Matrix Profile VI: Meaningful Multidimensional Motif Discovery

Chin-Chia Michael Yeh, †Nickolas Kavantzas, Eamonn Keogh
University of California, Riverside, †Oracle Corporation
myeh003@ucr.edu, nickolas.kavantzas@oracle.com, eamonn@cs.ucr.edu

***Abstract*—Time series motifs are approximately repeating patterns in real-valued time series data. They are useful for exploratory data mining and are often used as inputs for various time series clustering, classification, segmentation, rule discovery, and visualization algorithms. Since the introduction of the first motif discovery algorithm for univariate time series in 2002, multiple efforts have been made to generalize motifs to the multidimensional case. In this work, we show that these efforts, which typically attempt to find motifs on *all* dimensions, will not produce meaningful motifs except in the most contrived situations. We explain this finding and introduce *m*STAMP, an algorithm that allows meaningful discovery of multidimensional motifs. Beyond producing objectively and subjectively meaningful results, our algorithm has a host of additional advantages, including being much faster, requiring fewer parameters and supporting streaming data. We demonstrate the utility of our mSTAMP-based motif discovery framework on domains as diverse as audio processing, industry, and sports analytics.**

Keywords—Time Series; Motif Discovery; Multidimensional Data

I. Introduction

Time series motifs are approximately repeating patterns in real-value data, Fig. 2 shows some examples highlighted in the top two time series. They are useful in exploratory data mining. If a time series pattern is *conserved*, we may assume that there is some high-level atomic mechanism/behavior that causes that pattern to be conserved. That behavior may be desirable in certain cases (e.g., a perfect badminton shot [22]) or undesirable in others (e.g., the cough of a sick pig [7]). In either case, the discovery of motifs is often the first step in various kinds of higher-level time series analytics [26].

Since the introduction of the first motif discovery algorithm for univariate time series in 2002 [12], many researchers have generalized motifs to the multidimensional case [1][4][22][24]. However, almost all of these efforts attempt to find motifs on *all* dimensions. We believe that using all dimensions will generally not produce meaningful motifs, except in the most contrived situations. To see this in an intuitive setting, consider Fig. 1.

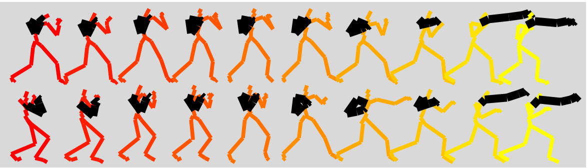

Fig. 1. Two motion-capture traces [6]. While the right-hand punch is nearly identical in both moves, the boxer in the top trace is throwing a cross. In contrast, the boxer in the bottom trace is throwing a one-two combo. A video of these motions is available in [19].

If we focus solely on the boxer's dominant hand, the two behaviors are almost identical. However, if we look that the full set of Mo-Cap markers on all of the limbs, the differences in the non-dominant hand and in the footwork "swamp" the similarity of the punch, making this repeated behavior impossible to find with the classic multidimensional motif discovery algorithms, that use all the available dimensions [1][4][22][24].

To see why this is, consider the multidimensional time series shown in Fig. 2 (we will formalize our definitions in Section III).

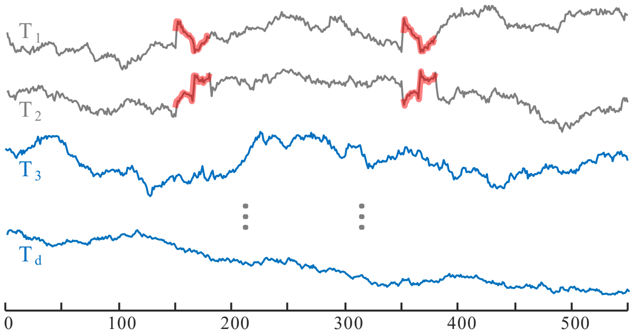

Fig. 2. A running example of a multidimensional time series. Both of the first two dimensions have a motif of length 30 embedded at locations 150 and 350. All remaining time series (just two are shown above) are simply random walks.

If we run the classic *single dimensional* motif discovery [26] on either of the first two dimensions, we correctly find the visually obvious motifs at locations 150 and 350. If we generalize the motif definition to *Multidimensional Time Series data* (MTS), and consider the best motif in the two dimensions $\{T_1,T_2\}$, then unsurprisingly, we still find the same best motif location. However, what will happen as we add in some random walks to the *multidimensional* dataset we consider? With just one random walk added to create a three-dimensional times series, we can still robustly find the correct motif locations; the signal of the true subset $\{T_1,T_2\}$ is strong enough to resist the irrelevant information added by a single random walk. However, empirically averaging over 100 trials, we have found that if there are eight additional irrelevant dimensions, then we do about as well as random chance. Moreover, the above motifs make up about 5% of the data. However, motifs are often much rarer, which accelerates the rate at which increasing dimensionality masks motifs that exist in a subspace of the data.

This illustrates a problem that is ubiquitous in medicine, science, and industry. The analyst suspects that there are motifs in some subset of the time series, but does not know *which* dimensions are involved, or even *how many* dimensions are involved. Doing motif search on *all* dimensions is almost guaranteed to produce meaningless results, even if a subset of dimensions has clear and unambiguous motifs.

Informally, we would like any multidimensional motif framework to be able to support all the following types of queries. Given a large *k*-dimensional time series:

- **Guided Search**: Find the best motif on *k* dimensions, where the integer *k* is given by the user, but which *k* dimensions to use is unspecified.
- **Constrained Search**: Find the best motif on *k* dimensions, but explicitly include (or exclude) a given subset of dimensions.
- **Unconstrained Search**: Find the best motif on *k* dimensions, where *k* is not given by the user but is the "natural" subset of the data that has motifs.

The first two tasks mostly reduce to questions of speed and scalability; the last task is subtler, requiring us rank different tentative solutions and return the most natural one.

The need for such tools is based on our collaborations with domain experts. For example, in the oil and gas industry, a single distillation column typically has well over a hundred time series (*Tags*, in the parlance of the industry) monitoring various aspects of the system. However, motifs typically appear in just a handful of dimensions. As a concrete example, consider a known motif known to appear on distillation columns in Texas. Between April and September, Texas often has brief thunderstorms with large amounts of rain falling within short periods of time. This falling rain cools the distillation column, reducing the pressure inside, and invokes a change in flowrate, or some other part of the system that attempts to compensate for the reduced pressure. Thus the "rainstorm" motif may *only* show up on the {`temp`, `pressure`, `flowrate`} tags.

Before leaving this example, it is worth noting that the important dimensions for the motif depend on the user-specified motif length. In such datasets, a motif query of one hour may turn up the thunderstorm example, but a motif query of length one day may find the motif representing a monthly calibration/cleaning run, which affects many more dimensions.

II. Related Work

There is a large and growing body of work on single time series motif discovery [12][26][27]; however, there is much less work on the multidimensional case [1][18][22][24].

The work of Minnen et al. [18] is the closest in spirit to our work. Their work was the first to note the detrimental impact of irrelevant dimensions on multidimensional motif search, and they introduced a method that is shown to be somewhat robust for a small number of *smooth*, but irrelevant dimensions, or *just* one noisy irrelevant dimension. However, the algorithm introduced is *approximate*. Even in an ideal case, with just six dimensions, they report "*with no noise, (our approach) achieves roughly 80% accuracy.*" We want to consider much higher dimensionalities, with a much greater fraction of irrelevant dimensions, and we are unwilling to compromise accuracy. The work was notable at the time for being much faster than a brute-force search, but since the advent of the Matrix Profile, that advantage has narrowed or disappeared [27][28].

