Matrix Profile XXIII: Contrast Profile: A Novel Time Series Primitive that Allows Real World Classification

Ryan Mercer Sara Alaee Alireza Abdoli Shailendra Singh Amy Murillo Eamonn Keogh
University of California, Riverside
{rmerc002, salae001, aabdo002, shailendra.singh, amy.murillo} @ucr.edu, eamonn@cs.ucr.edu

***Abstract*—Time series data remains a perennially important datatype considered in data mining. In the last decade there has been an increasing realization that time series data can best understood by reasoning about time series subsequences on the basis of their similarity to other subsequences: the two most familiar such time series concepts being *motifs* and *discords*. Time series motifs refer to two particularly close subsequences, whereas time series discords indicate subsequences that are far from their nearest neighbors. However, we argue that it can sometimes be useful to simultaneously reason about a subsequence's closeness to certain data and its distance to other data. In this work we introduce a novel primitive called the *Contrast Profile* that allows us to efficiently compute such a definition in a principled way. As we will show, the Contrast Profile has many downstream uses, including anomaly detection, data exploration, and preprocessing unstructured data for classification.**

Keywords-Motifs; Multiple Instance; Classification

I. Introduction

In order to perform various data mining tasks on time series, it can be fruitful to annotate each subsequence with metadata indicating various properties. One such feature is a subsequence's distance to its nearest neighbor within the same dataset. That information can be represented by the Matrix Profile [1]. Small values in the Matrix Profile are called *motifs*, and large values are called *discords*. Both motifs and discords have each been used in hundreds of research efforts . However, we argue that it may be useful to score subsequences with a new piece of meta-data that reflects the property that a subsequence is simultaneously close to its nearest neighbor in certain data but far from its nearest neighbor in other "black-listed" data. We call this property *Contrast*, and the vector that represents it the *Contrast Profil*e. While the proposed representation has many uses, for clarity, we will introduce it in the context of subsequence extraction to allow *classification*.

When using the UCR archive or similar benchmark datasets [2], the work of extracting the exemplars from a longer time series has already been done. Here, we argue that extracting the exemplars is actually the most difficult and critical task. In a handful of cases, it may be obvious where the beginning and the end of an exemplar is within a longer time series. But, in many cases, these demarcations may not be clear. Consider Fig. 1.*bottom*, which shows a time series known to have several examples of chicken dustbathing behavior [3]. Even to experts in avian biomechanics, it is not obvious where the dustbathing behavior is.

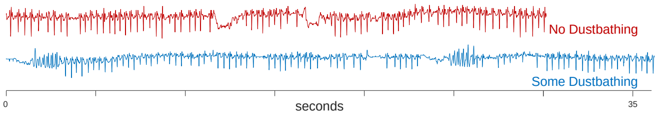

Fig. 1. Two short snippets of behavior from a chicken wearing a backpack accelerometer. The bottom time series is known to contain at least two examples of dustbathing behavior, whereas the top time series is known to be free of this behavior.

This suggests that a technique is needed to annotate each subsequence of the time series with a value that simultaneously represents how close that subsequence is to its nearest neighbor *within* the same time series and how far it is from its nearest neighbor in the time series known to be free of the target behavior. This score would reveal the location of the uniquely conserved behavior, in this case, *dustbathing*.

In Fig. 2, we give a visual intuition of the property of interest: abstracting time series subsequences to points in a high dimensional space. We explicitly consider three data points.

- Point **A** is far from its nearest neighbor in the non-target class, but it is also far from its nearest neighbor within its own target class. It is an anomaly that would score highly on the definition of time series *discord* [4].
- Point **B** in contrast is very close to its nearest neighbor in the target class, but it is also close to its nearest neighbors in non-target class. This point would score highly on the definition of time series *motif* [1].
- Point **C** *is* both very far from its nearest neighbor in the non-target class *and* very close to its nearest neighbor in the target class. This is exactly the property we desire.

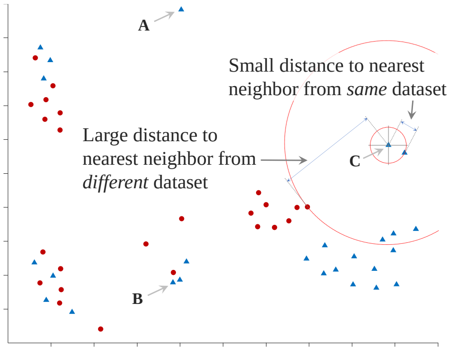
A
Small distance to nearest neighbor from *same* dataset
Large distance to nearest neighbor from *different* dataset
C
B

Fig. 2. A visual intuition of the "contrast" property. Of the three annotated points from the target class, only C is close to a member of its own class, while also being far from its nearest neighbor in the non-target class.

The rest of this paper is organized as follows. In Section II, we present the necessary definitions and notations. In Section III, we present several examples of data mining tasks that can exploit the Contrast Profile before experimentally demonstrating them in Section IV. Section V offers conclusions.

