

Elastic Time Series Motifs and Discords

Diego F. Silva

Departamento de Computação

Universidade Federal de São Carlos, São Carlos, Brazil

diegofs@ufscar.br

Gustavo E. A. P. A. Batista

Instituto de Ciências Matemáticas e de Computação

Universidade de São Paulo, São Carlos, Brazil

gbatista@icmc.usp.br

Abstract—The recent proposal of the Matrix Profile (MP) has brought the attention of the time series community to the usefulness and versatility of the similarity joins. This primitive has numerous applications including the discovery of time series motifs and discords. However, the original MP algorithm has two prominent limitations: the algorithm only works for Euclidean distance (ED) and it is sensitive to the subsequences length. In this work, we extend the MP algorithm to overcome both limitations. We use a recently proposed variant of Dynamic Time Warping (DTW), the Prefix and Suffix Invariant DTW (ψ -DTW) distance. The ψ -DTW allows invariance to warp and spurious endpoints caused by segmenting subsequences and has a side-effect of supporting the match of subsequences with different lengths. Besides, we propose a suite of simple methods to speed up the MP calculation, making it more than one order of magnitude faster than a straightforward implementation and providing an anytime feature. We show that using ψ -DTW avoids false positives and false dismissals commonly observed by applying ED, improving the time series motifs and discords discovery in several application domains.

Keywords—Time Series, Motifs, Discords, Similarity, Dynamic Time Warping.

I. INTRODUCTION

The similarity join for time series is a data mining procedure that retrieves the nearest neighbors of all possible subsequences with a given number of observations. Recently, Yeh et al. [1] proposed to organize the information obtained by the similarity join in a data structure known as Matrix Profile (MP). More importantly, the MP uses a fast Euclidean distance (ED) calculation algorithm to speed up the computation of the similarity join [2].

However, for a series of relevant applications domains, such as motion tracking, music processing, and sports monitoring, warping is an intrinsically valuable feature because of the presence of nonlinear time accelerations. Our first objective is to show, with a series of experiments, the similarity join with Euclidean distance fails to retrieve the most significant motifs and discords in several application domains that typically require warping. Besides, we use a variant of the Dynamic Time Warping (DTW) distance to find more meaningful matches and propose a suite of simple methods to speed up the MP computation under this distance, making it more than one order of magnitude faster than a straightforward implementation.

We designed our solution to work with streaming data. Therefore, we do not assume the data is carefully pre-segmented as it happens with data from repositories such as UCR [3]. Instead, we extract the subsequences using a sliding window, which is a standard procedure. As a consequence, we might have problems with spurious endpoints. These endpoints

occur when the subsequences are not perfectly segmented, and degrade the quality of the motifs and discords discovered.

To circumvent this issue, we use the Prefix and Suffix Invariant DTW (ψ -DTW) distance [4]. The ψ -DTW allows matching two subsequences under DTW ignoring up to r endpoints, being r a user-defined parameter. Although the user still needs to specify the length of the sliding window, ψ -DTW has a side-effect of supporting the match of subsequences of different lengths up to $2r$ observations.

Our algorithm is exact, i.e., the values provided in the MP are the actual ψ -DTW distances. However, as we assume streaming data with a potentially vast number of observations, we designed our solution as an anytime algorithm. We show that we can obtain a solution close to the optimal in half of the time taken by the state-of-the-art ED-based algorithm.

In summary, in this paper, we propose building the Matrix Profile with ψ -DTW distance to find “elastic” motifs and discords, i.e., with warped patterns and possibly with different lengths. The advantages of our method are:

- It extends, for the first time, MP to a distance measure based on the non-linear alignment of observations;
- It finds warped patterns and avoids false positives and negatives caused by endpoints;
- It is more than one order of magnitude faster than a straightforward DTW implementation;
- It is exact if ran to completion or gives a good approximate solution if interrupted.

II. DEFINITIONS AND BACKGROUND

Before introducing our method, we will introduce some definitions and briefly present the necessary background. We begin by the definitions of time series, subsequence, nearest neighbor, and trivial match.

