value δ^τ, i.e. $P\left(d^\tau = \delta^\tau | n_j^\tau, g_j^\tau\right)$ for $\tau = \{H, L\}$.

Hence, we propose a statistical test with significance α based on the null hypothesis $H_0^\tau(j) =$ *the deviation δ computed for groups of type τ whose indices range from j^- to j^+ is obtained by chance.* Finally, we propose to accept the bounds j^- and j^+ only if $H_0^H(j)$ and $H_0^L(j)$ are rejected, i.e. if $P\left(d_j^H = \delta^H | n_j^H, g_j^H\right) \leq \alpha$ and $P\left(d_j^L = \delta^L | n_j^L, g_j^L\right) \leq \alpha$.

We establish that the probability that a random assignment of n points in g groups yields an average deviation of the group sizes equal δ is given by:

$$P\left(d=\delta|n,g\right)=\begin{cases}0 & \text{if } n<g \text{ or } g<1 \text{ or } g^2\delta\notin\mathbb{N}\\ \dfrac{\sum_{l\in\Lambda}\widetilde{N}\left(n,g,l,\delta\right)}{N\left(n,g,1,+\infty\right)} & \text{otherwise}\end{cases} \tag{7}$$

where the functions N and $\widetilde{N}$ are detailed together with the proof in Appendix A.

It must be underlined that this distribution is not an approximation, but an exact count of the favourable cases, divided by the total number of possible assignments of n points in g groups.

4.1.2. Auto-adaptive bounds for local periodicity computation

Based on the previous statistical test, we propose to determine the bounds j^- and j^+ starting as close as possible to j, and pushing them away from j while the value is obtained by chance, i.e. while $H_0^\tau(j)$ is not rejected for both H and L groups.

Given this procedure, we propose three estimates of the local periodicity π_j defining the left, centre and right periodicity fronts, respectively denoted πL_j, πC_j and πR_j: the left one captures the periodicity from j to its left, i.e. with j^- decreasing and $j^+=j$, the central one captures the periodicity around j, i.e. with j^- decreasing and j^+ increasing, and the right one captures the periodicity from j to its right, i.e. with $j^-=j$ and j^+ increasing. Formally, denoting $H_0\left(\alpha,j\right)$ "$H_0^L\left(j\right)$ and $H_0^H\left(j\right)$ are both rejected at significance level α":

$$\pi L_j=\arg\min_{k>0}\left\{\pi\left(j-k,j\right)\ \text{ s.t. }\ H_0\left(\alpha,j\right)\right\} \tag{8}$$

$$\pi C_j=\arg\min_{k>0}\left\{\pi\left(j-\lfloor k/2\rfloor,j+\lceil k/2\rceil\right)\ \text{ s.t. }\ H_0^\alpha\left(\alpha,j\right)\right\} \tag{9}$$

$$\pi R_j=\arg\min_{k>0}\left\{\pi\left(j,j+k\right)\ \text{ s.t. }\ H_0^\alpha\left(\alpha,j\right)\right\} \tag{10}$$

The series made of consecutive left, centre and right periodicity values for $j=1...g$ respectively define the left, centre and right periodicity fronts. It must be noted that the computational complexity of this method is low since successive values of π_j for increasing k values are easily computed incrementally, and that the probability computations for the three periodicity fronts are often equal from one front to another, and hence can be accelerated using a simple cache.

Fig. 3 shows the periodicity fronts computed on an example dataset with a significance level $\alpha=1\%$. They allow to capture important points further used in the periodic zone identification. Indeed, it can be observed that each one is high in the central periodic part of the data, and low otherwise. The left front is high until the end of the periodic zone, the right front is high from its beginning, and the centre one is high over most of the periodic zone except at its ends: it can be observed that a drop in the right or left front respectively indicates the beginning or the end of a periodic zone, while the centre front "balances" the results from the two other ones.

Fig. 4 illustrates a periodicity front computed using the same data but without the auto-adaptive window: here, the windows used to compute the local periodicity degrees are constantly defined as the smallest windows yielding a non trivial periodicity, i.e. including

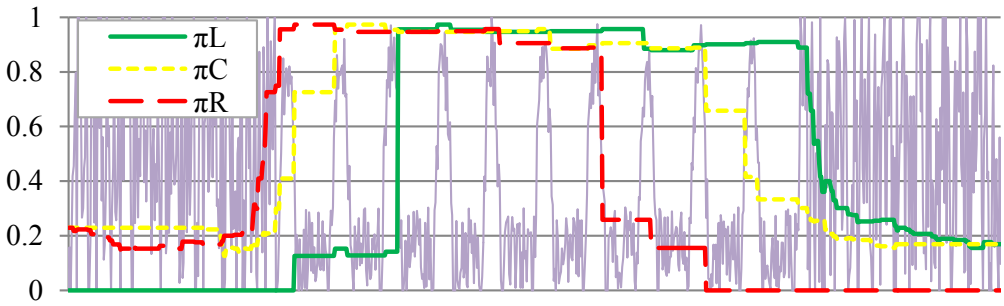

Figure 3: Periodicity fronts using auto-adaptive windows obtained with $\alpha = 1\%$, for the time series represented in light grey.

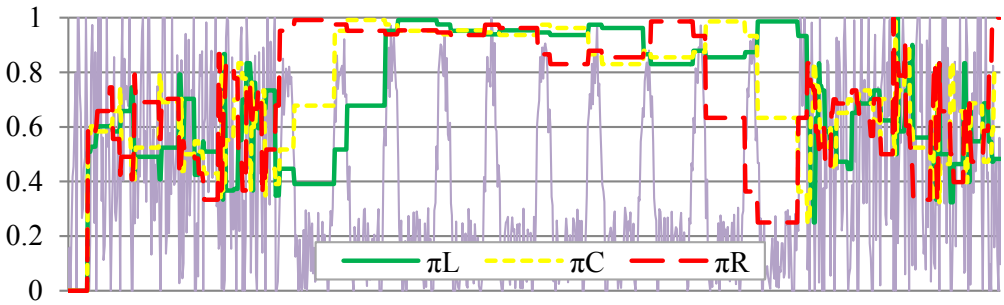

Figure 4: Periodicity fronts obtained with non adaptive windows such that $j^+ - j^- = 4$, for the time series represented in light grey.

at least two H or L groups, hence verifying $j^+ - j^- = 4$, or more precisely and using the same notations as in Eq. (8) to (10), $\pi L_j = \pi(j-4, j)$, $\pi C_j = \pi(j-2, j+2)$ and $\pi R_j = \pi(j, j+2)$. It can be observed on Fig. 4 that these small constant windows lead to periodicity front values that are high in non periodic zones, as shown on the left and right ends of the figure where periodicity fronts take values up to 0.8 for highly non periodic data. In comparison with Fig. 3, it illustrates an important property of the auto-adaptive windows: they enlarge the periodicity computation window in the noisy parts of the series, thus ensuring low periodicity degrees in these parts. Indeed high results obtained by chance over a too narrow window are prevented thanks to the statistical testing step.

4.2. Periodic group classification

Based on the periodicity fronts, each group is classified as P if it belongs to a periodic zone or as N if it does not. The methods proposed to carry out the classification are based on different combinations of the periodicity fronts and different reference values, as detailed in this section.

Reference values. Four reference values, each declined in two variants, are considered for group classification: either the average of the 3 periodicity fronts across all groups or the average of their maximum, possibly weighted according to the group sizes. Formally, the standard and weighted averages for each periodicity front are respectively denoted $\overline{\pi}k$ and

$\overline{\pi}k_w$ for $k = L, C, R$ and computed as:

$$\overline{\pi}k = \frac{1}{g}\sum_{j=1}^{g}\pi k_j \qquad \text{and} \qquad \overline{\pi}k_w = \frac{1}{n}\sum_{j=1}^{g} s_j \times \pi k_j \tag{11}$$

The standard and weighted averages of the periodicity front maximum are respectively denoted $\overline{\pi}M$ and $\overline{\pi}M_w$ and computed as:

$$\overline{\pi}M = \frac{1}{g}\sum_{j=1}^{g}\max\left(\pi L_j, \pi C_j, \pi R_j\right) \quad \text{and} \quad \overline{\pi}M_w = \frac{1}{n}\sum_{j=1}^{g} s_j \times \max\left(\pi L_j, \pi C_j, \pi R_j\right) \tag{12}$$

The underlying principle of the weighted variant is to try and compensate for the possible distortion that may appear compared to the time stamps from the original series, since periodicity is computed at the group level. More precisely, small and numerous groups covering a given part of the series may have their periodicity overweighted when compared to the one of less numerous large groups yet covering the same part of the series. Introducing the group size s_j as a multiplicative factor in Eq. (11) and (12) allows to balance this effect.

