Latent Time-Series Motifs

Josif Grabocka, ISMLL, University of Hildesheim
Nicolas Schilling, ISMLL, University of Hildesheim
Lars Schmidt-Thieme, ISMLL, University of Hildesheim

Motifs are the most repetitive/frequent patterns of a time-series. The discovery of motifs is crucial for practitioners in order to understand and interpret the phenomena occurring in sequential data. Currently, motifs are searched among series sub-sequences, aiming at selecting the most frequently occurring ones. Search-based methods, which try out series sub-sequence as motif candidates, are currently believed to be the best methods in finding the most frequent patterns.

However, this paper proposes an entirely new perspective in finding motifs. We demonstrate that searching is non-optimal since the domain of motifs is restricted, and instead we propose a principled optimization approach able to find optimal motifs. We treat the occurrence frequency as a function and time-series motifs as its parameters, therefore we *learn* the optimal motifs that maximize the frequency function. In contrast to searching, our method is able to discover the most repetitive patterns (hence optimal), even in cases where they do not explicitly occur as sub-sequences. Experiments on several real-life time-series datasets show that the motifs found by our method are highly more frequent than the ones found through searching, for exactly the same distance threshold.

Additional Key Words and Phrases: Time series, Repeated patterns, Motifs

ACM Reference Format:
Josif Grabocka, Nicolas Schilling and Lars Schmidt-Thieme, 2015. Latent Time-Series Motifs. *ACM Trans. Knowl. Discov. Data.* , , Article (May 2015), 21 pages.
DOI: 0000001.0000001

1. INTRODUCTION

Time-series are arguably the most widespread type of data which occur in virtually all the application domains of our modern lives, wherever measurements have associated time stamps (e.g.: physiological and medical, financial, meteorological, sound and video, monitoring system sensors, astronomy light intensities, and many more).

In many cases, the underlying patterns of those datasets are not known to the domain practitioners and a visual inspection is often infeasible given the complexity and size of the data. For this reason, finding the most repetitive patterns in time-series help the domain experts understand the underlying phenomena within diverse sources of data [Buhler and Tompa 2001; Syed et al. 2010]. The most repetitive time-series patterns are called *motifs* and their discovery has recently attracted considerable research [Patel et al. 2002; Mueen 2013; Yingchareonthawornchai et al. 2013; Li et al. 2012]. In brief terms, optimal motifs are those which repeat the most (i.e. have the highest frequency) given a distance/similarity threshold value. The approach of the current state-of-the-art motif discovery methods is to ***search*** the motifs from the segments (a.k.a sub-sequences) of time series [Patel et al. 2002; Yankov et al. 2007; Li

Authors' address: Josif Grabocka and Nicolas Schilling and Lars Schmidt-Thieme, Information Systems and Machine Learning Lab, Institute of Computer Science, University of Hildesheim, Universittsplatz 1, 31141 Hildesheim, Germany.

DOI: 0000001.0000001

et al. 2012; Castro and Azevedo 2010]. More concretely, series segments are considered to be motif candidates and the most frequent segments are sorted out.

In this paper we present an entirely new and orthogonally different perspective to the ***search-based*** approach. First of all, we treat frequency as a function and motifs as its variable. Naturally our task becomes finding the values of motifs which maximize the value of the frequency function. In this perspective we formalize motif discovery as a principled optimization problem and devise an optimization technique to ***learn*** the optimal motifs. The learning process uses the first order derivative of the frequency function, in order to find its maximum. In that way, our method can learn motifs which yield the maximum frequency (a.k.a the highest number of matches). The proposed ***learning*** method is theoretically superior to the ***search-based*** approach, because in the case of searching the motif candidates are limited to the domain of sub-sequences and cannot discover latent series patterns (Section 4.1) .

As the empirical results (Section 6) over various real-life datasets will indicate, our optimal motifs have significantly more matches (higher frequency) than the ones found through searching, for exactly the same distance threshold.

2. RELATED WORK

The research on discovering time-series motifs has suffered from a terminological ambiguity. Initially, motifs were defined to be the most frequently occurring patterns in a time-series [Patel et al. 2002]. However, another stream of papers redefined the term ”motif” as the closest pair among series segments [Mueen et al. 2009b; Mueen and Keogh 2010]. In this paper we mean ”the most frequently occurring patterns” [Patel et al. 2002] when referring to motifs. The closest pair of series segments, on the other hand, will be referred to as ”pair-motif” following the suggestion of [Mohammad and Nishida 2014].

2.1. Pair-motif discovery

The closest pair of series segments can be perceived as a sub-variation of the general motif discovery task. The brute-force search that computes the distance of every segment pair is computationally expensive, therefore efforts are devoted towards scaling the brute force up. A fast, yet exact, method that discovers pair-wise motifs has been introduced by [Mueen et al. 2009b]. Enumerations of all motifs having variable lengths has also been researched [Mueen 2013; Mohammad and Nishida 2014]. In a streaming scenario an algorithm can not rely on accessing the full past series, therefore we need to find the top-k motif search via an on-line method as in [Lam et al. 2011]. In addition, the statistical significance of the motifs found has also been a topic of interest [Castro and Azevedo 2011; 2012].

Note: Finding motif-pairs is equivalent to the problem of locating the closest pair of points in a geometrical space and is a historic problem in computational geometry [Cormen et al. 2001].

2.2. Motif Discovery

Repeating patterns in sequential data have initially been studied in bioinformatics [Buhler and Tompa 2001]. However, finding motifs is beneficial in understanding physiological human data [Syed et al. 2010], while being also useful in understanding behavioral patterns of living organisms [Brown et al. 2013]. The concept of recurrent patterns was transferred to the realm of time-series data under the term ”motifs” [Patel et al. 2002] and a search-based approach to discovering motifs was proposed. In order to find motifs that are immune to noisy variations, a probabilistic search of time-series motifs was based on random projections [Chiu et al. 2003]. Another work has explored the employment of uniform scaling as the similarity distance used for discovering the motifs [Yankov et al. 2007]. Furthermore, a hybrid combination of supervised and unsupervised learning has been used for searching recurring patterns [Oates 2002]. The first step involves a *teacher* which labels whether or not a time series includes a particular pattern, while in the next step an unsupervised learning from the series in order to reconstruct the *teacher* is exploited. The task of finding the most recurring motifs has also been tackled through searching for candidate motifs organized in a tree structure [Liu et al. 2005].

The brute-force approach which tries out every segment (sub-sequence) as a potential motif has a quadratic complexity in the number of segments. Therefore approximate motif discovery methods have been exploited. Conversion of motifs into a symbolic representation (named SAX) is a pre-processing alternative [Ferreira et al. 2006]. Over the new representation an agglomerative clustering can be used to find motifs [Ferreira et al. 2006]. A scalable alternative that can approximately discover multi-resolution motifs in a single scan utilizes different cardinalities of the symbolic representation [Castro and Azevedo 2010]. Last but not least, a scalable version of the pair-wise motifs has been extended to the general motifs discovery for large-scale data [Mueen et al. 2009a].

Given the widespread of multi-dimensional time series, there has also been interest in mining multi-dimensional motifs too. Several strategies were inspected, where motifs span all versus a subset of the dimensions, with or without temporal overlap [Minnen et al. 2007a]. The algorithm is based on random projections of the symbolic sub-sequence representations [Minnen et al. 2007a]. Discovering regions of high density in the space of sub-sequencies is another alternative to mining multivariate motifs [Minnen et al. 2007b]. Graph clustering implemented as a two-staged algorithm was also employed in detecting multidimensional motifs [Vahdatpour et al. 2009]. In the first step single-dimensional motifs are discovered and later blended through clustering [Vahdatpour et al. 2009].

Since motifs are previously unknown patterns, there is little information on the motifs' lengths too. Under such a reality authors attempted to discover the optimal motif length, for instance by inspecting the compressibility of the data [Yingchareonthawornchai et al. 2013]. In addition, variable-length motifs can be extracted using a grammar-inspired inference process [Li and Lin 2010]. Interest has been attracted in terms of visualizing variable-length motifs [Li et al. 2012], finding them in linear time [Catalano et al. 2006], or using them for classification purposes [Yin et al. 2014].

2.3. Difference to Symbolic Sequences

Another stream of papers finds latent/hidden motifs for symbolic sequences in DNA and protein data (e.g. [Tata and Patel 2008; Sahli et al. 2014]). However, those works operate with symbolic sequences, not real-valued series. Real-valued series are a different problem to symbolic sequences, for instance you cannot build fast sequence suffix/prefix trees out of real values. On the other hand, discretizing a real-valued series destroys local patterns which might be crucial for the application domain, e.g. discretized symbolic heart beats are meaningless for cardiologists.

