On Periodicity Detection and Structural Periodic Similarity

Michail Vlachos Philip Yu Vittorio Castelli

IBM T.J. Watson Research Center
19 Skyline Dr, Hawthorne, NY

Abstract

This work motivates the need for more flexible structural similarity measures between time-series sequences, which are based on the extraction of important periodic features. Specifically, we present non-parametric methods for accurate periodicity detection and we introduce new periodic distance measures for time-series sequences. The goal of these tools and techniques are to assist in detecting, monitoring and visualizing structural periodic changes. It is our belief that these methods can be directly applicable in the manufacturing industry for preventive maintenance and in the medical sciences for accurate classification and anomaly detection.

1 Introduction

In spite of the fact that in the past decade we have experienced a profusion of time-series distance measures and representations [9], the majority of them attempt to characterize the similarity between sequences based solely on shape. However, it is becoming increasingly apparent that *structural* similarities can provide more intuitive sequence characterizations that adhere more tightly to human perception of similarity.

While shape-based similarity methods seek to identify homomorphic sequences using the original raw data, structure-based methodologies are designed to find latent similarities, possibly by transforming the sequences into a new domain, where the resemblance can be more apparent. For example, in [6] the authors use change-point-detection signatures for identifying sequences that exhibit similar structural changes. In [7] Kalpakis, et al., use the *cepstrum* for clustering sequences that share a similar underlying ARIMA generative process. Keogh, et al. [10], employ a compression-based dissimilarity measure that is effectively used for clustering and anomaly detection. Finally, Vlachos, et al. [15] consider structural similarities that are based on burst features of time-series sequences.

In this work we consider methods for efficiently capturing and characterizing the periodicity and periodic similarity of time-series. Such techniques can be applicable in a variety of disciplines, such as manufacturing, natural sciences and medicine, which acquire and record large amounts of periodic data. For the analysis of such data, first there is a need for accurate periodicity estimation, which can be utilized either for anomaly detection or for prediction purposes. Then, a structural distance measure should be deployed that can effectively incorporate the periodicity for quantifying the degree of similarity between sequences. A periodic measure can allow for more meaningful and accurate clustering and classification, and can also be used for interactive exploration (and visualization) of massive periodic datasets. Let us consider areas where periodic measures can be applicable:

- In *natural sciences*, many processes manifest strong or weak periodic behavior, such as tidal patterns (oceanography), sunspots (astronomy), temperature changes (meteorology), etc. Periodic analysis and periodicity estimation is an important aspect in these disciplines, because they can suggest potential anomalies or help understand the causal relationship between different processes. For example, it is well established that solar variability greatly affects the climate change. In fact the solar cycle (sunspot numbers) presents striking resemblance to the northern hemisphere land temperatures [4].

- In *medicine*, where many biometric measures (e.g., heartbeats) exhibit strong periodicities, there is a great interest in detecting periodic anomalies. Disturbances of similar periodic patterns can be noted in many degenerative diseases; for example, it has been noted that Tourette's syndrome patients exhibit elevated eyeblink rate [14], while people affected by Parkison's disease show symptoms of gait disturbances [1]. The tools that we provide here, can significantly enhance the early detection of such changes.

- Finally, periodic analysis is an indispensable tool in *automotive, aviation and manufacturing* industries for machine monitoring and diagnostics [12]. Predictive maintenance can be possible by examination of the vibration spectrum caused by its rotating parts. Therefore, a change in the periodic structure of machine vibrations can be a good indicator of machine wear and/or of an incipient failure.

This work targets similar applications and provides tools that can significantly ease the "mining" of useful information. Specifically, this paper makes the following contributions:

1. We present a novel automatic method for accurate periodicity detection in time-series data. Our algorithm is the first one that exploits the information in both periodogram and autocorrelation to provide accurate periodic estimates without upsampling.

2. We introduce new periodic distance measures that exploit the power of the dominant periods, as provided by the Fourier Transform. By ignoring the phase information we can provide more compact representations, that also capture similarities under time-shift transformations.

3. Finally, we present comprehensive experiments demonstrating the applicability and efficiency of the proposed methods, on a variety of real world datasets (online query logs, manufacturing diagnostics, medical data, etc.).

2 Background

We provide a brief introduction to harmonic analysis using the discrete Fourier Transform, because we will use these tools as the building blocks of our algorithms.

2.1 Discrete Fourier Transform. The normalized Discrete Fourier Transform of a sequence $x(n), n = 0, 1 \dots N-1$ is a sequence of complex numbers $X(f)$:

$$X(f_{k/N}) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}}, \quad k = 0, 1 \dots N-1$$

where the subscript k/N denotes the frequency that each coefficient captures. Throughout the text we will also utilize the notation $\mathcal{F}(x)$ to describe the Fourier Transform. Since we are dealing with real signals, the Fourier coefficients are symmetric around the middle one (or to be more exact, they will be the complex conjugate of their symmetric). The Fourier transform represents the original signal as a linear combination of the complex sinusoids $s_f(n) = \frac{e^{j2\pi fn/N}}{\sqrt{N}}$. Therefore, the Fourier coefficients record the amplitude and phase of these sinusoids, after signal x is projected on them.

We can return from the frequency domain back to the time domain, using the inverse Fourier transform $\mathcal{F}^{-1}(x) \equiv x(n)$:

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(f_{k/N}) e^{\frac{j2\pi kn}{N}}, \quad n = 0, 1 \dots N-1$$

Note that if during this reverse transformation we discard some of the coefficients (e.g., the last k), then the outcome will be an approximation of the original sequence (Figure 1). By carefully selecting which coefficients to record, we can perform a variety of tasks such as compression, denoising, etc.

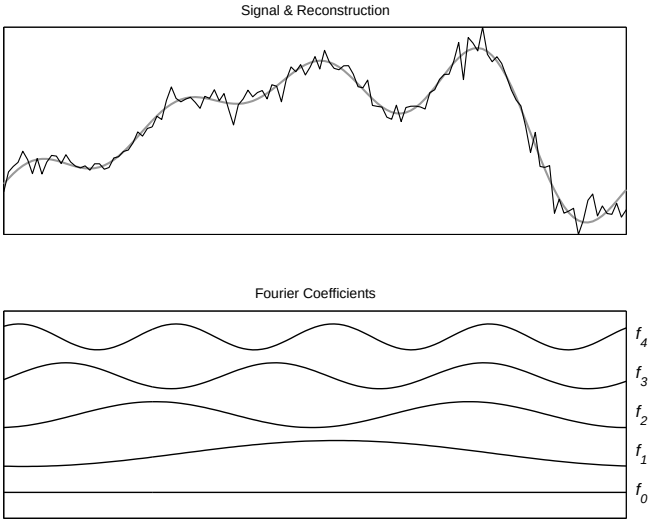

Figure 1: Reconstruction of a signal from its first 5 Fourier coefficients

2.2 Power Spectral Density Estimation. In order to discover potential periodicities of a time-series, one needs to examine its *power spectral density* (PSD or power spectrum). The PSD essentially tells us how much is the expected signal power at each frequency of the signal. Since period is the inverse of frequency, by identifying the frequencies that carry most of the energy, we can also discover the most dominant periods. There are two well known estimators of the PSD; the periodogram and the circular autocorrelation. Both of these methods can be computed using the DFT of a sequence (and can therefore exploit the Fast Fourier Transform for execution in $O(N \log N)$ time).

