Robust Multi-Variate Temporal Features of Multi-Variate Time Series

Robust Multi-variate Temporal (RMT) Features of Multi-variate Time Series

SICONG LIU, Arizona State University
SILVESTRO ROBERTO POCCIA, University of Torino
K. SELÇUK CANDAN, Arizona State University
MARIA LUISA SAPINO, University of Torino
XIAOLAN WANG, University of Massachusetts, Amherst

Many applications generate and/or consume multi-variate temporal data and experts often lack the means to adequately and systematically search for and interpret multi-variate observations. In this paper, we first observe that multi-variate time series often carry localized multi-variate temporal features that are robust against noise. We then argue that these multi-variate temporal features can be extracted by simultaneously considering, at multiple scales, temporal characteristics of the time-series *along with external knowledge*, including variate relationships that are known a priori. Relying on these observations, we develop data models and algorithms to detect *robust multi-variate temporal* (RMT) features that can be indexed for efficient and accurate retrieval and can be used for supporting data exploration and analysis tasks. Experiments confirm that the proposed RMT algorithm is highly effective and efficient in identifying *robust* multi-scale temporal features of multi-variate time series.

ACM Reference format:
Sicong Liu, Silvestro Roberto Poccia, K. Selçuk Candan, Maria Luisa Sapino, and Xiaolan Wang. 2017. Robust Multi-variate Temporal (RMT) Features of Multi-variate Time Series. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1, Article 1 (January 2017), 27 pages.
DOI: 0000001.0000001

1 INTRODUCTION

Many applications, such as motion recognition [34], generate temporal data and, in many of these applications, (a) the resulting time series data are multi-variate, (b) relevant processes underlying these time series are of different scales [17, 29, 48], and (c) the variates (i.e., observation parameters) are dependent on each other in various ways [40].

Analysis and exploration of time series (as well as other types of data) often start with extraction of patterns and features that describe salient properties of the data. Popular approaches in the literature include extraction of global features of the time series (such as spectral properties quantified using a transformation; e.g. Discrete Cosine or Wavelet Transforms) and the use of these global features (which describe properties of the time series as a whole) for indexing [5].

This work is supported by
NSF#1339835 "SI2-SSE: E-SDMS: Energy Simulation Data Management System Software", NSF#1318788 "Data Management for Real-Time Data Driven Epidemic Spread Simulations", NSF#1610282 "DataStorm: A Data Enabled System for End-to-End Disaster Planning and Response", NSF#1633381 "BIGDATA: Discovering Context-Sensitive Impact in Complex Systems", and EU-H2020 Marie Sklodowska-Curie grant agreement No 690817 "FourCModeling".
 1551-6857/2017/1-ART1 $15.00
DOI: 0000001.0000001

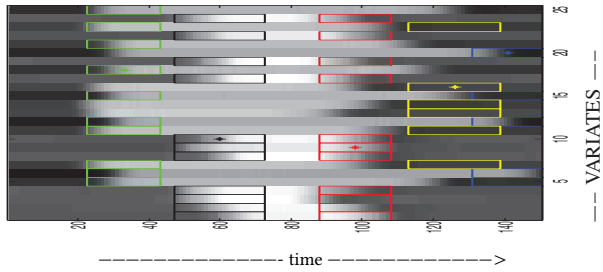


Fig. 1. A multi-variate time series data set, where each variate is plotted as a row of gray scale pixels and sample multi-variate features identified on the data set (each feature is marked with a different color): the figure shows 26 variates of length 150 and 5 local multi-variate features on these time series: note that some of the features correspond to the onset of a rise in amplitude, whereas other correspond to the drop in the series amplitude. For each time series involved in a given multi-variate feature, we plot the corresponding temporal scope (i.e., duration) of that feature.

Correlations, transfer functions, variate clusters, and spectral properties [47], SVD and similar eigen-decompositions can be used for extracting global fingerprints of multi-variate time series data [27]. The analogous analysis operation on a tensor, which can be used to represent temporal evolution of multi-modal data, is known as tensor decomposition [23]. Both matrix and tensor decomposition operations, as well as other techniques, such as probabilistic techniques (such as Dynamic Topic Modeling, DTM [3]), and AutoRegressive Integrated Moving-Average (ARIMA) based analyses (which separate a time series into autoregressive, moving-average, and integrative components for modeling and forecasting [33]) are expensive.

