MBPD: Motif-Based Period Detection

Rasaq Otunba[1], Jessica Lin[1], and Pavel Senin[2]

[1] George Mason University, Fairfax, VA 22030, USA
{rotunba, jessica}@gmu.edu
[2] University of Hawaii, Honolulu, HI 96822, USA
senin@hawaii.edu

Abstract. Massive amounts of data are generated daily at a rapid rate. As a result, the world is faced with unprecedented challenges and opportunities on managing the ever-growing data. These challenges are prevalent in time series for obvious reasons. Clearly, there is an urgent need for efficient solutions to mine large-scale time series databases. One of such data mining tasks is periodicity mining. Efficient and effective periodicity mining techniques in big data would be useful in cases such as finding animal migration patterns, analysis of stock market data for periodicity, and outlier detection in electrocardiogram (ECG), analyses of periodic disease outbreak etc. This work utilizes the notion of time series motifs for approximate period detection. Specifically, we present a novel and simple method to detect periods on time series data based on recurrent patterns. Our approach is effective, noise-resilient, and efficient. Experimental results show that our approach is superior compared to a popularly used period detection technique with respect to accuracy while requiring much less time and space.

Keywords: Motif, period, time series, Big Data

1 Introduction

Periodicity is the tendency of a pattern to recur at regular intervals, which are referred to as periods. Periodic patterns occur in many natural phenomena or human activities. Examples include an employee's daily work schedule, yearly migration pattern of animals, regional sunspot cycle etc. These data can be so large and complex that it becomes difficult to process using traditional database management tools or data processing applications. Such data is typically referred to as big data. The fact that we are in the era of big data cannot be overemphasized especially with the large amount of available data from the internet and ubiquitous computing devices that are now parts of our everyday lives. Detecting the period in data, big or "*not so big*", can provide useful insight on the data, help make better predictions, detect anomalies and improve similarity matching [16] among other things. It's imperative to point out that the focus of this work being time series makes it suitable for other kinds of data such as multimedia because they can be converted to time series e.g. the extraction of MFCC from audio as it is used for one of the datasets in our experiments. Several methods have been proposed to detect periods in data. Most of the existing methods are particularly suitable for perfect periods, which is hardly the case in most natural phenomena. While the related problem of finding the exact period of a time series is a simpler one to solve, it may be too restrictive for real-world phenomena. Periods in real datasets are typically noisy and incomplete. That is, while the periodic patterns exhibit tangible similarity, they may not always be identical and equally distributed. These factors warrant robust approximate period detection schemes like our solution and it has even been shown that in many applications, approximate solutions are sufficient [30]. A robust solution should also be able to detect periods in an efficient manner. In general, three types of periodic patterns can be detected in a time series as illustrated by Rasheed et al. [1] and they are described as follows:

- a time series exhibits partial periodicity if at least one symbol in addition to at least one variable symbol is periodic. For instance, in time series T = wxyz wxxy wxyy wxwz, the sequence wx is periodic with period p = 4; and the partial periodic pattern wx ** exists in T, where * denotes a variable symbol.
- a time series exhibits symbol periodicity if at most one symbol is repeated periodically. For example, in time series T = xyz xzy xxy xyy, symbol x is periodic with period p = 3. We consider this to be a special case of partial periodicity when the periodic subsequence has one symbol and argue that a technique that can detect subsequence periodicity can detect symbol periodicity.
- a time series exhibits segment periodicity if an entire pattern is periodic. For instance, the time series T = wxyz wxyz wxyz wxyz has a segment period p = 4. The periodic segment is wxyz.

Most techniques are suitable for discrete sequences. However, time series are real-valued data. To adapt the periodicity definitions described above, we need a pre-processing step that discretizes the real-valued time series into a symbolic representation. The standard pre-processing approach is to use SAX (Symbolic Aggregate approXimation) [2], a well-known discretization technique, to convert a time series into a string or a set of strings [3]. In this work, we propose a novel technique to detect periods in time series data by learning the repeated patterns (motifs) from data. To the best of our knowledge, this work is the first to use motif discovery as a means to period detection in time series.