2.2.1 Periodogram Suppose that X is the DFT of a sequence x. The *periodogram* $\mathcal{P}$ is provided by the squared length of each Fourier coefficient:

$$\mathcal{P}(f_{k/N}) = \|X(f_{k/N})\|^2 \quad k = 0, 1 \dots \lceil \tfrac{N-1}{2} \rceil$$

Notice that we can only detect frequencies that are at most half of the maximum signal frequency, due to the Nyquist fundamental theorem. In order to find the k dominant periods, we need to pick the k largest values of the periodogram. [1]

[1] Due to the assumption of the Fourier Transform that the data is periodic, proper windowing of the data might be necessary for achieving a more accurate harmonic analysis. In this work we will sidestep this issue, since it goes beyond the scope of this paper. However, the interested reader is directed to [5] for an excellent review of data windowing techniques.

Each element of the periodogram provides the power at frequency k/N or, equivalently, at period N/k. Being more precise, each DFT 'bin' corresponds to a *range* of periods (or frequencies). That is, coefficient $X(f_{k/N})$ corresponds to periods $[\frac{N}{k} \dots \frac{N}{k-1})$. It is easy to see that the resolution of the periodogram becomes very coarse for longer periods. For example, for a sequence of length $N = 256$, the DFT bin margins will be $N/1, N/2, N/3, \dots = 256, 128, 64$ etc.

Essentially, the accuracy of the discovered periods, deteriorates for large periods, due to the increasing width of the DFT bins (N/k). Another related issue is *spectral leakage*, which causes frequencies that are not integer multiples of the DFT bin width, to disperse over the entire spectrum. This can lead to 'false alarms' in the periodogram. However, the periodogram can still provide an accurate indicator of important short (to medium) length periods. Additionally, through the periodogram it is easy to automate the extraction of important periods (peaks) by examining the statistical properties of the Fourier coefficients (such as in [15]).

2.2.2 Circular Autocorrelation. The second way to estimate the dominant periods of a time-series x, is to calculate the circular AutoCorrelation Function (or ACF), which examines how similar a sequence is to its previous values for different τ lags:

$$ACF(\tau) = \frac{1}{N} \sum_{n=0}^{N-1} x(\tau) \cdot x(n+\tau)$$

Therefore, the autocorrelation is formally a convolution, and we can avoid the quadratic calculation in the time domain by computing it efficiently as a dot product in the frequency domain using the normalized Fourier transform:

$$ACF = \mathcal{F}^{-1} < X, X^* >$$

The star ($*$) symbol denotes complex conjugation.

The ACF provides a more fine-grained periodicity detector than the periodogram, hence it can pinpoint with greater accuracy even larger periods. However, it is not sufficient by itself for automatic periodicity discovery for the following reasons:

1. Automated discovery of important peaks is more difficult than in the periodogram. Approaches that utilize forms of autocorrelation require the user to manually set the significance threshold (such as in [2, 3]).

2. Even if the user picks the level of significance, multiples of the same basic period also appear as peaks. Therefore, the method introduces many false alarms that need to be eliminated in a post-processing phase.

3. Low amplitude events of high frequency may appear less important (i.e., have lower peaks) than high amplitude patterns, which nonetheless appear more scarcely (see example in fig. 2).

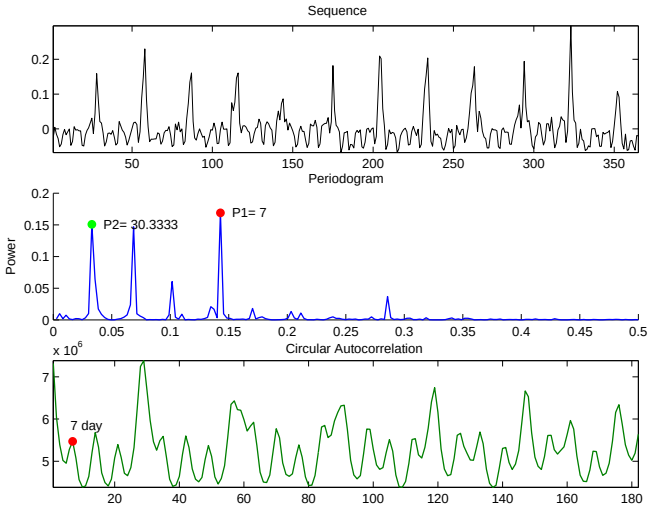

Figure 2: The 7 day period is latent in the autocorrelation graph, because it has lower amplitude (even though it happens with higher frequency). However, the 7 day peak is very obvious in the Periodogram.

The advantages and shortcomings of the periodogram and the ACF are summarized in Table 1.

From the above discussion one can realize that although the periodogram and the autocorrelation cannot provide sufficient spectral information separately, there is a lot of potential when both methods are combined. We delineate our approach in the following section.

3 Our Approach

We utilize a two-tier approach, by considering the information in both the autocorrelation and the periodogram. We call this method `AUTOPERIOD`. Since the discovery of important periods is more difficult on the autocorrelation, we can use the periodogram for extracting period candidates. Let's call the period candidates 'hints'. These 'hints' may be false (due to spectral leakage), or provide a coarse estimate of the period (remember that DFT bins increase gradually in size); therefore a verification phase using the autocorrelation is required, since it provides a more fine-grained estimation of potential periodicities. The intuition is that if the candidate period from the periodogram lies on a hill of the ACF then we can consider it as a valid period, otherwise we discard it as false alarm. For the periods that reside on a hill, further refinement may be required if the periodicity hint refers to a large period.

Figure 3 summarizes our methodology and Figure 4 depicts the visual intuition behind our approach with a working example. The sequence is obtained from the

Method	Easy to threshold	Accurate short periods	Accurate large periods	Complexity
Periodogram	yes	yes	no	$O(NlogN)$
Autocorrelation	no	yes	yes	$O(NlogN)$
Combination	yes	yes	yes	$O(NlogN)$

Table 1: Concise comparison of approaches for periodicity detection.