In contrast to the related work, our novel contribution relies in computing an optimal set of motifs given a threshold distance and the motifs' length. We are the first to propose a principled optimization method for the task. As a consequence, our approach leads to significantly improved motif quality (frequency) compared to brute-force search.

3. PRELIMINARIES

3.1. Notations

3.1.1. Time Series and Motifs. A time series is a long ordered sequence of real-valued measurements. Such a series is abstracted as a list of J-many Z-normalized sliding-window segments of length L and is denoted as $S \in \mathbb{R}^{J \times L}$. On the other hand, a repetitive pattern, a.k.a motif, is simply a sequence of L points. The definition can be generalized to a set of K-motifs and consecutively denoted as $M \in \mathbb{R}^{K \times L}$.

3.1.2. Motif Frequency. The occurrence frequency of a motif is defined as the *nontrivial* (see Section 3.2.2) number of matches between a motif and all the normalized segments of the time series. The current approach of counting the matching frequency of the k-th motif, denoted $M_{k,:} \in \mathbb{R}^L$, iterates over all the $j \in \{1, \ldots, J\}$ sliding window segments $S_{j,:}$ and check whether the motif of interest matches the segments within a ***threshold distance*** $T \in \mathbb{R}^+$.

$$\mathcal{F}(M) = \sum_{k=1}^{K} \sum_{j=1}^{J} \mathcal{F}_{k,j} \tag{1}$$

$$\mathcal{F}_{k,j} = \begin{cases} 1 & \text{if } \left(\sum_{l=1}^{L} \left(M_{k,l} - S_{j,l} \right)^2 \right) < T \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Equation 1 presents the formalism for the overall frequency as a sum of motifs' frequencies, while Equation 2 encapsulates the concept of a *match*. If the distance between a segment $S_{j,:}$ and a motif $M_{k,:}$ is less than the threshold T, then a matching value of one is granted. We would like to point out that maximizing frequency is by definition the aim of motif discovery task. In fact, our definition follows the established motif formulation routinely addressed in the related literature [Patel et al. 2002; Yankov et al. 2007; Oates 2002].

3.2. Problem Definition

3.2.1. Optimal Motifs. Following the established literature definition, the only optimality criterion of a motif is its frequency at a particular distance threshold. Therefore, the only legitimate metric to compare the qualities of motifs is frequency (a.k.a. support, or number of matches). The optimal motifs M^* for a time series are defined in Equation 3 as the candidate motifs M that achieve the maximum frequency value $\mathcal{F}(M)$ from Equation 1. There is, nevertheless, an important constraint in the search for motifs: The K motifs should be different from each other [Patel et al. 2002], otherwise, the motifs risk being close variations of the single most repetitive motif. Such a constraint is presented under a "such that (s.t.)" clause in Equation 3, which enforces each pair of motifs $(M_{k,:}, M_{p,:})$ to be different from each other by a distance of at least $2T$ (so each pair does not overlap within a threshold T, details in [Patel et al. 2002]).

$$\begin{aligned} M^* \; &:= \underset{M \in \mathbb{R}^{K \times L}}{\operatorname{argmax}} \; \mathcal{F}(M) \\ &\text{s.t.:} \left(\sum_{l=1}^{L} \left(M_{k,l} - M_{p,l} \right)^2 \right) > 2T, \\ &\forall k \in \{1, \ldots, K\}, \forall p \in \{k+1, \ldots, K\} \end{aligned} \tag{3}$$

3.2.2. Trivial Matches. Stated shortly, trivial matches are consecutive segments which match the same motif [Patel et al. 2002]. For instance, this case might happen if the sliding window is incremented by one. In that case two subsequent segments will share exactly $L - 1$ points and therefore the distance of any motif to those close-by segments will be very similar. Some related work increment the sliding window by an offset of points, therefore trivial matches can be trans-passed at the risk of potentially missing certain matches [Castro and Azevedo 2010; Minnen et al. 2007b]. However, in our paper all the reported figures on frequency do not include any trivial match throughout the experiments.

3.3. Searching The Motifs

The state-of-the-art methods referred in Section 2 focusing on searching motifs are primarily concerned with trying candidate motifs from the series segments. Despite proposing important novelties in their scope (scalability, length analysis, etc ...) still these techniques are upper bounded in terms of quality by the brute-force motif search.

Algorithm 1 BruteForceMotifSearch()

```
Input: Threshold $T \in \mathbb{R}^+$, Motif length $L \in \mathbb{N}^+$, Number of Motifs $K \in \mathbb{N}^+$, Segments $S \in \mathbb{R}^{J \times L}$
Output: $M \in \mathbb{R}^{K \times L}$
// Precompute frequencies of all segments
for $j = 1, \ldots, J$ do
  $\mathcal{F}_j \leftarrow 0$
  lastMatchIndex $\leftarrow -\infty$
  for $r = 1, \ldots, J$ do
    if $||S_{j,:} - S_{r,:}||_2^2 < T$ then
      // Avoid trivial matches
      if $r -$ lastMatchIndex $> 1$ then
        $\mathcal{F}_j \leftarrow \mathcal{F}_j + 1$
      end if
      lastMatchIndex $\leftarrow r$
    end if
  end for
end for
// Select top-K motifs
for $k = 1, \ldots, K$ do
  best$_j \leftarrow 0$
  for $j = 1, \ldots, J$ do
    // Check if the j-th segment is diverse
    if $||S_{j,:} - M_{p,:}||_2^2 > 2T, \ \forall p = k-1, \ldots, 1$ then
      if $\mathcal{F}_{\text{best}_j} > \mathcal{F}_j$ then
        best$_j \leftarrow j$
      end if
    end if
  end for
  $M_{k,:} \leftarrow S_{\text{best}_j,:}$
end for
return $M$
```

Algorithm 1 describes a speed-wise naive, yet qualitatively search-optimal implementation of a brute-force motif search. We can pre-compute the frequencies of all series segments in $\mathcal{O}(J^2L)$ runtime complexity and then search the top-K motifs using the computed frequencies in $\mathcal{O}(K^2JL)$ time. Since K is typically a small number compared to the segments $J >> K$, therefore the overall brute-force search has a complexity of $\mathcal{O}(J^2L + K^2JL) \sim \mathcal{O}(J^2L)$, meaning quadratic in the number of segments. In this paper we propose a learning (not searching) method that outputs motifs having higher frequencies than those discovered by the brute-force approach.

4. PROPOSED METHOD

4.1. Motivation

The state-of-the-art methods used for finding motifs are based on *searching* for the most frequently occurring candidate *segment*. In other words, any motif has to explicitly occur as a series segments $M_{k,:} \in S,\ \forall k \in \mathbb{N}_{k=1}^{K}$. Unfortunately, such constrained motifs are very restricted in the finite space of possible values they can have, compared to the space of real matrices $M \in \mathbb{R}^{K \times L}$ (infinitely more candidates than $M \in S$). In this paper, we hypothesize and empirically show that the optimal motifs are located in the space of real numbers $M \in \mathbb{R}^{K \times L}$, while the space of segments contains sub-optimal motifs. Figure 1 provides a hint for the comparison between restricted motifs $(M \in S)$ and un-restricted optimal ones. From a geometrical perspective the segments and the motifs are points in an L-dimensional space. In the example of Figure 1 the segments and motifs have a length of 2, thus the scenario is 2-dimensional.

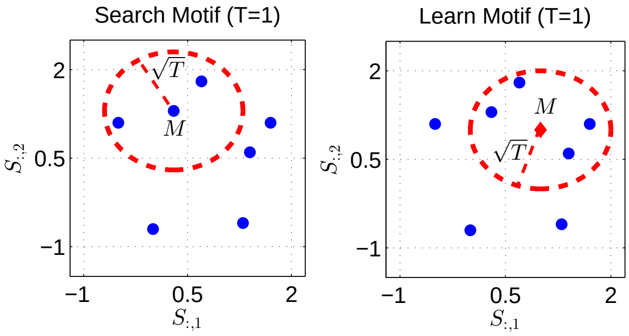

Fig. 1. Motif found by searching (left) yields 3 matches while learning a latent motif (right) yields 4 matches

The frequency of a motif M, given a threshold T, can be interpreted as the number of segment points (blue in the illustration) that lies within a radius of the threshold distance from the motif (shown in red). The radius is $\sqrt{T}$ because we used the squared-Euclidean distance in Equation 2, however this poses no problems since T is anyway a hyper-parameter of our method. The most frequently occurring motif is defined to be the point that covers the maximum number of blue points (segments) inside the circle of radius $\sqrt{T}$ that is centered at the motif, hence the densest geometrical ball [Liu et al. 2005]. The best segment-motif is shown in the left plot of Figure 1 and has a frequency of three. However, the optimal motif is located in the right plot and has a frequency of four. As clearly seen, the optimal solution is *hidden* in the space of real numbers, outside the very restricted set of segment points. The method proposed in this paper *learns* optimal motifs lying in the real-numbers space through a tailored numerical optimization technique. Even though the aforementioned 2-dimensional example was *created* to awake the reader on the need for *learning* motifs, still empirical results of Section 6.2 will demonstrate that learning motifs yields more frequently occurring patterns, compared to searching them, on real-life time series.