Tanaka et al. propose to perform multidimensional motif discovery by "*transforming multi-dimensional time-series data into 1-dimensional time-series data*" [22]. The idea is attractive for its simplicity, but it requires all (or at least *most*) of the dimensions to be relevant, as the algorithm is brittle to even a handful of irrelevant dimensions. Moreover, both the speed and accuracy of Tanaka's algorithm depend on careful tuning of five parameters.

In a series of papers, Vahdatpour and colleagues introduce an MTS motif discovery tool and apply it to a variety of medical monitoring applications [24]. Their approach is based on computing time series motifs for each individual dimension and using clustering to "stitch" together various dimensions. However, even when the motifs are quite obvious, the problems are small and simple, and at most three irrelevant dimensions are considered, they never achieved greater than 85% accuracy on the three domains in which they were tested. To be sure, this is much better than the 17% they achieve with the strawman of only considering a single dimension. But given that seven parameters need to be tuned to achieve this result, accuracy is likely to be further compromised in more challenging data sets.

It is worth restating that the multidimensional motif discovery algorithms in which we are aware have the weakness of being *approximate*. For example, [1][18][22] and [24] all achieve scalability by searching over a reduced time resolution/reduced cardinality symbolic approximation of the original data, and [4] achieves scalability by searching over a piecewise linear approximation of the data. While it is known that such methods *can* produce high precision results in the univariate case, with carefully chosen parameters, on relatively smooth data, it is less clear how well they work in the more general case. In contrast to these approaches, our *m*STAMP algorithm is *exact*; thus, we can ignore such considerations.

To summarize, all current multidimensional motif discovery algorithms in the literature are slow, approximate, and brittle to irrelevant dimensions. In contrast, we desire an algorithm that is fast, exact, and robust to hundreds of irrelevant dimensions.

A. Dismissing Apparent Solutions

Before continuing, we will take the time to dismiss some *apparent* solutions to our problem.

It may appear that we could use the correlation (or some other measure of mutual dependence) between the times series to guide our search for subsets of dimensions likely to yield *k*-dimensional motifs. However, this is not the case. Recall {T_1, T_2} from Fig. 2. Their correlation is effectively zero (-0.0052). However, if we create 10 random walks of the same length, then on average, we expect that about 22 of the 45 pairwise combinations will have a higher correlation. We are interested in repeated *local* patterns; statistics about *global* tendencies are unlikely to be informative.

III. Definitions and Notation

We begin by defining the data type of interest, *time series*:

Definition 1: A *time series* $T \in \mathbb{R}^n$ is a sequence of real-valued numbers $t_i \in \mathbb{R} : T = [t_1, t_2, \dots, t_n]$ where n is the length of T.

For motif discovery, we are not interested in the global properties of a time series, but in the local *subsequences*:

Definition 2: A *subsequence* $T_{i,m} \in \mathbb{R}^m$ of a T is a continuous subset of the values from T of length m starting from position i. Formally, $T_{i,m} = [t_i, t_{i+1}, \dots, t_{i+m-1}]$.

The particular local properties that we seek to capture are *time series motifs*:

Definition 3: A *time series motif* is the most similar subsequence pair of a time series. Formally, $T_{a,m}$ and $T_{b,m}$is the motif pair iff $dist(T_{a,m}, T_{b,m}) \leq dist(T_{i,m}, T_{j,m})\ \forall\ i, j \in [1, 2, \dots, n-m+1]$, where $a \neq b$ and $i \neq j$, and $dist$ is a function that computes the z-normalized Euclidean distance between the input subsequences.

We store the distance between a subsequence of a time series with all the other subsequences from the same time series in an ordered array called *distance profile*.

Definition 4: A *distance profile* $D \in \mathbb{R}^{n-m+1}$ of a time series T and a subsequence $T_{i,m}$ is a vector that stores $dist(T_{i,m}, T_{j,m}) \forall\ j \in [1, 2, \dots, n-m+1]$.

The distance profile can be computed efficiently by using a convolution-based method such as MASS [17].

The most efficient method of locating time series motifs exactly, is to compute the *matrix profile* [27][28].

Definition 5: A *matrix profile* $P \in \mathbb{R}^{n-m+1}$ of a time series T is a meta time series that stores the z-normalized Euclidean distance between each subsequence and its nearest neighbor, where n is the length of T, and m is the given subsequence length. The time series motif can be found by locating the two lowest values in P (they will have tying values). Note that other definitions of motifs (range motifs, top-*K* motifs etc.) can also be extracted trivially from the matrix profile [27][28].

The time complexity to compute P is $O(n^2)$ [28]. This may seem unscalable, but the following facts mitigate this. The time complexity does not depend on the *length* of the motifs[1]. In contrast, [1][18][22][24], all scale poorly for longer motif lengths. Moreover, the matrix profile can be computed with a variety of algorithms/computational frameworks, including STAMP [27], STAMP*I* [27], STOMP [28], and GPU-STOMP [28], which can exploit both the available computational resources and domain constraints for optimal performance. Even without resorting to high-performance hardware, our algorithm is at least two orders of magnitude faster than [1][18][22][24]. Fig. 3 shows the matrix profile of T_1.

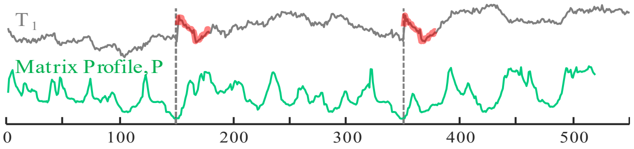

Fig. 3. Matrix profile of T_1. The two lowest points on P correspond to the locations of embedded motif pair (red).

Although the motif pair (red) is visually similar to the background random walk (black), the matrix profile still reveals the locations of the motif pair by strongly minimizing at the appropriate locations.

In addition to the special case of a single dimensional time series, our algorithm generalizes and extends [26][28] to find motifs in *multidimensional time series*.

Definition 6: A *multidimensional time series* $\boldsymbol{T} \in \mathbb{R}^{d \times n}$ is a set of co-evolving time series $T^{(i)} \in \mathbb{R}^n$: $\boldsymbol{T} = [T^{(1)}, T^{(2)}, \dots, T^{(d)}]^T$ where d is the dimensionality of $\boldsymbol{T}$ and n is the length of $\boldsymbol{T}$.

Similarly, the definition of a subsequence in multidimensional setting becomes the following:

Definition 7: A *multidimensional subsequence* $\boldsymbol{T_{i,m}} \in \mathbb{R}^{d \times m}$of a multidimensional time series $\boldsymbol{T}$ is a set of univariant subsequences from $\boldsymbol{T}$ of length m starting from position i. Formally, $\boldsymbol{T_{i,m}} = [T_{i,m}^{(1)}, T_{i,m}^{(2)}, \dots, T_{i,m}^{(d)}]^T$.

As demonstrated in Section I, using all dimensions for motif discovery is generally guaranteed to fail (A similar observation, but for time series *classification*, is forcefully made in [9]). In general, only a subset of all dimensions should be used for multidimensional motif discovery.