Definition 1: A time series T is an ordered set of values $t = (t_1, t_2, \dots, t_N)$ such that $t_i \in \mathbb{R}$ are referred as *observations*. The number of observations N is referred as its *length*.

Definition 2: A subsequence $t_{q,m}$ is a continuous subset of T of length m starting from the observation q , i.e., $t_{q,m} = (t_q, t_{q+1}, \dots, t_{q+m-1}) \mid q \in [1, N - m + 1]$.

Definition 3: The nearest neighbor (NN) of $t_{q,m}$ is the subsequence $t_{p,m}$ such that $t_{p,m}$ is the subsequence with the smallest distance to $t_{q,m}$. Formally, $NN(t_{q,m}) = t_{p,m} \Leftrightarrow dist(t_{q,m}, t_{p,m}) = \min(dist(t_{q,m}, t_{k,m})) \mid k \in [1, N - m + 1]$ excluding trivial matches.

Definition 4: Given two subsequences $t_{q,m}$ and $t_{p,m}$, they constitute a *trivial match* if $|p - q| \leq \xi$, $\xi \in \mathbb{N}$ and $\xi < m$.

We continue by defining the patterns found by our method, starting by the time series discord [5] and the best pair motif.

Definition 5: The *discord* $t_{d,m}$ is the subsequence with the largest distance to its nearest neighbor. Formally, $\text{dist}(t_{d,m}, \text{NN}(t_{d,m})) = \max(\text{dist}(t_{q,m}, \text{NN}(t_{q,m})))$, $q \in \{1 \dots N - m - 1\}$.

Definition 6: The *best pair motif* is the pair of subsequences $(t_{q,m}, t_{p,m})$ which returns the smallest distance of any pair of subsequences with no trivial matches. Formally, $\text{dist}(t_{q,m}, t_{p,m}) = \min(\text{dist}(t_{k,m}, t_{l,m}))$, $l, k \in 1 \dots N - m - 1$ and $|l - k| > \xi$.

For the cases in which the user requires more than one motif, we define the K -th best pair motif.

Definition 7: The K -th best pair motif is the pair $(t_{q,m}, t_{p,m})$ with the smallest distance in the dataset with no overlap with any I -th motif pair such that $I < K$.

There are (at least) two definitions of time series motifs in the literature [6]. For simplicity, we focus on the best pair motifs. However, our method is easily extended to find k -motifs, which we intend to discuss in a future work.

Most of the presented definitions are directly or indirectly related to a distance measure between subsequences. The most common distance measure to search for motifs and discord is the Euclidean distance (ED) [7]. However, the ED is very sensitive to small distortions in the time axis [8], usually referred as *warping*. Warping commonly occurs in time series data, for instance, in motion tracking from subjects with different paces and music data with tempo differences. These applications require a more flexible matching of observations, referred as *elastic* alignment, in which the k -th observation of $t_{q,m}$ can be associated with the l -th observation of $t_{p,m}$, such as $l \cong k$. The Dynamic Time Warping (DTW) is arguably the most relevant and widely used distance with this property.

The alignment of DTW has to respect a few constraints. In particular, the endpoints constraint requires the matching of all observations, starting at the pair $(t_{q,p})$ and finishing at (t_{q+m-1}, t_{p+m-1}) . However, Silva, Batista, and Keogh [4] noticed a negative impact of this restriction to the time series classification, given that the segmentation of time series usually leaves data from previous and/or next events in the segmentation window.

The Prefix and Suffix Invariant DTW (ψ -DTW) was proposed to circumvent this problem. This distance measure is a subtle modification DTW's endpoints constraints, allowing the algorithm to skip a matching of some observations, limited to the maximum number of points that can be skipped, referred as *relaxing factor*.