4.2. Smooth (Differentiable) Motif Frequency

We are going to find the optimal motif through a mathematical maximization of the frequency as a function of the motifs. Unfortunately, the frequency of Equation 1 has two problems (i) it is not continuous at point $||M_{k,:} - S_{j,:}|| = T$ and (ii) first derivative is zero in all other points (i.e. frequency is flat having values 1 or 0). Therefore we cannot compute the optimal motifs using gradient-based optimization. However, we can use a differentiable approximation for the frequency function using the Gaussian kernel of Equations 4-5.

$$\hat{\mathcal{F}}(M) \;=\; \frac{1}{KJ}\sum_{k=1}^{K}\sum_{j=1}^{J}\hat{\mathcal{F}}_{k,j} \tag{4}$$

$$\hat{\mathcal{F}}_{k,j} \;=\; e^{-\frac{\alpha}{T}\sum_{l=1}^{L}(M_{k,l}-S_{j,l})^2} \tag{5}$$

The smooth frequency function of Equation 5 is both an accurate approximation to the frequency measure from Equation 2, but also a differentiable alternative, as illustrated in Figure 2 (left plot). The parameter α controls the smoothness of the soft frequency. For optimization reasons (details in Section 4.4) the frequency sum of Equation 4 is divided by KJ to limit the value of $\hat{\mathcal{F}}$ between 0 and 1. In terms of notation, the approximated frequency is distinguished by a hat ($\mathcal{F}$ vs $\hat{\mathcal{F}}$).

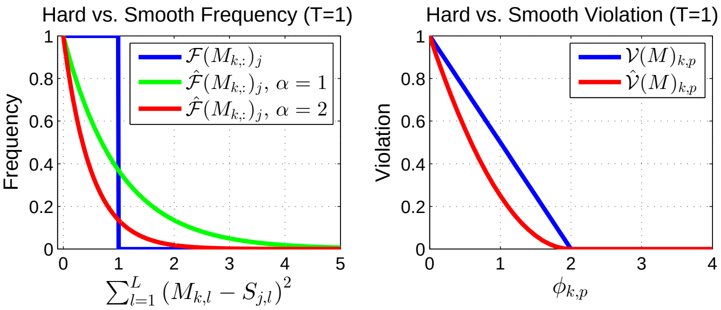

Fig. 2. Smooth vs. Hard Variants of Frequency (left) and Diversity Violation (right)

4.3. Motif Diversity Violation

As previously described in Equation 3 the motifs need to be distant by a margin of $2T$. We call such a property as *motif diversity*. In that line, this section is devoted to formalizing a differentiable penalty function for the violations of the distances among motifs from the diversity threshold of $2T$. As a first step, the distance between two motifs $M_{k,:} \in \mathbb{R}^L$ and $M_{p,:} \in \mathbb{R}^L$ is defined as $\phi_{k,p} : \left(\mathbb{R}^L \times \mathbb{R}^L\right) \rightarrow \mathbb{R}$ and formalized in Equation 6.

$$\phi_{k,p} = \sum_{l=1}^{L}\left(M_{k,l} - M_{p,l}\right)^2 \tag{6}$$

The distance $\phi_{k,p}$ of any pair of motifs $M_{k,:}, M_{p,:}$ should obey to the diversity constraint shown in Equation 7.

$$\phi_{k,p} > 2T, \quad \forall k \in (\{1, \ldots, K\}, \, \forall p \in \{k+1, \ldots, K\}) \tag{7}$$

We introduce the concept of *diversity violation* by Equations 8-9. For each of the $\frac{K(K-1)}{2}$ pairs of motifs, the violation is 0 if the distance between the pair motifs is greater than $2T$. Otherwise, if the distance is zero then the motifs are identical (hence not at all diverse) and a maximum violation of one is returned. For all the distances between 0 and $2T$ a linear violation between 0 and 1 is returned as formalized in Equation 9. The constant term $\frac{2}{K(K-1)}$ makes sure that the violation function has a range between 0 and 1, the same range as the approximative frequency.

$$\mathcal{V}(M) = \frac{2}{K(K-1)} \sum_{k=1}^{K} \sum_{p=k+1}^{K} \mathcal{V}_{k,p} \tag{8}$$

$$\mathcal{V}_{k,p} = \begin{cases} 1 - \frac{\phi_{k,p}}{2T} & \phi_{k,p} < 2T \\ 0 & \phi_{k,p} \geq 2T \end{cases} \tag{9}$$

Despite achieving its aim, the violation penalty of Equations 8-9 still it suffers in terms of differentiability at the point $\phi_{k,p} = 2T$. Therefore, we are proposing a smooth and differentiable variant of the violation penalty in Equations 10-11 by squaring the hard violation of Equation 9.

$$\hat{\mathcal{V}}(M) = \frac{2}{K(K-1)} \sum_{k=1}^{K} \sum_{p=k+1}^{K} \hat{\mathcal{V}}_{k,p} \tag{10}$$

$$\hat{\mathcal{V}}_{k,p} = \begin{cases} \left(1 - \frac{\phi_{k,p}}{2T}\right)^2 & \phi_{k,p} < 2T \\ 0 & \phi_{k,p} \geq 2T \end{cases} \tag{11}$$

As in the case of the frequency, we denote the smooth approximative version of the violation penalty by a hat ($\mathcal{V}$ for hard and $\hat{\mathcal{V}}$ for smooth). The violation penalty as a function of the distance between motif pairs is depicted in the right plot of Figure 2.

4.4. Motif Learning Through Optimization

This section fuses the smooth motif frequency and smooth motif diversity violation into a meaningful objective function. Our aim is to learn a set of K motifs that *maximize* the frequencies and *minimize* (*have no*) violations. Such an objective can be constructed as the maximization task of Equation 12.

$$\begin{aligned} M^* &= \underset{M}{\text{argmax}} \;\; \mathcal{O}(M) \\ &= \underset{M}{\text{argmax}} \;\; \hat{\mathcal{F}}(M) - \hat{\mathcal{V}}(M) \end{aligned} \tag{12}$$

The universally optimal motifs are those which achieve the universal maximum value of our objective function $O(M) = \hat{\mathcal{F}}(M) - \hat{\mathcal{V}}(M)$. As both terms are positive, the objective is maximized for the highest motif frequencies and zero violations. In this paper we will optimize the objective function through gradient ascent motif updates in a series of iterations. Since both ranges of $\hat{\mathcal{F}}$ and $\hat{\mathcal{V}}$ are between 0 and 1, no term over-scales the other and the overall learning does converge. In our preliminary experiments we found out that a trade-off coefficient β in the form $\hat{\mathcal{F}}(M) - \beta\hat{\mathcal{V}}(M)$ was not needed as both terms converge quickly.

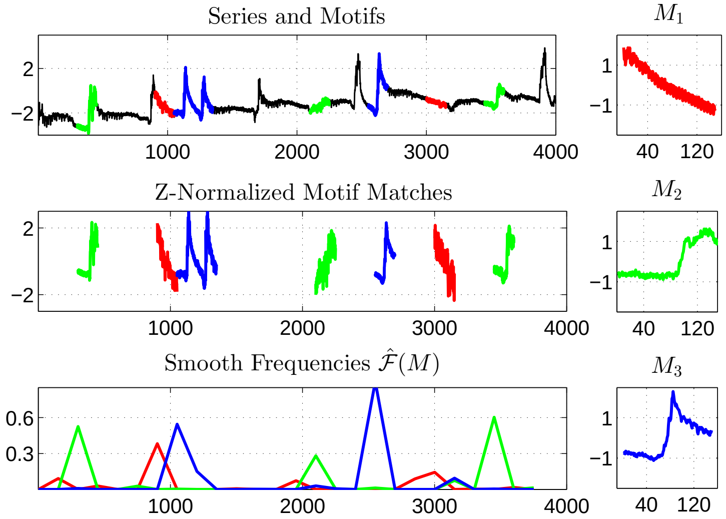

Fig. 3. Top-3 motifs from the "Insect B" time series $(L = 150, T = 61, \eta = 0.3, I = 300, \alpha = 2)$

The output of the learning process is a set of motifs M, as shown in Figure 3 for the "Insect B" time series. In this illustration the top three motifs $(K = 3)$ are shown on the right plots, while the matches of the motifs on the time series are shown in the upper-left plot. Z-normalized versions of the matched segments are shown in the middle-left plot and the lower-left plot illustrates the per-segment smooth frequency scores of the motifs.

