Nonparametric Extrema Analysis in Time Series for Envelope Extraction, Peak Detection and Clustering

Kaan Gokcesu, Hakan Gokcesu

Abstract—In this paper, we propose a nonparametric approach that can be used in envelope extraction, peak-burst detection and clustering in time series. Our problem formalization results in a naturally defined splitting/forking of the time series. With a possibly hierarchical implementation, it can be used for various applications in machine learning, signal processing and mathematical finance. From an incoming input signal, our iterative procedure sequentially creates two signals (one upper bounding and one lower bounding signal) by minimizing the cumulative drift. We show that a solution can be efficiently calculated by use of a Viterbi-like path tracking algorithm together with an optimal elimination rule. We consider many interesting settings, where our algorithm has near-linear time complexities.

I. INTRODUCTION

A. Extrema Analysis

The problem of extrema analysis of time series [1]–[3], whether in the form of envelope extraction [4], peak-burst detection [5], clustering [6] or signal bands/channels [7] have become prominent in machine learning, data mining and mathematical finance. The identification and analysis of the extrema in a given time series is essential in many applications because of its useful topological implications and features [5] in a myriad of fields, including but not limited to mass spectrometry [8], signal processing [9], [10], image processing [11], bioinformatics [12]–[14] and astrophysics [15].

The extrema can implicate certain increases (or decreases) in a time series, which can be used in the problem of classification [16]. For example, the workload (utilization) of a server CPU (processors) can be analyzed to extract idle or busy states, which can be used to forecast future states [17]. The extrema analysis can efficiently track the nominal behavior of a time series, which can be used in the problem of anomaly detection [18]–[20]. For example, the volumes of traffic in network data can be analyzed for anomalous or malicious behavior (such as attacks), where the analysis of periodicities or similarities can be meaningful [21]. The extrema analysis can model the dependencies in a time series, which can be used in the problem of online learning [22]–[24]. For example, the trading volume and prices in financial data can be analyzed for the prediction of oversold or overbought markets, where their correlating burst events can be meaningful [25]. The extrema can track the most informative values in the time series, such as the envelope of the electromyography (EMG) signal for use in robotic systems and prosthesis control [26] to achieve a perfect collaboration between man and robot. In general, the extrema can provide useful robust features [3], especially for robust learning [27]. The extrema are very important in these applications whether in the form of maxima or minima [5].

B. Envelope Extraction

The first of many applications of the extrema analysis is the envelope extraction, which is of utmost importance in the informative data analysis of a modulated signal, where the envelope is extracted by use of a demodulator [28]. Moreover, its usefulness is most apparent in the analysis of random signals, which is commonly encountered in the study of seakeeping and oceanography (or ocean engineering), where waves often occur in groups, which can result in severe damage to the offshore structures. The theory of the envelope of a random signal have been historically used to investigate the statistical properties of such wave groups and correspondingly the wave amplitude distribution [29]–[31]. Their use is of high importance in the prediction of the deck motion (which is necessary for safe landing and take-off for helicopters), where the extraction of the deck motion envelope is used in the platform movement predictions [32]. The works of [33], [34] has established the basis for the envelope statistics and the expression for its spectra was obtained by [4]. The efficient calculation of the envelope is addressed in [35].

Furthermore, the envelope extraction problem has become prominent with the rise of the field of bioinformatics, especially the electrical activity of the human body and specifically the skeletal muscle tissue, i.e., the electromyography (EMG) signal. The most widely used techniques to extract the EMG envelope are the Hilbert transform [36], mean square error (MSE) criterion [37] and the rectified signal waveform [38]. Many algorithms have been developed in literature to extract the EMG envelope, which primarily exploit the signal’s moving average activity together with a noise reduction [39]. However, such methods may suffer large bias errors and be unable to avoid measurement spikes [40]. The Savitsky-Golay smoother combined with a low-pass filter was used to address such problems [41], where the smoothing improves envelope shaping but introduces time delays, which may be intolerable in certain applications (such as robotics). In [26], the signal envelope has been extracted with time-frequency analysis on finite time windows (using Short-time Fourier Transform). In [42], the authors suggest rectifying the EMG signal a priori, which can enhance the muscle activation information but still struggles with high envelope variability [43]. In addition, Kalman and Wiener filter based algorithms have been developed for EMG by many works [44]–[48]. While the traditional Kalman setting has effective tracking when noise is white Gaussian; it provides no meaningful advantage against other methods, since EMG noise is strictly non-white [40]. Another approach by [49] completely forgoes the unbiased noise assumption and is considered more robust [50].
C. Peak-Burst Detection

Another application of interest for the extrema analysis is the well-studied peak (and burst) detection, which is the study of sudden increases or decreases in the data set for informative purposes. Note that while it is easy to visually identify peaks in small datasets or time-series, the notion of peak need to be formalized for objectivity and for the development of automated peak detection algorithms. The automation becomes important in applications involving large datasets and variables, such as the data center monitoring where there are many large time-series about the utilization of CPU and memory in many servers, which need to be analyzed in real-time. In [5], several different notions of peak was introduced together with their corresponding algorithms.