A time series motif is a pattern that consists of two or more similar subsequences based on some distance threshold [3]. While our approach can work with any motif discovery algorithms, in this work we focus on a recently proposed variable-length motif algorithm based on grammar induction called GrammarViz[1] [3]. GrammarViz consists of two major steps: (1) extracting subsequences via a sliding window converting them to strings via SAX; and (2) infers a set of context-free grammar rules on the sequence of strings using Sequitur [4]. The grammar rules represent repeated patterns in the sequence, each of which can be regarded a time series motif. GrammarViz is an ideal basis for our work because of its simplicity, space- and time-efficiency, and most importantly, its ability to detect variable-length motifs, as these properties are transferred to our periodicity detection method.

In summary, we propose a simple and elegant approximate periodicity detection algorithm, Motif-Based Period Detection (MBPD). Our work makes the following novel contributions:

- We propose to use time series motif discovery on the string representation as an antecedent to approximate periodicity detection on the original time series.
- Our algorithm is both time- and space-efficient, and is suitable for streaming data.
- We introduce a simple ranking method for the most significant period.
- We extended GrammarViz and implemented the periodicity visualization feature that allows users to navigate and sort the detected periods.
- We conducted experiments to compare the performance of our technique against other popular techniques on synthetic and real datasets.

The rest of the paper is organized as follows. Section 2 discusses related work while section 3 outlines preliminaries. We describe our approach in Section 4. Section 5 describes the experiments performed. We conclude with limitations of our approach and make recommendations for improvement as future work in Section 6.

2 Related Work

Existing periodicity detection methods can be categorized based on a number of factors [1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 24, 25]. These factors include parameter dependency, the type of periodicity detected, the span of periodicity detected and the domain in which the periodicity is detected. Some methods require the specification of the period value [12, 13, 14]. This is not ideal as the detection of the period value is in itself a task worthy of due consideration. Some of these methods detect only symbol periodicity [8, 26]. Yang et al. [9, 24] proposed a linear time distance-based technique for discovering the potential periods in a time series. However, their method fails to detect some valid periods because only adjacent intervals are considered. Rasheed at al. [1] proposed an algorithm to detect periodicity in time series using suffix trees. The time requirement for their proposed method can rise to the order of O(n^3). Most algorithms detect only a subset of the types of periodicity (symbol, sequence or segment) mentioned earlier. The method proposed by Han et al. in [7]

[1] Although not explicitly named in the paper, the authors refer to it as GrammarViz on their website: http://www.cs.gmu.edu/~jessica/GrammarViz.html

detects only segment periodicity. Certain techniques [12, 13], which are based on another technique, ParPer [14], are suitable for detecting sequence periodicity in time series. ParPer makes use of peculiar properties e.g. apriori property related to sequence periodicity in a time series for periodicity detection. Most of the aforementioned techniques suffer from noise sensitivity. WARP [11] was developed to be noise resilient but it detects only segment periodicity. Few techniques [12, 14] detect subsection periodicity while most are meant for full-cycle periodicity detection.

Periodicity detection algorithms can also be classified into time domain and frequency domain methods. Time domain methods are based on autocorrelation functions while frequency domain methods are based on spectral density functions. The premise for using time domain methods is that the autocorrelation function of a periodic data has the same period as the data with peaks obtained at time t=0, period T, and multiples of T. Time domain methods are suitable for sinusoidal signals and they are not noise resilient. Frequency domain methods, on the other hand, decompose signals into constituent frequency components. The result of frequency domain methods is a power spectral density with impulses determined by the corresponding Fourier coefficients. These Fourier coefficients can be extracted to create a periodogram [17].

Autocorrelation and Fourier Transforms are two of the most popular periodicity detection techniques [15]. Autocorrelation is able to detect short and long periods, but creates difficulty in identifying the true period due to the fact that the multiples of the true period will have the same power as the true period. On the other hand, Fourier transforms suffer from a number of problems: spectral leakage, which causes a lot of false positives in the periodogram, and poor estimation of long periods due to issues with low frequency regions or sparseness in data [18]. Some methods combine both autocorrelation and Fourier transforms [6, 16].