4.4.1. Gradient Ascent Optimization. Since the objective function of Equation 12 is a subtraction of frequency and diversity violations, the partial gradient of the objective function with respect to each point l of any k-th motif is decomposable as shown in Equation 13.

$$\frac{\partial \mathcal{O}(M)}{\partial M_{k,l}} = \frac{\partial \hat{\mathcal{F}}(M)}{\partial M_{k,l}} - \frac{\partial \hat{\mathcal{V}}(M)}{\partial M_{k,l}} \tag{13}$$

The partial derivative of the smooth frequency with respect to the motif is computed as the first derivative of Equation 4 in terms of M and shown below in Equation 14.

$$\frac{\partial \hat{\mathcal{F}}(M)}{\partial M_{k,l}} = \frac{-2\alpha}{KJT} \sum_{j=1}^{J} \left(M_{k,l} - S_{j,l}\right) \hat{\mathcal{F}}_{k,j} \tag{14}$$

Similarly the partial derivative of the diversity violation with respect to each motif's point is defined in Equation 15.

$$\frac{\partial \hat{\mathcal{V}}(M)}{\partial M_{k,l}} = \frac{2}{K(K-1)} \sum_{q=1}^{K} \frac{\partial \hat{\mathcal{V}}_{k,q}}{\partial M_{k,l}} \tag{15}$$

$$\frac{\partial \hat{\mathcal{V}}_{k,q}}{\partial M_{k,l}} = \begin{cases} \frac{(\phi_{k,q}-2T)(M_{k,l}-M_{q,l})}{T^2} & \phi_{k,q} < 2T \\ 0 & \phi_{k,q} \geq 2T \end{cases}$$

4.5. Learning Algorithm

Having defined the partial derivative needed for gradient ascent, we can present the complete learning method. Our method is detailed in Algorithm 2 and in this section we will explain the steps of the algorithm in detail. There are a set of hyper-parameters to the learning process, starting with the frequency smoothness α. The other important hyper-parameters are the number of motifs K, the threshold T and the motif length L, to be set by a practitioner. The learning rate η and the number of iterations I are less critical hyper-parameters that control the number of steps needed until convergence. For small learning rates and large number of iterations, the convergence is safely achievable.

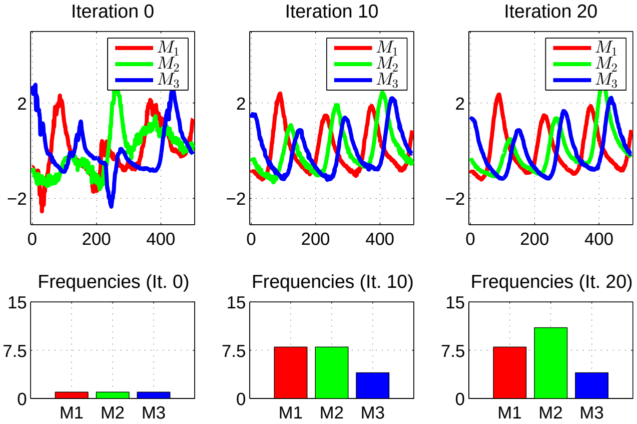

Fig. 4. Metamorphosis of three motifs on the "EOG" time series $(L = 150, T = 58, \eta = 0.3, I = 300, \alpha = 2)$

The algorithm starts with a set of motifs initialized from random segments and updates them in the direction of the partial gradients using a learning rate step size. The learning rate is dynamically updated per each point of each motif using an adaptive technique known as AdaGrad [Duchi et al. 2011]. We accumulate the square of the partial gradients into accumulators denoted by ∇. In order to speed-up the updates we pre-compute the per-segment frequencies $\hat{\mathcal{F}}_{k,j}$ and pair distances $\phi_{k,q}$ in lines 9-12. Then every point of each motif $M_{k,l}$ is updated in the positive direction of the derivative in lines 13-25. The partial gradients correspond to the ones previously explained in Section 4.4.1. The update of line 24 adjusts the learning rate by the square root of the accumulated square gradients [Duchi et al. 2011].

As a consequence of the gradient ascent updates, the motifs undergo a *metamorphosis* as is shown in Figure 4 for the "Full EOG" time series. The illustrative motifs are learned on the first 10000 non-overlapping segments of the time series having length $L = 150$. At the beginning (Iteration 0) the motifs are random and the corresponding

Algorithm 2 LearnMotifs()

```
1: Input: Threshold $T \in \mathbb{R}^+$, Motif length $L \in \mathbb{N}^+$, Number of Motifs $K \in \mathbb{N}^+$, Segments $S \in \mathbb{R}^{J \times L}$, Learning Rate $\eta \in \mathbb{R}^+$, Number of iterations $I \in \mathbb{N}^+$, Smoothness $\alpha \in \mathbb{R}^+$
2: Output: Motif $M \in \mathbb{R}^{K \times L}$
3: // Initialize random motifs and gradient accumulators:
4: $M \leftarrow \left(S_{\mathcal{U}(1,J),:}\right)^K, \nabla \leftarrow 0^{K \times L}$
5: // Initialize constant values:
6: $c_{\hat{\mathcal{V}}} \leftarrow \frac{2}{K(K-1)T^2},\ \ c_{\hat{\mathcal{F}}} \leftarrow \frac{-2\alpha}{KJT}$
7: // Iterate the learning method:
8: for iter= $1, \ldots, I$ do
9:     // Precompute the per-segment occurence scores:
10:    $\hat{\mathcal{F}}_{k,j} \leftarrow e^{-\frac{\alpha}{T}\sum_{l=1}^{L}(M_{k,l}-S_{j,l})^2} \quad \forall k \in \mathbb{N}_1^K, \forall j \in \mathbb{N}_1^J$
11:    // Precompute the pair-wise motif distances:
12:    $\phi_{k,q} \leftarrow \sum_{l=1}^{L} \left(M_{k,l} - M_{q,l}\right)^2, \quad \forall k \in \mathbb{N}_1^K, \forall q \in \mathbb{N}_1^K$
13:    // Update the motifs :
14:    for $k = 1, \ldots, K;\ \ l = 1, \ldots, L$ do
15:        // Gradient of frequency w.r.t. the motif:
16:        $\frac{\partial \hat{\mathcal{F}}(M)}{\partial M_{k,l}} = c_{\hat{\mathcal{F}}} \sum_{j=1}^{J} \left(M_{k,l} - S_{j,l}\right) \hat{\mathcal{F}}_{k,j}$
17:        // Gradient of diversity violation w.r.t. the motif:
18:        $\frac{\partial \hat{\mathcal{V}}(M)}{\partial M_{k,l}} = c_{\hat{\mathcal{V}}} \sum_{q=1}^{K} \begin{cases} \left(\phi_{k,q} - 2T\right)\left(M_{k,l} - M_{q,l}\right) & \phi_{k,q} < 2T \\ 0 & \phi_{k,q} \geq 2T \end{cases}$
19:        // Gradient of the final objective w.r.t. the motif:
20:        $\frac{\partial \mathcal{O}(M)}{\partial M_{k,l}} \leftarrow \frac{\partial \hat{\mathcal{F}}(M)}{\partial M_{k,l}} - \frac{\partial \hat{\mathcal{V}}(M)}{\partial M_{k,l}}$
21:        // Update the history of gradients:
22:        $\nabla_{k,l} \leftarrow \nabla_{k,l} + \left(\frac{\partial \mathcal{O}(M)}{\partial M_{k,l}}\right)^2$
23:        // Update the motif point:
24:        $M_{k,l} \leftarrow M_{k,l} + \frac{\eta}{\sqrt{\nabla_{k,l}}} \frac{\partial \mathcal{O}(M)}{\partial M_{k,l}}$
25:    end for
26: end for
27: return $M$
```

frequencies zero, however the motifs start to take form after approximately 20 iterations and converge after 40 iterations. The metamorphosis of the motifs is conducted such that their matching frequencies (lower plots) are maximized.

4.6. Convergence of The Learning Algorithm

The learning algorithm converges by updating the motifs so that the approximative frequency is maximized and the diversity violations minimized to zero as shown in Figure 5 (left plot) for an execution on the ”Insect B” dataset. It is worth noting that the inclusion of the penalty on the diversity violation is crucial for preserving the diversity constraint. An experiment is shown on the right plot of Figure 5. In this experiment the line 24 of Algorithm 2 is edited so the motifs are updated only with respect to the frequency and not diversity violation (see plot title). As we can clearly see, maximizing the frequencies without penalizing diversity violations causes the motifs to be similar

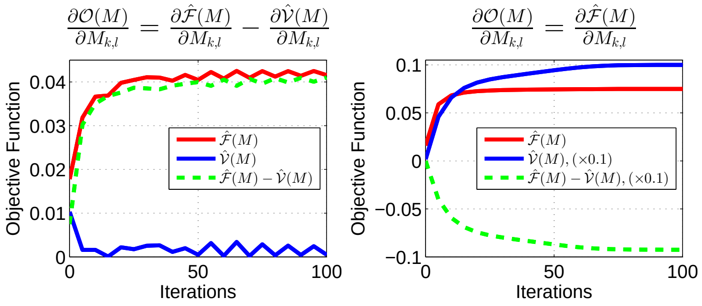

Fig. 5. Convergence on "Insect B" dataset ($K = 5, T = 382, \eta = 0.3$)

to each other. That is demonstrated by the fact that the violation measure increases, as shown in the right plot of Figure 5.