II. Definitions And Notation

Our data type of interest is *time series*.

Definition 1: A *time series* $\mathbf{T} = t_1, t_2, \ldots, t_n$ is a sequence of real-valued numbers.

Typically, we are not interested in global properties of a time series but rather shapes of small regions called *subsequences*.

Definition 2: A *subsequence* $\mathbf{T}_{i,m}$ is a contiguous subset of values from $\mathbf{T}$ starting at index i with length m.

We can measure the distance between any two time series of equal length using a distance measure. In this work, we use the ubiquitous z-normalized Euclidean distance [1]. One minor modification to the Euclidean distance is that we clip it at $\sqrt{(2 * m)}$ because values above this are anti-correlated in the Pearson Correlation space. This is done in order to make the greatest use of the normalized range when working with the Contrast Profile. If we need to measure the distance between a short time series and every subsequence from a long time series, we can produce a *distance profile.*

Definition 3: A *distance profile* $\mathbf{DP}_{i,m}^{(\mathrm{AB})}$ is the vector of distances between each subsequence in reference time series $\mathbf{T}^{(\mathrm{A})}$ and a query subsequence $\mathbf{T}_{j,m}^{(\mathrm{B})}$.

The distance can be computed very efficiently using the MASS algorithm [5]. Fig. 3 illustrates these definitions on a running example of a noisy electrocardiogram (ECG).

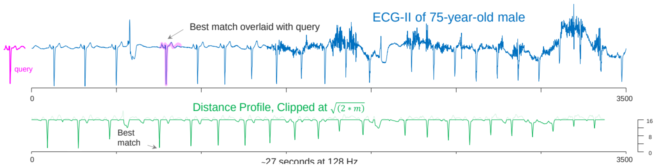

Fig. 3. *top*) A 27-second snippet of an ECG time series. *bottom*) A single heartbeat from earlier in the same dataset was used as a query to produce a distance profile, which has low values when the "sliding" query is similar to a subsequence and is minimized at the best match about five seconds in.

Our proposed ideas leverage the *self-join Matrix Profile* [1].

Definition 4: A *self-join Matrix Profile* $\mathbf{MP}_m^{(\mathrm{AA})}$ of a time series $\mathbf{T}^{(\mathrm{A})}$ is a vector of Euclidean distances between every subsequence $\mathbf{T}_{i,m}^{(\mathrm{A})}$ and its nearest neighbor $\mathbf{T}_{j,m}^{(\mathrm{A})}$. Formally, $\mathbf{MP}_m^{(\mathrm{AA})} = [\boldsymbol{min}(\mathbf{DP}_{1,m}^{(\mathrm{AA})}), \boldsymbol{min}(\mathbf{DP}_{2,m}^{(\mathrm{AA})}), \ldots, \boldsymbol{min}(\mathbf{DP}_{n-m+1,m}^{(\mathrm{AA})})]$

Fig. 4 shows $\mathbf{MP}_{128}^{(\mathrm{AA})}$ for our running example. We can see that the top motifs are a pair of normal heartbeats. Using some out-of-band data (including advice of cardiologist Dr. Greg Mason), we annotated the location of two premature ventricular contractions. While these two beats are similar, they are not as well conserved as normal beats.

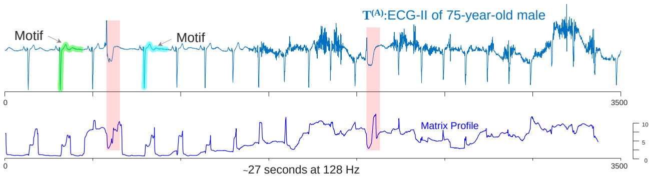

Fig. 4. *top*) The ECG shown in Fig. 3 with its $\mathbf{MP}_{128}$ (*bottom*). The lowest values of $\mathbf{MP}_{128}$ are the Top-1 motif pair, here two normal beats. Also, two PVCs shown highlighted with red bars for future reference.

In addition to subsequence comparisons *within* a time series, it can also be fruitful to make comparisons *between* two time series using the *AB-join Matrix Profile.*

Definition 5: An *AB-join Matrix Profile* $\mathbf{MP}_m^{(\mathrm{AB})}$ between reference time series $\mathbf{T}^{(\mathrm{A})}$ and a query time series $\mathbf{T}^{(\mathrm{B})}$ is a vector of Euclidean distances between each subsequence $\mathbf{T}_{i,m}^{(\mathrm{A})}$ and its nearest neighbor $\mathbf{T}_{j,m}^{(\mathrm{B})}$. Formally,

$$\mathbf{MP}_{128}^{(\mathrm{AB})} = [\boldsymbol{min}(\mathbf{DP}_{1,m}^{(\mathrm{AB})}), \boldsymbol{min}(\mathbf{DP}_{2,m}^{(\mathrm{AB})}), \ldots, \boldsymbol{min}(\mathbf{DP}_{n-m+1,m}^{(\mathrm{AB})})]$$

Note that in general, $\mathbf{MP}_m^{(\mathrm{AB})} \neq \mathbf{MP}_m^{(\mathrm{BA})}$: even with equal lengths, they correspond to different reference time series.