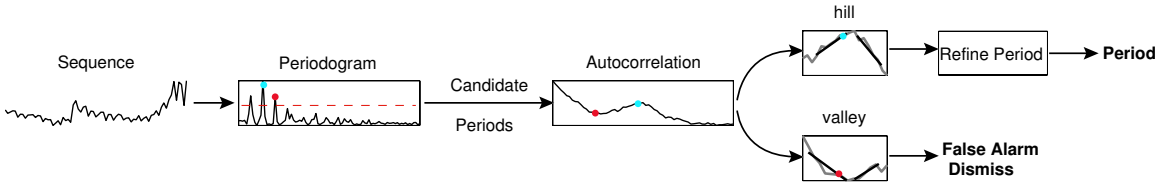

Figure 3: Diagram of our methodology (`AUTOPERIOD` method)

MSN query request logs and represents the aggregate demand for the query 'Easter' for 1000 days after the beginning of 2002. The demand for the specific query peaks during Easter time and we can observe one yearly peak. Our intuition is that periodicity should be approximately 365 (although not exactly, since Easter is not celebrated at the same date every year). Indeed the most dominant periodogram estimate is 333.33 = (1000/3), which is located on a hill of the ACF, with a peak at 357 (the correct periodicity -at least for this 3 year span). The remaining periodic hints can be discarded upon verification with the autocorrelation.

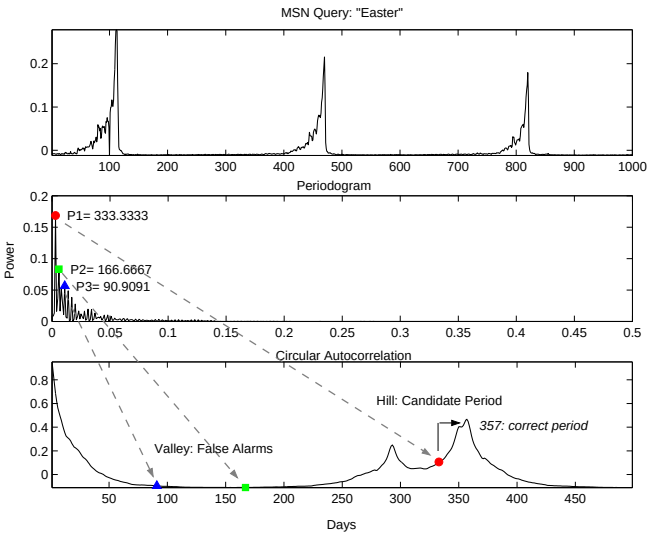

Figure 4: Visual demonstration of our method. Candidate periods from the periodogram are verified against the autocorrelation. Valid periods are further refined utilizing the autocorrelation information.

Essentially, we have leveraged the information of both metrics for providing an accurate periodicity detector. In addition, our method is computationally efficient, because both the periodogram and the ACF can be directly computed through the Fast Fourier Transform of the examined sequence in $O(N \log N)$ time.

3.1 Discussion. First, we need to clarify succinctly that the use of the combined periodogram and autocorrelation does not carry additional information than each metric separately. This perhaps surprising statement can be verified by noting that:

$$< X, X^* >= \|X\|^2$$

Therefore, the autocorrelation is the inverse Fourier transform of the periodogram, which means that the ACF can be considered as the dual of the periodogram, from the time into the frequency domain. In essence, our intention is to solve each problem in its proper domain; (i) the period significance in the frequency domain, and (ii) the identification of the exact period in the time domain.

Another issue that we would like to clarify is the reason that we are not considering a (seemingly) simpler approach for accurate periodicity estimation.

Looking at the problem from a signal processing perspective, one could argue that the inability to discover the correct period is due to the 'coarse' sampling of the series. If we would like to increase the resolution of the DFT, we could 'sample' our dataset at a finer resolution (upsampling). Higher sampling rate essentially translates into padding the time-series with zeros, and calculating the DFT of the longer time-series. Indeed, if we increase the size of the example sequence from 1000 to 16000, we will be able to discover the correct periodicity which is 357 (instead of the incorrect 333, given in the original estimate).

However, upsampling also imposes a significant performance overhead. If we are interested in obtaining online periodicity estimates from a data stream, this alternative method may result in a serious system

bottleneck. We can see this analytically; the time required to compute the FFT of a sequence with length 2^x is in the order of $2^x log 2^x = x2^x$. Now let's assume that we pad the sequence with zeros increasing its length 16 times (just like in our working example). The FFT now requires time in the order of $(x+4)2^{x+4}$, which after algebraic calculations translates into 2 orders of magnitude additional time.

Using our methodology, we do not require higher sampling rates for the FFT calculation, hence keeping a low computational profile.

3.2 Discovery of Candidate Periods. For extracting a set of candidate periodicities from the periodogram, one needs to determine an appropriate power threshold that should distinguish only the dominant frequencies (or inversely the dominant periods). If none of the sequence frequencies exceeds the specific threshold (i.e., the set of periodicity 'hints' is empty), then we can regard the sequence as non-periodic.

In order to specify which periods are important, we first need to identify how much of the signal energy is attributed to random mechanisms, that is, everything that could not have been attributed to a random process should be of interest.

Let us assume that we examine a sequence x. The outcome of a permutation on the elements of x is a sequence $\tilde{x}$. The new sequence will retain the first order statistics of the original sequence, but will not exhibit any pattern or periodicities, because of the 'scrambling' process (even though such characteristics may have existed in sequence x). Anything that has the structure of $\tilde{x}$ is not interesting and should be discarded, therefore at this step we can record the maximum power (p_{max}) that $\tilde{x}$ exhibits, at any frequency f.

$$p_{max} = arg \max_f \|\tilde{X}(f)\|^2$$

Only if a frequency of x has more power than p_{max} can be considered interesting. If we would like to provide a 99% confidence interval on what frequencies are important, we should repeat the above experiment 100 times and record for each one the maximum power of the permuted sequence $\tilde{x}$. The 99^{th} largest value of these 100 experiments, will provide a sufficient estimator of the power threshold p_T that we are seeking. Periods (in the original sequence periodogram) whose power is more than the derived threshold will be considered:

$$p_{hint} = \{N/k : \mathcal{P}(f_{k/N}) > p_T\}$$

Finally, an additional period 'trimming' should be performed for discarding periods that are either too large or too small and therefore cannot be considered reliable. In this phase any periodic hint greater than N/2 or smaller than 2 is removed.

Figure 5 captures a pseudo-code of the algorithm for identifying periodic hints.

```
1  periods = getPeriodHints(Q)
2  {
3    k = 100;  // number of permutations
4    maxPower = {};  // empty set
5    periods = {};
6
7    for i = 1 to k
8    {
9      Qp = permute(Q);
10     P = getPeriodogram(Qp);
11
12     power = max(P.power);
13     maxPower.add(power);
14   }
15
16   percentile = 99;
17   maxPower.sort; // ascending
18   P_threshold = maxPower(maxPower.length*(percentile/100));
19
20   P = getPeriodogram(Qp);
21
22   for i = 1 to P.length
23   {
24     if (P[i].power > P_threshold)
25       periods.add(P);  // new candidate period
26   }
27
28   // period trimming
29   N = Q.length;
30   for i = 1 to periods.length
31   {
32     if (periods[i].hint >= N/2 || periods[i].hint <= 2)
33       periods[i].erase();
34   }
35
36   return periods;
37 }
```

Figure 5: Algorithm `getPeriodHints`

In [15] another algorithm for detection of important periods was proposed, which follows a different concept for estimating the periodogram threshold. The assumption there was that the periodogram of non-periodic time-series will follow an exponential distribution, which returned very intuitive period estimates for real world datasets. In our experiments, we have found the two algorithms to return very comparable threshold values. However, because the new method does not make any assumptions about the underlying distribution, we expect it to be applicable for a wider variety of time-series processes.