5. OPTIMALITY OF OUR METHOD

The objective function of Equation 12 is not concave, because the frequency function is a sum of Gaussians and not concave. We demonstrate the non-concavity of the frequency function in Figure 6 using the TAO and EEG LSF5 datasets. Here we generate all possible motifs of length 500 using two values, (for the sake of a 3d-plot), one value for all the first 250 points in X-axis and another value for the last 250 points in the Y-axis. As can be clearly seen, frequency is not a concave function in terms of motifs and has multiple local maxima.

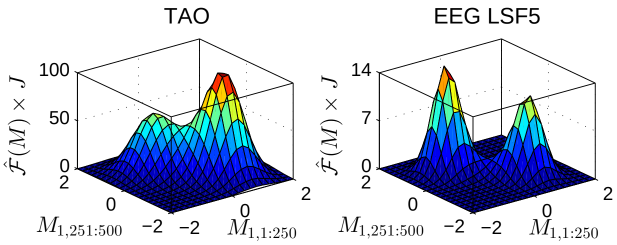

Fig. 6. Non-concave frequency $\hat{\mathcal{F}}(M)$ as a function of motif values $M_{1,:}$ on TAO and EEG LSF5 time-series datasets, Parameters: $L = 500, T = 100, \alpha = 2$

In case of non-concave functions (or non-convex for minimization problems), an effective cook-book solution is to combine gradient descent with a *random-restart* strategy [Lones 2011]. In order to avoid getting stuck in local maxima, the gradient descent optimization is restarted multiple times with random initial values for the motifs. The run that achieves the highest $\mathcal{F}(M)$ is selected, as is formalized in Equation 16, where the number of restarts is denoted by $R \in \mathbb{N}$. It is important to recognize that we select the motifs yielding the highest hard frequency $\mathcal{F}$, not the proxy smooth one $\hat{\mathcal{F}}$. The hard frequency $\mathcal{F}$ does **avoid** counting **trivial** matches in our implementation.

$$M^* := \underset{M^{(r)},\ r=1,\ldots,R}{\text{argmax}} \ \mathcal{F}(M^{(r)}) \qquad (16)$$
$$s.t. \ \ M^{(r)} \leftarrow \text{LearnMotif() from Alg 2}$$

Figure 7 illustrates the effect of 50 random restarts on the frequency function $\mathcal{F}(M)$ values over the TAO dataset. On the left plot we see that the maximum values of the objective are reached after a few restarts. The distribution of the frequency values, shown in the right plot, demonstrates that the histogram is normally distributed. That means there is a normal probability that a restart will yield an optimal value on the right portion (maximal) of the values within.

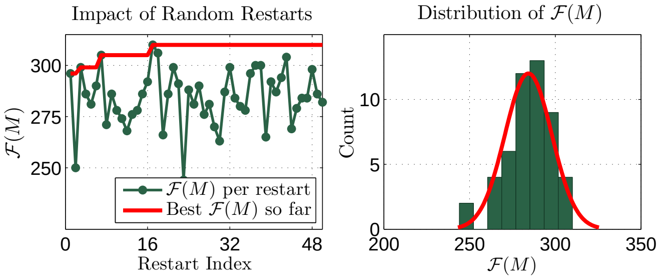

Fig. 7. Impact of Random Restarts on $\mathcal{F}(M)$; "TAO" time-series dataset with hyper-parameters $L = 500, K = 10, \eta = 0.3, I = 300, T = 109.6$

5.1. Runtime Algorithmic Complexity

The runtime complexity of Algorithm 2 is determined by the pre-computation steps and the update steps. Computing of the frequency terms has an algorithmic complexity order of $\mathcal{O}(RIKJL)$, while computing the pairwise distances has a computational complexity of $\mathcal{O}(RIK^2L)$. The computation of the partial gradients of the frequency with respect to the motifs has a complexity of $\mathcal{O}(RIKJL)$. Similarly the complexity of computing the gradients of the diversity violation with respect to the motif has a complexity of $\mathcal{O}(RIK^2L)$. The overall complexity of the algorithm is $\mathcal{O}(RIKJL + RIK^2L + RIKJL + RIK^2L)$, which translates to $\mathcal{O}(2RIK\,(J + K)\,L) \sim \mathcal{O}(RIKJL)$ since $K << J$. The brute force search on the other hand, has a complexity of $\mathcal{O}(J^2L)$ which is quadratic in terms of the number of segments J. In contrast our method is linear in terms of the number of segments J and faster than the brute-force search in case $RIK < J$. It is worth reminding that our algorithm learns optimal motifs (brute-force finds non-optimal motifs) and the primary strength is quality at a feasible runtime.

6. EMPIRICAL RESULTS

6.1. Experimental Setup

We compare the quality of the proposed methods against the brute-force search strategy using a battery of six time-series datasets from diverse application domains. In addition, we employ an evaluation protocol which compares the frequencies of the computed motifs per different number of motifs, motif lengths and distance thresholds.

6.1.1. Datasets

- **Insect B** is a time series of insect behavior data and has a length of 73929 points [Mueen et al. 2009b].
- **TAO** is a long time series representing Tropical Atmosphere Ocean temperature measurements having 741528 measurements[1].

[1] www.pmel.noaa.gov/tao

- **RandomWalk** is a time-series dataset consisting of 1000000 points, among which motifs at randomly selected time-stamps are implanted [Mueen et al. 2009b].
- **EEG** is a series of 1802136 continuous measurements from electroencephalographic sensors, measuring voltage differences across the scalp [Mueen et al. 2009b].
- **Salinity** is a time series containing recordings on the level of oceanic salt concentration. The data has a length of 2324134 points and is provided by the National Oceanographic Data Center[2].
- **EOG** is the longest series in our collection consisting of 8099500 points. The data is collected by an Electro-Oculogram and represent electrical potential between the front and the back of a human eye [Goldberger et al. e 13].

6.1.2. Baseline. Many motif discovery method are based on searching for frequent patterns among the series segments (e.g. [Patel et al. 2002; Yankov et al. 2007; Chiu et al. 2003; Li et al. 2012; Li and Lin 2010], enumerated in a broader scope in Section 2). While those search-based methods are successful in terms of scalability, data representation, on-line learning, etc..., they are still upper bounded in quality (a.k.a. frequency) by the Brute-Force search. That is trivial to show, because all the frequent sub-sequences those methods could find are also detectable by Brute-Force search. In that aspect, it is sufficient to demonstrate that our method is superior to Brute-Force searching in terms of **quality** (a.k.a. **frequency**) and that naturally translates into qualitative superiority against all the other scalable/approximate/on-line search-based methods.

6.1.3. Evaluation Protocol. We will compare against the brute-force search algorithm as the most qualitative *search-based* baseline. Our protocol involves comparisons across all the parameters of both the searching- and learning- based methods.

Three different number of motifs will be computed $K \in \{3, 10, 30\}$ having two different lengths $L \in \{500, 1000\}$. Furthermore, the threshold (T) of the experiments is chosen as a percentile in the distribution of distances between segments. To illustrate the setup, a length corresponding to the 1%-th percentile, (denoted Pct $= 1$ in Table I) means that 1-% of segments pairs have a pairwise Euclidean distance smaller than the threshold. In that way we can compare our method against the brute-force search across a range of thresholds computed by different percentiles $T \in \{0.001\%, 0.01\%, 0.1\%, 1\%\}$ of the pairwise distances of segments. In that way we avoid hand-picking different thresholds values per dataset and select the threshold in a data-driven neutral manner. In order to ensure convergence, the learning rate was set to an initial value of $\eta = 0.1$ and the number of iterations to $I = 1000$. In addition, the optimization was restarted $R = 200$ times. The segments were extracted from the series by sliding a window and normalizing the clipped segment, while the window is slid by half of the motif length. For every combination of the number of motifs K, length L and threshold T (computed from the percentile), three different values of frequency smoothness were searched $\alpha \in \{1, 2, 3\}$, keeping the one yielding the highest $\mathcal{F}$ value.

The brute-force search baseline was executed using the **same** $K, L, T(\text{Pct})$ combination parameters as the learning-based approach, and for both methods the final frequency $\mathcal{F}$ does **not** include trivial matches. In order to be entirely transparent to the research community we publicly shared our source code and the data used in this paper in an on-line repository[3].