Our method is able to detect the most significant period in a dataset without requiring the period value as a parameter and this is done in the time domain. Our method also detects the different types of periodicity.

3 Preliminaries

In this section we define periodicity, approximate periodicity and the problem addressed in this work.

Definition 1. Let S = $t_0, t_1 \ldots t_{n-1}$ be a string representation of a time series with length of *n*, i.e. |S| = n. S is said to be periodic if S(t) = S(t+p), where $t \in N$, $t \geq 0$, $t < n\text{-}p$, T is a subsequence of S such that, T= $t_0, t_1 \ldots t_{p-1}$, |T| = p, $p \geq 1$ and $p \leq n/2$. The smallest such subsequence T is called the period of S. If no such period T can be found in S, S is said to be aperiodic. For example, if S = wxywxywxy, the period T=wxy, p=3.
For clarity, we do not consider the substring of length (m*p) to be a period of S for any m such that $m \neq 1$.

Definition 2. Let S be n-long string over alphabet Σ. Let r be an error function defined on strings. S is called periodic with k error on T if there exists a string T over Σ, such that r(S, T) = k i.e. r evaluate the error of assuming T is the period of S. The string T that evaluates to smallest such k is called the approximate period of S. For example, let S = wxywxywxy, r(S, A) $\geq$ r(S, B) where A = wxy occurring at positions 0, 3 and 6, and B = wxy occurring at positions 0 and 6, A and B are candidate periods, A is the approximate period of S otherwise referred to as the most significant period of S.

Problem Definition. Given a string function r, and String S of length n over alphabet compute the approximate period T under the function r.

3.2 Motif Discovery

Time series motifs are repeated similar patterns. We argue that detecting the frequent patterns could serve as an antecedent to periodicity detection. Many algorithms have been proposed to find motifs in time series data [3, 19, 20, 21, 22]. In this work, we focus on GrammarViz, a fast, approximate variable-length motif discovery algorithm based on grammar induction [3]. The factors we considered in choosing a motif discovery method for this work include efficiency with respect to space and time, the ability to detect the motifs in a streaming fashion, the ability to detect variable length motifs, the ability to detect periodicity in string representation of time series, and simplicity. GrammarViz utilizes Sequitur [4], a context-free grammar induction technique, to derive rules considered to be motifs from string representation of time series. These motifs are mapped back to the original time series to show their occurrences. A benefit derived from the ability to work on string representation is the fact that the technique is applicable to many kinds of data whose dimensionality can be reduced by discretization to string symbols. GrammarViz is the first time series motif discovery algorithm that can detect variable-length motifs in an effective and efficient manner, and it is able to do so in a streaming fashion. The authors of GrammarViz also created a visualization tool for clarity and easy navigation of the produced results. GrammarViz achieves variable-length motif discovery as a result of numerosity reduction, thus making MBPD suitable for cases where a periodic pattern may occur with variable lengths in a time series.

4 Our Approach

The fundament premise of our approach is to first discover the motifs in the time series with high efficiency and effectiveness and then detect the most periodic motif.

Algorithm 1: Motif-Based Period Detection

```
INPUT:  String S of length n over alphabet Σ.
OUTPUT:  The approximate period of S, T.
/* Find the rule/motif objects M = {$m_1$, $m_2$, . . . $m_d$} from GrammarViz algorithm */
M = grammarViz(S);
/* Compute periods and errors for each motif, return the one with the smallest error*/
$m_1$ = periodicity($m_1$);
$p_1$ = $m_1$.getPeriod();
$r_1$ = $m_1$.getError();
```

```
7.  rMin = $r_1$  // store the minimum error in rMin
8.  for each $m_i \in M$ do
9.      $m_i$ = periodicity($m_i$);
10.     $p_i$ = $m_i$.getPeriod();
11.     $r_i$ = $m_i$.getError();
12.     if ($r_i$ < rMin)
13.         approxP = $p_i$;
14.         rMin = $r_i$;
15. end for
16. return approxP;
```