We refer to such subsets of subsequences *subdimensional subsequences*.

Definition 8: A *subdimensional subsequence* $\boldsymbol{T_{i,m}}(X) \in \mathbb{R}^{k \times m}$ is a multidimensional subsequence for which only a subset of dimension is selected, where X is an indicator vector that shows which dimension is included, and k is the number of dimension included (i.e., $\|X\|_0 = k$).

We want to compute the distance between two multidimensional subsequences by using only their corresponding subdimensional subsequences. The distance function that measures this relation is called *k-dimensional distance*.

Definition 9: The *k-dimensional distance* function or $dist^{(k)}$ computes the distance between two multidimensional subsequences by using only the "best" k out of d dimensions. Formally, $dist^{(k)}(\boldsymbol{T_{i,m}}, \boldsymbol{T_{j,m}}) \coloneqq \min_X dist\left(\boldsymbol{T_{i,m}}(X), \boldsymbol{T_{j,m}}(X)\right)$, where $\|X\|_0 = k$.

The definition of a distance profile is changed slightly for the multidimensional setting, and it is thus renamed the *k-dimensional distance profile*.

Definition 10: A *k-dimensional distance profile* $D \in \mathbb{R}^{n-m+1}$ of a time series $\boldsymbol{T}$ and a subsequence $\boldsymbol{T_{i,m}}$ is a vector that stores $dist^{(k)}(\boldsymbol{T_{i,m}}, \boldsymbol{T_{j,m}}) \forall\ j \in [1, 2, \dots, n-m+1]$.

Multidimensional motifs must also be redefined slightly to allow for representing within subdimensional setting.

Definition 11: A *k-dimensional motif* is the most similar subdimensional subsequence pair of a multidimensional time series when the distance is computed by using the *k*-dimensional distance function. Formally, $\boldsymbol{T_{a,m}}$ and $\boldsymbol{T_{b,m}}$ is the *k*-dimensional

[1] The word "dimensionality" is overloaded for MTS. It is used both to refer to the number of time series and to the number of data points in a subsequence. For clarity, we only use it in the former sense.

motif pair iff $dist^{(k)}(\boldsymbol{T_{a,m}}, \boldsymbol{T_{b,m}}) \leq dist^{(k)}(\boldsymbol{T_{i,m}}, \boldsymbol{T_{j,m}}) \forall\, i, j \in [1, 2, \ldots, n-m+1]$, where $a \neq b$ and $i \neq j$.

To find the k-dimensional motif, we modify the matrix profile for the k-dimensional motif problem.

Definition 12: A *k-dimensional matrix profile* $P \in \mathbb{R}^{n-m+1}$ of a multidimensional time series $\boldsymbol{T}$ is a meta time series that stores the z-normalized Euclidean distance between each subsequence and its nearest neighbor (the distance is computed using k-dimensional distance function), where n is the length of $\boldsymbol{T}$, d is the dimensionality of $\boldsymbol{T}$, k is the given number of dimension, and m is the given subsequence length. Formally, the i th position in P stores $dist^{(k)}(\boldsymbol{T_{i,m}}, \boldsymbol{T_{j,m}}) \forall\, j \in [1, 2, \ldots, n-m+1]$, where $i \neq j$. The k-dimensional motif can be found by locating the two lowest values in P (these two lowest values must be a tie [27]).

Fig. 4 shows the k-dimensional matrix profile of the running example for all possible settings of k.

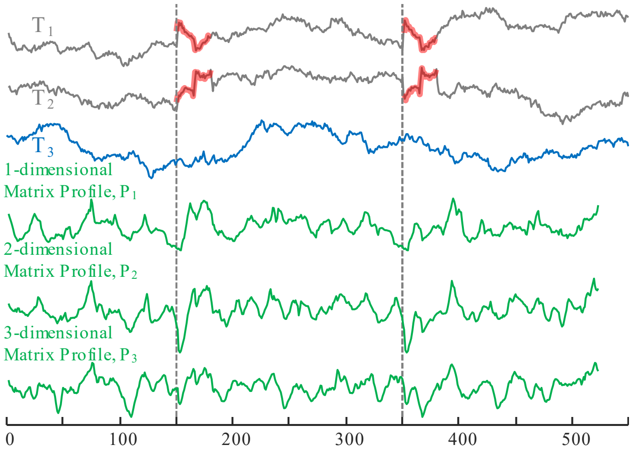

Fig. 4. *top*) The multidimensional time series shown in Fig. 2. *Bottom*) The multidimensional matrix profiles of various subsets of the data. Note that the (implanted) semantically meaningful motif can be spotted visually by inspecting the lowest points of the 1-dimensional or 2-dimensional matrix profile, but the 3-dimensional case has the motifs in an effectively *random* location.

Note, the correct motif pair only appears in P_1 and P_2 (as the lowest point in the curve), since the inserted motif is 1-dimensional and 2-dimensional motif by definition.

A k-dimensional matrix profile only reveals the location of motifs in time, but it fails to reveal which k out of the d dimension contains the motif pair. To store this information, we define another meta time series called the *k-dimensional matrix profile subspace*.

Definition 13: A *k-dimensional matrix profile subspace* $\boldsymbol{S} \in \mathbb{R}^{k \times n-m+1}$ is a multidimensional meta time series that stores the selected k dimension for each subsequence when computing the distance with others.

With these definitions formalized, we are ready to introduce our algorithms. Before continuing, we wish to clarify our claimed contributions. Our algorithm *is* orders of magnitude faster than existing works [1][18][22][24]; however, this is simply a property we inherit from the use of the matrix profile [27], which is *not* a claimed original contribution. Our contribution is in producing semantically meaningfully multidimensional motifs on a subset of a large MTS, which may comprise *mostly* of irrelevant and spurious data.

IV. The Multidimensional Motif Discovery Framework

The mSTAMP-based motif discovery framework is inspired by the idea of the multidimensional matrix profile. Similar to the original matrix profile [26][28], the multidimensional matrix profile can be computed by multiple algorithms and can be adopted in various time series data miming tasks with appropriate modification and/or postprocessing. The specific algorithm, modification, and postprocessing described in this section is just one realization for using multidimensional matrix profile in motif discovery.

A. The mSTAMP Algorithm

Our definitions allow a naïve solution. We could compute the matrix profile (the multidimensional variant using all dimensions [16]) to all d choose k combinations of dimensions and choose the best one under some ranking function. However, this naïve solution is only computable for trivially small datasets due to the combinatorial explosion inherent in this approach.

Fortunately, the combinatorial search space can be searched efficiently and admissibly in a greedy fashion. Our algorithm can compute the k-dimensional matrix profile for every possible setting of k (i.e., 1 to d) simultaneously in $O(d \log d\, n^2)$ time and $O(dn)$ space. The algorithm is outlined in ALGORITHM I. To simplify the presentation, we omit the operations related to storing of the k-dimensional matrix profile subspace. Before explaining the algorithm, we note that the source code of mSTAMP in both MATLAB and Python is publicly available in [19] and that the correctness of the algorithm is formally demonstrated in Section IV.B.