The ψ -DTW is calculated using a dynamic programming algorithm, which recurrence relation is defined by Equation 1.

$$dtw(i, j) = c(i, j) + \min \begin{cases} dtw(i-1, j) \\ dtw(i, j-1) \\ dtw(i-1, j-1) \end{cases} \quad (1)$$

where $i, j = 1 \dots m$ and $c(i, j)$ is the cost of matching the i -th and j -th observation of the subsequences, calculated with the squared difference between the observations. Equation 2 defines the initial condition for the algorithm.

$$dtw(i, j) = \begin{cases} 0, & (i = 0 \wedge j \leq r) \vee (j = 0 \wedge i \leq r) \\ \infty, & \text{otherwise} \end{cases} \quad (2)$$

where $r \in \mathbb{N}$ is the relaxing factor. Finally, Equations 3 and 4 define the ψ -DTW between two subsequences.

$$\psi\text{-DTW}(t_{q,m}, t_{p,m}) = \min_{i,j \in \text{finalSet}} dtw(i, j) \quad (3)$$

$$\text{finalSet} = \{(m - c, m)\} \cup \{(m, m - c)\} \forall c \in [0, r] \quad (4)$$

In addition to reducing the impact of prefixes and suffixes, ψ -DTW allows the matching of patterns with different lengths when assessing fixed-length subsequences. Specifically, the largest difference occurs when m observations from one subsequence matches $m - 2r$ observations from the other.

Figure 1 illustrates the matching of observations obtained by ED and ψ -DTW. Note that ED erroneously matches the slightly displaced valley and the spurious prefix. On the other hand, the non-linear alignment obtained by ψ -DTW is robust to the warped valley and disregards several of the first observations of one subsequence, ignoring the spurious prefix.

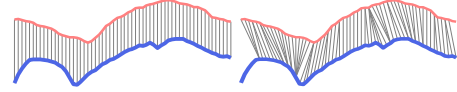


Fig. 1. Alignment obtained by matching two subsequences by the ED (left) and the ψ -DTW (right)

Finally, a commonly applied technique within the DTW algorithm is the warping window [9]. In a practical standpoint, it limits the alignment distance in the time axis by adding the constraint $|i - j| \leq w$ to the recurrence relation (Equation 1), in which the parameter w is referred as warping window length. This constraint makes the DTW calculation faster and more accurate in several application domains [10].

III. PROPOSED METHOD

The main motivation for our method is the recent proposal of the matrix profile (MP) [1]. The MP is a representation of the subsequences similarities in a time series, comprised in a vector which stores the ED from each subsequence to its nearest neighbor. A linear search in this simple structure finds several different patterns in the time series. Specifically, the smallest and the highest values in the MP points to a motif and to the discord, respectively.

The MP is accompanied by a vector of pointers to the nearest neighbor of each subsequence, referred as MP index. For clarity, consider that $t_{p,m}$ is the NN of the subsequence $t_{q,m}$. While the MP in the position q stores the value obtained by $\text{ED}(t_{q,m}, t_{p,m})$, the MP index stores p .

In this work, we develop a method to find motifs and discords that may suffer from warping and may have slightly different lengths, which we call *elastic motifs* (ELMO) and

elastic discords (ELD). For this purpose, we construct a ψ -DTW-based MP, which we refer as *elastic distance matrix profile* (EMP). We emphasize that this is the first work to construct the MP using any elastic distance measure.

In a practical standpoint, the construction of the EMP is a series of similarity searches. To speed up this procedure, we adapt techniques to accelerate the DTW-based search and propose two techniques that are specific for the EMP.

In this work, we speed up the motif and discord discovery based on the techniques used by the UCR Suite [11], which comprises cascading pruning and early abandoning algorithms to avoid the costly calculation of DTW. The main modification to adapt the suite for ψ -DTW is the use of suitable lower bound (LB) functions (c.f. [4]). Besides, we propose two techniques that can further accelerate the search for motifs and discords.

The effectiveness of pruning and early abandoning depends on one major factor: the value of the best-so-far distance (*bsf*), i.e., the distance to the nearest neighbor of a query in a given step of the search. Specifically, quickly decreasing the *bsf* value induces a faster ELMO and ELD discovery.

The fact that ψ -DTW is a symmetric distance measure may be useful in this situation. Consider we are currently searching for the NN of $t_{q,m}$. For every subsequence $t_{p,m}$ not pruned or early abandoned, we verify the p -th position of the EMP. If this value is greater than ψ -DTW($t_{q,m}, t_{p,m}$), we update the EMP and EMP-index in this position.