Algorithm 1 shows the pseudocode for the MBPD algorithm. The motif objects returned in Line 2 are stored along with the start and stop positions of each occurrence in the time series. We consider only periods that occur at least 3 times in a time series for this work since anything less has a higher probability of being a false positive but it's trivial to modify the algorithm to detect periods that occur twice if desired. Lines 3-7 and the loop from Lines 8-14 computes the period of each motif, the error (our r function from definition 2) defined by the standard deviation of the intervals of all occurrences and the approximate period. The period of each motif is calculated as the mean of intervals (between the start positions of two consecutive occurrences) of all occurrences in the time series. Both computations of the approximate period and error are done on the original time series after the derived string motifs are mapped back to the original time series. The approximate period of the time series is the period corresponding to the lowest error. The periodicity function called on Lines 4 and 9 of algorithm 1 is shown in algorithm 2.

The efficiency of MBPD largely depends on the efficiency of GrammarViz (Line 2), which has Sequitur at its core. GrammarViz has linear time and space complexity. As a result, the time complexity of MBPD is O(n*k) for a time series of size n, where k is the average number of instances for each motif rule produced by Sequitur. The space complexity is still O(n) because the memory space needed for variables used in algorithms 1 and 2 are negligible. Compared to most existing methods for time series periodicity detection, MBPD has a competitive space and time complexity.

Algorithm 2. Periodicity Algorithm

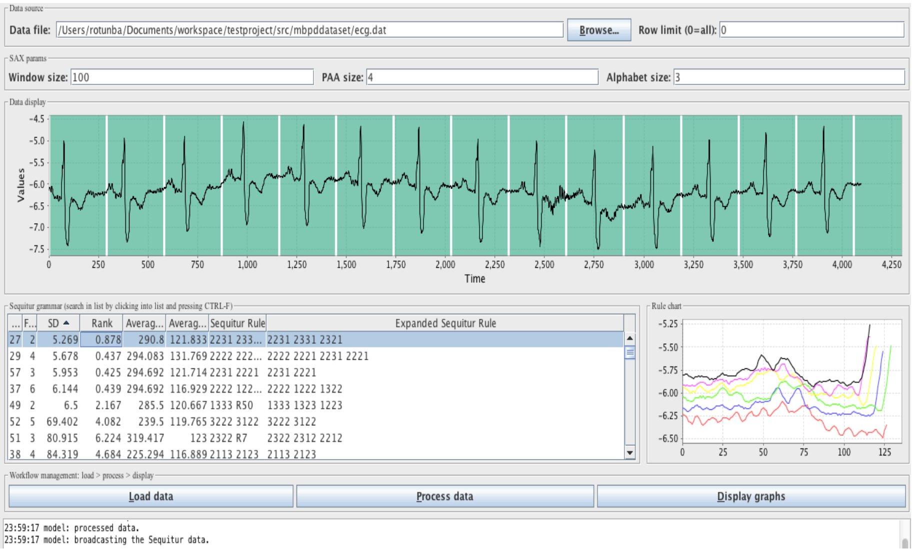
```
INPUT:  Motif M with start positions $A = \{a_1, a_2, . . . a_b\}$ for all b occurrences
OUTPUT:  Motif M with the period and error set respectively

sum_Interval = 0, sqd = 0;
for each $a_i \in A$ do
    sum_Intervals = sum_Intervals + $a_i - a_{i-1}$;
end for
M.period = sum_Intervals/(b-1);
for each $a_i \in A$ do
    sqd = sqd + (($a_i - a_{i-1}$ - M.period) ^ 2);
end for
M.error = (sqd/(b-1)) ^ 0.5;
return M;
```

5 Experiment

In this section we evaluate MBPD on synthetic, pseudo-synthetic and real datasets. Our periodicity detection and visualization software is an extension of GrammarViz. More details about the visualization tool and the GrammarViz algorithm in line 2 of algorithm 1 can be found in [3]. Experiments were performed on a 2.7GHz, Intel Core i7, MAC OS X version 10.7.5 with 8GB memory.