Fig. 5 shows $\mathbf{MP}_{128}^{(\mathrm{AB})}$ for our running example with a region of normal ECG from the same patient.

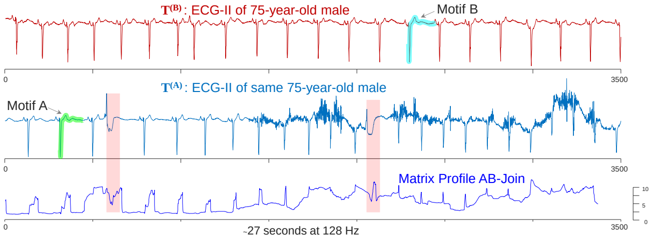

Fig. 5. *top*) Time series $\mathbf{T}^{(\mathrm{B})}$ is a normal ECG time series from the same patient. *center*) Time series $\mathbf{T}^{(\mathrm{A})}$,which contains the behavior of interest, is the original ECG introduced in Fig. 3. *bottom*) The top motif pair, where motif$^{(\mathrm{A})}$ is the unrequited nearest neighbor of motif$^{(\mathrm{B})}$. The red bars foreshadow discovery of two PVCs.

We now exploit an important observation. Note that $\mathbf{MP}_{128}^{(\mathrm{AA})}$ and $\mathbf{MP}_{128}^{(\mathrm{AB})}$ from the last two figures are very similar in most regions. This makes sense. A noisy $\mathbf{T}_{i,m}^{(\mathrm{A})}$ will tend to be just as far from any other $\mathbf{T}_{j,m}^{(\mathrm{A})}$ as it is from any $\mathbf{T}_{k,m}^{(\mathrm{B})}$ (An implication of theorem 1 of [6]). Moreover, a normal heartbeat in $\mathbf{T}^{(\mathrm{A})}$ will tend to have approximately the same low distance to another normal heartbeat, whether that beat happens to come from $\mathbf{T}^{(\mathrm{A})}$ or $\mathbf{T}^{(\mathrm{B})}$. The *only* places showing a significant difference are the locations corresponding to behaviors that are *unique* to $\mathbf{T}^{(\mathrm{A})}$: in this case, the two PVC beats.

We formalize these observations with our proposed representation, the *Contrast Profile*, specializing from the generic $\mathbf{T}^{(\mathrm{A})}$ and $\mathbf{T}^{(\mathrm{B})}$, to consider two time series $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ which have a mild assumption about their contents.

Definition 6: A *Contrast Profile* $\mathbf{CP}_m$ is the difference between Matrix Profiles $\mathbf{MP}_m^{(+-)}$ and $\mathbf{MP}_m^{(++)}$, where $\mathbf{MP}_m^{(+-)}$ joins $\mathbf{T}^{(+)}$ with $\mathbf{T}^{(-)}$, and $\mathbf{MP}_m^{(++)}$ is the self-join of $\mathbf{T}^{(+)}$.

$$\mathbf{CP}_m = (\,\mathbf{MP}_m^{(+-)} - \mathbf{MP}_m^{(++)}\,)/\sqrt{(2*m)}$$

The Contrast Profile is defined for any two time series so long as *m* is shorter than the time series' lengths. However, we proposed to compute the Contrast Profile only when we believe that the two following assumptions are likely to be true:

- $\mathbf{T}^{(+)}$ contains at least two behaviors that are unique to the phenomena of interest.
- $\mathbf{T}^{(-)}$ contains zero behaviors of interest.

Under these assumptions, large values of $\mathbf{CP}_m$ indicate behaviors that appear two or more times in $\mathbf{T}^{(+)}$ while absent from $\mathbf{T}^{(-)}$. Fig. 6 gives a visual intuition of these definitions. Note that $\mathbf{CP}_{128}$ peaks at the locations of the shape that is unique to $\mathbf{T}^{(+)}$ (i.e., the two PVC heartbeats).

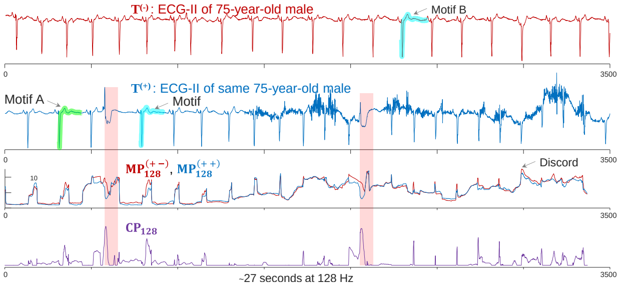

Fig. 6. *top-to-bottom*) Query time series $\mathbf{T}^{(-)}$ contains normal heartbeats. Time series $\mathbf{T}^{(+)}$ contains at least two instances of a behavior of interest. The top discord of the AB-join Matrix Profile (the highest peak), results from a noisy region in $\mathbf{T}^{(+)}$, far from the ground truth labeled with red bars. The top two candidates peak within the ground truth.