Examples: We use sequences from the MSN query logs (yearly span) to demonstrate the usefulness of the discovered periodic hints. In Figure 6(a) we present the demand of the query *'stock market'*, where we can distinguish a strong weekly component in the periodogram. Figure 6(b) depicts the query *'weekend'* which does not contain any obvious periodicities. Our method can set the threshold high enough, therefore avoiding false

alarms.

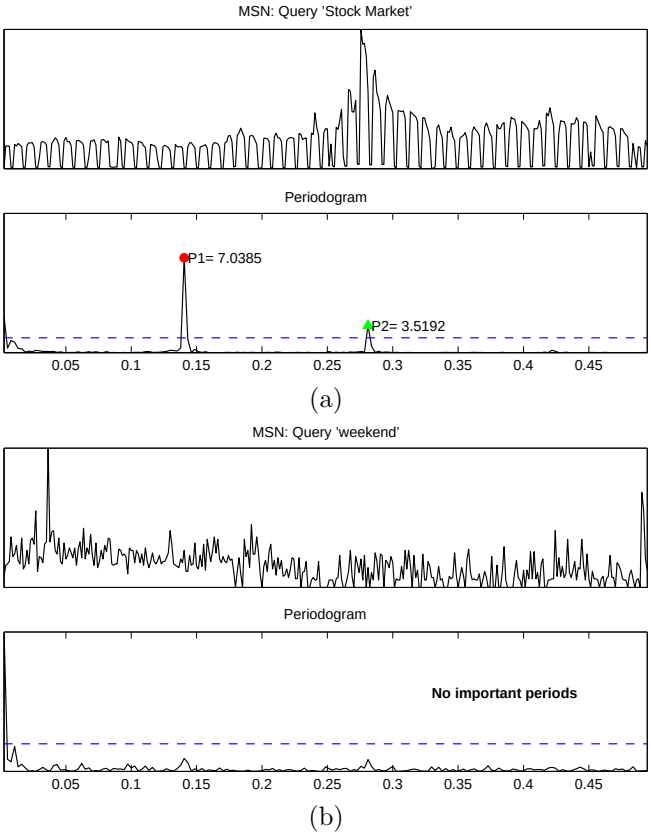

Figure 6: (a) Query 'stock market' (2002): Weekly periodic hint is identified. (b) Query 'weekend' (2002): No significant periodicities are spotted.

3.3 Verification of Candidate Periods. After the periodogram peaks have been identified, we have obtained a candidate set of periodicities for the examined sequence. The validity of these periods will be verified against the autocorrelation. An indication that a period is important, can be the fact that the corresponding period lies on a *hill* of the autocorrelation. If the period resides on a *valley* then it can be considered spurious and therefore safely discarded.

After we discover that a periodicity 'hint' resides on a hill of the autocorrelation, we can refine it even further by identifying the closest peak (i.e., local maximum). This is a necessary step, because the correct periodicity (i.e., peak of the hill) might not have been discovered by the periodogram, if it was derived from a 'wide' DFT bin. This is generally true for larger periods, where the resolution of the DFT bins drops significantly. We will explicate further, how to address the above issues:

3.3.1 Validity of Periodicity Hint. The significance of a candidate period ideally can be determined by examining the curvature of the ACF around the candidate period p. The autocorrelation is concave downward, if the second derivative is negative in an open interval $(a \dots b)$:

$$\frac{\partial^2 ACF(x)}{\partial x^2} < 0, \text{for all } x \in (a \dots b), a < p < b$$

Nevertheless, small perturbations of the ACF due to the existence of noise, may invalidate the above requirement. Will will seek a more robust estimator of the curvature by approximating the ACF in the proximity of the candidate period with two linear segments. Then it is sufficient to examine if the approximating segments exhibit an upward-downward trend, for identifying a concave downward pattern (i.e., a hill).

The segmentation of a sequence of length N into k linear segments can be computed optimally using a dynamic programming algorithm in $O(N^2k)$ time, while a greedy merge algorithm achieves results very close to optimal in $O(N \log N)$ time [8]. For this problem instance, however, one can employ a simpler algorithm, because we require only a two segment approximation for a specific portion of the ACF.

Let $\hat{S}_a^b$ be the linear regression of a sequence x between the positions $[a \dots b]$ and $\epsilon(\hat{S}_a^b)$ be the error introduced by the approximating segment. The best split position t_{split} is derived from the configuration that minimizes the total approximation error:

$$t_{split} = arg \min_t \epsilon(\hat{S}_1^t) + \epsilon(\hat{S}_{t+1}^n)$$

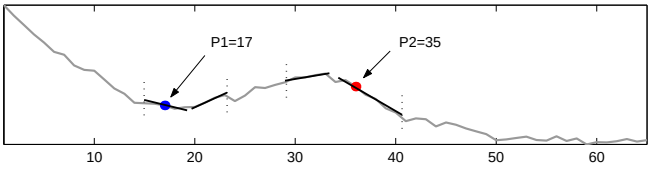

Figure 7: Segmentation of two autocorrelation intervals into two linear segments. The left region indicates a concave upward trend ('valley') while the right part consists of a concave downward trend ('hill'). Only the candidate period 35 can be considered valid, since it is located on a hill.

There is still the issue of the width of the search interval on the ACF, that is how much should we extend our search for a hill around the candidate period. Since the periodicity hint might have leaked from adjacent DFT bins (if it was located near the margin of the bin) we also examine half of the adjacent bins as well. Therefore, for a hint at period N/k, we examine the range $R_{N/k}$ of the ACF for the existence of a hill:

$$R_{N/k} = [\tfrac{1}{2}(\tfrac{N}{k+1} + \tfrac{N}{k}) - 1, \ldots, \tfrac{1}{2}(\tfrac{N}{k} + \tfrac{N}{k-1}) + 1]$$

3.3.2 Identification of closest Peak. After we have ascertained that a candidate period belongs on a hill and not on a valley of the ACF, we need to discover the closest peak which will return a more accurate estimate of the periodicity hint (particularly for larger periods). We can proceed in two ways; the first one would be to perform any hill-climbing technique, such as gradient ascent, for discovering the local maximum. In this manner the local search will be directed toward the positive direction of the first derivative. Alternatively, we could derive the peak position directly from the linear segmentation of the ACF, which is already computed in the hill detection phase. The peak should be located either at the end of the first segment or at the beginning of the second segment.

We have implemented both methods for the purposes of our experiments and we found both of them to report accurate results.