Fig. 1 is a snapshot of the visualization tool showing the approximate periodicity detected in an ECG dataset in the data display section of the figure. Other periods can be viewed by navigating the list of rules in the sequitur grammar section of the figure.



Data file: /Users/rotunba/Documents/workspace/testproject/src/mbpddataset/ecg.dat

Row limit (0=all): 0

Window size: 100

PAA size: 4

Alphabet size: 3

...	F...	SD ▲	Rank	Averag...	Averag...	Sequitur Rule	Expanded Sequitur Rule
27	2	5.269	0.878	290.8	121.833	2231 233...	2231 2331 2321
29	4	5.678	0.437	294.083	131.769	2222 222...	2222 2221 2231 2221
57	3	5.953	0.425	294.692	121.714	2231 2221	2231 2221
37	6	6.144	0.439	294.692	116.929	2222 122...	2222 1222 1322
49	2	6.5	2.167	285.5	120.667	1333 R50	1333 1323 1223
52	5	69.402	4.082	239.5	119.765	3222 3122	3222 3122
51	3	80.915	6.224	319.417	123	2322 R7	2322 2312 2212
38	4	84.319	4.684	225.294	116.889	2113 2123	2113 2123

```
23:59:17 model: processed data.
23:59:17 model: broadcasting the Sequitur data.
```

Fig. 1. Snapshot of MBPD visualization tool detecting periodicity in ECG dataset.

We compare our method with Fast Fourier Transform (FFT). The frequency with the highest spectral power from FFT of the dataset is converted into time domain and considered as the most significant period. FFT is chosen for its suitability for real-valued datasets.

It is worth mentioning that while we considered other state-of-the-art techniques such as STNR [1], WARP [11], and the probability-based method in [18] for comparison, we found that they are not suitable for our purpose for the following reasons other than their unsuitableness for real values. WARP caused an out of memory exception for the large datasets (65636 data points) used in our experiments and returned unintuitive results for most of the other datasets e.g. 515 (43 years) for the Zürich sunspot dataset whereas the proper period for the dataset is 132 (11 years). The out of memory exception is most probably due to WARP's O(m^2) space complexity for a time series of size m.; STNR on the other hand returns many candidate periods even with the pruning techniques suggested, which deviates from our goal of finding the most significant period. Finally the method proposed in [18] is meant for binary sequences representation of Boolean-type observations and not real-valued sequences.

6.2 Datasets

We used 12 datasets of various periodicity, noises, and lengths in our experimental evaluation. We ensured the length of each dataset is a power of 2 to avoid introducing bias by padding the dataset with zeros in order to use FFT for comparison. A subplot of all 12 datasets is shown in Figure 2. Figures 3 and 4 show the periodicity detected in 2 of the datasets used in our experiments.