Peak detection is a common task in many applications involving time series analysis. Standard approaches include either fitting a known function to the (possibly smoothed) time series; or matching a known peak shape in the time series. Another widely used approach to detect the peaks is the detection of the zero-crossings of the gradient in the time series (i.e., local extrema), which is found by analyzing the differences between a point and its neighbors for the change in the gradient sign. However, such methods are susceptible to errors in noisy observations, which needs to be solved by the use of a signal to noise ratio (SNR) threshold [9], [51]. The work in [11] parametrically learns such threshold by adapting it to the noise levels in the time-series.

It is also possible to analyze the time series according to different properties of a possible peak and detecting the ones which satisfy all these requirements. One such work is [13], where the analysis of peaks in gene expression microarray time series data (specifically, for malaria parasite Plasmodium falciparum) is studied using multiple methods, each of which assigns a score to every point of the time series. In one of the methods, the score is set as the rate of change (gradient) at each point. In another one, it is the fraction of the area under a certain candidate peak point. For each method, the top 10 performing candidates are selected and the peaks detected by multiple methods are estimated as the true peaks, i.e., it is assumed that the true peaks need to satisfy more than one condition. In that application, the peaks detected are used to identify genes, and support vector machine (SVM) is used to assign each estimated gene to a group. The main problems encountered in peak detection are the existing noise in the time series and the fact that peaks can have varying amplitudes and scales, which may result in a high false positive rate [5].

There are some algorithms, which utilize the shape characteristics of the peaks in certain applications. One such approach is [52], which proposes a continuous wavelet transform (CWT) based algorithm to match preset pattern shapes to the possible peaks in mass spectroscopy data. The two dimensional CWT coefficients of the dataset is calculated using a Mexican-Hat mother wavelet (which has a basic peak shape) for the incoming time-series at multiple distinct scales. The changes in the wavelet transform are examined to identify possible peaks. Other wavelet based approaches to analyze spectroscopy data also exists [8], [53].

Another approach is to analyze the rate of change in the time series. One example is [10], which proposes a ‘momentum’ based algorithm to detect possible peaks. The main idea is to compute the ‘velocity’, i.e., gradient, and ‘momentum’, i.e., the product of the functional value and the ‘velocity’ at various points. If a ball were to be dropped from a previous peak, it should gain momentum as it rolls down and similarly lose momentum as it climbs up the next peak. The point where it comes to rest and stop, i.e., when the momentum is zero, should be the next peak. Physical world analogs of the Newton’s laws are proposed to calculate the changes in momentum as the ball follows along the time-series.

The burst detection is also a prominent problem in machine learning, where the bursts are, in a sense, wider versions of the peaks. While a burst is made up of a wide region of extreme values with fast changes on both sides, a peak is made up of a narrow region of extreme values with fast changes on both sides. In [15], the authors propose a wavelet based burst detection algorithm. The wavelet coefficients for Haar wavelets together with some time-domain statistics such as the windowed average are used in a data structure called the shifted wavelet tree (SWT). Each level of the tree represents a resolution (time scale) and each node of a level corresponds to a certain window. By iterating over different window sizes and time scales, the bursts are detected. The work in [15] apply this method to detect burst in Gamma ray time series in real-time for the Milagro astronomical telescope, where the burst durations vary greatly from minutes to days.

The work in [21] propose a moving average based algorithm for burst detection. The time-series is smoothed with a moving average filter and the values which are larger than some preset parameter \( \theta \) times the standard deviation of the smoothed time-series are detected. They choose \( \theta \) typically between 1.5 and 2. The smoothing window is selected by using domain knowledge (e.g., 30 for daily data). In [25], the authors detect burst in real-time streaming stock market data and analyze correlations between the detected bursts.

D. Clustering

The extrema analysis and specifically peak detection based approaches are also widely used in applications considering clustering (or unsupervised classification). One such popular application is the detection of the QRS complex in electrocardiogram (ECG) signals. Many different methods have been developed for detection of such events, which can be summarized as those based on wavelet transform [8], [8], [62], [54], [57], traditional window-threshold techniques [58]–[60], Hilbert transform [61], combining Hilbert and wavelet transform [62], artificial neural networks [63], [64], techniques using templates [65], [66], morphology filtering [67], [68], nonlinear filtering [69]–[72], Kalman filtering [73], Gabor filtering [74], Gaussian second derivative filtering [75], linear prediction analysis [76], higher-order statistics [77], K-Means clustering [78], fuzzy C-Means clustering [79], Empirical Mode Decomposition [80], hidden Markov models [81], and techniques using histogram/cumulative distribution function [82], intensity weighted variance [83], stochastic resonance [84], or a smoothed nonlinear energy operator [85].
E. Signal Bands

The time-series bands or channels is another (maybe most widely used) application (especially in the stock market). Within economics and mathematical finance, the trading bands are used for a variety of different trading approaches because of the various studies in support of them [85].