ALGORITHM I. THE *m*STAMP ALGORITHM

```
Procedure mSTAMP($\boldsymbol{T}$, $m$)
Input: Inputted time series $\boldsymbol{T} \in \mathbb{R}^{d \times n}$, interested subsequence length $m \in \mathbb{Z}$
Output: A set of $k$-dimensional matrix profile $\boldsymbol{P} \in \mathbb{R}^{d \times n-m+1}$
1  $\boldsymbol{P} \leftarrow$ size $d \times n-m+1$ inf matrix
2  $idxes \leftarrow$ integers from 1 to $n-m+1$
3  for each $idx$ in $idxes$   // random order if anytime algorithm used
4      $\boldsymbol{D} \leftarrow$ size $d \times n-m+1$ zero matrix
5      for $i$ from 1 to $d$
6          $Q \leftarrow \boldsymbol{T}[i, idx\colon idx+m-1]$
7          $\boldsymbol{D}[i,:] \leftarrow$ distanceProfile($Q$, $\boldsymbol{T}[i,:]$)
8      end for
9
10     $\boldsymbol{D} \leftarrow$ columnWiseAscendingSort($\boldsymbol{D}$)
11     $D' \leftarrow$ length $n-m+1$ zero array
12     for $i$ from 1 to $d$
13         $D' \leftarrow D' + \boldsymbol{D}[i,:]$
14         $D'' \leftarrow D' \div i$
15         $\boldsymbol{P}[i,:] \leftarrow$ elementWiseMin($\boldsymbol{P}[i,:]$, $D''$)
16     end for
17 end for
18 return $\boldsymbol{P}$
```

In line 1, the memory for the k-dimensional matrix profile for each setting of k is allocated and initialized as an array filled with *infinity*. For each iteration in the main loop (line 3 to line

17), we select one subsequence from $\boldsymbol{T}$ as the query for further processing. The subsequences are selected in a random order if the anytime-algorithm property is desired [28]. From line 5 to line 8, the dimension-wise distance profile using the query and $\boldsymbol{T}$ is computed and stored in matrix $\boldsymbol{D}$. If the query is selected in a random order, MASS [17] is used for the distance profile computation; otherwise, the method proposed by Zhu et al. [28] is used for distance profile computation. This is because that method (with time complexity of $O(n)$) is faster than MASS (with time complexity of $O(n \log n)$) but requires the subsequences to be selected in order (line 3), which nullifies the anytime-algorithm property. Next, in line 10, a column-wise sort in ascending order is applied to the matrix $\boldsymbol{D}$. Finally, from line 12 to line 15, each k-dimensional matrix profile is updated with the corresponding k-dimensional distance profile (i.e., D'') if the corresponding element in D'' is smaller.

B. Demonstration of Correctness

The basic strategy of *m*STAMP is simple. In each iteration of the main loop (line 3 to line 17 in ALGORITHM I.), the algorithm computes the k-dimensional distance profile for a given subsequence under every possible setting of k (from 1 to d). Therefore, it is sufficient to justify the algorithm's overall correctness by demonstrating the correctness of the computed k-dimensional distance profile.

Given a multidimensional subsequence $\boldsymbol{T_{i,m}}$ and its parent time series $\boldsymbol{T}$, the algorithm first computes the distance profiles for each dimension independently and stores them in matrix $\boldsymbol{D}$ (line 4 to line 8 in ALGORITHM I.). In other words, the (l, j) position of $\boldsymbol{D}$ stores the distance between $T_{i,m}^{(l)}$ and $T_{j,m}^{(l)}$. Note that each row of $\boldsymbol{D}$ is the dimension-wise distance profile (**Definition 4**) instead of the k-dimensional distance profile (**Definition 9**). Naïvely, the k-dimensional distance profile can be produced by solving $\min_X \|\boldsymbol{D}[j, X]\|_{\boldsymbol{0}} \; \forall \, j \in [1, 2, \ldots, n - m + 1]$ for each setting of k, where X is an indicator vector that shows which dimensions are included ($\|X\|_0 = k$.) However, computing the k-dimensional distance by enumerating all possible combination would be extremely inefficient.

Because the z-normalized Euclidean distance is non-negative, every number in $\boldsymbol{D}$ is non-negative. By taking this fact into account, the 1-dimensional distance profile is the smallest value in each column of $\boldsymbol{D}$, the 2-dimensional distance profile is the two smallest values in each column of $\boldsymbol{D}$, and the rest can be solved trivially after $\boldsymbol{D}$ is sorted column-wise. As a result, applying column-wise sort (line 10 in ALGORITHM I.) and column-wise cumulative sum (line 13 in ALGORITHM I.) to $\boldsymbol{D}$ can produce the k-dimensional distance profile. Therefore, the algorithm ultimately computes the correct k-dimensional matrix profile.

C. The Expressiveness of our Model

With the correctness of the algorithm demonstrated, now we are ready to discuss the expressiveness of the discovered multidimensional motifs. It may seem counterintuitive, but as demonstrated in Fig. 5, the lower dimensional motif may or may not be a subset of the higher dimensional motif, since the lower dimensional motif pair could be closer than any subset of dimensions in the higher dimensional motif pair.

For clarity, here the best 3-dimensional motif pair is the patterns occurring at times '3' and '4' of all three time series, but the best 2-dimensional motif pair is the patterns occurring at times '1' and '2' of just B and C.

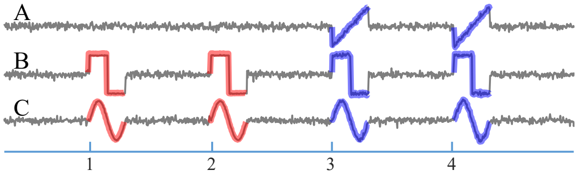

Fig. 5. When the 2-dimensional motif and 3-dimensional motif are extracted using the multidimensional matrix profile, the 2-dimensional motif may or may not be a subset of the 3-dimensional motif. In this example, the motif with lower dimensionality is not a subset of the higher dimensional motif.

This property is unfortunate, since it excludes the possibility to use various pruning and dynamic programing techniques to speed up the computation. However, as we will see, it is this expressiveness that allows the discovery of semantically *meaningful* motifs in high-dimensional data.

D. Constrained Search

There are two types of constraints that are useful in multidimensional motif searches: *exclusion* and *inclusion*. The exclusion constraint "blacklists" a predetermined set of dimensions from the search; therefore, no motif can span the excluded dimensions. Conversely, the inclusion constraint "whitelists" a predetermined set of dimensions, and all motifs must span the included dimensions. The implementation of exclusion is simple; we simply remove the blacklisted dimension before calling *m*STAMP. The implementation of inclusion is slightly more complicated, as we must move the distance computed by using whitelisted dimensions up to the front after a column wise-ascending sort has been applied (see line 10 in ALGORITHM I.).

These constraints are similar to the "*must-link*" and "*cannot-link*" operators in constrained clustering [25]. They allow the user to give domain specific "hints" to the algorithm. We developed this tool in collaboration with Dr. John Criley (UCLA School of Medicine), who gave us the following example. The reader may not understand the intricacies of the following examples, but our main point is that domain experts *will* appreciate the ability to do constrained search.

Dr. John Criley noted that a cardiologist searching a heavily telemetered archive of sleep studies for evidence of predictors of *Pulsus Paradoxus* might need to insist on the inclusion *RESPIRATION*, but be agnostic as to which other time series could be a part of a motif [8]. In contrast, a neurosurgeon searching the same dataset may wish to exclude explicitly *one* of the two *ELECTROOCULOGRAM* (EOG) time series (eye movement). Because the two eyes typically move in tandem, they are redundant, and the pairing of {EOG_{left}, $\text{EOG}_{\text{right}}$} will tend to report a strong, but spurious 2-dimenional motif.