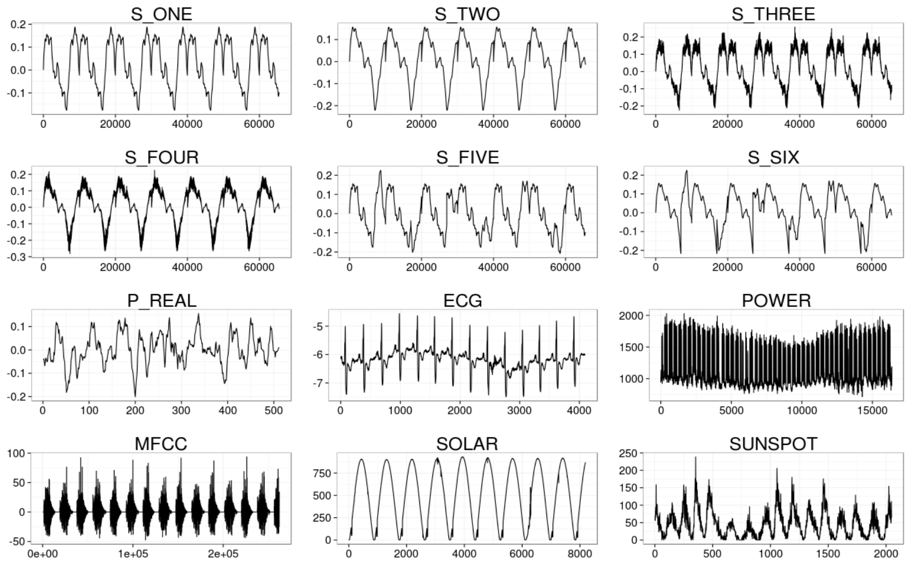


Fig. 2. A subplot of all 12 datasets used for experiments

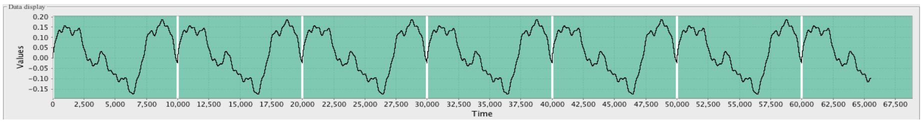


Fig. 3. Snapshot of MBPD visualization tool detecting periodicity in S_ONE dataset.

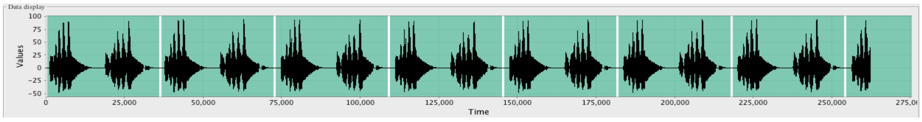


Fig. 4. Snapshot of MBPD visualization tool detecting periodicity in MFCC dataset.

Synthetic datasets. We created 6 synthetic datasets with various properties that could affect performance.

- S_ONE: 65636 points to depict perfect segment periodicity by repeating 10000 points 7 times except for the last 4364 points.
- S_TWO: 65636 points to depict perfect segment periodicity by repeating 10000 points 7 times and except for the last 4364 points.
- S_THREE: 65636 points to demonstrate sensitivity to noise by introducing noise in the form of insertion, deletion and replacement into S_ONE, 10% each.
- S_FOUR: 65636 points to demonstrate sensitivity to noise by introducing noise in the form of insertion, deletion and replacement into S_TWO, 10% each.
- S_FIVE: 65636 points to depict subsequence periodicity by repeating a portion of the repeated 10000 points in S_ONE and leaving the remaining as variable points.
- S_SIX: 65636 points to depict subsequence periodicity by repeating a portion of the repeated 10000 points in S_TWO and leaving the remaining as variable points.

Real datasets. We used 5 real datasets.

- ECG: 4096 points of ECG (electrocardiogram) data
- POWER: 16384 points of power consumption data
- MFCC: 262144 points of MFCC (Mel-Frequency Cepstral Coefficients) extracted from Rufous-collared Sparrow bird song which can be found at http://www.xeno-canto.org/120810
- SOLAR: 8192 points of solar data
- SUNSPOT: 2048 points of sunspots on Zürich.

Pseudo-Real dataset (P_REAL**)**. This dataset has 512 points. We use this dataset to depict periodicity detection in a short dataset. It's a sea surface temperature dataset for North Atlantic Ocean (simulated data for 1000 years) with 10-year moving average smoothing.

6.3 Results

We evaluated the performance of our method against FFT with respect to the ranking error rates on both the synthetic and real datasets as shown in Table 1. In Table 2, we show the error rates of the period value detected on the synthetic datasets.

Ranking error rate is computed by dividing the rank position, *i* (starting from zero) of the most significant period by the size of the time series e.g. if the candidate periods returned by MBPD are 3, 4, 7 and 9 in that order when ranked and 4 is the most significant period, the ranking error rate of MBPD for this dataset is ¼ (0.2500). Error rate of the period values is computed as follows in equation 1:

$$Error\ rate\ of\ period\ = \frac{|\mathbf{Expected\ value - Actual\ value}|}{\mathbf{Actual\ value}} \qquad (1)$$

Since we do not know the exact periods in the real datasets, we did not evaluate the error rate of the period values. Table 3 contains the periods detected on the real datasets as well as the expected range of values. As shown in Table 1 and Table 2, MBPD ranks the most significant period better and detects the period more accurately than FFT. Since we are concerned with the most significant period, an improper ranking otherwise referred to as false dismissal experienced by using FFT is undesirable. Even though we do not know the exact period for the real datasets, the visualization tool helps by allowing us to visualize the results, e.g. the highlighted patterns in Figures 1, 3, and 4.