A trading band is, in its most simple form, an envelope (or channel) around a certain evolving metric (property) of the time series. Traditionally, these kinds of bands are constructed by creating two signals that are individually a certain distance away from a centralizing signal, which consequently define an upper and a lower band. The distance between the upper and lower bands is typically dependent on the trading band type, which is usually either a measure of volatility or a function of some other related parameter [89]. The upper and lower bands could be considered as some forms of support and resistance lines; where support and resistance are the values from which a quantity struggles to descend and ascend respectively [90].

One of the most popular signal bands/channels is the Donchian Price Channels [7], which defines a simple band with a set of three bands in total: one upper, one lower and a middle band, which is the direct average of them. For a preset window size \( n \), the upper and the lower bands are the maximum and the minimum respectively of the last \( n \) samples.

Bollinger bands are another popular technical analysis tool in mathematical finance. Aside from the field of economics, Bollinger bands have also been used to measure the accident rate (as a safety indicator) in the air travel industry and as a method of inspection in patterned fabrics [88]. Bollinger bands consist of two main values, which are a measure of the central tendency together with a measure of the volatility [89]. The traditionally used central tendency measure is the moving average \( \mu_t \) and volatility measure is the moving standard deviation \( \sigma_t \). These two values create three bands in total, where the middle band is the moving average \( \mu_t \) and the upper and lower bands are \( \mu_t \pm 2\sigma_t \). Another popular signal band is Keltner channels, which again uses the moving average for the measure of the central tendency. However, for the measure of the volatility, they adopt the Average True Range (ATR) [91].

F. Contributions and Organization

The problem with most of the algorithms in literature is in their parametric nature, where the more generally applicable and nonrestrictive they are, the more free parameters they have [5]. However, the selection of parameters not only depend on the application and the dataset, it is also non trivial. To this end, we propose a nonparametric approach for the extrema analysis problem, which can be used in many types of applications including envelope extraction, peak-burst detection and clustering. In Section II, we mathematically formalize the problem setting in a natural way. In Section III, we provide the methodology to find the solution for that formulation, which sequentially outputs two non-crossing time series for a given input time series. In Section IV, we provide the computational complexity analysis for many interesting settings and show its efficiency. In Section V, we provide some concluding remarks.

II. Problem Formulation

In this section, we formalize the extrema analysis problem for a given signal, mathematically. We start with an incoming time series, i.e., \( \{x_1, x_2, \ldots, x_T\} \), which is written for brevity as \( \{x_t\}_{t=1}^T \). For this incoming time series, we want to create two signals, which are \( \{a_t\}_{t=1}^T \) and \( \{b_t\}_{t=1}^T \). These two signals will be our upper and lower bounding time series respectively.

Thus, we want \( a_t \geq x_t \) and \( b_t \leq x_t \) for all \( t \).

We postulate that there is a natural nonparametric way of defining the upper and lower bounding time series. For a definition that is parameter-free, we create the sequences \( \{a_t\}_{t=1}^T \) and \( \{b_t\}_{t=1}^T \) directly from the time series \( x_t \), such that at any time \( t \), \( x_t \) is either \( a_t \) or \( b_t \). Hence, the problem is to split the time series \( \{x_t\}_{t=1}^T \) into two sequences \( \{a_t\}_{t=1}^T \) and \( \{b_t\}_{t=1}^T \) by finding a binary sequence \( \{s_t\}_{t=1}^T \), for which

\[
a_t = s_t x_t + (1 - s_t)a_{t-1},
\]

\[
b_t = (1 - s_t)x_t + s_tb_{t-1}.
\]

The sequence \( \{s_t\}_{t=1}^T \) is selected by minimizing the following objective:

\[
\min_{\{s_t\}_{t=1}^T} \left( \sum_{t=2}^T |a_t - a_{t-1}| + \sum_{t=2}^T |b_t - b_{t-1}| \right). \tag{3}
\]

Hence, the problem is to split \( x_t \) into two sequences such that the sum of their \( L_1 \) (Manhattan) drifts (sum of the successive samples distances) are minimized.

We have constructed the envelopes \( \{a_t\}_{t=1}^T \) and \( \{b_t\}_{t=1}^T \) with piecewise constant interpolation for the formulation, where \( a_t \) (and similarly \( b_t \)) is either \( x_t \) or \( x_{\tau} \) for some \( \tau < t \). However, we can just as easily use linear interpolation (which is more common and intuitive) for the missing the indices, which does not change the \( L_1 \) drift (and the objective function). With linear interpolation, we have the following beautiful property.

Lemma 1. The signals \( \{a_t\}_{t=1}^T \) and \( \{b_t\}_{t=1}^T \) never cross.