4 Extension for Streaming Data.

Even though we have presented the `AUTOPERIOD` algorithm for static time-series, it can be easily extended for a streaming scenario, by adapting an incremental calculation of the Fourier Transform. Incremental Fourier computation has been a topic of interest since the late 70s and it was introduced by Papoulis [13] under the term 'Momentary Fourier Transform' (MFT). MFT covered the aggregate (or growing) window case, however recent implementations also deal with the sliding window case, such as in [16, 11]. Incremental `AUTOPERIOD` requires only constant update time per DFT coefficient, and linear space for recording the window data.

5 Accuracy of Results

We use several sequences from the MSN query logs to perform convincing experiments regarding the accuracy of our 2-tier methodology. The specific dataset is ideal for our purposes because we can detect a number of different periodicities according to the demand pattern of each query.

The examples in Figure 8 demonstrate a variety of situations that might occur when using both the periodogram and autocorrelation.

■ *Query 'Easter'(MSN):* Examining the demand for a period of 1000 days, we can discover several periodic hints above the power threshold in the periodogram. In this example, the autocorrelation information refines the original periodogram hint (from 333 → 357). Additional hints are rejected because they reside on ACF valleys (in the figure only the top 3 candidate periods are displayed for reasons of clarity).

■ *Query 'Harry Potter'(MSN):* For the specific query although there are no observed periodicities (duration 365 days), the periodogram returns 3 periodic hints, which are mostly attributed to the burst pattern during November when the movie was released. The hints are classified as spurious upon verification with ACF.

■ *Query 'Fourier'(MSN):* This is an example where the periodogram threshold effectively does not return candidate periods. Notice that if we had utilized only the autocorrelation information, it would have been more troublesome to discover which (if any) periods were important. This represents another validation that our choice to perform the period thresholding in the frequency space was correct.

■ *Economic Index (Stock Market):* Finally, this last sequence from a stock market index illustrates a case where both the periodogram and autocorrelation information concur on the single (albeit weak) periodicity.

Through this experimental testbed we have demonstrated that `AUTOPERIOD` can provide very accurate periodicity estimates without upsampling the original sequence. In the sections that follow, we will show how it can be used in conjunction with periodic similarity measures, for interactive exploration of sequence databases.

6 Structure-Based Similarity and Periodic Measures

We introduce structural measures that are based on *periodic* features extracted from sequences. Periodic distance measures can be used for providing more meaningful structural clustering and visualization of sequences (whether they are periodic or not). After sequences are grouped in 'periodic' clusters, using a 'drill-down' process the user can selectively apply the `AUTOPERIOD` method for periodicity estimation on the sequences or clusters of interest. In the experimental section we provide examples of this methodology using hierarchical clustering trees.

Let us consider first the utility of periodic distance measures with an example. Suppose that one is examining the similarity between the two time-series of Figure 9. When sequence A exhibits an upward trend, sequence B displays a downward drift. Obviously, the Euclidean distance (or inner product) between sequences A and B, will characterize them as very different. However, if we exploit the frequency content of the sequences and evaluate their periodogram, we will discover that it is almost identical. In this new space, the Euclidean distance can easily identify the sequence similarities. Even though this specific example could have been addressed in the original space using the Dynamic Time Warping

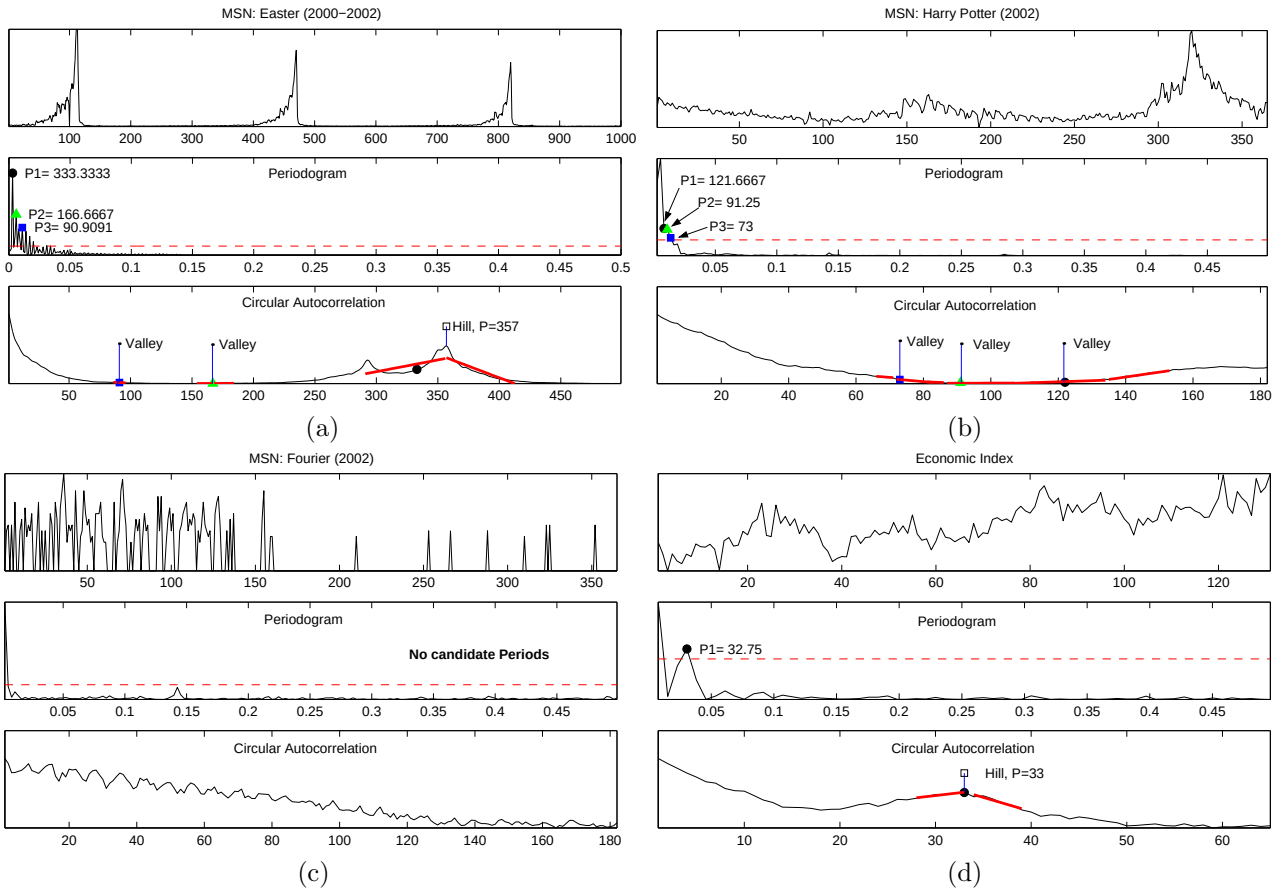

Figure 8: Periodicity detection results of the `AUTOPERIOD` method.

(DTW) distance, we have to note that our method is significantly more efficient (in terms of both time and space) than DTW. Additionally, periodic measures can address more subtle similarities that DTW cannot capture, such as *different* patterns/shapes occurring at periodic (possibly non-aligned) intervals. We will examine cases where the DTW fails in the sections that follow.