Since the reference value is computed using an average of the periodicity front, it may be very low if the whole dataset is non periodic, so non periodic parts of the dataset may be classified as periodic whereas they are only "less non periodic" than others. In order to avoid this effect, we propose to use a minimum threshold for periodicity denoted π_{min}: denoting πk_{ref} the final reference value with $k = L, C, R, M$ and $\overline{\pi}$ one of the values given in Equations (11) or (12):

$$\pi M_{ref} = \max\left(\pi_{min}, \overline{\pi}k\right) \tag{13}$$

Classification methods. Based on the reference values, we define three classification methods m_1, m_2 and m_3, based on three different combinations of the periodicity fronts πL, πC, πR.

The first method, denoted m_1, considers that if one of the three fronts is high enough, then the group belongs to a periodic zone. Formally:

$$m_1\left(j\right) = \begin{cases} P & \text{if } \max\left(\pi L_j, \pi C_j, \pi L_j\right) \geq \overline{\pi}M_{ref} \\ N & \text{otherwise} \end{cases} \tag{14}$$

where P indicates group j belongs to a periodic zone and N it does not.

The second method, denoted m_2, is based on a simple voting system, stating that if the left and centre fronts or the right and centre fronts are greater than their respective average values, then the group belongs to a periodic zone. Formally:

$$m_2\left(j\right) = \begin{cases} P & \text{if } \left(\left(\pi L_j \geq \overline{\pi}L_{ref}\right) \wedge \left(\pi C_j \geq \overline{\pi}C_{ref}\right)\right) \vee \left(\left(\pi C_j \geq \overline{\pi}C_{ref}\right) \wedge \left(\pi R_j \geq \overline{\pi}R_{ref}\right)\right) \\ N & \text{otherwise} \end{cases} \tag{15}$$

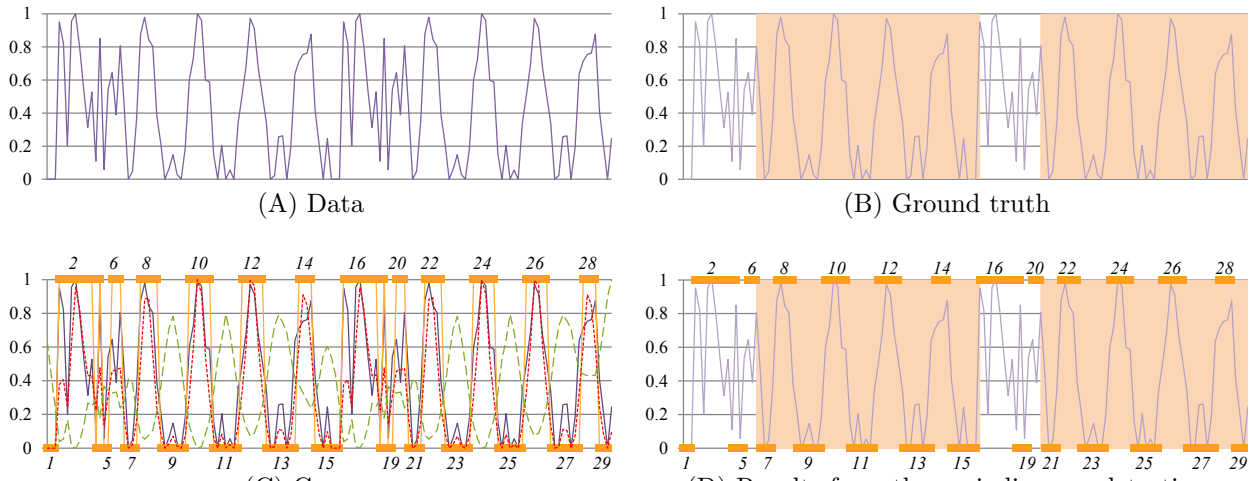

Figure 5: Periodic zone detection indicated with a coloured background for the time series represented as a purple line. (A) Studied dataset. (B) Ground truth, i.e. expected results, consisting of two periodic zones separated by a non periodic one, after an initial non periodic one. (C) Identified groups, as described in Fig. 1. (D) Obtained results: very similar to the ground truth.

The third method is the most optimistic one: it asserts that a group belongs to a periodic zone if one of the 3 fronts is greater than its average value. Formally m_3 is defined as:

$$m_3(j) = \begin{cases} P & \text{if } (\pi L_j \geq \overline{\pi} L_{ref}) \vee (\pi C_j \geq \overline{\pi} C_{ref}) \vee (\pi R_j \geq \overline{\pi} R_{ref}) \\ N & \text{otherwise} \end{cases} \tag{16}$$

These three methods are respectively denoted m_{1w}, m_{2w} and m_{3w} when π^k_{ref} is computed using the weighted average instead of the standard one.

It can be noticed that, since the goal of these methods is to extract the approximately flat parts of the fronts, standard change detection methods [62] could be used. However, these methods rely on model hypotheses for the data, which are not known and probably not compatible with periodicity fronts. Moreover a straightforward approach as the one presented here can be implemented to segment the time series since the auto-adaptive window and the periodicity front computation provide a robust analysis.

4.3. Clustering of periodic groups

Once each group is classified as belonging or not to a periodic zone, respectively labelled N or P, the consecutive periodic groups are gathered in *periodic zones*.

After these periodic zones are determined, a post processing step is proposed in order both to discard the ones containing less than *minSize* groups and to merge successive zones separated by less than *minSep* groups. When this filtering post processing step is performed, the method is prefixed with an f, for example fm_1 or fm_{3w}.

The output of these two first steps is a list of triplets containing an interval considered periodic, its periodicity degree π and its period p. Fig. 5 illustrates a case where two zones

are identified, as expected from visual inspection of the data. In this example, the two returned triplets are:

$$Z_1 = ([23, 73], 0.83, 11.90)$$
$$Z_2 = ([93, 140], 0.78, 11.30)$$

meaning that the first periodic zone Z_1 spans from point 23 (beginning of group 7) to point 73 (end of group 15), has a periodicity degree $\pi = 0.83$ and an estimated period $p = 11.90$, and the second periodic zone Z_2 spans from point 93 (beginning of group 21) to point 140 (end of group 29), has a periodicity degree $\pi = 0.78$ and an estimated period $p = 11.30$.

5. Linguistic rendering

The linguistic rendering step aims at giving a linguistic representation of each of the previous triplets, representing three core aspects of the periodic zones: their time context or location in time, their periodicity and their period.

5.1. Proposed protoform to express local periodicity information

We propose to linguistically express each periodic zone on the base of a protoform instantiated with the values computed in the previous steps. More specifically, we propose the following structure for the protoform:

$$\underbrace{Prec_1\, TimeCtxt}_{Time\,context}\ \text{the series is}\ \underbrace{Pdty\ (\pi)}_{Periodicity\,part}\ \Big[\underbrace{\text{with a period of } Prec_2\, \hat{p}\, units}_{Period\,part}\Big]$$

where $Prec_1$ and $Prec_2$ are precision adverbs, as "exactly", "approximately", "roughly", *TimeCtxt* a time contextualisation expression, as "the first quarter of" or "during summertime", *Pdty* a linguistic periodicity assessment, as for instance "highly periodic", or simply "periodic", π the computed value for the periodicity, $\hat{p}$ a convenient approximation of computed value for the period p, and *unit* the unit deemed the most adequate to represent the period, as "months", "weeks" or "seconds".

The period part is included only if the periodicity π is high enough. Indeed, a computed period is meaningful and relevant to include in the result sentence only if the data is actually periodic, i.e. if its periodicity degree is high.

The variety of generated sentences can be illustrated by the following examples:

- "The first two months, the series is highly periodic (0.89) with a period of approximately 1 week"
- "During the first quarter, the series is periodic (0.78) with a periodicity of exactly 1 month"
- "From September to November, the series is not very periodic"

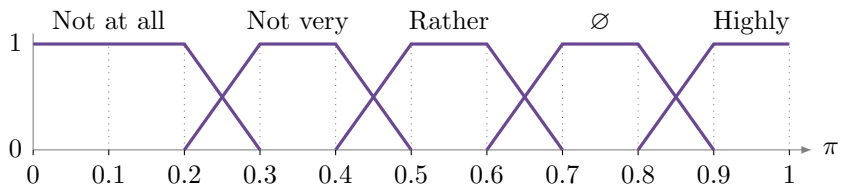

Figure 6: Linguistic variable for the periodicity degree π

The two following sentences are generated from the example time series illustrated in Fig. 5, to respectively characterise the two identified periodic zones:

- "Approximately from its first quarter to its half, the series is periodic (0.83) with a period of approximately 12 points"
- "Approximately from its third quarter to its end, the series is periodic (0.78) with a period of approximately 11 points"

The rendering of the different parts of the protoform are detailed in the following subsections: first, the period part, then, the periodicity part and the time context. Lastly, Subsection 5.5 describes the expression of the approximation quality, selecting a precision adverbs both for period and time context.

5.2. Period measure rendering

The period linguistic expression is part of the DPE method [14]. Its three main steps aim at representing the period in a way familiar to a human user. To do so, three interconnected aspects of time formulation are taken into account: the choice of a relevant time unit, the selection of a good approximation and the enrichment with a precision adverb.