Table 1. Ranking error rate on synthetic and real datasets.

Datasets	**MBPD**	**FFT**
S_ONE	0.0000	0.0000
S_TWO	0.0000	0.0000
S_THREE	0.0000	0.0000
S_FOUR	0.0000	0.0000
S_FIVE	0.0000	0.0000
S_SIX	0.0000	0.0000
P_REAL	0.0000	0.0000
ECG	0.0000	0.0002
POWER	0.0000	0.0001
MFCC	0.0000	-
SOLAR	0.0000	0.0001
SUNSPOT	0.0000	0.0005

Table 2. Period error rate on synthetic datasets.

Datasets	**MBPD**	**FFT**
S_ONE	0.0000	0.0637
S_TWO	0.0000	0.0637
S_THREE	0.0009	0.0637
S_FOUR	0.0011	0.0637
S_FIVE	0.0000	0.0637
S_SIX	0.0000	0.0923

Table 3. Period values on real datasets.

Datasets	**MBPD**	**FFT**	**EXPECTED VALUES**
P_REAL	77.00	85.33	75 - 85
ECG	290.80	292.57	290 - 295
SOLAR	870.60	910.22	870 - 875
MFCC	36340	-	36000 - 36500
POWER	328.98	334.37	325 - 330
SUNSPOT	135.85	136.53	132 - 137

For some of the real datasets, we have some prior knowledge on what to expect for the periodicity. The Zürich sunspot data, for example, is known to have a period of about 11 years (132 months) as described in [28]. All three techniques produced reasonable approximations for the dataset. The MFCC data is extracted from a bird song, which has a period between 30K - 40K when visualized in the software and listened to meticulously. As seen in table 3, all 3 techniques performed competitively on 4 of the real datasets but only MBPD detected the period in the MFCC extraction. This also makes MBPD stand out as a superior technique. We did not record the period and ranking error rate for FFT on the MFCC dataset because the 10 most significant periods (3.33, 4.00, 3.33, 4.00, 2.86, 2.86, 2.86, 3.33, 3.34, 4.00) were spurious altogether and we don't know which of them should be selected for evaluation as they are all far from the expected period. We consider only the first 10 coefficients in this work because the first 10 coefficients are known to contain approximately 90% of the energy [29]. We attribute the poor result of FFT to the known issue of FFT regarding the low frequency regions which translates to issues in detecting long periods as is the case with the MFCC dataset (35000 - 36000).

7 Conclusion and Future Work

We present an approximate periodicity detection scheme in this work. We evaluated our approach against popular techniques on synthetic and real datasets. Our technique is highly competitive with respect to efficiency and effectiveness as well as being robust to noise. We also utilized a visualization tool for this work. Even though the intention is to detect the most significant approximate period in the dataset, our visualization tool permits the navigation of other periods. As future work, we would like to extend the work to detect the exact or at least approximate span of the periods detected in addition to detecting the periodic pattern with high confidence. As this work seeks to motivate the use of motif discovery as an antecedent to periodicity detection, we do not claim that GrammarViz is the best choice of algorithm for motif discovery. Since GrammarViz is an approximate motif discovery algorithm, it may not find all of the motifs, which in turn may impact the quality of periods detected by our algorithm. We believe that using a better grammar induction algorithm or, more generally, a more aggressive motif discovery technique as an antecedent could enhance the performance of MBPD. Nevertheless, the benefit of finding variable-length patterns and the ability to do so efficiently as permitted by GrammarViz is highly desirable and beneficial to our algorithm.