Table I. Hard Frequencies: Learning Motifs (LM) vs. Brute Force Motifs (BFM)

	L=500								*L=1000*							
	Pct=0.001		*Pct=0.01*		*Pct=0.1*		*Pct=1*		*Pct=0.001*		*Pct=0.01*		*Pct=0.1*		*Pct=1*	
	BFM	**LM**	**BFM**	**LM**	**BFM**	**LM**	**BFM**	**LM**	**BFM**	**LM**	**BFM**	**LM**	**BFM**	**LM**	**BFM**	**LM**
Datasets	***Top-3 (K=3)***								***Top-3 (K=3)***							
Insect B	4	**9**	6	**10**	16	**45**	44	**151**	4	**11**	4	**13**	9	**27**	19	**51**
TAO	12	**24**	29	**45**	86	**119**	313	**429**	10	**12**	18	**35**	56	**98**	219	**284**
RandomWalk	25	**43**	74	**125**	239	**321**	697	**855**	9	**23**	27	**64**	114	**165**	327	**458**
EEG LSF5	17	**42**	47	**101**	150	**199**	388	**442**	11	**34**	27	**73**	96	**125**	232	**238**
Salinity	39	**48**	151	**184**	497	**590**	1462	**1718**	18	**32**	72	**94**	269	**330**	683	**876**
EOG	153	**190**	504	**669**	1646	**2168**	4957	**8042**	67	**102**	196	**340**	676	**1390**	2171	**5998**
Datasets	***Top-10 (K=10)***								***Top-10 (K=10)***							
Insect B	11	**18**	14	**23**	35	**78**	81	**189**	11	**29**	11	**28**	17	**54**	46	**97**
TAO	30	**48**	62	**95**	192	**314**	780	**1164**	18	**29**	44	**55**	112	**203**	344	**584**
RandomWalk	40	**79**	132	**206**	313	**579**	1314	**1502**	23	**48**	52	**118**	223	**310**	603	**768**
EEG LSF5	42	**109**	131	**273**	400	**557**	1118	**1266**	32	**96**	84	**212**	234	**379**	634	**810**
Salinity	100	**105**	291	**358**	1000	**1149**	2797	**2995**	47	**59**	136	**198**	456	**597**	1222	**1564**
EOG	263	**283**	973	**1296**	3128	**4130**	11181	**13439**	122	**164**	417	**685**	1552	**2206**	4321	**5729**
Datasets	***Top-30 (K=30)***								***Top-30 (K=30)***							
Insect B	31	**40**	36	**47**	68	**107**	200	**221**	32	**49**	32	**49**	42	**72**	89	**110**
TAO	65	**95**	133	**209**	432	**698**	1720	**2193**	38	**55**	65	**93**	202	**336**	577	**932**
RandomWalk	61	**117**	158	**279**	471	**764**	1778	**2249**	45	**87**	83	**174**	256	**421**	989	**1151**
EEG LSF5	110	**281**	275	**646**	850	**1442**	2541	**3505**	72	**205**	153	**428**	417	**879**	1304	**1914**
Salinity	162	**199**	428	**540**	1260	**1456**	3270	**3855**	91	**107**	233	**284**	660	**779**	2038	**2150**
EOG	427	**557**	1494	**2028**	5200	**5681**	**17442**	17075	247	**338**	787	**1186**	2306	**2955**	6227	**7349**
Wins	0	**18**	0	**18**	0	**18**	1	**17**	0	**18**	0	**18**	0	**18**	0	**18**

6.2. Results

In the conducted experiments, for all the different thresholds T (computed through the percentile), for all the different number of motifs K and for different motif lengths L, the motifs learned through our method **almost always** had a higher frequency than the ones found through brute-force search. Table I displays the empirical results comparing the frequency score of the optimal learned motif (denoted LM) against the motifs found through brute-force search (denoted BFM).

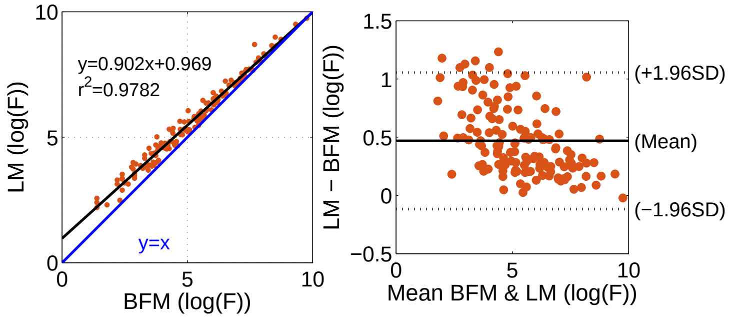

Fig. 8. Bland-Altman plot showing significance of LM vs BFM frequencies (log-scale for visual comprehension)

The results of Table I indicate that learning the motifs (LM) is better than searching (BFM) them in 99.31% of the experiments (143/144). The improvement arising from

[2] http://www.nodc.noaa.gov/General/salinity.html
[3] http://fs.ismll.de/publicspace/LearnMotifs/

learning motifs (LM) in terms of motif frequencies is in average $67 \pm 56\%$ better than the search-based approach (BFM). The famous Bland-Altman plot is used to assess the significance of the improvements. Figure 8 (left plot) shows the dominating ratio of LM through least-squares fitting. Moreover, the right plot shows that the difference LM-BFM and its standard deviations are above zero, thus we have a significant difference in terms of frequencies.

Table II. Running Times (seconds) of Learning Motifs

	L=500				***L=1000***			
	Pct=0.001	***Pct=0.01***	***Pct=0.1***	***Pct=1***	***Pct=0.001***	***Pct=0.01***	***Pct=0.1***	***Pct=1***
Datasets	***Top-3 (K=3)***				***Top-3 (K=3)***			
Insect B	14.3	14.9	14.6	14.5	21.3	22.0	20.4	21.4
TAO	231.8	234.0	236.5	234.9	233.5	229.6	239.6	229997
RandomWalk	319.4	319.4	338.1	317.5	326.2	328.0	324.4	349844
EEG LSF5	11990.3	711.5	46473.8	10472.3	688.2	7160.4	6652.2	6817.2
Salinity	4690.6	2918.8	2837.9	3085.0	3710.4	1042.3	6925.6	1184.0
EOG	74114.1	19786.1	107585.0	74479.9	4679.0	4365.0	35988.8	35642.7
Datasets	***Top-10 (K=10)***				***Top-10 (K=10)***			
Insect B	47.1	251.1	48.0	47.9	74.8	179.5	73.5	197797
TAO	763.0	2309.3	798.3	780.0	776.6	2293.1	5627.6	764.9
RandomWalk	1097.0	1074.4	44425.9	44318.9	1056.6	1074.4	34507.4	10894.9
EEG LSF5	154632.2	154325.2	2477.9	8778.3	22054.0	22651.2	2482.0	8634.1
Salinity	9805.0	206640.0	9232.4	10696.9	3605.6	3274.3	3323.6	3382.3
EOG	59876.8	65480.0	58343.4	291463.4	122449.0	40395.9	122389.2	43407.9
Datasets	***Top-30 (K=30)***				***Top-30 (K=30)***			
Insect B	306.9	460.8	286.3	333.8	367.8	575.6	388.5	253.5
TAO	2429.9	2400.8	5061.9	5422.3	97017.7	2526.7	5018.9	5440.9
RandomWalk	3114.1	3524.7	3188.2	3216.4	11319.4	3281.6	3184.1	75287.5
EEG LSF5	7558.7	7478.6	7488.1	7328.3	130566.3	49818.7	7560.4	7207.7
Salinity	41372.2	37847.7	36486.3	23483.1	32050.3	9914.4	89407.1	88720.4
EOG	122752.6	140252.5	558185.7	120871.9	40991.9	42518.8	67828.3	44654.8

Even though the proposed method is significantly better in quality that the search-based alternatives, it is not the fastest method in the literature. We are emphasizing that learning the motifs in our experiments was in general up to two/three orders slower than searching the motifs. However, since our method is always better in terms of quality than searching, our primary objective is to show that our approach is *practically feasible* in terms of run-time. In that context, learning the Top-30 motifs of Insect_B (smallest dataset) took 4.7 minutes, while learning the Top-30 motifs of EOG (largest dataset) took 33.57 hours, in a cluster having Intel Xeon E5-2670v2 processors with speed 2.50GHz. The full table of runtimes can be accessed from Table II, while the runtime results for searching motifs are shown in Table III.

7. CASE STUDY: AUDIO MOTIFS

In this case study we extract motifs from audio files. The case discussed in this thread is a poem by Edgar Allen Poe, titled ”The Bells” and famous for its onomatopoeic nature in terms of repeating the word ”Bells”. We extract a time-series representation of the audio file through the first channel of the Mel-frequency cepstral coefficients (MFCC). For the sake of illustration we took the first 300000 measurements of the original WAV file, corresponding to a 68 seconds audio reading of the poem.