Proof. Suppose they cross at some \( t_c \), which is a real number and both \( a \) and \( b \) are interpolated there. For that, let us have \( t_{a,1} < t_{c} < t_{a,2} \) and \( t_{b,1} < t_{c} < t_{b,2} \), where \( a_{t_{a,1}} = x_{t_{a,1}}, a_{t_{a,2}} = x_{t_{a,2}}, b_{t_{b,1}} = x_{t_{b,1}}, b_{t_{b,2}} = x_{t_{b,2}} \) and \( a_t \) and \( b_t \) are interpolated between the samples \( \{t_{a,1}, t_{a,2}\} \) and \( \{t_{b,1}, t_{b,2}\} \) respectively. This will have the following loss

\[
L_T = \sum_{t=2}^{t_{a,1}} |a_t - a_{t-1}| + \sum_{t=t_{a,2}}^T |a_t - a_{t-1}| + |a_{t_{a,1}} - a_{t_{a,2}}|,
\]

\[
+ \sum_{t=2}^{t_{b,1}} |b_t - b_{t-1}| + \sum_{t=t_{b,2}}^T |b_t - b_{t-1}| + |b_{t_{b,1}} - b_{t_{b,2}}|.
\]

Now, if we switch the classes of the samples after \( t_c \), i.e., \( a_{t_{b,1}} \leftarrow b_{t_{b,1}} \) and vice-versa, we will have a loss which has a \( \triangle \) difference resulting from the cross at \( t_c \), which is

\[
\Delta = |a_{t_{a,1}} - b_{t_{b,2}}| + |b_{t_{b,1}} - a_{t_{a,2}}| - |a_{t_{a,1}} - a_{t_{a,2}}| - |b_{t_{b,1}} - b_{t_{b,2}}|.
\]

Since the original \( a_t \) and \( b_t \) were crossing each other, we want to compare the sum of the diagonals of a butterfly with the sum of its sides. From the triangle inequality, we have \( \Delta < 0 \), i.e., a smaller loss, which concludes the proof. \( \square \)
III. METHODOLOGY

A. Path Tracking Algorithm

The brute force approach is to try for all possible \( \{s_t\}_{t=1}^T \) sequence, which are \( 2^T \) in total (i.e., exponential), and thus intractable.

A tractable approach is to use a path tracking method similar in spirit to the Viterbi algorithm \[23\]. In this approach, we sequentially calculate the losses of the sequences (or paths) \( \{s_t\}_{t=1}^T \) and only keep the paths with the minimal losses (the survival rule).

**Definition 1.** At any time \( t' \), let \( \{s_{1,t}\}_{t=1}^{t'} \) and \( \{s_{2,t}\}_{t=1}^{t'} \) be two distinct sequences, which are identical after some time \( t_s \), such that

- \( s_{1,t} = s_{2,t} = b \) for \( t_s < t \leq t' \) (where \( b \in \{0,1\} \)),
- \( s_{1,t_s} = s_{2,t_s} = 1 - b \),
- \( \{s_{1,t}\}_{t=1}^{t-1} \) is distinct to \( \{s_{2,t}\}_{t=1}^{t-1} \).

**Proposition 1.** For two sequences \( \{s_{1,t}\}_{t=1}^{T} \) and \( \{s_{2,t}\}_{t=1}^{T} \) as in **Definition 1** let their losses be \( L_{1,t} \) and \( L_{2,t} \). We need only to keep track of the sequence with the smaller loss.

**Proof.** We observe that the last time \( \{s_{1,t}\}_{t=1}^{t'} \) and \( \{s_{2,t}\}_{t=1}^{t'} \) is 0 and 1 coincide, thus, the additive loss resulting from the new sample \( x_{t+1} \) is identical. Without loss of generality, assume that the loss of \( \{s_{1,t}\}_{t=1}^{t'} \) up to \( t' \), \( L_{1,t'} \), is smaller. Then, there is no reason to keep track of the loss of \( \{s_{2,t}\}_{t=1}^{t'} \), since for any sequence \( \{s_{1,t}\}_{t=1}^{t+1} \), the loss of \( \{s_{1,t}\}_{t=1}^{t+1} \cup \{s_{2,t}\}_{t=1}^{t+1} \) will be smaller than the loss of \( \{s_{2,t}\}_{t=1}^{t+1} \cup \{s_{2,t}\}_{t=1}^{t+1} \), which concludes the proof.