We envision that domain experts in other areas will be interested in experimenting with similar domain-based constraints, based on their experience and knowledge.

E. Unconstrained Search

It is possible that a user knows, even if only approximately, the "expected" dimensionality of patterns in her domain. For example, suppose the user wishes to find repeated saxophone elements in a musical performance that is represented in twelve-dimensional Mel-frequency cepstral coefficient (MFCCs) space. The user can be sure that the motif will span about three dimensions, but *which* three depends on whether the instrument is a soprano, alto, tenor, baritone, or bass saxophone [11].

However, it is also possible that a user exploring a dataset has little idea about the plausible dimensionality of the repeated structure in their time series; therefore, it is necessary to support an *unconstrained* search for multidimensional motif search.

To be clear, by *unconstrained* search, we mean that mSTAMP searches the full d dimension space and returns the multidimensional motif on k dimensions, with $1 \leq k \leq d$, and typically $k \ll d$; where k is not a user input, but it is chosen by an algorithm as the "natural" dimensionality of a repeated structure in the data. Because the mSTAMP algorithm searches for motifs in all possible subsets of dimensions of a given multidimensional time series, the problem of an *unconstrained* search reduces to selecting the best motif of all possible k-dimensional motifs.

Before describing our selection method for choosing the "natural" motif dimensionality in a dataset, we note that since all k multidimensional motifs are found by the time the selection method is invoked by the user. If the user is not satisfied by the output of the selection method, finding it to be too conservative, or too liberal, the user can "nudge" the solution to examine the other possibilities without any significant (re)computational effort.

Our selection method is inspired by the elbow (or knee) finding method [23], which is commonly used for model selection, for example choosing between alterative clusterings. We visually or algorithmically locate the inflection point when we plot the "score" for each k-dimensional motif. By adopting an elbow-finding framework, we further reduce the problem to which statistics about the motifs can be used as the score. We claim that the matrix profile value for each k-dimensional motif is a convenient and suitable score for this purpose.

Let us revisit the toy example shown in Fig. 2, with the number of random walk time series set to four in addition to the two random walks that have an embedded motif. We note in passing that even this simple and small example is not trivial for humans to process. In [19], we remove the color clue that helps in Fig. 2 and shuffled the order of the time series. We invite the reader to see how difficult it is to find the correct answer by visual inspection.

We locate all k-dimensional motifs by using mSTAMP and plot their corresponding matrix profile values in Fig. 6. The matrix profile value for 3-dimensional motif is noticeably greater than the 2-dimensional motif's matrix profile value; therefore, the figure has suggested that the natural dimensionality is 2, coinciding with the ground truth dimensionality of the embedded motif.

Beyond the visual inspection used above, there are multiple suggestions in the literature on how to automatically locate the turning point in an elbow plot [15]. We use the Minimum Description Length (MDL) principle [14] to determine the most preferable k. In essence, the MDL principle states that the model, that allows the observed data to be compressed the most, is likely to be the true model. In other words, the MDL principle has cast the elbow-finding problem into a maximum compression (or minimum model size) finding problem.

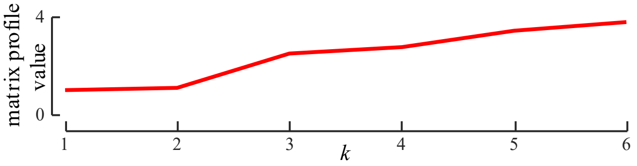

Fig. 6. The matrix profile value for each k-dimensional motif. Notice how the value dramatically increases when k is greater than 2 (the natural dimensionality of the embedded motif).

The compression (encoding) technique we consider is similar to the difference-encoding scheme used in [27]. We encode a given time series T by storing the difference between T and the reference time series T_r. For example, given two discrete time series T and T_r (with 4-bit integers):

$T =$	1	2	0	12	4	5	2	1	10	15
$T_r =$	1	2	0	11	4	5	1	0	10	15

This would take 80 bits to store, as there are 20 4-bit integers. We can compute the difference $\Delta = T - T_r$:

$\Delta =$	0	0	0	1	0	0	1	1	0	0

Since Δ only contains 0s and 1s, we can use 10 1-bit integers to store Δ, and compression can be achieved by storing the same information indirectly with T_r and Δ (which requires 50 bits to store) instead of storing T and T_r directly.

The MDL principle can be applied trivially in this problem. We compute the number of bits required to store each of the k-dimensional motifs by compressing the subsequence pair that spans the motif subspace suggested by the k-dimensional matrix profile. Fig. 7 shows the bit information of the same k-dimensional motifs (the motifs used to plot Fig. 6), and the embedded motif (i.e., 2-dimensional motif) can be identified by looking for the minimum point in the bit information curve.

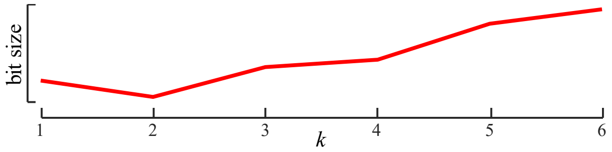

Fig. 7. The required bit value for storing each k-dimensional motif. Notice the 2-dimensional motif required the minimal bit to store.

In the case where multiple semantically meaningful k-dimensional motifs are presented in the multidimensional time series (e.g., Fig. 5), we can just interactively apply the MDL-based method to discover the motif. There are two steps in each iteration: 1) apply the MDL-based method to find the k-dimensional motif with the minimum bit size and 2) remove the found k-dimensional motif by replacing the matrix profile values of the found motif (and its trivial match) to infinity. If we apply the two steps above to the time series shown in Fig. 5, the 3-dimensional motif would be discovered in the first iteration, and the 2-dimensional motif would be discovered in the second iteration. In terms of the terminal condition for the iterative method, it can be either be an input for the user or a more advanced technique could be applied. Due to space limitations, we will have to leave the discussion on termination condition to future works. An example of applying such iterative algorithm on real-world physical activity monitoring time series is shown in Section V.E.

V. Experimental Evaluation

We begin by stating our experimental philosophy. We have designed all experiments such that they can be easily reproduced. To this end, we have built a webpage [19] that contains all datasets and code used in this work with some supporting videos. Our experiments are designed to show the following:

- In real-world domains, the naïve approach of using all the dimensions to find motifs rarely produces useful information. In contrast, our approach allows the user to find an appropriate subset of the dimension to reveal latent structure in the data.
- Our approach is versatile and works with real-world data from various domains without modification.
- Our approach is scalable enough to allow us to tackle real-world problems.

The last point requires careful qualification. The scalability of our approach is something we *inherit* from our use of the Matrix Profile, which as we noted earlier, can be computed efficiently with a variety of computational paradigms, including STAMP [27], STOMP [28], and their GPU versions [28]. Thus, we do not take credit for the scalability of our algorithm; other than to note that we carefully designed it to exploit the Matrix Profile's noted celerity.

All experiments are performed on a server with Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz, and the algorithm is implemented with MATLAB. However, we also have a Python version of our algorithm freely available in [19].