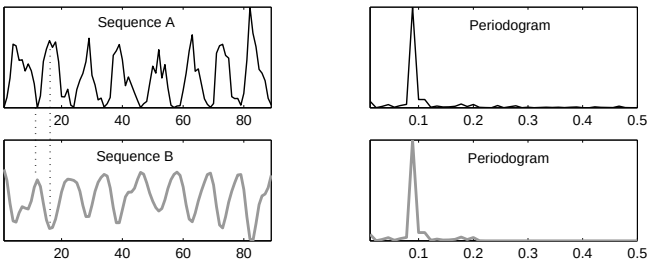

Figure 9: Sequences A and B are very distant in a Euclidean sense in the time domain. A transformation in the frequency domain (using a view of the periodogram) reveals the structural similarities.

The new measure of structural similarity that we present, exploits the *power* content of only the most dominant periods/frequencies. By considering the most powerful frequencies, our method concentrates on the most important structural characteristics, effectively filtering out the negative influence of noise, and eventually allowing for expedited distance computation. Additionally, the omission of the phase information renders the new similarity measure shift invariant in the time domain. We can therefore discover time-series with similar patterns, which may occur at different chronological instants.

6.1 Power Distance (`pDist`). For comparing the periodic structure of two sequences, we need to examine how different is their harmonic content. We achieve this by utilizing the periodogram and specifically the frequencies with the highest energy.

Suppose that X is the Fourier transform of a sequence x with length n. We can discover the k largest coefficients of X by computing its periodogram $\mathcal{P}(X)$ and recording the position of the k frequencies with the highest power content (parameter k depends

on the desired compression factor). Let us denote the vector holding the positions of the coefficients with the largest power p^+ (so $p^+ \subset [1 \dots n]$). To compare x with any other sequence q, one needs to examine how similar energies they carry in the dominant periods of x. Therefore, we evaluate $\mathcal{P}(Q(p^+))$, that describes a sequence holding the equivalent coefficients as the vector $\mathcal{P}(X(p^+))$. The distance `pDist` between these two vectors captures the periodic similarity between sequences x and q:

$$pDist = \|\mathcal{P}(Q(p^+)) - \mathcal{P}(X(p^+))\|$$

Example: Let x and q be two sequences and let their respective Fourier Transforms be $X = \{(1+2i), (2+2i), (1+i), (5+i)\}$ and $Q = \{(2+2i), (1+i), (3+i), (1+2i)\}$. The periodogram vector of X is: $\mathcal{P}(X) = \|X\|^2 = (5, 8, 2, 26)$. The vector holding the positions of X with highest energy is $p^+ = (2, 4)$ and therefore $\mathcal{P}(X(p^+)) = (0, 8, 0, 26)$. Finally, since $\mathcal{P}(Q) = (8, 2, 10, 5)$ we have that: $\mathcal{P}(Q(p^+)) = (0, 2, 0, 5)$ [2].

In order to meaningfully compare the power content of two sequences we need to normalize them, so that they contain the same amount of total energy. We can assign to any sequence $x(n)$ unit power, by performing the following normalization:

$$\hat{x}(n) = \frac{x(n) - \frac{1}{N}\sum_{i=1}^{N} x(i)}{\sqrt{\sum_{i=1}^{N}(x(n) - \frac{1}{N}\sum_{i=1}^{N} x(i))^2}}, \quad n = 1, \dots, N$$

The above transformation will lead to zero mean value and sum of squared values equal to 1. Parseval's theorem dictates that the energy in the time domain equals the energy in the frequency domain, therefore the total energy in the frequency domain should also be unit:

$$\|\hat{x}\|^2 = \|\mathcal{F}(\hat{x})\|^2 = 1$$

After this normalization, we can more meaningfully compare the periodogram energies.

Indexability: Although in this work we are not going to discuss now to index the `pDist`, we would like to note that this is possible. The representation that we are proposing, utilizes a different set of coefficients for every sequence. While indexing might appear problematic using space partitioning indices such as R-trees (because they operate on a fixed set of dimensions/coefficients), such representations can be easily indexed using metric tree structures, such as VP-Tree or M-Tree (more details can be found in [15]).

[2] The zeros are placed in the vectors for clarity reasons. In the actual calculations they can be omitted.

7 Periodic Measure Results

We present extensive experiments that show the usefulness of the new periodic measures and we compare them with widely used shape based measures or newly introduced structural distance measures.

7.1 MSN query logs. Using 16 sequences which record the yearly demand of several keywords at the MSN search engine, we perform the hierarchical clustering which is shown in Figure 10. In the dendrogram derived using the `pDist` as the distance function, we can notice a distinct separation of the sequences/keywords into 3 classes. The first class contains no clear periodicities (no specific pattern in the demand of the query), while the second one exhibits only bursty seasonal trends (e.g., during Christmas). The final category of queries are requested with high frequency (weekly period) and here we can find keywords such as 'cinema', 'bank', 'Bush' etc.

We utilize an extended portion of the same dataset for exploring the visualization power of periodic distance measures. Using the pairwise distance matrix between a set of MSN keyword demand sequences (365 values, year 2002), we evaluate a 2D mapping of the keywords using Multidimensional Scaling (Figure 11). The derived mapping shows the high discriminatory efficacy of the `pDist` measure; seasonal trends (low frequencies) are disjoint from periodic patterns (high frequencies), allowing for a more structural sequence exploration. Keywords like *'fall'*, *'Christmas'*, *'lord of the rings'*, *'Elvis'*, etc, manifest mainly seasonal bursts, which need not be aligned in the time axis. On the contrary, queries like *'dry cleaners'* or *'Friday'* indicate a natural weekly repeated demand. Finally, some queries do not exhibit any obvious periodicities within a year's time (e.g., *'icdm'*, *'kdd'*, etc).