Regarding the time unit, it seems from general observations that speakers prefer using small numbers and thus adapt the considered unit. For instance, the statement "I meet her every week" seems preferable to "I meet her every 168 hours". Moreover, an approximation is usually used to express time: one would rather say "This happens every 45 minutes" than "This happens every 44.2 minutes".

Therefore, a representation of the period is achieved by first selecting the most convenient unit among a scale given by the user, then computing an approximation of p, $\hat{p}$, as the nearest multiple of 5 or the nearest integer if too far from p. Finally a precision adverb is added, as detailed in Section 5.5, to express the approximation error, computed as $err = |p - \hat{p}| / p$ (see [14] for a more detailed justification of this error measure).

5.3. Periodicity degree rendering

The periodicity degree π is a dimensionless value in $[0, 1]$, defining a scale from a "worst" to a "best" result.

We propose to define a linguistic variable for $\pi \in [0, 1]$ with linguistic labels *highly,* $\emptyset$*, rather, not very* and *not at all* [14]. The empty label represents the "normal" qualifier.

Indeed, to express the periodicity of a given zone, we can say that it is "highly periodic", "rather periodic", or simply "periodic", the latter case corresponding to the empty label. The labels *not at all* and *not very* are relevant only if the user chooses to have a linguistic representation of all parts of the dataset, including the non periodic ones (defined as the complement of the periodic ones).

These linguistic labels are associated with the modalities shown on Fig. 6 that indicates the membership functions $\mu_M(x)$ with $M = \{highly, \varnothing, rather, not\ very, not\ at\ all\}$.

The modality to which the periodicity to be expressed π has the highest membership value is selected, i.e.:

$$modality(\pi) = \arg\max_{m \in M} \mu_m(\pi)$$

Given the selected modality, the periodicity part is instantiated as "*modality* periodic (π)", as for instance "rather periodic (0.51)".

5.4. Time context rendering

The time context rendering, specific to LDPE, represents the location in time of the considered zone characterised by its periodicity. It is represented as an interval, i.e., in LDPE, a couple of values in the set of time stamps from X denoting the first and last indices of the periodic zone. It can be expressed both in an absolute and a relative way.

Absolute interval rendering. The absolute interval rendering is the location of the considered zone in the dataset, as for instance "the two first quarters" or "from the second to the fourth month".

Using the hierarchical approach of interval rendering described in [63], the user can define temporal hierarchies as *quarters*, *months* and so on. Moreover, single intervals not belonging to the hierarchy can be defined as well, like *Easter holidays* or *Summertime*.

The fit between a considered zone and a candidate interval can for instance be measured with the Moore distance [64]. The candidate that minimises this distance, interpreted as a representation error, is selected and returned as the time context.

Alternatively, the candidate selection can exploit knowledge about interval granularity: when two intervals fit the zone to be characterised, the one with the most precise granularity level can be selected, as suggested in [63].

Relative interval rendering. Relative interval rendering allows to generate sentences as "the first two thirds" or "the last tenth". They can also be based on hierarchical intervals but do not need to be specified since they are dimensionless and relevant for any dataset. They are defined with ratios as for instance *halves* = ([0, 0.5], [0.5 1]), or *thirds* = ([0, 0.33], [0.33, 0.66], [0.67, 1]) .

Depending on the dataset size, the ratios contained in the relative intervals are converted into intervals similar to the absolute ones, and the same principles as the ones used in the absolute interval rendering are used to find relevant intervals and generate the linguistic expression.

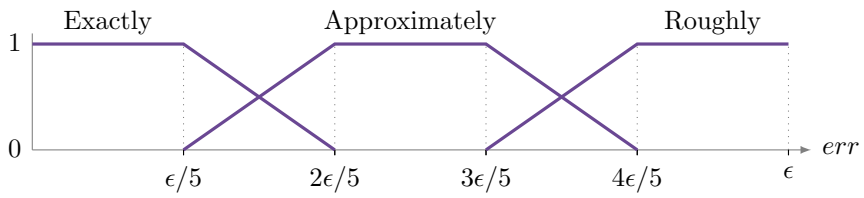

Figure 7: Linguistic variable "Precision" for the period and the periodicity degree

5.5. Precision adverb rendering

For both period (see Section 5.2) and time context rendering (see Section 5.4), a precision adverb, part of the DPE method [14], can be included so as to represent the approximation error made during the linguistic rendering, where the error is respectively measured by the *err* function or the Moore distance.

The precision is represented as a linguistic variable depending on a user defined precision threshold ϵ specifying the maximal accepted error. During period and time context rendering, this value is used in order to choose an acceptable approximation: the approximation error can not be greater than ϵ. The linguistic variable *Precision*, illustrated on Fig. 7, is thus defined on the universe $[0, \epsilon]$.

Finally, the selected modality is the one to which the approximation error has maximal membership degree.

6. Experiments

This section presents the experimental study of the proposed LDPE methodology on artificial data and an illustrative example of results obtained on real data. In order to study the behaviour of the method in different configurations, all LDPE variants are tested over different types of generated datasets.

The experimental protocol is first presented, followed by the quality measures used to compare the methods, the scenarios creating different contexts are then described. The artificial data generator used to populate the scenarios is then introduced and the obtained results are commented. The last subsection illustrates the case of a real time series.

6.1. Experimental protocol

12 variants of LDPE are considered and divided in four categories, depending on whether they use the standard or the weighted average (as described in Section 4.2) and whether they use the post processing filter or not (see Section 4.3). The basic variants, with standard average and no filtering, are denoted m_1, m_2 and m_3. The variants using weighted average are denoted by the addition of the index w, i.e. m_{1w}, m_{2w} and m_{3w}. The use of filtering is denoted by prefixing the name with f, leading to fm_1, fm_2 and fm_3 in the case of standard average and to fm_{1w}, fm_{2w} and fm_{3w} in the case of weighted average.

Each method is tested with the 256 combinations of method parameters: $\alpha = \{1\%, 5\%, 10\%, 15\%\}$ for the statistical test (see Eq. (9)), $\pi_{min} = \{0.2, 0.4, 0.6, 0.8\}$, minimum value for the average reference value (see Eq. (13)), *minSep*$= \{2, 4, 6, 8\}$ and *minSize*$= \{2, 4, 6, 8\}$,

used in the filtering step (see Section 4.3). Note that these two method parameters do not influence the variants not using filtering.

Each LDPE variant with each set of parameters is run with 6 scenarios, declined over 5 configurations depending on the value of the scenario parameter (as detailed in Section 6.3 below). Moreover, each configuration for each scenario is repeated 20 times.

As a consequence, the total number of generated datasets equals 256 parameters combinations times 6 scenarios times 5 configurations times 20 repetitions, i.e. 153,600 datasets.

6.2. Quality measures

Using artificial datasets is relevant since they allow to define a ground truth, setting a reference against which the obtained results can be compared. In particular, both the expected number of periodic zones and their positions in time are available.

First, the error on the number of periodic zones is computed, comparing the ground truth value, denoted Z^T, and the identified one, denoted Z. The zone error measure zE is defined as their relative difference

$$zE = \frac{\left|Z - Z^T\right|}{Z} \tag{17}$$

zE is thus a positive value that must be minimised.

Second, a point to point comparison between the zones identified by the method and the ones expected from the ground truth is performed: it evaluates, for each time stamp i, if the tested LDPE variant correctly assigns the point x_i as belonging to a periodic zone or to a non periodic zone. It thus corresponds to an accuracy measure where the 2 classes are 'belonging to a periodic zone' or not and equivalently is the proportion of common points in or out the periodic zones. Denoting n the total number of points in the considered time series, per the classification function provided as final result by LDPE and per^T the ground truth one, the point classification measure pC can be written:

$$pC = \frac{1}{n}|\{\, x_i \mid per(x_i) = per^T(x_i) \,\}| \tag{18}$$

where $pC \in [0, 1]$ must be maximised.

zE and pC are computed for each LPDE variant, parameter configuration and generated dataset. They are then aggregated computing their averages and standard deviations at different levels: first at the configuration level over the 20 repetitions, then at the scenario level over their 5 configurations, then at the parameter level over the 6 scenarios and finally at the global level over the 256 parameters combinations.

The evaluation of the linguistic rendering part of LDPE is a difficult task: it cannot be performed numerically and requires to involve a human assessment, due to its subjective and semantic nature. From a theoretical point of view, the validity of the proposed approach comes from the large variety of the possible sentences, induced by the richness of the considered protoform. The examples of possible sentences, illustrated in Section 5, indicate a satisfying description of the time series. An experimental validation is ongoing, raising challenging issues in terms of protocol and interviews with panels of users.