A. Synthetic Data

The *m*STAMP algorithm can be built on top of either the STAMP or STOMP algorithm; therefore, it inherits all the positive characteristics from its parent algorithm, including:

- the runtime does not depend on data's properties (noise, stationarity, periodicity etc.), only it length n.
- the runtime does not depend on the *subsequence* length, m.
- the algorithm is easy to parallelize.
- the algorithm can be cast as an *anytime algorithm* [26].

To empirically confirm the aforementioned characteristics, we have performed a scalability test.

We begin by testing scalability of the *m*STAMP algorithm with a randomly generated 4-dimensional time series of length 2^{14} with multiple subsequence lengths. The resulting runtimes are shown in Fig. 8. Unsurprisingly, the change of subsequence length does not impact the runtime, concurring with the claims of both STAMP [26] and STOMP [28].

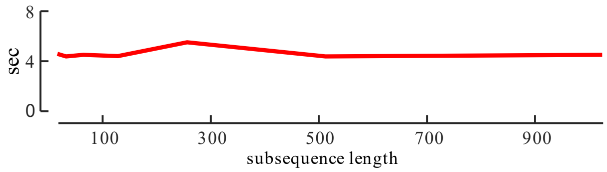

Fig. 8. The runtime does not vary significantly as we change the subsequence length.

Before moving on, it is worth reminding ourselves how remarkable and unexpected this property is. We can perform motif search with complete freedom from the curse of dimensionality (unlike everywhere else in this paper, here the term *dimensionality* is used to denote subsequence length) that plagues all other approaches [1][4][12][22][24].

Next, we fix the subsequence length to 256 and test the *m*STAMP on a 4-dimensional time series of increasing lengths. As shown in Fig. 9, the runtime grows quadratically with time series length, which coincides with the claimed time complexity of the parent algorithm, STOMP [28].

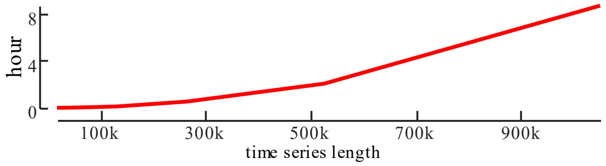

Fig. 9. The runtime increases quadratically with the length of the time series.

Before further mitigating this time complexity, it is worth noting that it may already be fast enough for most applications. For example, an oil distillation column may have four dimensions, say {TEMP, PRESS, FLOW-RATE, REFLUX-RATE} and be sampled once a minute. Fig. 9 indicates that it will take about two hours of CPU time to find motifs in a full year of historical data (525,600 data points). This is almost certainly acceptable in this domain; given the potential cost savings an actionable motif could lead to.

Nevertheless, we can offer the user a further significant speed-up by processing the data in an anytime fashion. Like one of its parent algorithm STAMP [26], *m*STAMP can be trivially modified to be an effective anytime algorithm. Fig. 10 shows the convergence rate of *m*STAMP on a 3-dimensional time series with 2-dimensional motifs embedded. The Root Mean Squared Error (RMSE) decreases quickly in the first few percent of iterations. After only 10 percent of the computations have been completed, the current "best-so-far" matrix profile is not only visually similar to the exact matrix profile (the inset images in Fig. 10), but the RMSE is also very low. This property is useful for interactive data exploration as the user can terminate the algorithm early when satisfied by the discovered motifs using the current approximate matrix profile [26].

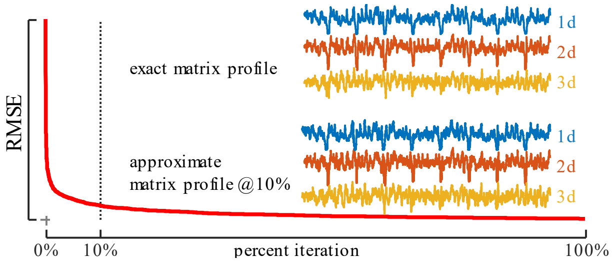

Fig. 10. Like its parent STAMP, *m*STAMP converges quickly. The approximated multidimensional matrix profile achieves a low Root Mean Square Error (RMSE) when just 10% of the iteration are completed. The inset images are the multidimensional matrix profiles.

Because the input time is multidimensional, we need to test the scalability of *m*STAMP when we vary the dimensionality of the input time series. Here, we fixed the time series length to 2^{14} and subsequence length to 256. The runtime shown in Fig. 11 confirms our claim in Section IV.A as the runtime has a linearithmic relationship with the time series dimensionality.

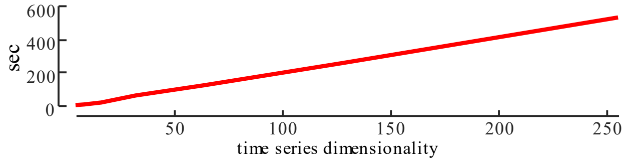

Fig. 11. The runtime increases linearithmically with the dimensionality of time series.

In addition to the runtime and anytime property, we also consider that the *accuracy* of unconstrained motif search is also tested. Note that the *m*STAMP algorithm does compute the multidimensional matrix profile exactly. However, the unconstrained motif search could still fail to find the semantically correct motifs. For example, this could happen if the motifs are subtle and the large number of irrelevant dimensions happens to produce a spuriously similar pair of subsequences. Thus, here we test the MDL-based heuristic's ability to find an embedded 4-dimensional motif among a set of multidimensional random walks. Fig. 12 shows the average accuracy as we increase the number of irrelevant dimensions for both the MDL-based method and the original matrix profile by using *all* dimensions. Note the latter is an upper bound for the performance of all known rival methods [22] that use all dimensions, since they are using *all* dimensions, *and* are approximate.

The MDL-based algorithm almost always finds the correct embedded motif, while the *all* dimensions algorithm failed in most cases. Even if we increase the number of irrelevant dimensions to 64 times the number of relevant dimensions, the accuracy is still near perfect.

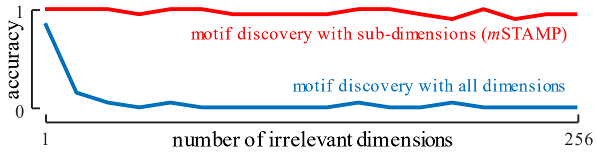

Fig. 12. The accuracy of the MDL-based unconstrained motif search algorithm as we vary the number of irrelevant dimensions while keeping the number of relevant dimension (i.e., 4) fixed. The results are averaged over forty trials. The method is robust against irrelevant dimensions.

Becase the multidimensional matrix profile is already computed exactly for the MDL-based algorithm, a manual inspection of the matrix profile value curve (see Fig. 6) could also be performed as a safeguard measure.

B. Motion Capture Case Study

The creation of *motion graphs* is a fundamental problem in computer animation/gaming [10]. The task is as follows: Given a large corpus of motion capture data, automatically construct a directed graph called a *motion graph* that encapsulates connections among the database. This allows a finite repertoire of motions to be synthesized into an infinite set of plausible motions, which can be "steerable" to some goal, or adaptive to changing inputs [2][10] (This video [3], which accompanies [2], offers a more visual and intuitive explanation of motion graphs).

We demonstrate how *m*STAMP can help the user create higher quality motion graphs by discovering subdimensional motifs, rather than being forced to consider all dimensions. We applied the *m*STAMP algorithm to subject 13's motion capture recording (where the subject performs various boxing moves for 40 seconds) from the CMU Motion Capture Database [6]. The recording consists of a multidimensional time series with 38 dimensions, each corresponding to the motion of a given joint.