Using this property, we keep track of the sequences with equivalence classes. At any time \( t' \), \( A_{t'}; b_{t'} \) for \( \tau < t' \) denote the class of the sequences \( \{s_t\}_{t=1}^{t'} \) where the last index of 1 is \( t' \) and the last index of 0 is \( \tau \). Similarly, we have \( B_{t'}; a_{t'} \). At any time \( t \), we will have \( 2(t-1) \) classes with the corresponding losses, which are given as

\[
A_t; b_{t<\tau} : L_{t<\tau}^a, \quad B_t; a_{\tau<t} : L_{\tau<t}^b.
\]

Note that each class keeps track of a single surviving sequence with the best loss. With a new incoming sample \( x_{t+1} \), we update the classes and their losses as follows:

\[
A_{t+1}; b_{t<\tau} : L_{t<\tau}^a + |x_{t+1} - x_{\tau}|, \quad & A_{t+1}; b_t : \min_{\tau<t} (L_{t<\tau}^a + |x_{t+1} - x_{\tau}|), \\
B_{t+1}; a_{\tau<t} : L_{\tau<t}^b + |x_{t+1} - x_{\tau}|, \quad & B_{t+1}; a_{t} : \min_{\tau<t} (L_{\tau<t}^b + |x_{t+1} - x_{\tau}|).
\]

The minimum operations at \(8\) and \(10\) are where the Viterbi behavior comes from. At the end of \( T \) rounds, we will have

\[
A_T; b_{t<T} : L_{t<T}^a, \quad & B_T; a_{t<T} : L_{T<T}^b.
\]

We can select the class with the minimum cumulative loss, which will minimize \[3\]. This approach has quadratic \( O(T^2) \) computational and memory complexity.

B. Efficient Algorithm

First of all, we observe that keeping both the equivalence classes \( A_t; b_t \) and \( B_t; a_t \) at time \( t \) is unnecessary (since there is no distinction between them). To this end, we revamp the design and only keep the following losses at each time \( t \):

\[
L_{t,\tau}, \quad t \in \{0,1,\ldots,t-1\},
\]

where \( L_{t,\tau} \) represents the loss of the best sequence where \( \{s_t\}_{t=\tau+1}^{t+1} \) is \( b \in \{0,1\} \) and \( s_{\tau} = 1 - b \), i.e., if \( |x_{t'}|_{t'=\tau+1}^{t+1} \) belongs to the time series \( a \) then \( x_{\tau} \) belongs to the time series \( b \) and vice-versa (thus, at \( \tau \), the assignment changes). The sequential update of \( L_{t,\tau} \) is given as the following:

\[
L_{t+1,\tau} = L_{t,\tau} + |x_{t+1} - x_{\tau}|, \quad \tau \in \{0,1,\ldots,t-1\}, \quad & L_{t+1,t} = \min(L_{t,0}, \min_{1 \leq \tau < t} (L_{t,\tau} + |x_{t+1} - x_{\tau}|)).
\]

We observe that the individual updates to \( L_{t+1,\tau} \) for \( \tau \in \{0,1,\ldots,t-1\} \) is unnecessary since we only need to compare the losses at the end. Thus, we modify it as the following:

\[
M_{t+1,\tau} = \left\{ \begin{array}{ll} M_{t,\tau}, & \tau \in \{0,1,\ldots,t-1\}, \\
\min_{0 \leq \tau < t} (M_{t,\tau} + d_{t,\tau}), & \tau = t \end{array} \right.
\]

where \( d_{t,0} = -|x_{t+1} - x_{\tau}|, d_{t,\tau} = -|x_{t+1} - x_{\tau}| + |x_{t+1} - x_{\tau}| \).

Note that we have \( M_{t,0} = 0, M_{t,t} = 0 \). For further efficiency, we also keep the minimizer \( \tau \) at each time \( t \)

\[
\tau_t = \arg\min_{0 \leq \tau < t} (M_{t,\tau} + d_{t,\tau}).
\]

At the end, we recreate the sequence as follows: We calculate

\[
\tau^* = \arg\min_{0 \leq \tau \leq T-1} M_{t,\tau},
\]

which means the samples between \( t \in \{\tau^*+1,\ldots,T\} \) belong to the first time series. Then, we get the next index, which is \( \tau_{\tau^*} \), which means the samples between \( t \in \{\tau_{\tau^*}+1,\ldots,\tau^*\} \) belong to the second time series. Then, setting \( \tau^* \leftarrow \tau_{\tau^*} \), we recursively re-calculate the time indices and setting the samples in the corresponding time series alternatively.

We can see that the double indexing is unnecessary since

\[
M_{t,\tau} = M_{t+1,\tau} = \min_{0 \leq \tau < t} (M_{t,\tau} + d_{t,\tau}),
\]

Thus, we can remove one of the indexing and just write that

\[
M_t = \min_{0 \leq \tau \leq t-1} (M_{t,\tau} + d_{t,\tau}), \quad \tau_t = \arg\min_{0 \leq \tau \leq t-1} (M_{t,\tau} + d_{t,\tau})
\]

where

\[
d_{t,\tau} = \begin{cases} -|x_{t+1} - x_{\tau}| + |x_{t+1} - x_{\tau}|, & 1 \leq \tau < t-1 \\ -|x_{t+1} - x_{\tau}|, & \tau = t-1 \end{cases}.
\]

This approach has \( O(T) \) memory complexity and \( O(T) \) backtracking. However, it will still have \( O(T^2) \) complexity in total because of the calculation in \( M_t \).
C. Optimal Elimination

We observe that the reason for the quadratic time complexity is because the complexity at each time \( t \) is proportional with the number of equivalence classes at time \( t \). To make it efficient, we can eliminate certain equivalence classes. Suboptimal ways include the elimination of \( M_\tau \) where \( \tau \leq t - w \) for some window \( w \), or similarly, keeping the best \( w \) classes (with the minimum \( M_\tau \)).