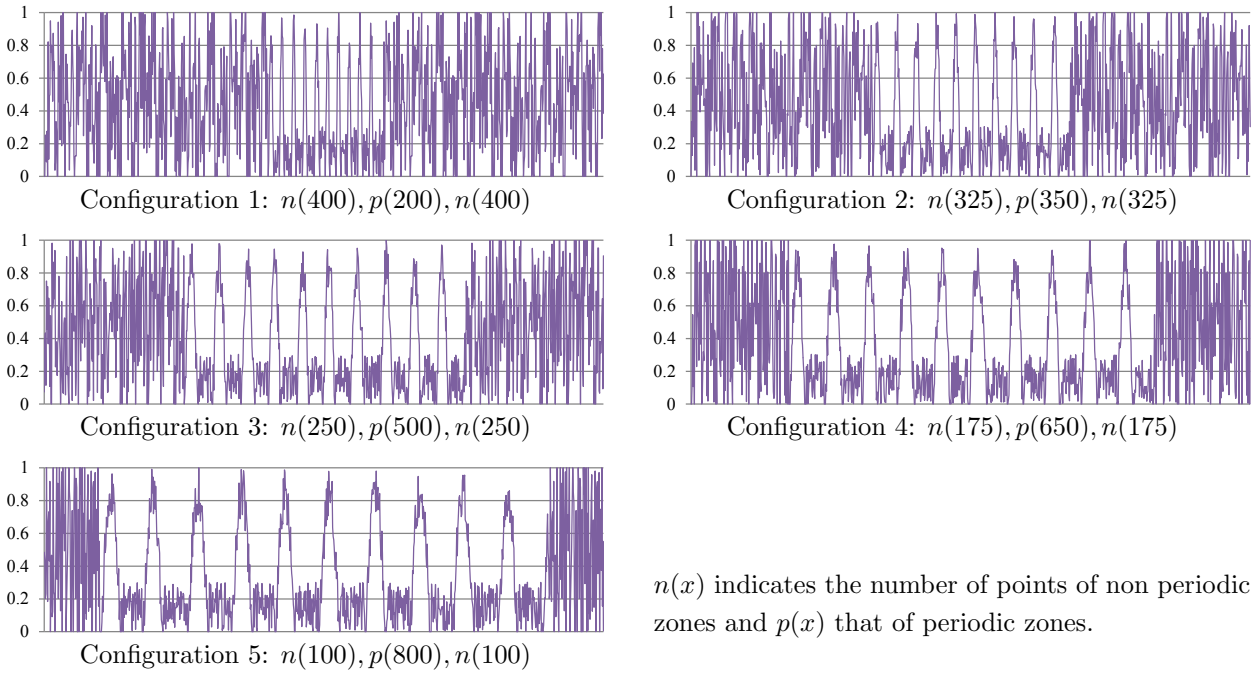

Figure 8: Example of time series datasets generated by the 5 configurations of S1.They differ by the size of the alternated non periodic and periodic zones.

6.3. Scenarios

Six scenarios, denoted S1 to S6, are designed in order to represent six typical use cases. They are defined by the alternation of periodic and non periodic zones with varying periods. Each of them is declined over 5 configurations, depending on the size of each zone.

Scenario 1 (S1) is made of 3 zones, one non periodic, one periodic, and one non periodic, shortly written *npn*. Its goal is to test the LDPE method in a simple configuration, where one periodic zone lies in the middle of two non periodic ones. In order to test different cases, the periodic zone gets larger from one configuration to the next one, thus reducing the surrounding non periodic zones. More precisely, the non periodic, periodic and non periodic zones respectively occupy 40%, 20% and 40% in the first configuration, to 10%, 80% and 10% at the fifth one, for a total number of 1,000 points. Thus the zones are respectively 400, 200 and 400 points in the first configuration, 325, 350, 325 in the second one, 250, 500, 250 in the third one, 175, 650, 175 in the fourth one, and 100, 800, 100 in the last one. Fig. 8 illustrates these 5 configurations of S1.

The 5 other scenarios are defined similarly. S2 aims at testing a slightly more complex case, where a non periodic zone is followed by a periodic one, then by a non periodic one and lastly by a periodic one, shortly written *npnp*. S3, summarised as *pnpnp*, adds a periodic zone at the beginning of the previous scenario. S4, summarised as $pp'npn$, tests another case, where two periodic zones are consecutive, but with different periods. This aims at testing the efficiency of the periodicity fronts, since they should be able to detect the change of periodicity. S5, summarised as n, defines a totally non periodic dataset, where sizes do not change from one step to another, since only one non periodic zone is defined. Finally,

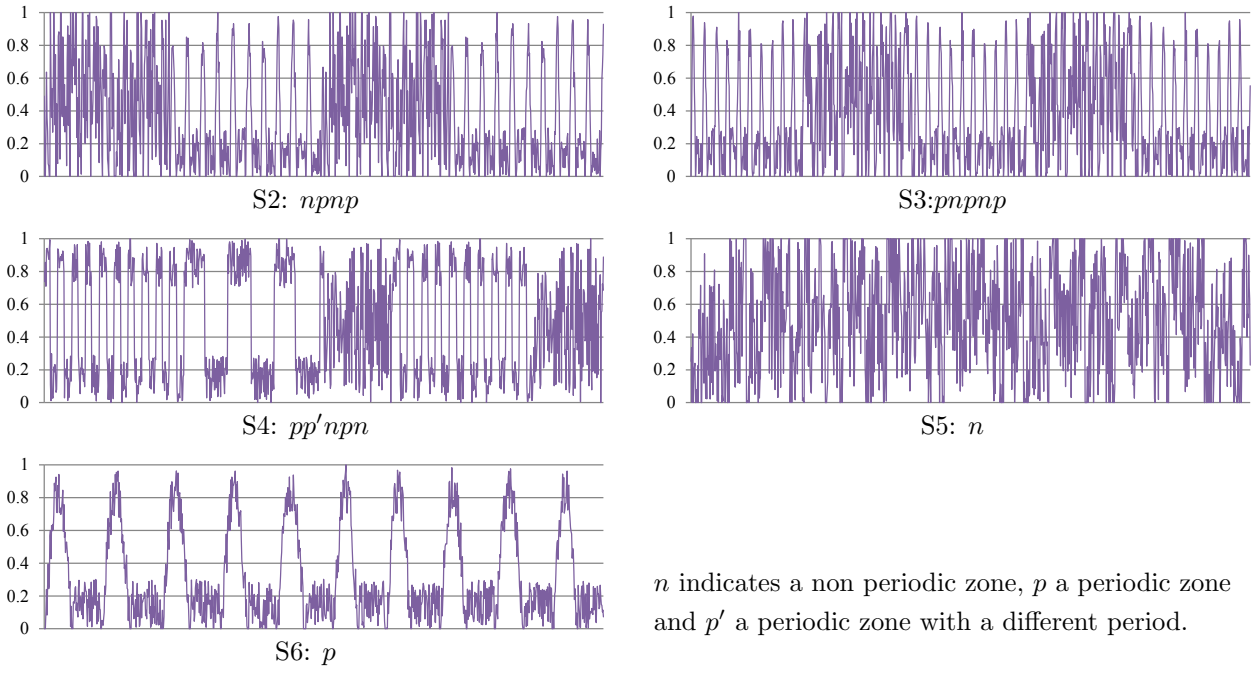

Figure 9: Example of time series generated for Scenarios 2 to 6

S6, summarised as p, symmetrically tests a totally periodic dataset. Fig. 9 illustrates the time series generated with Scenarios 2 to 6.

6.4. Artificial data generation

The data generation is based on the engine described in [14], extended to allow to add consecutive datasets, generated along the same principle one after the other, so as to generate alternate periodic and non periodic zones.

The datasets are generated as noisy series of periodic shapes, either sines or "wave", i.e. sines with flat parts [14]. They are created as a succession of high and low value groups on which two types of noise are applied: the first one modifies the group size and the second one the time series values. At the end of the artificial generation, the dataset is normalised in $[0, 1]$. All the datasets illustrated on Fig. 8 and 9 are generated according to this method.

6.5. Results

Due to the important number of experiments run, this section presents the most important results while Appendix B contains the numerical values of zE and pC for all of them.

From these tables, it can be generally observed that the LDPE method works well for the task of identifying periodic zones across the different scenarios. More precisely, the best method, according to the zE and pC criteria, appears to be fm_{2w}, i.e. using the post processing filter and the weighted average. Indeed, on average for all scenarios, it returns a 21% error rate in zone identification and 91% correct point classification (see Tables B.3 and B.5 in Appendix B).

The following subsections comment in more details the influence of the method parameters and the comparison of the LDPE variants.

6.5.1. Method parameter influence

For each of the 12 LDP variants, we consider the influence of its parameters, α, π_{min} and possibly *minSep* and *minSize*. We measure a parameter importance by assigning a score to each of its considered values, computed as the number of times this value is used in the parameter combinations that leads to the 30 best results for each variant. This count is weighted so that if the parameter value is used in the best combination it receives a score of 30, in the 2^{nd} best a score of 29, and more generally a score of $30 - position$.

These results are assessed according to the two measures zE and pC, detailed in Tables B.2 and B.4 and commented below.