Using this simple technique, we provide a reasonable initial guess of a *bsf* to the subsequence $t_{p,m}$. When searching for its NN, having this value since the beginning improves the pruning power of the LB and early abandoning techniques.

Given that a good initial guess of the *bsf* tends to improve the runtime of the similarity search, we propose a heuristic order for assessing the subsequences. While a straightforward implementation would follow the temporal order of the subsequences, we use the EMP information to choose the next subsequence as the one with best initial *bsf*. Specifically, while searching for the NN of the current subsequence, we store the position of the smallest value in the EMP which was not assessed yet. Then, in the next iteration, our method will skip to such a subsequence.

The advantages of this method are twofold: it reduces the runtime for the calculation of the EMP and it improves the efficiency of the anytime motifs discovery (c.f. Section V-C).

IV. ON USING ψ -DTW INSTEAD OF DTW

A question that may arise at this point is “*why should one choose a DTW variation instead of the traditional algorithm?*” The answer to this question is based on the fact that DTW is a special case of the ψ -DTW. Specifically, both algorithms calculate the same distance when the endpoints cause no impact in the alignment. Otherwise, ψ -DTW is able “fix” the DTW by ignoring these spurious endpoints.

As a practical consequence, DTW may discard several relevant motifs by considering them distant, because their endpoints are disproportionately contributing to the distance. For instance, consider the subsequences in Figure 2. This pair is one of the most relevant motif pairs found by ψ -DTW in

the athlete positioning dataset (c.f. V-B1). However, the DTW consider these subsequences a distant pair in comparison to other matches.

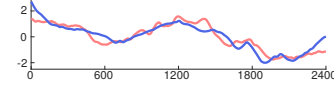


Fig. 2. A pair of motifs found by ψ -DTW considered distant by DTW

In opposite, the DTW may consider observations of similar events as discords. In general, this fact represents cases in which the endpoints should not be part of the target pattern, leading the DTW to false positives regarding discords. Figure 3 shows a relevant discord found by DTW and its nearest neighbor according ψ -DTW in the athlete positioning dataset. Note that, except by a clear prefix and few discrepant observations at the end, both subsequences describe a similar behavior, so the red subsequence should not be considered a discord.

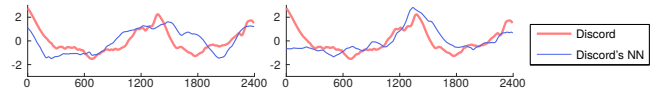


Fig. 3. A discord found by DTW (red) and its nearest neighbor (blue) according the traditional DTW (left) and ψ -DTW (right)

V. EXPERIMENTAL EVALUATION

In this section, we evaluate the scalability of our method and the quality of its solutions. For reproducibility, we developed a website¹, containing source codes and datasets.

It is important to notice that evaluating the significance of time series motifs and discords is a difficult matter. Since discovering these patterns is an unsupervised task, it is impossible to use evaluation measures which depend on a ground truth. For this reason, we provide subjective ways to evaluate the presented motifs and discords.

A. Scalability

In order to evaluate the scalability of our method, we measured the runtime² of the EMP calculation for random walk data varying the length of the time series (N) and the length of the subsequence (m). We set the warping window length (w) and the relaxing factor (r) as 5% of N , based on the evidence that small windows are a good choice for most cases of nearest neighbor search [10].

We compared the runtime of our method against an implementation of DTW-based MP without the previously described optimizations and STOMP [12], the fastest algorithm for ED-based MP calculation.