The subsequence in $\mathbf{T}^{(+)}$ corresponding to the highest point in the Contrast Profile is called the *Plato*, a backronym of *Pattern likely able to organize*, which is suggestive of a *platonic* ideal for some behavior of interest.

While we use the Matrix Profile as the core function to compute the Contrast Profile, the value optimized is rather simple. The Plato is the subsequence in $\mathbf{T}^{(+)}$ with maximum difference between its nearest neighbor distance in $\mathbf{T}^{(-)}$ and nearest neighbor distance in $\mathbf{T}^{(+)}$. This could be discovered by a classic nested-loop, brute-force algorithm, requiring O($|\mathbf{T}^{(+)}|$ ($|\mathbf{T}^{(-)}|$ + $|\mathbf{T}^{(+)}|$) *m*). As *m* could be in the thousands, this is clearly intractable. As we will later show, by exploiting the Matrix Profile, we can completely remove the dependence on *m* to produce a highly scalable algorithm.

To summarize, we have shown that at least for our running example, the Contrast Profile can be used to extract discriminating subsequences. This clearly has implications for several downstream algorithms, including classification and novelty/anomaly detection. However, before discussing these, in the next two sections we will consider the Contrast Profile's robustness to noise and the plausibility of the assumptions that warrant its use.

A. *General Contrast Profile Observations*

Note that while the two time series that are input into the Contrast Profile are denoted $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$, there is nothing pejorative about the "negative" time series. It is simply a snippet of data which we know does not have some behavior. That behavior *could* be undesirable, say a seizure, or it could be desirable, say a critical depressurization phase in an industrial process.

The Contrast Profile is bound between zero and one. A value of one corresponding to $\mathbf{T}_{i,m}^{(+)}$ means that $\mathbf{T}_{i,m}^{(+)}$ is a perfect motif in $\mathbf{MP}_m^{(++)}$ while also a maximum discord in $\mathbf{MP}_m^{(+-)}$ [1]. A value of zero means that $\mathbf{T}_{i,m}^{(+)}$ is conserved at least as much in $\mathbf{MP}_m^{(+-)}$ as $\mathbf{MP}_m^{(++)}$.

This property is critically different from that of TS-Diff [7], which is optimized solely by maximizing $\mathbf{MP}_m^{(+-)}$, a definition that simply tends to point to the noisiest subsequence.

A useful property of the Contrast Profile is that it is length invariant and sampling-rate invariant. For example, we can meaningfully compare scores for length 50 and for length 60, and state which subsequence is better conserved. This provides us with the opportunity to remove the Contrast Profile's only parameter, the subsequence length. We propose the Pan-Contrast Profile (in the spirit of [8]). We can simply compute all Contrast Profiles in some range, and choose the Plato from the one that produces the highest value. To see why we can expect this to work, consider the two extreme cases.

- If *m* is too small, then we are only comparing tiny fragments of the time series. These are very unlikely to be discriminating.
- If *m* is too large, then we are comparing the most discriminating subsequence along with extra non-discriminating shapes padded to its prefix or suffix. These non-discriminating sections can only dull the contrast property.

In Fig. 7, we show the Pan-Contrast Profile for an ECG, which example bears out our intuition above. The optimal Plato has a length of 313, which is about the length of the PVC, excluding the QRS peak, which it shares with healthy beats.

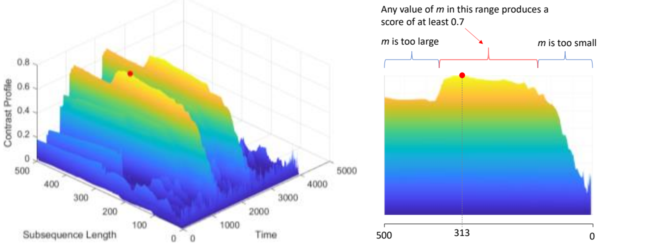

Fig. 7. *left*) The Pan-Contrast Profile for an ECG. A red dot indicates the largest value. *right*) A side view shows that the Contrast Profile is very robust to its only input parameter. Any subsequence length from 131 to 424 would have produced a score of at least 0.7.

One additional takeaway from this experiment is the relative insensitivity of the Contrast Profile definition to its only parameter. Over a huge range of values (131 to 424) it produces nearly identical values in nearly identical locations.