Zone identification results. Regarding zone identification, π_{min} appears to be the most important parameter, since the largest tested value $\pi_{min} = 0.8$ yields from 70% to 90% of the 30 best results for all the methods, with the notable exception of fm_{2w}, for which 60% of the 30 best results are achieved with $\pi_{min} = 0.6$.

This is a very strong result since it implies that using the average value of the periodicity fronts as the reference (see Section 4.2 and Eq. (11) to (16)) is not adequate to identify the periodic zones. Indeed, since π_{min} is the minimum for the used reference value, it means that this value, computed as an average, weighted or not, has to be increased (cf. Eq. (13)), since the highest value of π_{min} yields the best results. It also means that using the average returns many false positive, which have to be filtered out by a threshold. Given this result, the 3^{rd} quartile for instance may be more appropriate.

Furthermore, the importance of α is very clear in the case of non filtered methods. Indeed, 80% of their 30 best results are obtained using $\alpha = 1\%$ or 5%, i.e. the 2 smallest tested values. This can be interpreted as a relation to the filtering power of the statistical test performed through the auto-adaptive window determination (see Section 4.1.2). Indeed, the smaller α, the larger the considered window when computing the periodicity fronts. Since the window gets larger in noisy area than in periodic ones, the periodicity degree computed with a larger window in a noisy area will be smaller using smaller α, entailing a compensation for the absence of filtering thanks to the statistical testing.

Regarding the filtered methods, the influence of α is less clear, since filtering is executed afterwards using the parameters *minSep* and *minSize*. Quite interestingly, these three parameters values yielding the 30 best results are more or less uniformly distributed across the different tested values. This is a good point since it means that, when using both the statistical test and a simple post filtering, the results are more or less equivalent over a range of used parameters values, so they can be considered robust.

Point classification results. In point classification, π_{min} is also an important parameter, having more or less the same influence as in the zone identification, with the exception of the m_2 methods, regardless of the filtering used.

Rather surprisingly, the α parameter does not have the same effect as for zone identification. Indeed, the 30 best results for non filtered methods are obtained for $\alpha = 5\%$ or 10%

Table 1: Best methods for all scenarios and all method parameters

	1^{st}	2^{nd}	3^{rd}
zE (μ, σ)	fm_{2w} $(21\%, 11\%)$	fm_{1w} $(25\%, 13\%)$	fm_2 $(26\%, 7\%)$
pC (μ, σ)	fm_{2w} $(91\%, 5\%)$	m_{2w} $(89\%, 4\%)$	fm_1 $(89\%, 5\%)$

and for $\alpha = 5\%$ to 15% for the filtered ones. This can be explained by the fact that the point measure is very optimistic since randomly affecting the periodic points yields on average a 50% result for pC. So if the strictest $\alpha = 1\%$ never yields one of the 30 best results in point classification, it may be because the filtering is too strict in this case and, even though it gives good results for zone identification, it does not take into account the points located at the ends of the zone, hence yielding a bad point classification.

For the filtered methods, the *minSize* parameter returns the best results with its largest tested values, since 80% of the 30 best results are obtained with *minSize*= 6 or 8. On the contrary, the *minSep* parameter gives the best results with its smallest tested values, since 80% of the 30 best results are obtained with *minSep*= 2 or 4.

These two observations seem to imply that some not so small zones and some small gaps are erroneously identified. This is a classical consequence of the use of a crisp threshold to distinguish between periodic and non periodic groups. Some more sophisticated methods for segmentation might be useful to improve the classification [62].

6.5.2. LDPE variant comparison

Table 1 indicates the best three LDPE variants, when ranked according to their results in terms of the zE and pC criteria, averaged over tested parameters for all scenarios.

General result. For both zE and pC, fm_{2w}, i.e. m_2 using filtering and the weighted average, comes first: it has the smallest number of errors in zone identification and the highest point classification score, over all parameters and all scenarios. Scores obtained with other methods are given in Appendix B.

More generally, the m_2 variant, regardless of the filtering or kind of used average, seems to be the most efficient method since it is present four times in one of the 3 first positions either in zE or pC. However m_1 appears to behave well too, since it is present each time in 2^{nd} and 3^{rd} position. Clearly, the m_3 method is not adapted for zone identification, since it never appears in this ranking. Indeed, as the most optimistic method among the three defined (see Section 4.3), it may be too optimistic for the zone identification task.

As for the methods parameters, the usage of the post processing filter is discriminant. Indeed, the filtered method are much more efficient than the non filtered ones, since the 6 best methods regarding zones identification, and 4 of the best with respect to point classification use the post processing filter.

Zone results. Regarding the zE results, their standard variation is rather high as their average error, even for the best method. This is due to the number of zones to be detected in each scenario, ranging from 1 in S5 and S6 to 5 in S3 and S4. If one extra zone, even

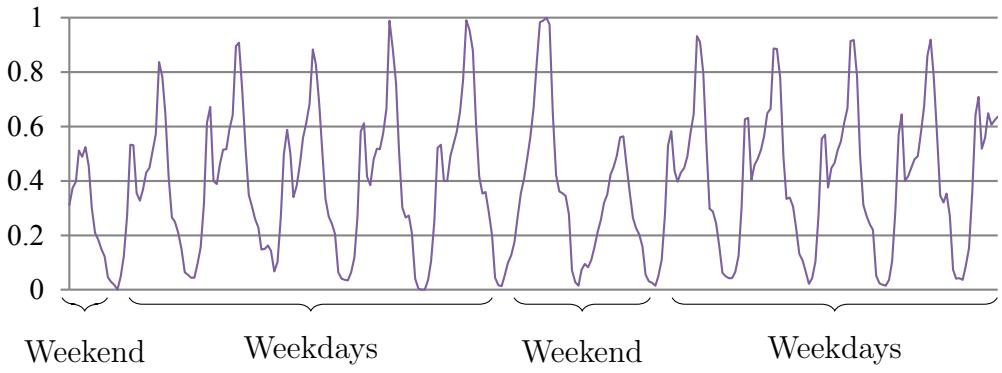

Figure 10: Two week measures of the quantity of CO_2 per hour in the station Châtelet.

small, is detected, then the error rate is increased by 20% with the 5 zone scenarios, and 100% with the 1 zone scenarios. This measure is thus very sensitive to error, hence the large results and deviation.

Point results. Conversely, the pC results are high and their standard variation are low, especially given the fact that more than 150,000 experiments run. This is due to the non severity of the count of points, as mentioned above. In that respect, it complements the much more severe zE measure.

6.6. Experimentation with real data

In this subsection, an illustrative experiment using real data is presented. The first paragraph describes the used dataset, and the second one comments the obtained results.

6.6.1. Data presentation

The RATP is the main public transport operator company in Paris, France and monitors among others the air quality underground. Through its Open Data service [65], it released hourly measurements carried out during the first quarter of 2012 in different metro stations.

For the test, we use a 12 days sample of the normalised amount of CO_2 in the station Châtelet, illustrated on Fig. 10.

Visual inspection indicates that the data are indeed periodic, with different patterns for the weekdays and the weekends indicated below the figure.

6.6.2. Results

The dataset is processed with the method fm_{2w} and the parameters $minSep = 2$, $minSize = 2$, $\alpha = 10\%$ and $\pi_{min} = 0.8$, yielding the result shown on Fig. 11. The result is interesting since the periodicity of weekends is sufficiently different from the one of weekdays for LDPE to create different zones for each. The result is not flawless however since the first weekday on the left is omitted. Moreover, a small part of the weekend is identified as weekday.

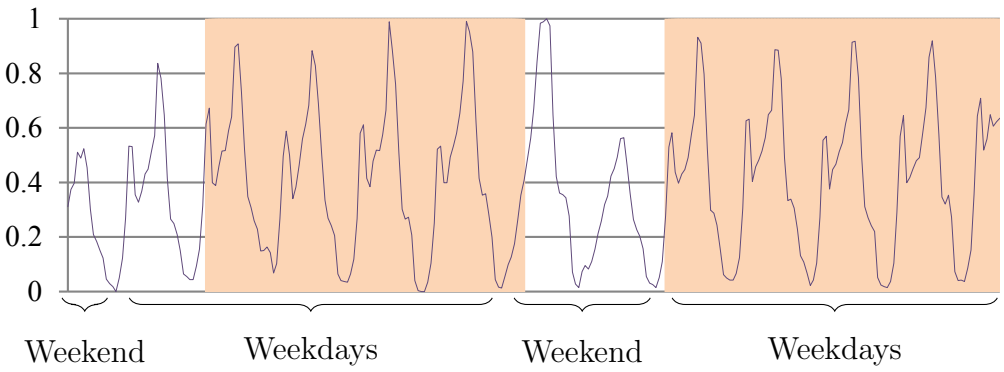

Figure 11: Result of local periodicity detection on the real dataset from Fig. 10

This behaviour is due to a wrong clustering during the first step of DPE (cf. Section 3.1) rather than a wrong zone identification in LDPE. Some of our ongoing works are precisely related to enhance this first step.