The brute force DTW and the STOMP respectively constitute a baseline and a topline for our method. Except for $\mathcal{O}(N \log N)$ calculations for the first subsequence, STOMP only requires $\mathcal{O}(N)$ operations for each subsequence. Then, its time complexity is $\mathcal{O}(N^2)$. On the other hand, the time complexity for the brute force DTW is $\mathcal{O}(N^2mw)$, given by

¹<https://sites.google.com/view/elastic-motifs-discords>

²All the experiments were taken on the same computer – 40-core Intel Xeon E5-2690v2@3.00GHz with 130GB of RAM– with only operating system-related processes running in parallel.

the $\mathcal{O}(N^2)$ pairs of subsequences and the $\mathcal{O}(mw)$ complexity of the DTW distance. All the speed up methods presented in Section III are heuristics and may eventually fail on pruning DTW. For this reason, the worst case complexity of our method is the same than the naive DTW. However, our experimental evaluation shows that, in practice, the runtime of our method is much closer to the baseline than to its worst case.

Figure 4 shows the runtimes for a random walk with 100,000 points and varied subsequence length.

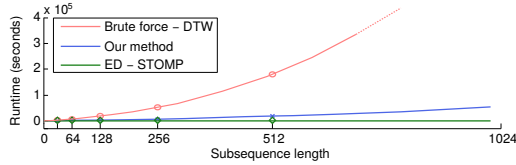


Fig. 4. Runtime for calculating the MP using a brute force DTW algorithm (red), our method (blue), and STOMP (green) for different subsequence lengths. The markers point to real runtime values and the lines were obtained by the Spline interpolation

The reason for the (approximately) constant runtime for STOMP is that its time complexity is independent on the subsequence length. The runtime of our method increases according to the subsequence length, but we can notice a moderate increasing in comparison to the brute force DTW.

The other experimented parameter, which affects all three methods, is the length of the time series. Figure 5 shows the results for the subsequence length fixed in 128 observations.

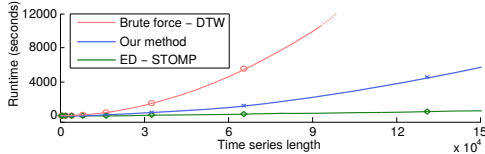


Fig. 5. Runtime for calculating the MP using a brute force DTW algorithm (red), our method (blue), and STOMP (green), varying the time series length

Although we compared our method against STOMP, a direct comparison is not a fair approach. While the ED-based method is faster, it is not suitable for datasets that suffer from warping. In contrast, ψ -DTW suppresses this issue while guarantees a good runtime in relatively long time series. For applications with long time series and in which the runtime is a critical issue, our method provides an excellent approximate result faster than STOMP (c.f. Section V-C).

B. Case studies

In order to validate the quality of the patterns discovered by our method, we experimented it on different domains and compared the results to ED-based motifs and discords. We set the warping length and the relaxing factor as 5% of and the range for trivial matches as 25% of the subsequence length.

1) *Athlete Positioning*: The first dataset used in our experiments monitors a soccer player's trajectory in the attack/defense axis during a single game [13]. We present the patterns with 2400 observations (2 minutes of the match). Figure 6 shows the first ED-motifs and ELMO. We can notice

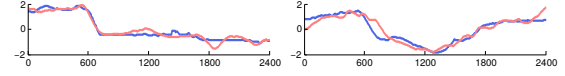


Fig. 6. First ED-based motif pair (left) and the first pair of ELMO (right) in the athlete positioning data

that even (apparently) subtle endpoints and warping distortions affect the the ED, making it disregards the ELMO as motifs.

Another interesting behavior of the ED in the presence of warped data is that it tends to favor “conservative” patterns. Subsequences composed mostly of flat lines and/or single diagonals, like the presented in Figure 6 (left), are usually considered motifs. In this study case, these patterns indicate that the player stands in the same position. Certainly, it does not represent a move in which this player performed an important role. On the other hand, complex subsequences tend to be far from any other patterns in the Euclidean space, which favors such kind of pattern to be considered a discord. This is easily seen by the discovered discord, presented in Figure 7.

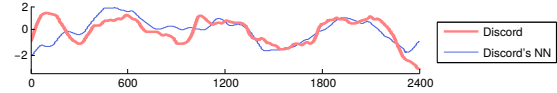


Fig. 7. The discord found by ED (red) and its nearest neighbor (blue) in the athlete positioning data

The discord found by the ED has quite similar neighbors if we apply ψ -DTW to align the observations. Figure 8 shows the first ED-discord and its nearest neighbor according to ψ -DTW. Note that the subsequences are very similarly shaped, but affected by a subtle warping and a clear suffix.