We observe that there is a trimming approach to the equivalence classes without loss of optimality.

**Lemma 2.** We observe that we can directly eliminate all \( M_{\tau_1} \) for \( \tau_1 \neq \tau \) such that

\[
M_{\tau_1} \geq M_{\tau_2} + |x_{\tau_2} - x_{\tau_1}|,
\]

for some \( \tau_2 \notin \{0, \tau_1\} \).

**Proof.** From the algorithm, we know that for any time \( t \), if

\[
t^* = \arg \min \{M_\tau + d_{t, \tau}\},
\]

we have

\[
M_t = M_{t^*} + d_{t, t^*}.
\]

Let \( \tau_1, \tau_2 < t \). Then, we have

\[
M_{\tau_1} \geq M_{\tau_2} + |x_{\tau_2} - x_{\tau_1}|,
\]

(24)

from the triangle inequality. Thus,

\[
M_{\tau_1} + d_{t, \tau_1} \geq M_{\tau_2} + d_{t, \tau_2}.
\]

(26)

Thus there exists a \( t^* \), which is distinct from \( \tau_1 \). Hence, we can eliminate \( M_{\tau_1} \), which concludes the proof.

**Theorem 1.** If, for some \( t_1 < t_2 \),

\[
x \in [\min(x_{t_2}, x_{t_2+1}), \max(x_{t_2}, x_{t_2+1})],
\]

\( M_{t_1} \) does not survive.

**Proof.** From [Lemma 2] for survivability, we require

\[
M_{t_1} < M_{t_2} + |x_{t_1} - x_{t_2}|,
\]

(27)

for all nonzero \( t_1 \neq t_2 \) pairs in the running. If \( t_1 < t_2 \), we have

\[
M_{t_2} \geq M_{t_1} - |x_{t_2+1} - x_{t_1}| + |x_{t_2+1} - x_{t_1}|,
\]

(28)

from triangle inequality. However, (28) holds with equality, i.e.,

\[
M_{t_1} - |x_{t_1} - x_{t_2}| = M_{t_1} - |x_{t_2+1} - x_{t_1}| + |x_{t_2+1} - x_{t_1}|
\]

(30)

when \( x_{t_1} \) is between \( x_{t_2} \) and \( x_{t_2+1} \), which, together with (28) results in

\[
M_{t_1} - |x_{t_1} - x_{t_2}| \geq M_{t_2}.
\]

(31)

Thus, \( M_{t_1} \) breaks the condition in (27), which concludes the proof.

IV. Complexity Analysis

In this section, we analyze the computational complexity of our optimal trimming. We analyze the number of surviving equivalence classes since the computational complexity per time is linearly dependent on that. While the complexity is straightforward to see for periodic signals, random processes need more analysis.

A. Uniform i.i.d. Process

Let \( x_t \) be a uniform i.i.d. process such that \( x_t \sim U([0, 1]) \), i.e., \( f(x) = 1 \) for \( x \in [0, 1] \) and 0 otherwise.

**Corollary 1.** For a uniform i.i.d. process, the expected computational complexity of our algorithm is \( O(\log T) \) per time.

**Proof.** At any arbitrary time \( T \), let \( \tau = T - t \) be the number of samples that come after \( x_t \). From [Theorem 1] the only way \( M_t \) will survive at time \( T \) is if all the samples \( x_{t'+\tau} \) are in \([0, x_t]\) or in \((x_t, 1]\). Thus, the probability that \( M_t \) survives is

\[
P_t = x_t^\tau + (1 - x_t)^\tau,
\]

(32)

where \( \tau = T - t \). Since the survival is a Bernoulli trial with \( P_t \), the expected survivability \( z_t \) of \( M_t \) is

\[
z_t \triangleq E[x_t | P_t],
\]

(33)

\[
= \int_0^1 (x_t^\tau + (1 - x_t)^\tau) dx_t,
\]

(34)

\[
= \left( \frac{x_t^{\tau+1}}{\tau+1} - \frac{(1-x_t)^{\tau+1}}{\tau+1} \right) \bigg|_0^1,
\]

(35)

\[
= \frac{2}{\tau+1}.
\]

(36)

Hence, the expected number of survivors \( S_T \) at time \( T \) is

\[
S_T = O(\log T),
\]

(37)

which concludes the proof.

B. Arbitrary i.i.d. Process

Let \( x_t \) be i.i.d. with \( x_t \sim D \) for some distribution \( D \). Let its probability density function (PDF) be \( f(x) \) and its cumulative distribution function (CDF) be \( F(x) \).