The returned zones are $Z_1 = ([60, 141], 0.92, 24.00)$ and $Z_2 = ([202, 291], 0.83, 22.90)$, given that the period p in third position is expressed in hours.

Therefore, using an acceptable error $\epsilon = 5\%$ (see Section 5.5) and relative intervals dividing the dataset into 2 to 5 parts (see Section 5.4), the linguistic rendering of Z_1 is "Approximately from the fifth to the half, the series is highly periodic (0.92) with a period of exactly one day", and the linguistic rendering of Z_2 is "Approximately from the second third to the end, the series is highly periodic (0.83) with a period of approximately 1 day".

7. Conclusion and future works

This paper presents the LDPE method that allows to detect periodic zones in a time series, to generate sentences characterising them in terms of periodicity and period and locating them in time. Generated sentences can for instance be "Approximately from March to June, the series is highly periodic with a period of exactly 2 weeks". The proposed LDPE method allows to determine local periodic zones over model-free time series using robust parameters since reasonable changes in their values do not affect the good results of the method. It is based on an auto-adaptive window scheme allowing to compute local periodicity, guaranteeing their significance and based on a new probability distribution presented in this paper. More than 150,000 experiments have been run on artificial data as well as an illustrative example on real data, proving the efficiency of the proposed LDPE method and yielding very satisfying results.

Ongoing works include the experimental study of variants for the periodicity front segmentation as well as for the erosion score computation, so as to further improve the quality of the identified intervals and thus the global quality of the generated linguistic summaries. The relevance of more precise summaries, obtained by running the periodicity detection method within the identified periodic zones, will also be studied, taking into account the trade-off between increased computational cost and result quality. At a theoretical level,

future works will include the study of a formal expression of the probability distribution of the periodicity degree, for instance the computation of $P(\pi_j = \varpi \,|\pi_{j-1}, ..., \pi_1)$ for the segmentation, or $P(\pi_j = \varpi \,|n, g)$ for the auto-adaptive window.

References

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

[2] L. Zadeh, A computational approach to fuzzy quantifiers in natural languages, Computers & Mathematics with Applications 9 (1) (1983) 149–184.

[3] J. Kacprzyk, S. Zadrozny, Protoforms of linguistic data summaries: towards more general natural-language-based data mining tools, in: Soft computing systems, 2002, pp. 417–425.

[4] J. Lafler, T. D. Kinman, The calculation of RR Lyrae periods by electronic computer, The Astrophysical Journal Supplement Series 11 (1965) 216.

[5] C. E. Goldblum, R. C. Ritter, G. T. Gillies, Using the fast Fourier transform to determine the period of a physical oscillator with precision, Review of Scientific Instruments 59 (5) (1988) 778.

[6] V. Backmutsky, J. Blaska, M. Sedlacek, Methods of finding actual signal period time, in: Proc. of IMEKO'00, 2000, pp. 243–248.

[7] A. M. Noll, Cepstrum pitch determination, The Journal of the Acoustical Society of America 41 (2) (1967) 293.

[8] Z. Li, Mining periodicity and object relationship in spatial and temporal data, Ph.D. thesis, University of Illinois at Urbana-Champaign (2013).

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

[10] M. G. D. Geers, V. G. Kouznetsova, Scale transitions in solid mechanics based on computational homogenization, Mechanics 27 (2001) 37–48.

[11] R. U. Kiran, M. Kitsuregawa, Novel techniques to reduce search space in periodic-frequent pattern mining, in: Proc. of DSAA'14, 2014, pp. 377–391.

[12] P. Grosche, M. Muller, Computing predominant local periodicity information in music recordings, in: Proc. of WASPAA'09, 2009, pp. 33–36.

[13] J.-P. Duval, Périodes locales et propagation de périodes dans un mot, Theoretical Computer Science 204 (1-2) (1998) 87–98.

[14] G. Moyse, M.-J. Lesot, B. Bouchon-Meunier, Linguistic summaries for periodicity detection based on mathematical morphology, in: Proc. of IEEE SSCI FOCI'13, 2013, pp. 106–113.

[15] J. Kacprzyk, S. Zadrozny, Computing with words is an implementable paradigm: fuzzy queries, linguistic data summaries, and natural-language generation, IEEE Trans. on Fuzzy Systems 18 (3) (2010) 461–472.

[16] B. Bouchon-Meunier, G. Moyse, Fuzzy linguistic summaries: where are we, where can we go?, in: Proc. of CIFEr'12, 2012, pp. 317–324.

[17] J. Kacprzyk, R. Yager, "Softer" optimization and control models via fuzzy linguistic quantifiers, Information Sciences 34 (2) (1984) 157–178.

[18] J. Kacprzyk, A. Wilbik, Towards an efficient generation of linguistic summaries of time series using a degree of focus, in: Proc. of NAFIPS'09, 2009, pp. 1–6.

[19] J. Kacprzyk, R. Yager, Linguistic summaries of data using fuzzy logic, Int. Journal of General Systems 30 (2) (2001) 133–154.

[20] F. Díaz-Hermida, A. Ramos-Soto, A. Bugarín, On the role of fuzzy quantified statements in linguistic summarization of data, in: Proc. of ISDA'11, 2011, pp. 166–171.

[21] R. Castillo-Ortega, N. Marín, D. Sánchez, A. Tettamanzi, Quality assessment in linguistic summaries of data, in: Proc. of IPMU'12, 2012, pp. 285–294.

[22] L. Danlos, F. Meunier, V. Combet, EasyText: an operational NLG system, in: Proc. of ENLG'11, 2011, pp. 139–144.

[23] P. Cariñena, A. Bugarín, M. Mucientes, S. Barro, A language for expressing fuzzy temporal rules, Mathware & soft computing 7 (2) (2000) 213–227.
[24] R. Castillo-Ortega, N. Marín, D. Sánchez, E. Corchado, H. Yin, Fuzzy quantification-based linguistic summaries in data cubes with hierarchical fuzzy partition of time dimension, in: Intelligent Data Engineering and Automated Learning, Vol. 5788, 2009, pp. 578–585.
[25] J. Kacprzyk, A. Wilbik, S. Zadrozny, Linguistic summarization of time series using a fuzzy quantifier driven aggregation, Fuzzy Sets and Systems 159 (12) (2008) 1485–1499.
[26] A. Ramos-Soto, A. Bugarín, S. Barro, F. Díaz-Hermida, Automatic linguistic descriptions of meteorological data, in: Proc. of the Iberian Conf. on Information Systems and Technologies'13, 2013, pp. 1–6.
[27] G. Moyse, M.-J. Lesot, B. Bouchon-Meunier, Mathematical morphology tools to evaluate periodic linguistic summaries, in: Proc. of FQAS'13, 2013, pp. 257–268.
[28] D. Gerhard, Pitch extraction and fundamental frequency: history and current techniques, Tech. rep., Dept. of Computer Science, University of Regina (2003).
[29] B. Kedem, Spectral analysis and discrimination by zero-crossings, Proc. of the IEEE 74 (11) (1986) 1477–1493.
[30] M. Tsuji, E. Yamada, A wavelet approach to real time estimation of power system frequency, in: Proc. of SICE'01, 2001, pp. 58–65.
[31] M. K. Mahmood, J. E. Allos, 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 34 (3) (1985) 413–417.
[32] A. M. Zayezdny, Y. Adler, I. Druckmann, Short time measurement of frequency and amplitude in the presence of noise, IEEE Transactions on Instrumentation and Measurement 41 (3) (1992) 397–402.
[33] D. Sánchez, G. Triviño, Computational perceptions of uninterpretable data. A case study on the linguistic modeling of human gait as a quasi-periodic phenomenon, Fuzzy Sets and Systems 253 (2013) 101–121.
[34] L. Palmer, Coarse frequency estimation using the discrete Fourier transform, IEEE Trans. on Information Theory 20 (1) (1974) 104–109.
[35] B. Quinn, Estimating frequency by interpolation using Fourier coefficients, IEEE Transactions on Signal Processing 42 (5) (1994) 1264–1268.
[36] P. J. Kootsookos, A review of the frequency estimation and tracking problems, Tech. rep., Australian National University (1993).
[37] E. Jacobsen, P. J. Kootsookos, Fast, accurate frequency estimators, IEEE Signal Processing Magazine 24 (3) (2007) 123–125.
[38] B. Bernd, U. Ligges, C. Weihs, Frequency estimation by DFT interpolation: a comparison of methods, Tech. rep., Technische Universität Dortmund (2009).
[39] D. Gabor, Theory of communication. Part 1: The analysis of information, Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering 93 (26) (1946) 439–441.
[40] J. Allen, L. Rabiner, A unified approach to short-time Fourier analysis and synthesis, Proceedings of the IEEE 65 (11) (1977) 1558–1564.
[41] A. Grossmann, J. Morlet, Decomposition of Hardy functions into square integrable wavelets of constant shape, SIAM journal on mathematical analysis 15 (4) (1984) 723–736.
[42] S. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. on PAMI 11 (7) (1989) 674–693.
[43] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proceedings of the Royal Society 454 (1971) (1998) 903–995.
[44] H.-R. Ke, K.-C. Wang, C.-I. Yang, K.-F. Chang, Wavelet and Hilbert-Huang transform based on predicting stock forecasting in second-order autoregressive mode, Int. Journal of Applied Physics and Mathematics 4 (1) (2014) 9–14.
[45] G. Rilling, P. Flandrin, P. Goncalves, On empirical mode decomposition and its algorithms, in: Proc.