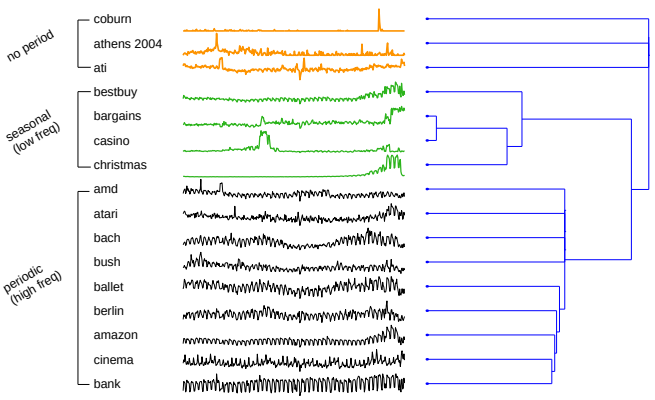

Figure 10: Dendrogram based on periodic features

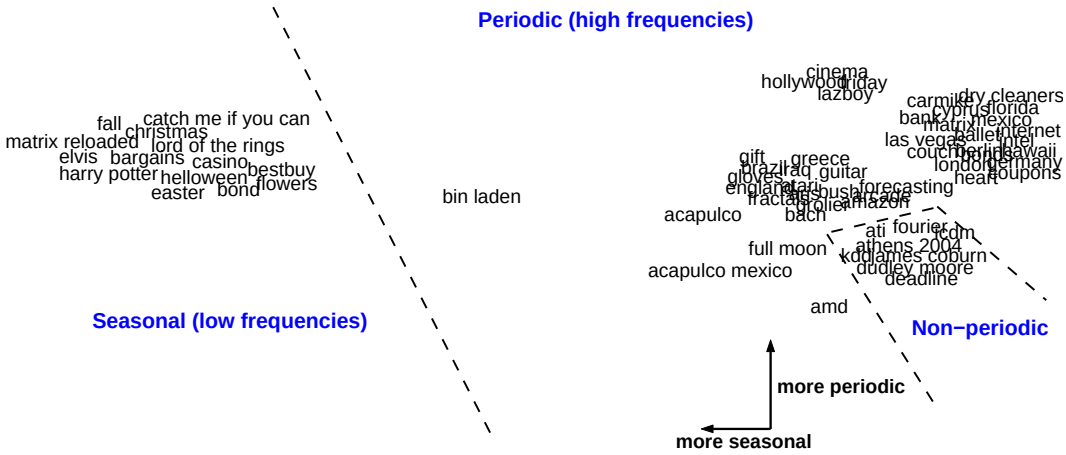

Figure 11: Mapping on 2D of pairwise distances between several sequences. The similarity measure utilized was the power based similarity. We can clearly distinguish a separation between periodic and seasonal trends.

7.2 Structured + Random Mixture. For our second experiment we use a combination of periodic time-series that are collected from natural sciences, medicine and manufacturing, augmented by pairs of random noise and random walk data.

All datasets come in pairs, hence, when performing a hierarchical clustering algorithm on this dataset, we expect to find a direct linkage of each sequence pair at the lower level of the dendrogram. If this happens we consider the clustering correct. The dataset consists of 12 pairs, therefore a measure of the clustering accuracy can be the number of correct pair linkages, over twelve, the number of total pairs.

Figure 12 displays the resulting dendrogram for the `pDist` measure, which achieves a perfect clustering. We can also observe that pairs derived from the same source/process are clustered together as well, in the higher dendrogram level (Power Demand, ECG, MotorCurrent etc). After the clustering, we can execute the `AUTOPERIOD` method and annotate the dendrogram with the important periods of every sequence. Some sequences, like the random walk or the random data, do not contain any periodicities, which we indicate with an empty set {}. When both sequences at the lower level display the same periodicity, a single set is displayed on the bifurcation for clarity.

For many datasets that came into 2 pairs (power demand, video surveillance, motor current), all 4 instances instances demonstrated the same basic period (as suggested by the `AUTOPERIOD`). However, the periodic measure can effectively separate them into two pairs, because the power content of the respective frequencies was different.

For example, in the video surveillance dataset, both actors display a periodic movement every 30 units (drawing a gun from a holster). However, because the male person performs the movement with wider 'arches' (because of different body structure), the periodic measure can distinguish his movement, due to the higher energy content. The above example indicates that analogous periodic measures could be effectively used for biometric characterization, since every individual tends to have a distinct intrinsic rhythm (e.g., when typing on the keyboard, performing repetitive moves, speaking, etc).

On the sunspot sequence set the `AUTOPERIOD` estimates of 89 and 84 units may appear erroneous at first glance, because of our knowledge that the solar cycles range from 10 to 12 years. However, this is not the case because the 1000 sequence points record sunspot measurements of approximately 120 years. After the proper rescaling the estimates of 89 and 84 yield periodicities close to 11 and 10 years respectively.

Euclidean	DTW	Cepstrum	CDM	pDist
0.16	0.66	0.75	1	1

Table 2: Clustering accuracy for the dataset of fig. 12

On the same dataset the accuracy results for Euclidean, DTW, Cepstrum and CDM compression based measure [10] are given in table 2. CDM is the only one that also achieves perfect clustering. However, it should be noted that while all other methods operate on the original dimensional space (using 1000 points), `pDist` works on a very lower dimensional space, using only 50 numbers to describe each sequence, after a 20x compression of the data.

7.3 ECG datasets. Our last experiment is performed on the MIT-BIH Arrhythmia dataset. We use

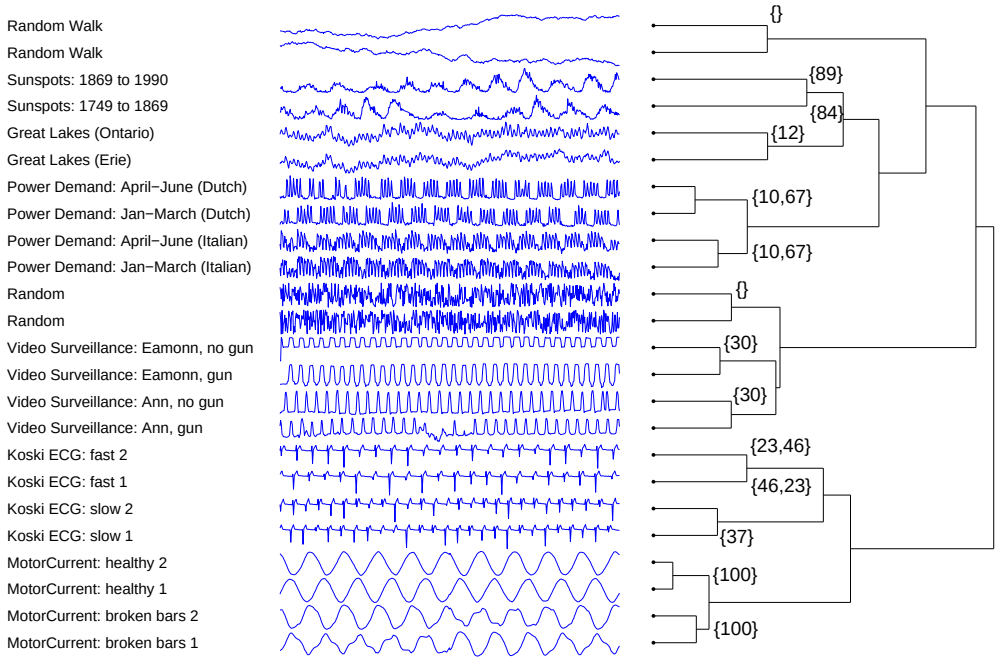

Figure 12: The `pDist` measure produces an accurate dendrogram based on the periodic structural characteristics of a dataset. The lower dendrogram levels are also annotated by the periods discovered as important, by a subsequent run of the `AUTOPERIOD` method.

two sets of sequences; one with 2 classes of heartbeats and another one with three (figures 13, 14). We present the dendrogram of the `pDist` measure and the DTW, which represents possibly one of the best shape based distance measures. To tune the single parameter of the DTW (corresponding to the maximum warping length) we probed several values and here we report the one that returned the best clustering.