First, we visually examined the video snippet corresponding to the motif pair discovered by using *all* 38 dimensions. We found that the subject is performing an uppercut punch in one of the snippets, but the other snippet consists of blocking/dodging motion. In retrospect, the results shown in Fig. 12 make this result unsurprising. This finding offers support for our claim that sometimes an algorithm needs to ignore a significant fraction of the dimensions to discover semantically meaningful motifs in multidimensional time series.

Next, we examine the video snippet corresponding to the 3-dimensional motif discovered by *m*STAMP. Here, the motif pair discovered consists of the subject performing a cross and a one-two combo. Our algorithm matches a simple cross with the cross in a one-two combo, and the three matching dimensions are from joints in the right humerus (right upper arm), right radius (right forearm), and left femur (left upper lag). The motif discovered within the subspace is much more meaningful comparing to the motif discovering using all dimension, and allows the construction of a seamless motion graph after blending all other limbs [10]. We have plotted the motions as a sequence of stick figures in Fig. 13. Note how the right arm of the subject is in a different position in latter frames within different occurrences of the motif. We invite the interested reader to refer to the supporting website for the motif pairs shown in the form of video [19].

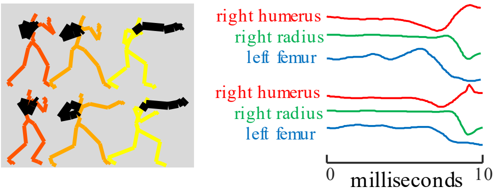

Fig. 13. (see also Fig. 1) *top)* The subject is throwing a cross. *bottom)* The subject is throwing a one-two combo (jab cross combo). The right arm is highlighted with black. Our algorithm is capable of discovering the cross in the one-two combo, because it explores the subspace rather than *all* dimensions.

C. *Music Processing Case Study*

The original matrix profile has been shown to be useful for music information retrieval (MIR) [16]. To demonstrate the potential utility of our enhanced multidimensional variant of matrix profile for MIR, we have performed a simple motif discovery experiment on the Mel-spectrogram of the song *Never gonna give you up* by Rick Astley. The Mel-spectrogram is extracted with the following parameters, which are commonly used in MIR: 46 milliseconds short time Fourier transform (STFT) window, 23 milliseconds STFT hop, and 32 Mel-scale triangular filters. When we apply the matrix profile to music by using all dimensions with a five-second subsequence length, it is unsurprising that the motif we discovered is the chorus of the song [16]. The discovered motif is shown in Fig. 14. Note how the extracted pairs match each other in all dimensions.

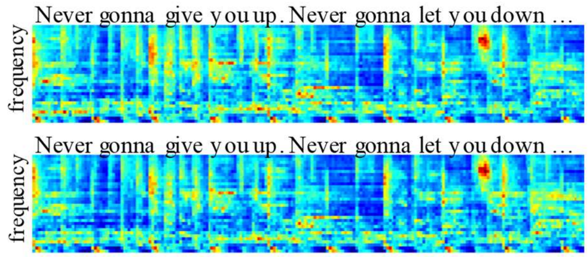

Fig. 14. When the motif is found using all dimensions, the chorus is discovered. This visualization of the data is compact and intuitive, but note that our algorithm is still operating on the raw time series signals.

Next, we applied *m*STAMP to discover motifs in subspaces, ranging from 1 dimension to 32 dimensions. We discovered that while most of the high dimensional motifs are parts of the chorus, both the 1-dimensional and 2-dimensional motif pair only represents the drum pattern. When we examine the exact subspace to which these lower dimensional motifs span, the motif pairs are in the space spanned by the two lowest frequency bands (i.e., typical frequency range for percussion), which confirms our intuition. Fig. 15 shows the 2-dimensional motif pair. Note how the lowest two frequency bands are matched, while the other frequency bands differ significantly.

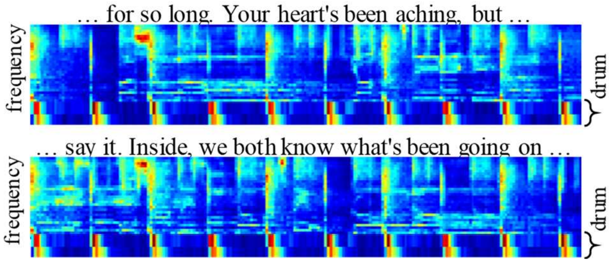

Fig. 15. When the motif is found by using only the two *best* dimensions, the repeated drum pattern in lowest two frequency bands is dissevered. The lowest two frequency bands are enlarged for better visibility.

This example showcases one of the advantages of our method: Once the multidimensional matrix profile is computed, users can explore the matrix profile for different dimensionalities without additional computational cost. In other words, the users can quickly explore the motifs mined from each matrix profile and decide the correct number of dimensions for the users' specific task at hand; whether it is audio thumbnailing (as in Fig. 14) or generating infinite playlists (Fig. 15) [5].

D. *Electrical Load Measurement Case Study*

To illustrate the unconstrained search functionality of our motif search method, we tested our method on an electrical load measurement dataset [21]. The dataset consists of electrical load measurements for individual appliances (and an *aggregated* load) from households in United Kingdom. Five appliances are considered: fridge-freezer, freezer, tumble dryer, dishwasher, and washing machine. The data was collected from April 19, 2014 to May 15, 2014, where the length is 17,000. The subsequence length was set to 4 hours.

As shown in Section IV.E, we can determine the natural dimensionality of a given multidimensional time series' motif by examining the matrix profile values of the *k*-dimensional motifs. Fig. 16 shows the matrix profile values for the motifs found in the electrical load measurement time series. According to the figure, it is likely that the natural dimensionality of the multidimensionality motif is 2. To confirm that this suggested dimensionality is semantically meaningful, we have examined the dimensions spanned by the 2-dimensional motif. The relevant dimensions are the electrical load measurements of tumble dryer and washing machine. Since both machines are typically used one after another in a short window of time, it is not surprising that the discovered 2-dimensional motif spanned the use of these related appliances.

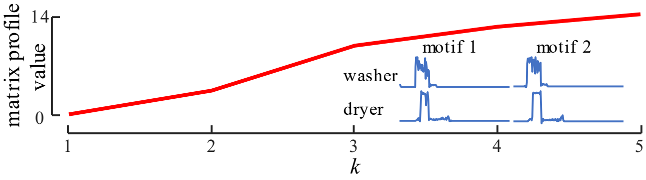

Fig. 16. The natural dimensionality of the multidimensional motif is 2 as suggested by this figure. The discovered motifs (inset) correspond to the electrical load from using a washer, followed by dryer.

E. *Physical Activity Monitoring Case Study*

Multiple types of human motion (e.g., walking, running, and rope jumping) can occur within a single recording session of physical activity, and the problem of extracting meaningful patterns is often formulated as a motif discovery problem [1][18][22]. As noted in Section IV.E, the MDL-based motif discovery algorithm can be applied multiple times to the precomputed multidimensional matrix profile to iteratively discover all top-*K* motifs (see [20] for definition of top-*K* motif). To showcase the effectiveness of the MDL-based dimension selection algorithm on such tasks, we consider the first subject (i.e., subject 101) of PAMAP2 dataset [13].