of NSIP'03, 2003, pp. 8–11.
[46] F. Cong, T. Sipola, T. Huttunen-Scott, X. Xu, T. Ristaniemi, H. Lyytinen, Hilbert-Huang versus Morlet wavelet transformation on mismatch negativity of children in uninterrupted sound paradigm, Nonlinear biomedical physics 3 (1).
[47] N. Padmaja, S. Varadarajan, Atmospheric signal processing using wavelets and HHT, Journal of Computations & Modelling 1 (1) (2011) 17–30.
[48] M. Vlachos, P. S. Yu, V. Castelli, On periodicity detection and structural periodic similarity, in: Proc. of SIAM'05, Vol. 119, 2005, pp. 449–460.
[49] S. Papadimitriou, A. Brockwell, C. Faloutsos, Adaptive, hands-off stream mining, in: Proc. of VLDB'03, 2003, pp. 560–571.
[50] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, Knowledge and information Systems 3 (3) (2001) 263–286.
[51] E. Keogh, K. Chakrabarti, S. Mehrotra, M. Pazzani, Locally adaptive dimensionality reduction for indexing large time series databases, ACM Trans. on Database Systems 27 (2) (2002) 188–228.
[52] Y. Cai, R. Ng, Indexing spatio-temporal trajectories with Chebyshev polynomials, in: Proc. of SIGMOD'04, 2004, pp. 1–12.
[53] B. Ozden, S. Ramaswamy, A. Silberschatz, Cyclic association rules, in: Proc. of ICDE'98, 1998, pp. 412–421.
[54] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proc. of ICDE'95, 1995, pp. 3–14.
[55] C.-H. Lee, M.-S. Chen, C.-R. Lin, Progressive partition miner: an efficient algorithm for mining general temporal association rules, IEEE Trans. on Knowledge and Data Engineering 15 (4) (2003) 1004–1017.
[56] N. Méger, C. Rigotti, Constraint-based mining of episode rules and optimal window sizes, in: Proc. of PKDD'04, 2004, pp. 313–324.
[57] C. Berberidis, I. P. Vlahavas, W. G. Aref, M. J. Atallah, A. K. Elmagarmid, On the discovery of weak periodicities in large time series, in: Proc. of PKDD'02, 2002, pp. 51–61.
[58] S. Ma, J. L. Hellerstein, Mining partially periodic event patterns with unknown periods, in: Proc. of ICDE'01, 2001, pp. 205–214.
[59] G. Moyse, M.-J. Lesot, Fast and incremental erosion score computation, in: Proc. of IPMU'14, 2014, pp. 376–385.
[60] J. Serra, Introduction to mathematical morphology, Computer Vision, Graphics, and Image Processing 35 (3) (1986) 283–305.
[61] S. Gorard, Revisiting a 90-year-old debate: the advantages of the mean deviation, British Journal of Educational Studies 53 (4) (2005) 417–430.
[62] M. Basseville, I. Nikiforov, Detection of abrupt changes: theory and application, Vol. 104, Prentice Hall Englewood Cliffs, 1993.
[63] R. Castillo-Ortega, D. Sánchez, N. Marín, Time series comparison using linguistic fuzzy techniques, Computational Intelligence for Knowledge-Based Systems Design 6178 (2010) 330–339.
[64] R. Moore, Interval arithmetic and automatic error analysis in digital computing, Ph.D. thesis, Stanford University (1963).
[65] RATP, Qualité de l'air mesurée dans nos stations - http://data.ratp.fr/ (2012).
[66] W. Feller, An introduction to probability theory and its application, 3rd Ed., Vol. 1, John Wiley & Sons, Inc., 1967.

Appendix A. Determination of $P\,(d = \delta|n, g)$

Denoting d the random variable measuring the absolute deviation of the sizes of g groups containing n points, we compute in this section $P\,(d = \delta\,|n, g\,)$, the probability that d equals δ given n and g. This probability is defined only if $n \geq g$ and $g \geq 1$.

The proof is split in two parts. First, a general expression for d is given allowing its computation by decomposing the group sizes into two, the ones whose size is smaller than

the average, and the ones whose size is larger than the average.

Based on this decomposition, the probability is computed as the ratio between the possible combinations of group sizes yielding a deviation δ divided by the total number of possible combinations of n points in g groups.

Appendix A.1. Alternate expression for d

In this subsection, an alternate expression is given for d, based on the splitting of the group sizes around the average.

Since each data point belongs to one and only one group, the average group size μ is:

$$\mu = \frac{1}{g}\sum_{j=1}^{g} s_j = \frac{n}{g} \tag{A.1}$$

For convenience in further expressions, we define $\mu^- = \lfloor \mu \rfloor$ and $\mu^+ = \lceil \mu \rceil$.

From the definition of the average deviation d (see Eq. (2)):

$$d = \frac{1}{g}\sum_{j=1}^{g} |s_j - \mu|$$

denoting L the set of group indices whose sizes are smaller or equal to μ and U the set of group indices whose size are greater than μ, l the cardinality of L and u the cardinality of U, it holds that $L \cup U = \{1, ..., g\}$ and $l + u = g$ and d can be written:

$$d = \frac{1}{g}\left(\sum_{j\in L}(\mu - s_j) + \sum_{j\in U}(s_j - \mu)\right) = \frac{1}{g}\left(l\mu - u\mu + \sum_{j\in U} s_j - \sum_{j\in L} s_j\right) \tag{A.2}$$

Moreover, as the sum of all group sizes equals n, $\sum_{j\in U} s_j = n - \sum_{j\in L} s_j$, so denoting $\theta = \sum_{j\in L} s_j$, (A.2) becomes:

$$d = \frac{1}{g}\Big(l\mu - (g-l)\mu + n - \theta - \theta\Big) = \frac{2}{g}\left(l\mu - \theta\right) \tag{A.3}$$

This expression directly yields $g^2 d = 2\left(nl - g\theta\right)$, so $g^2 d \in \mathbb{N}$ since n, l and θ are integers. This yields $g^2\delta \notin \mathbb{N} \Rightarrow P\left(d = \delta | n, g\right) = 0$. In the remainder of the proof, we assume $g^2\delta \in \mathbb{N}$.

Appendix A.2. Combinations yielding d = δ

The second part of the proof is a combinatorial problem aiming at finding the number of possible point distributions such that the deviation of their group sizes equals δ.

Eq. (A.3) implies that the deviation can be computed knowing only the sizes of groups whose indices belong to L. So the first step details the number of possible sizes of L sets, or values of l, for a given δ.

For each possible l, the number of possible point distribution is computed. Counting this number is a known arithmetic problem related to the number of compositions of θ into l groups verifying some constraints, as detailed in the second subsection.

The third step combines the two results in order to evaluate the number of distributions such that the deviation of the group sizes equals δ given the possible l.

Appendix A.2.1. Set of possible values for l given δ

The number of possible l given δ is determined thanks to the constraints on l and θ.

First of all, $1 \leq l < g$. The special case where $l = g$ only happens when all groups have μ^- elements, or equivalently when $\mu^- = \mu$, i.e. $n \mod g = 0$.

Moreover, the groups whose indices belong to L are such that $\forall j \in L$, $1 \leq s_j \leq \mu^-$, so by definition of θ, $l \leq \theta \leq l\mu^-$. Symmetrically, since $u = g - l$, $\forall j \in U$, $\mu^+ \leq s_j$, yielding $\theta \leq n - u\mu^+$. So $\theta \leq \min(l\mu^-, n - u\mu^+)$.

Furthermore, θ is an integer by definition, since it is the sum of the integer number of points in the groups of L. So the possible values for l are such that $\theta \in \mathbb{N}$.

With these constraints, the set of sizes of the possible L sets is denoted Λ and defined as:

$$\begin{aligned}\Lambda = &\left\{1 \leq l < g \text{ s.t. } l \leq \theta \leq \min\left(l\mu^-, n - u\mu^+\right) \wedge \theta \in \mathbb{N}\right\} \\ &\cup \{g \text{ if } n \bmod g = 0\}\end{aligned} \tag{A.4}$$

Appendix A.2.2. Number of compositions of n points into g groups of size in $[a, b]$

It can first be noted that the points in a time series are ordered, yielding constraints on the possible number of g groups built from n points. More precisely, creating g groups from n ordered points is actually equivalent to selecting $g - 1$ bars and placing them in the $n - 1$ gaps between each data points [66] p.38. So the number of ways of grouping n points into g groups is here equivalent the number of compositions of n into g due to the nature of the considered data.