Several researchers noticed that significant amount of waste in processing and exploration can be avoided if the attention is directed towards parts of a given time series that are likely to contain interesting patterns [4, 35]. One way to achieve this involves searching for frequently repeating patterns; this is commonly known as the *motif search* problem [14]. Most of the common approaches for motif search involve incrementally moving (or *shift*ing) a fixed-length time window starting from the beginning of the given time series. For each window interval a temporal signature (such as SAX words [28]) is generated (to speed up the matching of subsequences) and frequent sub-sequences are discovered using different indexing and hashing algorithms and leveraging pruning techniques for eliminating non-promising subsequences [54]. Other local features of uni-variate time-series include landmarks [38], perceptually important points (PIP) [10], patterns [1], shapelets [42, 55], snippets [50], longest common subsequences (LCSS [49]), and motif-based schemes (which search for frequently repeating temporal patterns) [6]. Noting that uni-variate time series often carry localized temporal features which can be used for efficient search and analysis, in our earlier work [4] we developed an sDTW algorithm for extracting salient local features (that are robust against various types of noise) of uni-variate time series and showed that these can help align similar time series more efficiently and effectively. RPM [52] and STS3 [37] are two recent approaches that also seek informative patterns from uni-variate time series.

1.1 Contributions of this Paper: Local Features of Multi-Variate Time Series

In this paper, we develop data models and algorithms to detect local, *robust multi-variate temporal* (RMT) features of multi-variate time series (Figure 1). Recently, in [51], we proposed a multi-variate feature extraction technique, which considered the relationships and dependencies between the individual uni-variate time-series that make up the multi-variate series. The local,

robust multi-variate temporal (RMT) features are extracted leveraging known correlations and dependencies among the variates. As in [4], for uni-variate series, [51] also (a) smoothes the data to generate different version of the input object corresponding to different scales and (b) compares neighboring points both in time (or in x and y dimensions) and scale to identify regions where the gradients are large. As in [31], SIFT-like feature descriptors are extracted to support search.

What makes the problem of extracting local features from multi-variate series difficult, however, is that the concepts of neighborhood, gradient, *and* smoothing *are not well-defined in the presence of multiple variates.* In [51], we argued that this difficulty can be overcome by leveraging metadata (known correlations and dependencies among the variates in the time series) to define neighborhoods, support data smoothing, and construct scale spaces in which gradients can be measured. Based on this observation, we proposed *topology-sensitive smoothing* and *topology-sensitive gradient computation* techniques to identify local features of multi-variate time series at different time/variate scales. In this paper, we show that unlike [51] (where the time and variate scales are shrank and expanded together), more effective local-feature sets can be located if we allow for the time and variate aspects of the multi-variate time series to be considered independently from each other – leading to multi-variate features with heterogeneous time- and variate-scales. This is also visualized in Figure 1, where we see some features that are short in time but contain a lot of variates (such as the feature highlighted in red), whereas others are longer, but contain fewer variates (such as the feature highlighted in yellow). We also provide a detailed discussion of RMT based multi-variate time series matching and inconsistency removal and experimentally evaluate their effectiveness in gesture and motion recognition [34].

1.2 Organization of the Paper

In the next section, we describe the related work. In Section 3.1, we introduce the metadata-enriched multi-variate time series (MMTS) model that forms the basis of the proposed work. In Section 3.2, we present an overview of the proposed approach to locate *robust multi-variate temporal* (RMT) features and extract their descriptors. Sections 4 through 8 provide details of the various steps of the proposed RMT feature identification and descriptor extraction algorithm. Section 9 formally defines the RMT feature set of a multi-variate time series and Section 10 describes how to use these features for matching multi-variate time series. We present experimental evaluations of the proposed approach in Section 11. We conclude the paper in Section 12.

2 RELATED WORK

Euclidean distance and, more generally L_p-norm measures, were among the first used to determine the similarity between two time series. They require that the time series being compared are of *same temporal length* and, since they assume strict synchrony among time series, they are not suitable when two time series can have different speeds or are shifted in time [7, 21]. Other measures that require equal length and perfect synchrony across time series include *cosine* and (Pearson's) correlation similarity [44]. In contrast, edit distance [24] measures aim to determine the minimum sequence of *edit* operations that are required to measure similarity. In 70s Sakoe [43] and then in mid 90s, Berndt [2] proposed dynamic time warping (DTW) technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions. Intuitively, DTW considers all possible warping paths that can warp (or transform) one series into the other and picks the warping path that has the lowest cost. DTW has found wide acceptance and last two decades have seen several innovations [8, 11, 20, 21, 41]. For example, while the original DTW is not metric (does not satisfy triangular inequality) [8] proposed an extended version of DTW that satisfies triangular inequality. Most of the above algorithms, including DTW, are initially designed