A computation of a single Contrast Profile requires O($|\mathbf{T}^{(+)}|^2 + |\mathbf{T}^{(+)}||\mathbf{T}^{(-)}|$) time. To concretely ground this, the example shown in Fig. 6 takes 0.182 seconds, and the full Pan-Contrast Profile shown in Fig. 7 takes 82 seconds. Note that because the Contrast Profile is based on the Matrix Profile, it inherits many of the Matrix Profile's desirable properties such as time complexity that is completely independent of the subsequence's dimensionality, and the possibility of anytime, online, and GPU-accelerated computation [9].

Thus far, we have only defined the Top-1 Plato. However, it is possible that we may be interested in the Top-K Platos, as we the behavior of interest is polymorphic. For example, unlike the simple PVC arrhythmia shown Fig. 6, some arrhythmias such as bidirectional ventricular tachycardia can present themselves with a handful of different shapes even from a single individual. If we are given $\mathbf{T}^{(+)}$ that has at least two examples of each manifestation, we would like to extract them all.

Recall that for time series discords, the Top-K discords correspond to the Top-K peaks in the Matrix Profile. However, that is not the case for the Contrast Profile. To discover the Kth Plato we must ensure that the influence of the Kth-1 Plato is first removed from the Contrast Profile. That is trivial to achieve, we simply concatenate the Kth-1 Plato to $\mathbf{T}^{(-)}$ and then recompute the Contrast Profile from scratch[1]. All subsequences in $\mathbf{T}^{(+)}$ that were similar to the Kth Plato will then be close to a subsequence in $\mathbf{T}^{(-)}$, and thus their original peaks will vanish.

B. *Online Contrast Profile*

The reader will appreciate that it may be useful to compute the Contrast Profile in an online fashion. While "online" could have several interpretations, we believe the most useful variant will be a fixed $\mathbf{T}^{(-)}$ with an incrementally updated $\mathbf{T}^{(+)}$ in the face of real-time data arrival.

Assume that we start with a computed $\mathbf{CP}_m$ of length n for $\mathbf{T}^{(+)}$, and some length for $\mathbf{T}^{(-)}$, and we wish to ingest an additional datapoint, the $n + 1$ datapoint. This will result in the creation of a new subsequence, NEW, which ends with the $n + 1$ datapoint.

What effect will subsequence NEW have on the current $\mathbf{CP}_m$, beyond lengthening it by one?

If NEW is sufficiently dissimilar to any other subsequence in $\mathbf{T}^{(+)}$, then the previous n values of $\mathbf{CP}_m$ will be unchanged regardless of NEW's distance to its nearest neighbor in $\mathbf{T}^{(-)}$.

If NEW is similar to one or more subsequences in $\mathbf{T}^{(+)}$, but also sufficiently close to its nearest neighbor in $\mathbf{T}^{(-)}$, then the previous n values of $\mathbf{CP}_m$ will again be unchanged.

If NEW is similar to one or more subsequences in $\mathbf{T}^{(+)}$, and it is far from any subsequence in $\mathbf{T}^{(-)}$, then we will have to update $\mathbf{CP}_m$ corresponding to those subsequences.

From this, we can see that the previously computed $\mathbf{CP}_m$ values can only increase or stay the same. They can never decrease. Then, adding the $n + 1$ value to $\mathbf{CP}_m$ requires computing every index in $\mathbf{DP}^{(+-)}_{NEW,m}$ and $\mathbf{DP}^{(++)}_{NEW,m}$. After outlining the algorithm that maintains the Contrast Profile *Incremental* (ContrastProfile*I*) in TABLE I. , we will explain how this process can be accomplished surprisingly efficiently by exploiting the MASS algorithm [5].

We denote the updated variables with ^{NEW} in the superscript. In line 1, each newly arriving time point $t^{(+)}$ is appended to the expanding time series $\mathbf{T}^{(+)}$. This completes the next subsequence NEW in $\mathbf{T}^{(+),NEW}_m$ in lines 2 and 3. Lines 4 and 5 correspond to updating the contrasting Matrix Profile by first calculating the distance profile $\mathbf{DP}^{(+-)}_{last,m}$ between $\mathbf{T}^{(-)}$ and NEW, then appending the minimum of $\mathbf{DP}^{(+-)}_{last,m}$ to $\mathbf{MP}^{(+-)}_m$ and storing in $\mathbf{MP}^{(+-),\,NEW}_m$.