General case. The number of compositions of n into g groups such that each group contain at least a points and at most b points is denoted $N(n, g, a, b)$.

If $bg < n < ag$ or $b < a$ then $N(n, g, a, b) = 0$. Moreover, if $n = g$ and $a = 1 < b$, then $N(n, g, a, b) = 1$.

Subtracting $a - 1$ to the g groups, $N(n, g, a, b)$ can be evaluated as the number of compositions of $n - g(a - 1)$ points in g groups containing at least 1 and at most $b - a + 1$ points. So:

$$N(n, g, a, b) = N(n - g(a - 1), g, 1, b - a + 1) \tag{A.5}$$

The number of compositions of n into g groups containing at least 1 and at most b points can be computed as the sum of the number of groups having exactly k groups among g containing b elements times the number of compositions of $n - kb$ into $g - k$ groups containing at least 1 and at most $b - 1$ elements, i.e.:

$$N(n, g, 1, b) = \sum_{k=0}^{\lfloor n/b \rfloor} \binom{g}{k} \times N(n - kb, g - k, 1, b - 1) \tag{A.6}$$

The last argument $b-1$ in the recursive expression ensures its convergence thanks to the constraint $b < a \Rightarrow N(n, g, a, b) = 0$ shown above. Moreover, the summation stops when

$k = \lfloor n/b \rfloor$ since the constraint $bg < n \Rightarrow N(n, g, a, b) = 0$ implies that there cannot be more than $\lfloor n/b \rfloor$ groups containing exactly b elements[1].

In the case where no constraint is given for the group sizes, which we denote $N(n, g, 1, +\infty)$, the number of compositions equals:

$$N(n, g, 1, +\infty) = \binom{n-1}{g-1} \tag{A.7}$$

This comes from the gaps and bars argument given at the beginning of this subsection.

Appendix A.3. Number of compositions given l and δ

For fixed δ, n and g and a given l, the number $\widetilde{N}$ of compositions with deviation δ is equal to the number of ways of making L and U sets with their respective constraints. Since θ can be computed from Eq. (A.3) as $nl/g + g\delta/2$, it is the product of the number of compositions of θ in l groups having at least 1 and at most μ^- elements times the number of compositions of $n - \theta$ in u groups having at least μ^+ and at most $n - \theta - (u-1)\mu^+$ elements, times the number of ways of choosing l elements among g groups, i.e.:

$$\widetilde{N}(n, g, l, \delta) = \binom{g}{l} \times N\left(\theta, l, 1, \mu^-\right) \times N\left(n - \theta - \mu^-, u, 1, n - \theta - u\mu^+ + 1\right) \tag{A.8}$$

Appendix A.4. Probability of $d = \delta$ given n and g

The probability is finally computed as the ratio between the number of compositions of n into g groups such that their deviation is δ and the number of all compositions of n in g groups, i.e.:

$$P(d = \delta \,|n, g) = \begin{cases} 0 & \text{if } n < g \text{ or } g < 1 \text{ or } g^2\delta \notin \mathbb{N} \\ \dfrac{\sum_{l \in \Lambda} \widetilde{N}(n, g, l, \delta)}{N(n, g, 1, +\infty)} & \text{otherwise} \end{cases} \tag{A.9}$$

Appendix A.5. Implementation details for the computation of $P(d = \delta|n, g)$

The function dedicated to the computation of $P(d = \delta|n, g)$ takes δ, n and g as parameters and returns the corresponding probability.

In order to avoid floating point issues when computing $2nl - g^2\delta \mod 2g$ while determining Λ, the implemented solution keeps the integer value $g^2\delta$ in memory and passes it as a parameter to the function. Indeed, since g is a constant, $P(d = \delta \,|n, g) = P(dg^2 = \delta g^2 \,|n, g)$.

Moreover, in the main program, the value $g^2\delta$ is computed when evaluating the deviation for the high and low value groups. So the principle here is simply to keep this value instead of dividing it before multiplying it again in the probability computation function, thus removing the floating point computation potential errors.

[1]This proposal comes from a discussion on the Math Stack Exchange web site (`http://math.stackexchange.com/questions/900828/number-of-groups-containing-at-least-1-and-at-most-k-elements`)

Appendix B. Detailed experimental results

Table B.2: Percentage of occurrences of each value among the 30 best (i.e. minimum) zone determination error zE across all scenarios, configurations and repetitions, for each method and for each parameter

zE	m1		m2		m3		fm1		fm2		fm3	
%	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg
Alpha												
1%	81%	97%	27%	7%	74%	95%	18%	33%	0%	0%	10%	8%
5%	19%	3%	54%	73%	26%	5%	19%	21%	13%	22%	20%	21%
10%	0%	0%	19%	18%	0%	0%	41%	28%	39%	35%	45%	45%
15%	0%	0%	0%	2%	0%	0%	22%	17%	48%	43%	25%	26%
Pi min												
0,2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0,4	0%	0%	0%	0%	0%	0%	3%	5%	0%	13%	0%	0%
0,6	14%	31%	27%	10%	11%	18%	15%	26%	40%	58%	10%	8%
0,8	86%	69%	73%	90%	89%	82%	82%	69%	60%	29%	90%	92%
minSize												
2							21%	11%	16%	41%	30%	32%
4							31%	22%	16%	18%	30%	30%
6							25%	30%	39%	20%	23%	21%
8							22%	37%	29%	21%	17%	16%
minSep												
2							18%	37%	44%	28%	14%	15%
4							15%	37%	41%	12%	14%	11%
6							48%	26%	13%	23%	41%	42%
8							19%	0%	2%	37%	31%	32%

Table B.3: Average zone determination error zE, for each method for each scenario

zE	m1		m2		m3		fm1		fm2		fm3	
	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg
S1	354%	206%	206%	137%	406%	325%	46%	23%	33%	23%	66%	48%
S2	140%	105%	89%	83%	177%	155%	17%	13%	18%	18%	24%	22%
S3	97%	98%	71%	72%	107%	96%	20%	20%	25%	25%	21%	21%
S4	49%	46%	26%	25%	65%	62%	33%	33%	33%	33%	32%	33%
S5	69%	69%	54%	54%	69%	69%	49%	46%	28%	28%	49%	49%
S6	86%	151%	108%	0%	39%	0%	3%	16%	19%	0%	0%	0%
μ	133%	113%	92%	62%	144%	118%	28%	25%	26%	21%	32%	29%
σ	113%	58%	62%	48%	137%	113%	18%	13%	7%	11%	23%	19%
Rank	11	9	8	7	12	10	4	2	3	1	6	5

Table B.4: Percentage of occurrences of each value among the 30 best point classification result pC across all scenarios, configurations and repetitions, for each method and for each parameter

pC	m1		m2		m3		fm1		fm2		fm3	
%	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg
Alpha												
1%	0%	0%	0%	8%	0%	0%	0%	1%	0%	0%	0%	0%
5%	46%	55%	8%	22%	41%	69%	30%	35%	27%	14%	15%	3%
10%	53%	39%	56%	47%	56%	31%	38%	38%	39%	36%	45%	51%
15%	0%	6%	36%	23%	3%	0%	32%	25%	34%	51%	40%	46%
Pi min												
0.2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0.4	0%	0%	0%	8%	0%	0%	0%	0%	14%	4%	0%	0%
0.6	0%	0%	100%	92%	0%	0%	8%	28%	62%	94%	0%	0%
0.8	100%	100%	0%	0%	100%	100%	92%	72%	25%	2%	100%	100%
minSize												
2							1%	0%	1%	0%	1%	21%
4							17%	11%	16%	17%	23%	31%
6							30%	32%	36%	47%	25%	24%
8							52%	57%	47%	36%	51%	24%
minSep												
2							48%	57%	87%	50%	36%	27%
4							38%	43%	13%	17%	29%	31%
6							15%	0%	0%	15%	28%	30%
8							0%	0%	0%	18%	7%	12%

Table B.5: Average point classification result pC, for each method for each scenario

pC	m1		m2		m3		fm1		fm2		fm3	
	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg	Avg	wAvg
S1	89%	94%	92%	93%	82%	87%	92%	96%	94%	94%	86%	90%
S2	91%	93%	90%	90%	85%	86%	93%	94%	91%	90%	88%	88%
S3	86%	85%	83%	82%	81%	81%	88%	86%	84%	83%	83%	82%
S4	94%	94%	90%	90%	90%	90%	94%	94%	89%	89%	91%	91%
S5	80%	84%	91%	91%	76%	76%	86%	90%	95%	95%	81%	81%
S6	78%	63%	59%	90%	86%	96%	80%	55%	53%	97%	92%	100%
μ	86%	86%	84%	89%	83%	86%	89%	86%	84%	91%	87%	89%
σ	6%	12%	13%	4%	5%	7%	5%	16%	16%	5%	4%	7%
Rank	6	9	11	2	12	7	3	8	10	1	5	4