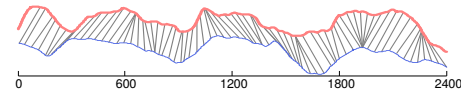


Fig. 8. The discord found by ED (red) and its nearest neighbor according to ψ -DTW (blue) in the athlete positioning data. The gray lines show a subset of aligned observations

On the other hand, the ELD represents a more reliable exceptional pattern. Figure 9 presents this ELD, which comprises the actions of turning off the sensor for the halftime and turning it on again after the interval, in a different location. This is the reason for the clear “jump” in the pattern.

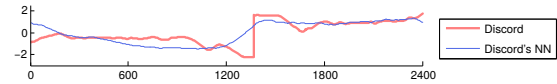


Fig. 9. ELD (red) with its respective nearest neighbor (blue) in the athlete positioning data

2) *Motion Capture*: In this case study, we used time series of tracking human motion from the HDM05 MoCap Database [14]. The session used in our work tracks the right wrist during repetitions of distinct exercises. The subsequence length is 300 observations, representing 2.5 seconds. Figure 10 presents the first motif pairs according ED and ψ -DTW.

The bias of ED for matching conservative patterns can be easily noticed again in these results. Similarly, ED considers

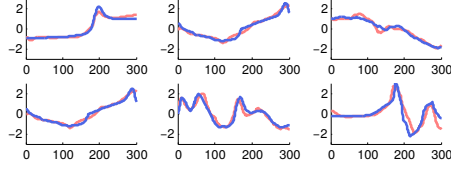


Fig. 10. Three first motifs according ED (*top*) and first pairs of ELMO (*bottom*) in the motion capture data

complex patterns as discords, which can be seen in Figure 11. While ED matches this subsequence with a simple pattern, ψ -DTW reveals another subsequence very similar to the discord.

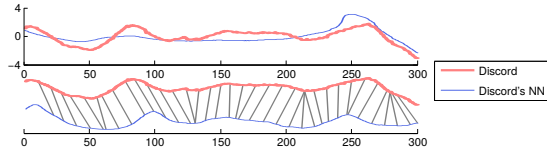


Fig. 11. Discord discovered by the ED with its nearest neighbors according ED (*top*) and ψ -DTW (*bottom*) in the motion capture data

Figure 12 shows the ELD and its NN. Note that the ELD is much simpler than the discord presented in Figure 11.

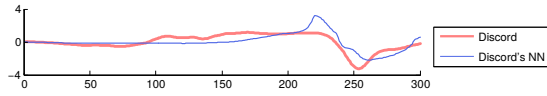


Fig. 12. ELD in the motion capture data and its nearest neighbor

To strengthen the evidence that ψ -DTW finds more relevant discords, we ran our method on the only session in the HDM05 data with annotated “strong artifacts.” While the ED does not consider the correct subsequence as discord, the ELD covers most part of the annotated anomaly.

3) *Gesture Analysis:* In this study, we used the Palm Graffiti Digits dataset [15], regarding subjects “drawing” digits in the air while facing a 3D camera. We concatenated the examples in random order to obtain a stream and searched patterns with 360 observations, which may comprises two gestures. Figure 13 presents the resulted motifs.

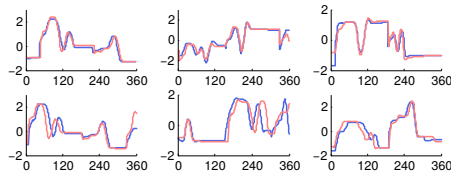


Fig. 13. Three first motifs according ED (*top*) and first pairs of ELMO (*bottom*) in the gesture data

Despite motif discovery is essentially an unsupervised task, we can use the labels for a better analysis of our motifs in this case. For both distance measures, the found patterns consist of two gestures of the same class. Therefore, in the point of view of class labels, the ED and ψ -DTW perform similar.