Figure 9 illustrates shows the MFCC representation time-series together with the results of the brute force search algorithm in blue and our proposed method in red. We extracted three motifs $K = 3$ of length $L = 300$ for both methods. The distance threshold used in the experiment is the 0.1%-th percentile of pair-wise segment distances corresponding to a value of $T = 171.56$. For each method, we display the location of

Table III. Running Times (msecs) of searching brute-force motifs

	L=500				***L=1000***			
	Pct=0.001	***Pct=0.01***	***Pct=0.1***	***Pct=1***	***Pct=0.001***	***Pct=0.01***	***Pct=0.1***	***Pct=1***
Datasets	***Top-3 (K=3)***				***Top-3 (K=3)***			
Insect B	60	63	60	57	43	40	36	42
TAO	8092	8986	8524	7963	3550	3328	3367	3855
RandomWalk	12637	14398	13129	17302	5326	6336	7479	6551
EEG LSF5	29825	45684	80684	46916	19170	16007	44147	48198
Salinity	371804	63604	75709	59381	100073	52378	22855	29939
EOG	1328921	974995	1870861	599013	432456	389050	883847	1002247
Datasets	***Top-10 (K=10)***				***Top-10 (K=10)***			
Insect B	65	75	63	81	44	55	44	54
TAO	7659	6022	7651	6404	4243	3068	2093	3231
RandomWalk	15876	13373	28573	25452	5178	5980	18486	4644
EEG LSF5	84019	78900	37632	39182	44384	39119	21335	41621
Salinity	90910	214055	64057	63665	31852	33435	30476	26541
EOG	906873	854037	722941	697854	882154	792652	878873	1523925
Datasets	***Top-30 (K=30)***				***Top-30 (K=30)***			
Insect B	248	412	112	115	101	100,	80	53
TAO	6806	5418	6894	7915	9288	3307	5607	2468
RandomWalk	15973	39291	8402	9885	7785	7532	7188	14683
EEG LSF5	45499	45629	50080	38360	40905	14501	18317,	16803
Salinity	67964	222415	76594	74409	34896	30701	82041	93272
EOG	970592	1137155	1324029	956936	402382	314477	344206	333424

the motif matches over series segments with a filled oval mark. Under the plots of the matches we show the found motifs together with the corresponding frequencies. For the same distance threshold, the learned motifs have totally 50 matches while the searched motifs have 35 matches, for an improvement of 42% in terms of frequency. Our method learns patterns that *for exactly the same distance threshold* match more frequently than the brute-force motifs.

An investigation of the motif sounds reveals that the top-K repetitive sounds are different pronunciations of the word bell. All the motifs are different from each other by 2T, so they are all legit motifs by definition. Let us analyze how optimality translates in concrete terms. For instance we can consider the segment between points 10000-15000 in the times series, which corresponds to the following poem text:

... Of the rapture that impels
To the swinging and the ringing
Of the *bells*, *bells*, *bells* -
Of the *bells*, *bells*, *bells*, *bells*,
Bells, *bells*, *bells* -
To the rhyming and the chiming of the *bells*! ...

Within the above segment, the brute-force motifs can find $7, 10, 7$ occurrences of the word bell within a threshold T. Our motifs can find $9, 11, 9$ matches within the same interval and for exactly the same distance threshold T. As the ground truth text above indicate, there are 11 ”bells” pronunciations in total. In average, given the specified threshold T, the brute-force motifs find similar sounds that match to the word Bells in 72% of the cases, the matches of our optimal motifs correspond to the word Bells in average on 88% of the cases. This is a very important detection accuracy given that we used only the first channel of the MFCC representation, which is a low-resolution representation that encapsulates only the overall loudness of the sound.

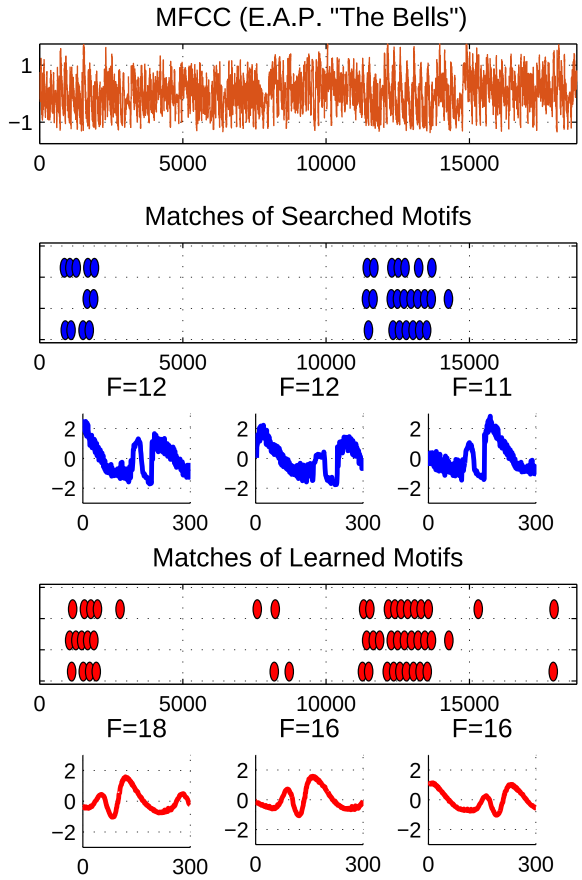

Fig. 9. Learning 3 audio motifs on a read version of the ”The Bells” poem from Edgar Allan Poe. The method parameters are: $T = 171.56$ $(Pct = 0.1\%)$, $L = 300, \alpha = 3, \eta = 0.3, I = 1000, R = 4$.

8. DISCUSSION ON MOTIF QUALITY CRITERION

Given the fact that motifs have been utilized in diverse ways (see Section 2), one could question whether frequency is best quality criterion for searching motifs. However, when searching for the most repetitive pattern (our problem definition), frequency is the only meaningful quality measure for finding those patterns. In the context of time series, repetitiveness is defined as the number of matches given a distance threshold T (a hyper-parameter). Such a threshold-based match is not necessary in the case of discrete valued sequences, such as strings (e.g. DNA) or transactions (e.g. frequent item set mining), where a match means direct equality. From an optimization perspective, the right outcome of this paper is the comparison of two numerical optimization approaches. Searching and learning motifs solve the same objective function (Equation 12), with a difference on the way they compute the parameters M (the motifs). To give a hint: Assume you have a list of empirical values Y and you need to find the value of a parameter x that achieves the maximum value of the function $f(x, Y)$ over Y (in our case the frequency of x on Y). The search approach finds x by guessing possible values among $x \in Y$, while the learning approach uses the slope indication of $\frac{d\,f(x,Y)}{d\,x}$ to update $x \in \mathbb{R}$. The standard mathematical approaches are often orthogonal to current practices of computer scientists (data miners), who often minimize problems by guess-searching for candidate solutions, instead of utilizing first or second order derivative (curvature) information on the surface of the objective functions. Yet, given the highly non-convex nature of the objective functions for various data mining problems, search-based heuristics still find competitive local optima solutions, compared to the slower and better local optima computed through derivative-based solvers. On other cases the situation arises from the lack of mathematically-principled problem formalizations which would enable derivative-based solvers. In the motifs case, we believe the community had previously failed to correctly formalize the problem as a parametric maximization function. Our contribution can be seen two folds: A) formalizing the problem and B) proposing a numerical optimization that computes better local optima than the existing guess-searching based numerical optimization.

9. CONCLUSION

This paper proposed a new perspective in learning time-series motifs. In contrast to current state of the art techniques which **searches** out motif candidates from series segments, our method **learns** them in a principled optimization. The motif frequency is approximated as a differentiable function and a gradient ascent method is proposed to find the motif values which maximize the objective function. In order to avoid local optima, a random restart strategy is combined with the gradient ascent learning of the motifs.

Learned optimal motifs have more segment matches than the motifs found through searching, for the same distance threshold. The optimal motifs represent latent patterns not necessarily present as sub-sequences in an explicit form, therefore can identify motifs which are in the center of the densest hyper-balls including segment points. Detailed experimental results demonstrate that learning optimal motifs **always** produces more qualitative motifs than searching them.

10. ACKNOWLEDGMENT

This research has been co-funded by the Seventh Framework Programme of the European Commission, through the REDUCTION project (#288254)[4] and by Deutsche Forschungsgemeinschaft within the project HyLAP[5]. The authors also acknowledge Prof. Eamonn Keogh, University of California, Riverside, and Dr. Lucas Drumond, ISMLL, University of Hildesheim, whose suggestions helped improving the quality of the work.