For both dataset instances, `pDist` again returns an accurate clustering, while DTW seems to perform badly on the high level dendrogram aggregations, hence not leading to perfect class separation. The Euclidean distance reported worse results. The CDM measure is accurate on the 2 class separation test but does not provide a perfect separation for the 3 class problem (see the original paper [10] for respective results).

7.4 Distance Measures Overview. The experiments have testified to the utility of periodic measures for exploration of sequence databases. The only real contender to the `pDist` measure is the compression-based CDM measure. However, compared to CDM our approach presents some favorable advantages: (i) it does not require any discretization phase (we operate on the original data), (ii) it is meaningful for both long and short sequences (CDM performs better on longer sequences) (iii) it can be easily extended for streaming sequences, using incremental Fourier Transform computation (iv) it provides additional sequence information in the form of periodic estimates.

8 Conclusion

We have presented methods for accurate periodicity estimation and for characterization of structural periodic similarity between sequences. It is our belief that these methods will find many applications for interactive exploration of time-series databases and for classification or anomaly detection of periodic sequences (e.g., in auto manufacturing, biometrics and medical diagnosis).

Acknowledgements: We are thankful to MSN and Microsoft for letting us use a portion of the MSN query logs. We also wish to thank Eamonn Keogh for numerous suggestions and for kindly donating his dendrogram code.

References

[1] G. Ebersbach, M. Sojer, F. Valldeoriola, J. Wissel, J. Müller, E. Tolosa, and W. Poewe. Comparative analysis of gait in parkinson's disease, cerebellar ataxia and subcortical arteriosclerotic encephalopathy. In *Brain, Vol. 122, No. 7, 1349-1355, July 1999.*

[2] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid.

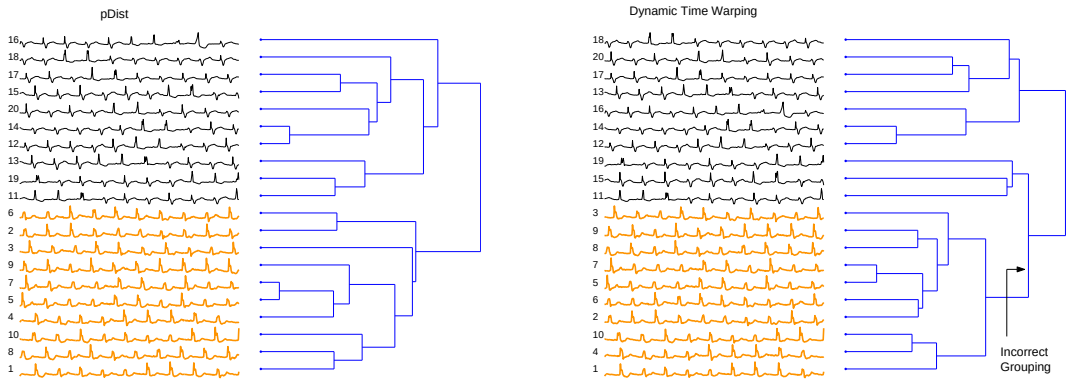

Figure 13: 2 class ECG problem: DTW provides incorrect grouping

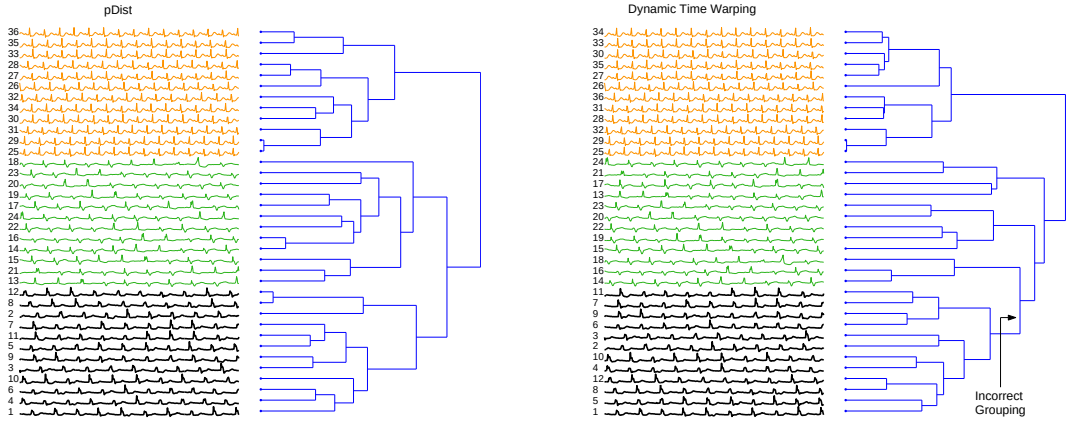

Figure 14: 3 class ECG problem: only `pDist` provides correct clustering into 3 groups

Using Convolution to Mine Obscure Periodic Patterns in One Pass. In *Proc. of EDBT*, 2004.

[3] F. Ergün, S. Muthukrishnan, and S. C. Sahinalp. Sublinear methods for detecting periodic trends in data streams. In *LATIN*, 2004.

[4] E. Friss-Cristensen and K. Lassen. Length of solar cycle - An Indicator of solar-activity closely related with climate. In *Science, 254*, pages 698–700, 1991.

[5] F. J. Harris. On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. In *Proc. of the IEEE, Vol. 66, No 1*, 1978.

[6] T. Idé and K. Inoue. Knowledge Discovery from Time-Series Data using Nonlinear Transformations. In *Proc. of the 4th Data Mining Workshop (Japan Soc. for Software Science and Technology*, 2004.

[7] K. Kalpakis, D. Gada, and V. Puttagunta. Distance Measures for Effective Clustering of ARIMA Time-Series. In *Proc. of ICDM*, 2001.

[8] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Proc. of ICDM*, 2001.

[9] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. In *Proc. of SIGKDD*, 2002.

[10] E. Keogh, S. Lonardi, and A. Ratanamahatana. Towards parameter-free data mining. In *Proc. of SIGKDD*, 2004.

[11] M. Kontaki and A. Papadopoulos. Efficient similarity search in streaming time sequences. In *SSDBM*, 2004.

[12] J. S. Mitchell. *An introduction to machinery analysis and monitoring*. PennWell Publ. Co., 1993.

[13] A. Papoulis. *Signal Analysis*. McGraw-Hill, 1977.

[14] J. Tulena, M. Azzolini, J. de Vriesa, W. H. Groeneveld, J. Passchier, and B. van de Wetering. Quantitative study of spontaneous eye blinks and eye tics in Gilles de la Tourette's syndrome. In *Journal of Neurol. Neurosurg. Psychiatry 1999,67:800-802*.

[15] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identification of Similarities, Periodicities & Bursts for Online Search Queries. In *Proc. of SIGMOD*, 2004.

[16] Y. Zhu and D. Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *VLDB*, 2002.