TABLE I. THE CONTRASTPROFILE*I* ALGORITHM

```
Algorithm: ContrastProfileI($\mathbf{T}^{(-)}$, $\mathbf{T}^{(+)}$, $t^{(+)}$, $\mathbf{MP}^{(+-)}_m$, $\mathbf{MP}^{(++)}_m$, $m$)
Input: negative time series $\mathbf{T}^{(-)}$, positive time series $\mathbf{T}^{(+)}$, a new positive time point $t^{(+)}$ following $\mathbf{T}^{(+)}$, Matrix Profile $\mathbf{MP}^{(+-)}_m$, Matrix Profile $\mathbf{MP}^{(++)}_m$, and subsequence length $m$.
Output: The Contrast Profile $\mathbf{CP}_m$, the incrementally updated Matrix Profiles $\mathbf{M}P^{(+-),NEW}_m$ and $\mathbf{M}P^{(++),NEW}_m$, and the current time series $\mathbf{T}^{(+),NEW}$.
1   $\mathbf{T}^{(+),NEW} = [\mathbf{T}^{(+)}, t^{(+)}]$
2   $last \leftarrow n - m + 1$ // index of last subsequence in $\mathbf{T}^{(+),NEW}$
3   $NEW \leftarrow \mathbf{T}^{(+),NEW}_{last,m}$ // last subsequence in $\mathbf{T}^{(+),NEW}$ of length m
4   $\mathbf{DP}^{(+-)}_{last,m} \leftarrow$ MASS($\mathbf{T}^{(-)}$, $NEW$))   // Begin AB-join update
5   $\mathbf{MP}^{(+-),\,NEW}_m \leftarrow$ [$\mathbf{MP}^{(+-)}_m$, Min($\mathbf{DP}^{(+-)}_{last,m}$)]
6   $\mathbf{DP}^{(++)}_{last,m} \leftarrow$ MASS($\mathbf{T}^{(+)}$, $NEW$)   // Begin self-join update
7   $\mathbf{MP}^{(++)\prime}_m \leftarrow$ ElemWiseMin($\mathbf{MP}^{(++)}_m$, $\mathbf{DP}^{(++)}_{last,m}$) // Update prev vals
8   $\mathbf{MP}^{(++),\,NEW}_m \leftarrow$ [$\mathbf{MP}^{(++)\prime}_m$, Min($\mathbf{DP}^{(++)}_{last,m}$)]
9   $\mathbf{CP}_m \leftarrow (\mathbf{MP}^{(+-),\,NEW}_m - \mathbf{MP}^{(++),NEW}_m)$/sqrt(2 * m)
10  return $\mathbf{CP}_m$
```

Because this is a Matrix Profile where the query time series is unchanging, the previously computed values are also unchanged. An extra line of work is done in lines 6 – 8 to update the self-join Matrix Profile because the query time series $\mathbf{T}^{(+)}$ has expanded. The self-join distance profile between $\mathbf{T}^{(+)}$ and NEW is stored in $\mathbf{DP}^{(++)}_{last,m}$. The element-wise minimum between $\mathbf{MP}^{(++)}_m$ and $\mathbf{DP}^{(++)}_{last,m}$ is stored in $\mathbf{MP}^{(++)\prime}_m$, which is then updated to $\mathbf{MP}^{(++),NEW}_m$ after concatenating the minimum value of $\mathbf{DP}^{(++)}_{last,m}$. Finally, in lines 9 and 10, $\mathbf{CP}_m$ is recomputed from the expanded $\mathbf{MP}^{(+-),NEW}_m$ and updated and expanded $\mathbf{MP}^{(++),NEW}_m$.

[1] This is what *logically* must be done, however by caching distance calculations and only recomputing values that could have changed, the time and space overhead for the Kth-1 Plato is inconsequential.

The time complexity of ContrastProfile*I* is dominated by the MASS function, which performs an O(*n*log*n*) FFT operation. The time complexity begins as O($|\mathbf{T}^{(+)}|$ log $|\mathbf{T}^{(+)}|$ + $|\mathbf{T}^{(-)}|$ log $|\mathbf{T}^{(-)}|$), but as the size of $\mathbf{T}^{(+)}$ dominates, the effective time complexity is O($|\mathbf{T}^{(+)}|$log$|\mathbf{T}^{(+)}|$). Each time the function is called, MASS searches a slightly longer time series with n becoming $n+1$. There are no conditional control statements, making the runtime value-invariant to the incoming data.

This time complexity discussion is a little indirect. A more intuitive way to measure the time requirements is by using the Maximum Time Horizon, which answers the question, "*How long can the Contrast Profile be maintained before the maintenance computation is slower than the sampling rate?*"

For example, consider the following two scenarios which refer to an Intel® Core i7-9700 CPU at 3.00GHz with 32 GB of memory (full worked details at [10]).

- If we have a Contrast Profile created with $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ both of length 10,000, and the data is arriving at 10Hz, then we can update the Contrast Profile for about 51 hours before the arrival rate is faster than our update time.
- Most automotive GPS loggers update at 1hz. If we have a Contrast Profile created with $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ both of length 10,000, with data arriving at 1Hz, then we can update the Contrast Profile for about 9.5 months before the arrival rate is faster than our update time.

Note that we do not specify the value of m in the above, as the update times are effectively invariant to the subsequence length due to the use of the MASS algorithm.

III. Algorithms That Exploit The Contrast Profile

We believe that the Contrast Profile may be a useful primitive within dozens of higher-level algorithms. In this section we give some concrete examples.