for comparing uni-variate time series. More recently, various extensions of DTW have been proposed for *multi*-dimensional time series [39, 45]. The most prevalent of these are the *vectorized* and *independent* extensions. In *vectorized DTW*, a multi-variate time series is considered as a sequence of vectors, where the length of vector is equal to number of variates in the time series. The DTW algorithm is then applied using the distances among these vectors instead of differences in signal amplitude. In *independent DTW*, however, each variate is treated independently from the others and DTW is applied separately to each; finally, these independent DTW distances are added to compute the overall distance between the given pair of multi-variate series.

An alternative approach to the above techniques is to extract *features* from the given time series and use these features to compute similarity/distance instead of the original series. *We provided a detailed discussion of the related work on global and local features in the previous section.* [18], for example, proposed a feature-extraction algorithm that extracts minimal distinguishing subsequences that can be used as features. Morchen in [36] proposed using DFT (Discrete Fourier Transform) and DWT (Discrete Wavelet Transform) for feature extraction. *PCA-Similarity Factor* [25] and EROS (*Extended Frobenius norm*) [53], that use matrix factorization techniques, such as singular vector decomposition (SVD) and principle component analysis, have also been proposed to transform the input multi-variate time series into equal length and then apply cosine similarity over them. Applications of SVD and DTW for various multimedia tasks, such as similarity search, classification, recognition, and watermarking, include [19, 22, 26, 32, 46].

As we discussed earlier, in [51], we proposed to extract and use SIFT [30, 31]-like robust multi-variate temporal features to determine similarity between time series. In this paper, we extend the approach in [51] with more general scale-space construction and pruning techniques to obtain better classification performances. The general framework is named RMT for both papers, but the underlying techniques here are significant extensions of [51]: In particular, [51] utilizes a special case of our proposed generalized scale-space construction and pruning techniques, where only diagonal scale space is considered. In this submission, however, we argue (and experimentally show) that, in general, using a more complete scale-space can be more effective. Indeed, as the experimental results reported in Section 11 show, we achieve better accuracy using a complete scale-space (enabling time and variate scales to be different) rather than using a diagonal scale space (where the time and variate scales are constrained to be identical). In addition, while [51] considers only one scheme for time series matching using RMT features, this paper introduces several alignment measures (including for alignment of features pairs and measuring feature significance) and provides a detailed study of the impact of these measures on the classification accuracies.

3 DATA MODEL AND OVERVIEW OF THE PROPOSED APPROACH

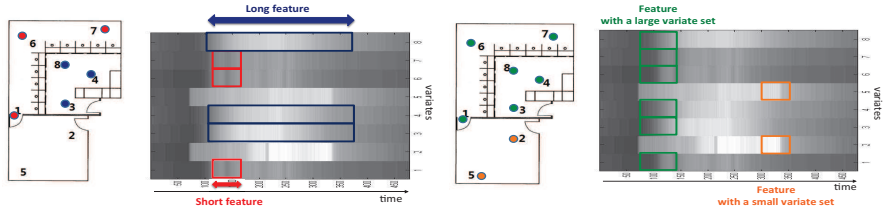
Before describing the process through which we extract RMT features, we first introduce the metadata-enriched, multi-variate time series model underlying the proposed approach.

3.1 Metadata-Enriched Multi-Variate Time Series (MMTS) Model

In this section, we present a *metadata-enriched multi-variate time series* (MMTS) model which minimizes the assumptions that need to be made about the structure of the data:

Definition 3.1 (Metadata-Enriched Multi-Variate Time Series (MMTS)). A metadata-enriched multi-variate (MM) time series is a four-tuple $\mathbf{Y} = (\mathcal{V}, \mathcal{M}, \mathbb{Y}, \mathcal{D})$, where

- $\mathcal{V}$ is a set of variates,
- $\mathcal{M} = \{M_1, \ldots, M_m\}$ is a set of metadata modalities, where each modality M_i describes how the corresponding subset $V_i \subseteq \mathcal{V}$ of variates are related to each other,
- $\mathbb{Y}$ is a $(d_1 + d_2 + \ldots + d_m) \times l$ data matrix, where
  - l is the temporal length of the multi-variate time series,