**Corollary 2.** For an arbitrary i.i.d. process, the expected complexity of our algorithm is \( O(\log T) \) per time.

**Proof.** For \( M_t \) to survive, we again need all of the upcoming samples \( x_{t'+\tau} \) be strictly less or more than \( x_t \). Thus, the probability of survival is

\[
P_t = (F(x_t))^\tau + (1 - F(x_t))^\tau,
\]

(38)

and its expectation is

\[
z_t = \int f(x) P_t dx_t,
\]

(39)

\[
= \int f(x_t) [(F(x_t))^\tau + (1 - F(x_t))^\tau] dx_t,
\]

(40)

\[
= \int_0^1 [(F)^\tau + (1 - F)^\tau] dF,
\]

(41)

\[
= \frac{2}{\tau+1}.
\]

(42)

Hence, the expected number of survivors at \( T \) is \( O(\log T) \).
C. Simple Symmetric Random Walk

Let \( x_t \) be a simple symmetric random walk, where the successive difference
\[
\delta_t \triangleq x_t - x_{t-1},
\]
is an unbiased random variable in \( \{-1, 1\} \), i.e.,
\[
\delta_t = \begin{cases} 
+1, & \text{with probability } 0.5 \\
-1, & \text{with probability } 0.5 
\end{cases}
\]

**Corollary 3.** For a simple symmetric lattice random walk with step size 1, we have \( O(\sqrt{T}) \) complexity.

**Proof.** At time \( T \), the survivability given \( x_T, z_t \), of \( M_t \) is again the sum of the probabilities of \( x_t \) being greater or less than all of the following samples, which is given by

\[
z_t = \begin{cases} 
\mathbb{P}(x_T > x_t; t < \tau \leq T), & x_T > x_t \\
\mathbb{P}(x_T < x_t; t < \tau \leq T), & x_T < x_t 
\end{cases}
\]

where \( \mathbb{P}(\cdot) \) is the probability operator. From the Ballot theorem [92], we have

\[
z_t = \frac{|x_T - x_t|}{T - t}.
\]

Since the sum \( \sum_{t'=t+1}^T \delta_{t'} \) converges in distribution to a Gaussian with variance \( T - t \), the expectation of its absolute value is \( O(\sqrt{T-t}) \). Thus, the total number of expected survivors \( S_T \) at \( T \) is

\[
S_T = O(\sqrt{T}),
\]

which concludes the proof. \( \square \)

**Remark 1.** The complexity bound will be the same for any arbitrary step size \( \delta \) since we only compare whether the values \( x_t \) are greater (or less) than the following samples \( x_{\tau>t} \).

D. General Symmetric Random Walk

Let \( x_t \) be a random walk, where the successive difference
\[
\delta_t \triangleq x_t - x_{t-1},
\]
is an unbiased, finite variance random variable with a distribution \( D \), i.e.,
\[
\delta_t \sim D, \quad \mathbb{E}[\delta_t] = 0, \quad \mathbb{E}[\delta_t^2] < \infty.
\]

**Corollary 4.** For a symmetric random walk with an unbiased, finite variance random step size \( \delta \), we have \( O(\sqrt{T}) \) complexity.

**Proof.** Similarly to the simple random walk, the survivability \( z_t \) of \( M_t \) is

\[
\mathbb{E}[s_t] = O \left( \frac{1}{\sqrt{T-t}} \right),
\]

from the extended Ballot theorem [93]. Thus, the expected number of survivors at time \( T \) is

\[
S_T = O(\sqrt{T}),
\]

which concludes the proof. \( \square \)

E. All-time Maximum and Minimum

Another interesting setting is, let \( x_t \) be the all time maximum with probability \( p_t \) and all time minimum with probability \( q_t \), i.e.,

\[
\mathbb{P}(x_t \geq x_T; \tau < t) = p_t,
\]

\[
\mathbb{P}(x_t \leq x_T; \tau < t) = q_t,
\]

where \( \mathbb{P}(\cdot) \) is again the probability operator.

**Corollary 5.** If at any time \( t \), \( x_t \) is the all time maximum with probability \( p_t \) and all time minimum with probability \( q_t \), where \( p_t \) and \( q_t \) are nonincreasing, i.e., \( p_t \leq p_{t-1}, q_t \leq q_{t-1} \); the expected number of survivors is

\[
S_T = O \left( p_T^{-1} + q_T^{-1} \right).
\]