A. *End-to-End Time Series Classification*

As we noted in the introduction, *given* discriminative subsequences (i.e., in the UCR format [2]) that characterize a behavior, time series classification is generally a simple task. We argue that *finding* such discriminative subsequences can be extremely difficult. Clearly the Contrast Profile has the potential to mitigate this difficulty. For concreteness, we outline a basic approach:

- Identify two snippets of time series that conform to the Contrast Profile assumptions.
- Run the Pan-Contrast Profile to discover the Plato.
- Use this Plato with a threshold t to discover similar instances, label them as the class that $\mathbf{T}^{(+)}$ represents.

Note that while the Euclidean distance is the natural distance measure to use, other measures such as DTW are possible [11]. We need to set a threshold; here we must resort to heuristics. For example we can use 3 × the distance for the Plato to its nearest neighbor (recall that we are assuming that the Plato's nearest neighbor is *also* an example of the desired behavior). Finally, the above assumes that there is a single template for the desired behavior. If we think it may be polymorphic, we can use the technique discussed in Section II.A to find the Top-K Platos instead. This is a very simple technique for end-to-end classification, but as we will show on diverse real-word problems, *extremely* effective.

IV. Experimental Evaluation

To ensure that our experiments are reproducible, we have built a website [10] which contains all data/code for the results.

A. *Insect Behavior Classification*

Sapsucking insects (insects in the orders Hemiptera and Homoptera) are insects that feed by sucking nutrients from plants. This behavior is typically not destructive by itself but can spread diseases from plant to plant. Worldwide, across all crops/insects, this results in billions of dollars in crop losses each year. The primary tool used to study these insects is the electrical penetration graph (EPG), which as shown in Fig. 8, produces a complex and noisy time series that reflects the insect's behavior [12].

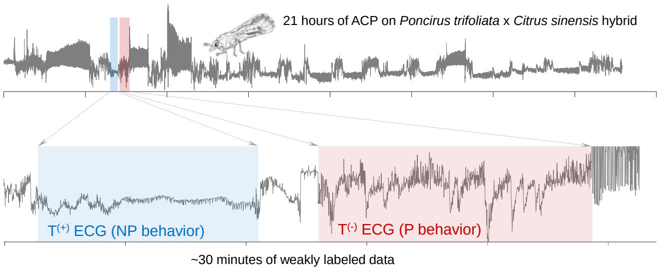

Fig. 8. *top*) 21 hours of Asian citrus psyllid (ACP) feeding behavior on citrus. *bottom*) A zoom-in of a small fraction of the data.

We managed to obtain 21 hours of such data that was annotated by a combination of algorithms and humans (exploiting out-of-band information). Using the two regions shown in Fig. 8.*bottom*, that conform to our algorithm's mild assumptions, we ran the Contrast Profile to produce the Plato shown in Fig. 9.

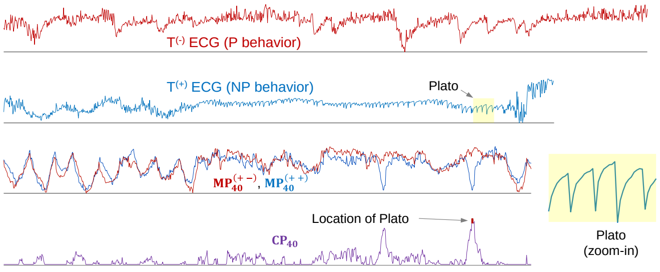

Fig. 9. *top-to-bottom*) The weakly-labeled instances shown in Fig. 8 have their $\mathbf{MP}_{40}^{(+-)}$and $\mathbf{MP}_{40}^{(++)}$computed to produce the $\mathbf{CP}_{40}$, which strongly peaks to indicate the location of the Plato.

Using this template to find the Top-100 instances in the full dataset (excluding training data), the Plato had an error-rate of 7%, whereas the Top-1 motif in $\mathbf{T}^{(+)}$ had an error-rate of 32%, not much better than the default error rate of 36.9%

B. *Chicken Behavior Classification*

Here we revisit the chicken behavior example considered in Fig. 1. First, we should explain why the data is weakly- labeled. The accelerometer worn by the bird *was* approximately synchronized with a video camera trained on the coop. However, technical limitations meant that the synchronization had an error of up to ± 3 seconds. By comparison, the dustbathing behavior we were tasked with quantifying is known to last about 0.5 to 3 seconds. Thus, a domain expert was able to locate 30-second regions *with* and *without* the behavior, but not provide annotations at a *finer* temporal resolution. In Fig. 10 we use the two time series shown in Fig. 1 to compute $\mathbf{CP}_{120}$ in an attempt to find a Plato that can act as a "signature" for dustbathing.