The dataset consists of multidimensional time series capturing both a heart rate monitor and three inertial measurement units (IMUs). The three IMUs are placed on the subject's wrist, chest, and ankle; each measures the temperature, 3-d acceleration data, 3-d gyroscope, and 3-d magnetometer while the subject is performing various physical activities. The activities performed by subject 101 during the recording are: lying, sitting, standing, walking, running, cycling, Nordic walking, ascending stairs, descending stairs, vacuum cleaning, ironing, and rope jumping.

Within the list of activities, the first three activities (i.e., lying, sitting, and standing) are more about the subject's passive posture rather than his or her action. As there are little or no repeated motion when the subject is not moving, the motif pairs that exist within these temporal regions should be less similar (and less meaningful) compared to the motif pairs occur during more dynamic activities. In other words, if our MDL-based method retrieves the motifs based on the similarity (i.e., from high similarity to low similarity), then we would expect the motifs from more dynamic events to rank higher than the more passive events. Fig. 17 shows the extracted motif pairs' class (i.e., dynamic versus passive) ordered based on the order in which they were retrieved, and the result largely coincides with our speculation.

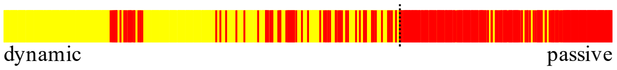

Fig. 17. The MDL-based algorithm prioritizes more active and meaningful motifs. If we stop the retrieving process at the dashed line, the F-measure for the retrieval would be 0.88.

To give a more quantitative evaluation on the motif retrieval result, we have computed the F-measure for each iteration (the MDL-based algorithm retrieves one item per iteration). The optimal stopping iteration is marked with dashed line in Fig. 17, and the corresponding F-measure is 0.88. Although the result F-measure is impressive given such simple MDL-based method, we cannot know the optimal stopping iteration without consulting the ground truth label. The F-measure provided here is for gauging the potential of the *m*STAMP-based motif discovery framework.

VI. Conclusion

We have shown that if the time series motif discovery is blindly applied to the multidimensional case, the results are likely to be unsatisfactory. To address this, we have introduced a *m*STAMP-based multidimensional motif discovery frame work, that solves this problem by returning the motifs that exist in natural subspaces of the higher dimensional data. The returned motifs are actionable, and they suggest at non-obvious latent structures in the data. We built our system on top of the recently introduced Matrix Profile and inherit all of its desirable properties, including anytime and incremental compatibility, low memory footprint, and scalability to large datasets [28].

Acknowledgment

We gratefully acknowledge funding from Oracle and NSF 1510741, and all data donors.

References

[1] Balasubramanian, A., Wang, J., and Prabhakaran, B. 2016. Discovering Multidimensional Motifs in Physiological Signals for Personalized Healthcare. *IEEE Journal of Selected Topics in Signal Processing*. vol. 10, no. 5, pp. 832-841.

[2] Beaudoin, P., van de Panne, M., Poulin, P., and Coros, S. 2008. Motion-Motif Graphs. *Symposium on Computer Animation* 2008.

[3] Beaudoin, P., van de Panne, M., Poulin, P., and Coros, S. 2008. Motion-Motif Graphs Video. www.youtube.com/watch?v=eiE1NFGzpzA.

[4] Berlin, E. and Laerhoven, and K. V. 2012. Detecting leisure activities with dense motif discovery. *UbiComp*. 250-259.

[5] Bohra, T., Kumar, V and Ganesan, S. 2015. Segmenting music library for generation of playlist using machine learning. *EIT* 2015: 421-425.

[6] CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/

[7] Exadaktylos, V., Silva, M., Ferrari, S., Guarino, M., Taylor, C. J., Aerts, J.-M., and Berckmans, D. 2008. Time-series analysis for online recognition and localization of sick pig (Sus scrofa) cough sounds. *Journal of the Acoustical Society of America*. Vol. 124, No. 6.

[8] Guntheroth, W., Morgan, B., Mullins, G. 1967. Effect of respiration on venous return and stroke volume in cardiac tamponade. Mechanism of pulsus parodoxus. *Circ. Res.* 20 (4): 381–90.

[9] Hu, B., Chen, Y., Zakaria, J., Ulanova, L., and Keogh, E. J. 2013. Classification of Multi-dimensional Streaming Time Series by Weighting Each Classifier's Track Record. *ICDM 2013*: 281-290.

[10] Kovar L., Gleicher, M., and Pighin. F. Motion Graphs. 2012. *ACM Transactions on Graphics*, 21, 3 (ACM SIGGRAPH '02).

[11] Loughran, R., Walker, J., O'Neill, M. and O'Farrell, M. 2008. The Use of Mel-frequency Cepstral Coefficients in Musical Instrument Identification. *International Computer Music Conference* (ICMC), 2008.

[12] Patel, P., Keogh, E. J., Lin, J., and Lonardi, S. 2002. Mining Motifs in Massive Time Series Databases. *ICDM 2002*. 370-377.

[13] Reiss, A. and Stricker, D. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. *The 16th IEEE International Symposium on Wearable Computers* (ISWC 2012).

[14] Rissanen, J. 1978. Modeling by shortest data description, *Automatica*, Volume 14, Issue 5, 1978, Pages 465-471.

[15] Salvador, S. and Chan, P. 2004. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. *International conference on tools with artificial intelligence*. pp 576–584.

[16] Silva, D. F., Yeh, C.-C. M., Batista, G. E. A. P. A., and Keogh, E. 2016. SiMPle: Assessing Music Similarity Using Subsequences. *ISMIR 2016*.

[17] MASS. http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html

[18] Minnen, D., Isbell, C. L., Essa, I. A., and Starner, T. Detecting Subdimensional Motifs: An Efficient Algorithm for Generalized Multivariate Pattern Discovery. *ICDM 2007*: 601-606.

[19] *m*STAMP project site. https://sites.google.com/view/mstamp/

[20] Mueen, A., Keogh, E., Zhu, Q., Cash, S., and Westover, B. 2009. Exact discovery of time series motif, *SDM* 2009.

[21] Murray, D. et. al. 2015. A data management platform for personalised real-time energy feedback. *In Proce of the 8th International Conference on Energy Efficiency in Domestic Appliances and Lighting*, 2015.

[22] Tanaka et al. 2005. Discovery of time-series motif from multi-dimensional data based on MDL principle. *Machine Learning*. 58 (2–3), pp. 269–300.

[23] Thorndike, R. L. 1953. Who belongs in the family? *Psychometrika*.

[24] Vahdatpour, A., Amini, N., and Sarrafzadeh, M. 2009. Toward Unsupervised Activity Discovery Using Multi-Dimensional Motif Detection in Time Series. *IJCAI*. 1261-1266.

[25] Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. 2001. Constrained K-means Clustering with Background Knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning*. pp. 577–584.

[26] Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., Silva, D. F., Mueen, A., and Keogh, E. 2016. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. *IEEE ICDM 2016*.

[27] Yeh, C.-C. M., Van Herle, H., and Keogh, E. 2016. Matrix Profile III: The Matrix Profile Allows Visualization of Salient Subsequences in Massive Time Series. *IEEE ICDM 2016*.

[28] Zhu, Y., Zimmerman, Z., Senobari, N., S., Yeh, C.-C. M., Funning, G., Mueen, A., Brisk, P., and Keogh, E. 2016. Matrix Profile II: Exploiting a Novel Algorithm and GPUs to break the one Hundred Million Barrier for Time Series Motifs and Joins. *IEEE ICDM 2016*.