**Proof.** At time \( T \), let \( t_1 \) be the last time we get an all time maximum and \( t_2 \) be the last time we get an all time minimum. Then, the number of surviving sequences are bounded by

\[
\sum_{t=1}^{T} z_t \leq \text{max}(t_{1,T}, t_{2,T}) \leq t_{1,T} + t_{2,T},
\]

where \( t_{1,T} \triangleq T - t_1 \) and \( t_{2,T} \triangleq T - t_2 \). Since the all time maximum and minimum probabilities are \( p_t \) and \( q_t \). Then, at time \( T \), the expectation of \( t_{1,T} \) is

\[
\mathbb{E}[t_{1,T}] = p_T + 2(1 - p_T)p_T - 1 + 3(1 - p_T)(1 - p_T - 1)p_T - 2 \ldots,
\]

\[
\leq 1 + (1 - p_T) + (1 - p_T)(1 - p_T - 1) + \ldots \quad (57)
\]

where we have the following recursion

\[
\mathbb{E}[t_{1,T}] \leq 1 + (1 - p_T)\mathbb{E}[t_{1,T-1}],
\]

and a similar recursion for \( t_{2,T} \) as well. Thus,

\[
\mathbb{E}[t_{1,T}] \leq \sum_{i=0}^{T} (1 - p_T)^i \leq \frac{1}{p_T},
\]

for nonincreasing \( p_t \) (similarly for \( \mathbb{E}[t_{2,T}] \) and \( q_T \), which concludes the proof. \( \square \)

V. Conclusion

In this paper, we proposed a nonparametric approach that can be used in envelope extraction, peak-burst detection and clustering in time series. Our proposed problem formulation naturally defines a splitting/forking of the time series, which can be used for various applications in machine learning, signal processing and mathematically finance (with a possibly hierarchical implementation via successive application of our method to the generated forks). We proposed an algorithm that sequentially creates two signals (one upper bounding and one lower bounding signal) from an incoming input signal by minimizing the cumulative \( L_1 \) drift. We showed that a solution can be efficiently calculated by use of a Viterbi-like path tracking algorithm and in many interesting settings, our algorithm has low time complexities.
Y. S. Shmaliy, S. Zhao, and C. K. Ahn, “Unbiased finite impulse response filtering: An iterative kalman filtering ignoring noise and initial conditions,” IEEE Control Systems Magazine, vol. 37, no. 5, pp. 70–80, 2017.

P. Du, W. A. Kibbe, and S. M. Lin, “Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching,” bioinformatics, vol. 22, no. 17, pp. 2059–2065, 2006.

E. Lange, C. Gröpl, K. Reinert, O. Kohlbacher, and A. Hildebrandt, “High-accuracy peak picking of proteomics data using wavelet technology,” in Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 1. IEEE, 2001, pp. 344–347.

J. M. Gregoire, D. Dale, and R. B. Van Dover, “A wavelet transform approach to automated peak detection,” in 2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 3. IEEE, 2001, pp. 2194–2195.

J. Excoffier and G. Guiocho, “Automatic peak detection in chromatography,” Chromatographia, vol. 15, no. 9, pp. 543–545, 1982.

D. Benitez, P. Gaydecki, A. Zaidi, and A. Fitzpatrick, “The use of the hilbert transform in eeg signal analysis,” Computers in biology and medicine, vol. 31, no. 5, pp. 399–406, 2001.

H. Rabbani, M. P. Mahjoob, F. Farahabadi, and A. Farahabadi, “R peak detection in electrocardiogram signal based on an optimal combination of wavelet transform, hilbert transform, and adaptive thresholding,” Journal of medical signals and sensors, vol. 1, no. 2, p. 91, 2011.

Q. Xue, Y. H. Hu, and W. J. Tompkins, “Neural-network-based adaptive matched filtering for qrs detection,” IEEE Transactions on biomedical engineering, vol. 39, no. 4, pp. 317–329, 1992.

G. Vijaya, V. Kumar, and H. Verma, “Ann-based qrs-complex analysis of ecg,” Journal of medical engineering & technology, vol. 22, no. 4, pp. 160–167, 1998.

N. Mietwa and L. S. Smith, “Smoothing and thresholding in neuronal spike detection,” Neurocomputing, vol. 9, no. 10, pp. 1366–1370, 2006.

V. P. Andreev, T. Rejtar, H.-S. Chen, E. V. Moskovets, A. R. Ivanov, and B. L. Karger, “A universal denoising and peak picking algorithm for r−ms based on matched filtration in the chromatographic time domain,” Analytical chemistry, vol. 75, no. 22, pp. 6314–6326, 2003.

F. Zhang and Y. Lian, “Qrs detection based on multiscale mathematical morphology for wearable eeg devices in body area network devices,” IEEE Transactions on Biomedical Circuits and Systems, vol. 3, no. 4, pp. 220–228, 2009.

———, “Qrs detection based on morphological filter and energy envelope for applications in body sensor networks,” Journal of Signal Processing Systems, vol. 64, no. 2, pp. 187–194, 2011.

Y. Sun, S. Suppappola, and T. A. Wurbslewski, “Microcontroller-based real-time qrs detection,” Biomedical instrumentation & technology, vol. 37, pp. 445–449, 1993.

Y. Ferdi, J. Herbeuval, A. Chairef, and B. Boucheham, “R wave detection using fractional digital differentiation,” Ithmb-Rbib, vol. 24, no. 5–6, pp. 273–280, 2003.