References

1. Rasheed, F., Al-Shalalfa, M. and Alhajj, R. (2011). Efficient Periodicity Mining in Time Series Databases Using Suffix Trees. In TKDE.
2. J. Lin, E. Keogh, S. Lonardi, and B. Chiu. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. Workshop on Research Issues in DMKD
3. Y. Li, J. Lin, and T. Oates. (2012). Visualizing variable-length time series motifs. In SDM.
4. C.G. Nevill-Manning and I.H. Witten. (1997). Identifying Hierarchical Structure in

Sequences: A linear-time algorithm. In Journal of Artificial Intelligence Research.

5. A. Amir, E. Eisenberg, and A. Elmagarmid. (2010). Approximate Periodicity. In ISAAC.
6. C. Berberidis, W. Aref, M. Atallah, I. Vlahavas, and A. Elmagarmid. (2002). Multiple and Partial Periodicity Mining in Time Series Databases. In ECAI.
7. J. Han, W. Gong, and Y. Yin. (1998). Mining Segment-Wise Periodic Patterns in Time Related Databases. In KDD.
8. S. Ma and J. Hellerstein. (2001). Mining Partially Periodic Event Patterns with Unknown Periods. In ICDE.
9. J. Yang, W. Wang, and P. Yu. (2002). Mining Partial Periodic Patterns with Gap Penalties. In ICDM
10. M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid. (2005). Periodicity Detection in Time Series Databases. In ICDE.
11. M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid. (2005). WARP: Time Warping for Periodicity Detection. In ICDM.
12. C. Sheng, W. Hsu, and M. L. Lee. (2006). Mining Dense Periodic Patterns in Time Series Data. In ICDE.
13. C. Sheng, W. Hsu, and M.L. Lee. (2005). Efficient Mining of Dense Periodic Patterns in Time Series. Technical report, Nat'l Univ. of Singapore.
14. J. Han, Y. Yin, and G. Dong. (1999). Efficient Mining of Partial Periodic Patterns in Time Series Database. In ICDE.
15. M. B. Priestley. Spectral Analysis and Time Series. (1981). London: Academic Press.
16. Vlachos M, Yu PS and Castelli V. (2005). On periodicity detection and structural periodic similarity. In SDM.
17. P. Stoica and R. L. Moses. (1997). Introduction to spectral analysis. Prentice-Hall, Upper Saddle River, NJ.
18. Zhenhui Li, Jingjing Wang and Jiawei Han. (2012). Mining Event Periodicity from Incomplete Observations. In KDD.
19. H. T. Lam, N. D. Pham, and T. Calders. (2011). Online discovery of top-k similar motifs in time series data. SIAM Conference on Data Mining. In SDM.
20. J. Lin, E. Keogh, S. Lonardi, and P. Patel. (2002). Finding motifs in time series. In Proc. of 2nd Workshop on Temporal Data Mining at KDD.
21. A. Mueen and E. J. Keogh. (2010). Online discovery and maintenance of time series motifs. In KDD.
22. P. Nunthanid, V. Niennattrakul, and C. Ratanamahatana. (2011). Discovery of variable length time series motif. In ECTICON.
23. W. F. Smyth. (2007). Computing Periodicities in Strings — A New Approach. In Proceedings of the 16th Australasian Workshop on Combinatorial Algorithms.
24. Yang J, Wang W and Yu PS. (2000). Mining asynchronous periodic patterns in time series data. In KDD.
25. M. Elfeky, W. Aref, and A. Elmagarmid. (2004). Using convolution to mine obscure periodic patterns in one pass. In EDBT.
26. J. D. Scargle. (1982). Studies in astronomical time series analysis. II - statistical aspects of spectral analysis of unevenly spaced data. In Astrophysical Journal.
27. https://jmotif.googlecode.com
28. http://solarscience.msfc.nasa.gov/SunspotCycle.shtml
29. Vlachos M (2005). A Practical Time-Series Tutorial with Matlab. In PKDD
30. S. Arora. (2003). Approximation Schemes for NP-hard Geometric Optimization Problems: A Survey. In Math Prog.