Figure 14 presents the discovered discord. Similarly to previous results, the discord found by the ED has a very similar pattern if we ignore spurious endpoints and a slight warping.

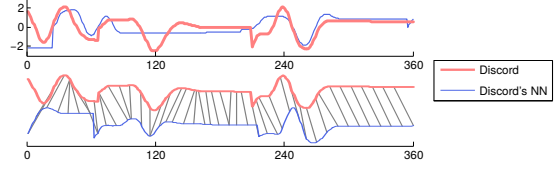


Fig. 14. Discord according the ED with its nearest neighbor (*top*) and the subsequence considered nearest neighbor according the ψ -DTW (*bottom*) in the gesture data

Figure 15 presents the discovered ELD. In both ED and ψ -DTW cases, the classes between the discovered discord and its nearest neighbor are divergent.

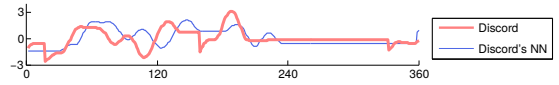


Fig. 15. ELD in the gesture data and its nearest neighbor

4) *Music Processing:* Applications of music data mining also commonly suffers from warping, mainly in live performances. In this case study, we experimented our method in a jazz concert data. The time series is composed of the tonal centroid variation, calculated by the Harmonic Change Detection Function (HCDF) [16]. The time series comprises around 1 hour and 8 minutes. The subsequences are composed of 200 data points, which represents 10 seconds.

We do not present visual comparisons in this case, since the visual analysis of HCDF is very difficult. However, we can evaluate the motifs according to the audio³ and their locations. The first three ED-based motifs have a feature in common: every pair appears in the same song, usually few seconds apart. It means that the ED is not revealing anything but the obvious. In the first motif, for instance, the HCDF represents excerpts of the guitar base of the same song, separated by improvisations by different musicians. In the second pair of motifs, the repetition happens in a sequence, i.e., just after the pattern is finished it is played again.

On the other hand, the ELMO pairs represent excerpts from different songs. It means that ψ -DTW is finding more interesting patterns, in which the tonal variations of subsequences from different tracks are similar.

Unfortunately, none of the experimented distance measures were able to find relevant/correct discords. It is easily verified by the fact that the discord found by the ED sounds very similar to its NN according ψ -DTW and vice-versa.

C. Anytime ELMO discovery

An anytime algorithm is able to be interrupted, providing a (good) partial solution to the user, at any point of its execution. The approximate solution is expected to be the best possible solution given the execution time of the algorithm.

The algorithm proposed in this work is an anytime algorithm. Given that it iteratively calculates the EMP, it may be interrupted at anytime to evaluate the current solution and

³The website for this paper contains audio files in which is possible to listen to the motifs and discords discovered in this dataset

we can continue the EMP calculation from the same point. Besides, different orders of picking the subsequences allow us to achieve different approximate solutions in the first steps. The heuristic order proposed in this work (c.f. Section III) privileges the fast ELMO discovery.

To ensure the quality of the anytime property of our method, we performed an experiment using random walk data with 100,000 points to find patterns with 128 observations. During this experiment, we assessed the EMP 10 times, interrupting the algorithm after analyzing every multiple of 10,000 subsequences.

Despite the first interruption on the algorithm was done after 10% of the time series had been assessed, the algorithm arrived such state in around 5% of the total execution time. At this interruption, the first 6 pairs of ELMO are the same than the optimal. From the next 4 pairs, the partial solution found 2 of the expected patterns. In summary, from the first 10 best motif pairs, 8 were already found after 5% of the total time.

Figure 16 presents the final and partial solutions. Notice that most of the lowest values were already found by our algorithm. The partial values that differ from the final were obtained by exploiting the symmetry, so the relative subsequences will be assessed in a future iteration. In addition, the missing values in the partial solution have no value associated by neither assessing the subsequence or using the symmetry. They are usually related to high distance values. It is important to notice that the presented results were obtained after 142 seconds of execution, while STOMP spent 283 seconds to construct the matrix profile for the Euclidean distance.