REFERENCES

André EX Brown, Eviatar I Yemini, Laura J Grundy, Tadas Jucikas, and William R Schafer. 2013. A dictionary of behavioral motifs reveals clusters of genes affecting Caenorhabditis elegans locomotion. *Proceedings of the National Academy of Sciences* 110, 2 (2013), 791–796.

Jeremy Buhler and Martin Tompa. 2001. Finding Motifs Using Random Projections. In *Proceedings of the Fifth Annual International Conference on Computational Biology (RECOMB '01)*. ACM, New York, NY, USA, 69–76. DOI:http://dx.doi.org/10.1145/369133.369172

N. Castro and P. Azevedo. 2010. Multiresolution Motif Discovery in Time Series. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, 2010, Columbus, Ohio, USA*. SIAM, 665–676.

N. Castro and P. Azevedo. 2011. Time Series Motifs Statistical Significance. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2011, 2011, Mesa, Arizona, USA*. SIAM, 687–698.

Nuno C. Castro and Paulo J. Azevedo. 2012. Significant Motifs in Time Series. *Stat. Anal. Data Min.* 5, 1 (Feb. 2012), 35–53. DOI:http://dx.doi.org/10.1002/sam.11134

[4] www.reduction-project.eu

[5] www.autonomous-learning.org

Joe Catalano, Tom Armstrong, and Tim Oates. 2006. Discovering Patterns in Real-Valued Time Series. In *Knowledge Discovery in Databases: PKDD 2006*, Johannes Frnkranz, Tobias Scheffer, and Myra Spiliopoulou (Eds.). Lecture Notes in Computer Science, Vol. 4213. Springer Berlin Heidelberg, 462–469. DOI:http://dx.doi.org/10.1007/11871637_44

Bill Chiu, Eamonn Keogh, and Stefano Lonardi. 2003. Probabilistic Discovery of Time Series Motifs. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*. ACM, New York, NY, USA, 493–498. DOI:http://dx.doi.org/10.1145/956750.956808

Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. 2001. *Introduction to Algorithms* (2nd ed.). McGraw-Hill Higher Education.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* 12 (July 2011), 2121–2159. http://dl.acm.org/citation.cfm?id=1953048.2021068

PedroG. Ferreira, PauloJ. Azevedo, CandidaG. Silva, and RuiM.M. Brito. 2006. Mining Approximate Motifs in Time Series. In *Discovery Science*, Ljupco Todorovski, Nada Lavrac, and KlausP. Jantke (Eds.). Lecture Notes in Computer Science, Vol. 4265. Springer Berlin Heidelberg, 89–101. DOI:http://dx.doi.org/10.1007/11893318_12

A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. 2000 (June 13). PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101, 23 (2000 (June 13)), e215–e220.

Hoang Thanh Lam, Ninh Dang Pham, and Toon Calders. 2011. *Online Discovery of Top-k Similar Motifs in Time Series Data*. Chapter 86, 1004–1015. DOI:http://dx.doi.org/10.1137/1.9781611972818.86

Yuan Li and Jessica Lin. 2010. Approximate Variable-length Time Series Motif Discovery Using Grammar Inference. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining (MDMKDD '10)*. ACM, New York, NY, USA, Article 10, 9 pages. DOI:http://dx.doi.org/10.1145/1814245.1814255

Yuan Li, Jessica Lin, and Tim Oates. 2012. Visualizing Variable-Length Time Series Motifs. In *Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012.* SIAM / Omnipress, 895–906. DOI:http://dx.doi.org/10.1137/1.9781611972825.77

Zheng Liu, JeffreyXu Yu, Xuemin Lin, Hongjun Lu, and Wei Wang. 2005. Locating Motifs in Time-Series Data. In *Advances in Knowledge Discovery and Data Mining*, TuBao Ho, David Cheung, and Huan Liu (Eds.). Lecture Notes in Computer Science, Vol. 3518. Springer Berlin Heidelberg, 343–353. DOI:http://dx.doi.org/10.1007/11430919_41

Michael Lones. 2011. Sean Luke: essentials of metaheuristics. *Genetic Programming and Evolvable Machines* 12, 3 (2011), 333–334.

David Minnen, Charles Isbell, Irfan Essa, and Thad Starner. 2007a. Detecting Subdimensional Motifs: An Efficient Algorithm for Generalized Multivariate Pattern Discovery. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining (ICDM '07)*. IEEE Computer Society, Washington, DC, USA, 601–606. DOI:http://dx.doi.org/10.1109/ICDM.2007.52

David Minnen, Charles L. Isbell, Irfan Essa, and Thad Starner. 2007b. Discovering Multivariate Motifs Using Subsequence Density Estimation and Greedy Mixture Learning. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1 (AAAI'07)*. AAAI Press, 615–620. http://dl.acm.org/citation.cfm?id=1619645.1619744

Yasser Mohammad and Toyoaki Nishida. 2014. Exact Discovery of Length-Range Motifs. In *Intelligent Information and Database Systems*, NgocThanh Nguyen, Boonwat Attachoo, Bogdan Trawiski, and Kulwadee Somboonviwat (Eds.). Lecture Notes in Computer Science, Vol. 8398. Springer International Publishing, 23–32. DOI:http://dx.doi.org/10.1007/978-3-319-05458-2_3

Abdullah Mueen. 2013. Enumeration of Time Series Motifs of All Lengths. *2013 IEEE 13th International Conference on Data Mining* 0 (2013), 547–556. DOI:http://dx.doi.org/10.1109/ICDM.2013.27

Abdullah Mueen and Eamonn Keogh. 2010. Online Discovery and Maintenance of Time Series Motifs. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*. ACM, New York, NY, USA, 1089–1098. DOI:http://dx.doi.org/10.1145/1835804.1835941

Abdullah Mueen, Eamonn Keogh, and Nima Bigdely-Shamlo. 2009a. Finding Time Series Motifs in Disk-Resident Data. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining (ICDM '09)*. IEEE Computer Society, Washington, DC, USA, 367–376. DOI:http://dx.doi.org/10.1109/ICDM.2009.15

Abdullah Mueen, Eamonn J Keogh, Qiang Zhu, Sydney Cash, and M Brandon Westover. 2009b. Exact Discovery of Time Series Motifs.. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2009*. SIAM.

T. Oates. 2002. PERUSE: An unsupervised algorithm for finding recurring patterns in time series. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. 330–337. DOI:http://dx.doi.org/10.1109/ICDM.2002.1183920

Pranav Patel, Eamonn Keogh, Jessica Lin, and Stefano Lonardi. 2002. Mining Motifs in Massive Time Series Databases. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM '02)*. IEEE Computer Society, Washington, DC, USA, 370–. http://dl.acm.org/citation.cfm?id=844380.844710

Majed Sahli, Essam Mansour, and Panos Kalnis. 2014. ACME: A scalable parallel system for extracting frequent patterns from a very long sequence. *The VLDB Journal* 23, 6 (2014), 871–893. DOI:http://dx.doi.org/10.1007/s00778-014-0370-1

Zeeshan Syed, Collin Stultz, Manolis Kellis, Piotr Indyk, and John Guttag. 2010. Motif Discovery in Physiological Datasets: A Methodology for Inferring Predictive Elements. *ACM Trans. Knowl. Discov. Data* 4, 1, Article 2 (Jan. 2010), 23 pages. DOI:http://dx.doi.org/10.1145/1644873.1644875

S. Tata and J.M. Patel. 2008. FLAME: Shedding Light on Hidden Frequent Patterns in Sequence Datasets. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. 1343–1345. DOI:http://dx.doi.org/10.1109/ICDE.2008.4497550

Alireza Vahdatpour, Navid Amini, and Majid Sarrafzadeh. 2009. Toward Unsupervised Activity Discovery Using Multi-dimensional Motif Detection in Time Series. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence (IJCAI'09)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1261–1266. http://dl.acm.org/citation.cfm?id=1661445.1661647

Dragomir Yankov, Eamonn Keogh, Jose Medina, Bill Chiu, and Victor Zordan. 2007. Detecting Time Series Motifs Under Uniform Scaling. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*. ACM, New York, NY, USA, 844–853. DOI:http://dx.doi.org/10.1145/1281192.1281282

MyatSu Yin, Songsri Tangsripairoj, and Benjarath Pupacdi. 2014. Variable Length Motif-Based Time Series Classification. In *Recent Advances in Information and Communication Technology*, Sirapat Boonkrong, Herwig Unger, and Phayung Meesad (Eds.). Advances in Intelligent Systems and Computing, Vol. 265. Springer International Publishing, 73–82. DOI:http://dx.doi.org/10.1007/978-3-319-06538-0_8

S. Yingchareonthawornchai, H. Sivaraks, T. Rakthanmanon, and C.A. Ratanamahatana. 2013. Efficient Proper Length Time Series Motif Discovery. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. 1265–1270. DOI:http://dx.doi.org/10.1109/ICDM.2013.111