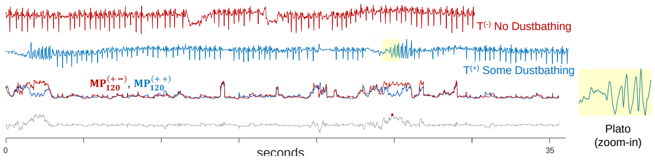

Fig. 10. *top-to-bottom*) The weakly-labeled instances shown in Fig. 1 have their $\mathbf{MP}_{120}^{(+-)}$and $\mathbf{MP}_{120}^{(++)}$ computed to produce the $\mathbf{CP}_{120}$, which strongly peaks to indicate the location of the Plato.

We used this Plato to search a 12,679,054,727 datapoint archive of chicken behavior for the one thousand best matches. The returned matches are shown in Fig. 11.

Domain experts examined the results and confirmed that all the returned subsequences are true positives.

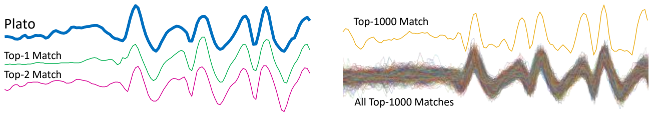

Fig. 11. The Plato used for dustbathing classification (*top.left*). Selected matches returned by a nearest neighbor search using the Plato discovered in Fig. 10. The Top-1000 matches (*bottom.right*).

The discovery of the Plato took 0.3 seconds. Surprisingly, the exact Top-1000 search in the 12.7 billion datapoints of disk-resident data (corresponding to four years of behavior) took only 55 minutes using the MASS algorithm.

V. Conclusions

We have introduced the Contrast Profile, a novel data structure that allows a user or algorithm to reason about the differences between two time series. We reiterate that the Contrast Profile is *not* a classification algorithm, but it can help any downstream time series classification algorithm by finding discriminative prototypes. Beyond allowing end-to-end time series classification with only the weakest possible assumptions/annotations of the data, we have shown that the Contrast Profile has several other uses in data mining, including anomaly detection and data exploration. We have shared all code and data with the community [10], to allow it to confirm and exploit our findings.

References

[1] C. M. Yeh *et al.*, "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec. 2016, pp. 1317–1322. doi: 10.1109/ICDM.2016.0179.

[2] "Welcome to the UCR Time Series Classification/Clustering Page." https://www.cs.ucr.edu/~eamonn/time_series_data_2018/ (accessed Jan. 17, 2021).

[3] A. C. Murillo, A. Abdoli, R. A. Blatchford, E. J. Keogh, and A. C. Gerry, "Parasitic mites alter chicken behaviour and negatively impact animal welfare," *Scientific Reports*, vol. 10, no. 1, Art. no. 1, May 2020, doi: 10.1038/s41598-020-65021-0.

[4] T. Nakamura, M. Imamura, R. Mercer, and E. Keogh, "MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives," p. 11.

[5] M. Abdullah, "The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance." https://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html (accessed Jan. 18, 2021).

[6] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is 'Nearest Neighbor' Meaningful?," in *Database Theory — ICDT'99*, Berlin, Heidelberg, 1999, pp. 217–235. doi: 10.1007/3-540-49257-7_15.

[7] J. Lin and E. Keogh, "Group SAX: Extending the Notion of Contrast Sets to Time Series and Multimedia Data," in *Knowledge Discovery in Databases: PKDD 2006*, Berlin, Heidelberg, 2006, pp. 284–296. doi: 10.1007/11871637_29.

[8] O. Yildirim, U. B. Baloglu, R.-S. Tan, E. J. Ciaccio, and U. R. Acharya, "A new approach for arrhythmia classification using deep coded features and LSTM networks," *Computer Methods and Programs in Biomedicine*, vol. 176, pp. 121–133, Jul. 2019, doi: 10.1016/j.cmpb.2019.05.004.

[9] Y. Zhu *et al.*, "Exploiting a novel algorithm and GPUs to break the ten quadrillion pairwise comparisons barrier for time series motifs and joins," *Knowl Inf Syst*, vol. 54, no. 1, pp. 203–236, Jan. 2018, doi: 10.1007/s10115-017-1138-x.

[10] "Contrast-Profile." https://sites.google.com/view/contrastprofile (accessed Jan. 05, 2021).

[11] C. M. Yeh, Y. Zhu, H. A. Dau, A. Darvishzadeh, M. Noskov, and E. Keogh, "Online Amnestic DTW to allow Real-Time Golden Batch Monitoring," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage AK USA, Jul. 2019, pp. 2604–2612. doi: 10.1145/3292500.3330650.

[12] D. S. Willett, J. George, N. S. Willett, L. L. Stelinski, and S. L. Lapointe, "Machine Learning for Characterization of Insect Vector Feeding," *PLOS Computational Biology*, vol. 12, no. 11, p. e1005158, Nov. 2016, doi: 10.1371/journal.pcbi.1005158.