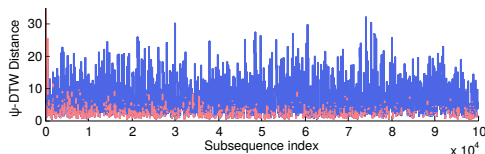


Fig. 16. Final EMP (blue) and the partial values obtained after 5% of the total execution time (red)

Although the results for the ELMO discovery are remarkably good, we could not achieve such outcome for discords. Specifically, the correct ELD is found in the ninth interruption, when 88% of the total runtime has already been spent. Therefore, the anytime characteristic of our method is only suitable for the motif discovery. We intend to study different heuristics in order to improve the anytime ELD discovery.

VI. CONCLUSION

In this paper, we proposed an approach to construct a subsequences distance matrix profile under the ψ -DTW in order to find motifs and discords on time series data. Our method allows the matching of subsequences with warping and spurious endpoints. We have shown that our method is able to find more relevant patterns on different domains. In addition, we proposed a suite of methods that is responsible for a significant speed up of the pattern discovery. Finally, our method is anytime and have shown excellent results on the motifs discovery even faster than the state-of-the-art to calculate the ED-based matrix profile.

As future work, we intend to investigate the proposed methods for different definitions of motifs and discords, as well for different tasks such as shapelet discovery and clustering. Besides, we plan to develop good heuristics for the anytime feature of our method for different purposes.

ACKNOWLEDGEMENTS

This work was funded by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) by grant #306631/2016-4 and São Paulo Research Foundation (FAPESP) by grants #2013/26151-5, #2016/04986-6, and #2017/24340-6.

REFERENCES

- [1] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets," in *ICDM*. IEEE, 2016, pp. 1317–1322.
- [2] A. Mueen, Y. Zhu, M. Yeh, K. Kamgar, K. Viswanathan, C. Gupta, and E. Keogh, "The fastest similarity search algorithm for time series subsequences under euclidean distance," August 2017, <http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html>.
- [3] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," July 2015, www.cs.ucr.edu/~eamonn/time_series_data/.
- [4] D. F. Silva, G. E. A. P. A. Batista, and E. Keogh, "Prefix and suffix invariant dynamic time warping," in *ICDM*, 2016, pp. 1209–1214.
- [5] E. Keogh, J. Lin, S.-H. Lee, and H. V. Herle, "Finding the most unusual time series subsequence: algorithms and applications," *Knowl Inf Syst*, vol. 11, no. 1, pp. 1–27, 2007.
- [6] A. Mueen, "Time series motif discovery: dimensions and applications," *Wires Data Min Knowl*, vol. 4, no. 2, pp. 152–159, 2014.
- [7] S. Torkamani and V. Lohweg, "Survey on time series motif discovery," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 2, 2017.
- [8] E. Keogh, "Exact indexing of dynamic time warping," in *VLDB*. VLDB Endowment, 2002, pp. 406–417.
- [9] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE T Acoust Speech*, vol. 26, no. 1, pp. 43–49, 1978.
- [10] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *SDM*. SIAM, 2005, pp. 506–510.
- [11] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *KDD*. ACM, 2012, pp. 262–270.
- [12] Y. Zhu, Z. Zimmerman, N. S. Senobari, C.-C. M. Yeh, G. Funning, A. Mueen, P. Brisk, and E. Keogh, "Matrix Profile II: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins," in *ICDM*. IEEE, 2016, pp. 739–748.
- [13] S. A. Pettersen, D. Johansen, H. Johansen, V. Berg-Johansen, V. R. Gaddam, A. Mortensen, R. Langseth, C. Griwodz, H. K. Stensland, and P. Halvorsen, "Soccer video and player position dataset," in *MMSys*. ACM, 2014, pp. 18–23.
- [14] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation Mocap Database HDM05," Universität Bonn, Tech. Rep. CG-2007-2, June 2007.
- [15] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE T Pattern Anal*, vol. 31, no. 9, pp. 1685–1699, 2009.
- [16] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *ACM Wksh on AMCMM*. ACM, 2006, pp. 21–26.