(a) Different lengths (b) Different numbers of variates

Fig. 2. Multi-variate features can be of different sizes (in this example, the multi-variate series represent temperature readings in a floor split into zones)

- $d_i = |V_i|$, and
- cells of the matrix $\mathbb{Y}$ take values from the data domain $\mathcal{D}$. ◇

As defined above, each variate is associated with a modality metadata describing how it is related with other variates. In this paper, without loss of generality, we consider *graph-organized* (G) representation of variate modalities: Each modality, i, has an associated graph $G_i(V_i, E_i, W_i)$ that relates the variates V_i of the given mode. Depending on the application, the graph maybe directed or undirected and weights may have *distance* or *similarity* semantics. If the underlying graph is unweighted, then for all $e_k \in E_i$, $W_i(e_k) = 1$.

Note that graph-based description of variate relationships is a common way of modeling temporal dynamics of multi-variate time series [12, 16, 47]. Note also that, while the metadata describes the relationship between the variates, this relationship may or may not have causal impact on the observed temporal characteristics of the data:

Definition 3.2 (Metadata-Defined Variate Causality Model). Let us assume that we have metadata $\mathcal{M}$ that describe the relationship between the variates in the data. Given $\mathcal{M}$, under the variate causality model, we have $\mathbb{Y}[t] = \mathbf{R}_{\mathcal{M}}\mathbb{Y}[t-1] + \vec{E}(t)$, where $\mathbb{Y}[t]$ is a column vector extracted from $\mathbb{Y}$ and corresponds to the observations at time t, $\mathbf{R}_{\mathcal{M}}$ is a (row-normalized) matrix defining how the values of $\mathbb{Y}$ at time $t-1$ impact the values of $\mathbb{Y}$ at time t, and $\vec{E}$ is a multi-variate time series denoting independent, external inputs. ◇

Intuitively, $\mathbf{R}_{\mathcal{M}}$ is a matrix describing how the values of one variate are impacted by the past values of the variates in the data. Alternatively, $\mathbf{R}_{\mathcal{M}}$ may be a matrix describing the relationships among simultaneous observations:

Definition 3.3 (Metadata-Defined Variate Correlation Model). Let us assume that we have metadata $\mathcal{M}$ that describe the relationship between the variates in the data. Given $\mathcal{M}$, under the variate correlation model, we have a matrix $\mathbf{R}_{\mathcal{M}}$ such that $\mathbf{R}_{\mathcal{M}}[i,j] = \Phi(\mathbb{Y}[*,i], \mathbb{Y}[*,j]) \in [0,1]$. Here $\mathbb{Y}[*,i]$ and $\mathbb{Y}[*,j]$ are rows corresponding to observations for variates i and j, respectively, and Φ is an application specific similarity function. ◇

Here, Φ may be computed by comparing (recent) historical data of the time series or may reflect available domain knowledge, such as the distance of the sensors recording the variates or known relationships parameters.

It is important to note that the algorithms presented in the paper are applicable under both of the above models[1] and we use the matrix $\mathbf{R}_{\mathcal{M}}$ to denote both relationships.

[1]Thus, without loss of generality, we sometimes focus on the dependency model and, other times, use the correlation model.

3.2 Overview: Extracting Local, Robust Multivariate Temporal (RMT) Features

In this paper, we propose algorithms to extract *robust multi-variate temporal* (RMT) features from metadata-enriched multi-variate time series (MMTS) data sets. Intuitively, as already visualized in Figure 1, an RMT feature is a fragment of a multi-variate time series that is maximally different from its immediate neighborhood (both in time and across variate relationships specified by the metadata). As in [31], we rely on a four step process to identify such RMT features and extract their feature descriptors: *(Step 1): Scale-space construction:* As shown in Figure 2, multi-variate temporal features of interest can be of different lengths and may cover different number of variates. In order to be able to locate such features of different sizes, the RMT features are extracted from a scale-space we construct for the given multi-variate time series through iterative smoothing[2]. Iterative smoothing of the multi-variate data in time and variates creates different resolution versions of the input data and, thus, helps identify features with different amount of details in time and in terms of the number of variates involved. While iterative smoothing techniques are well understood for uni-variate data [4, 48], this is not the case for multi-variate time series. Therefore, in Sections 4 and 5, we describe how to construct a scale-space by smoothing a multi-variate time series, leveraging available metadata that describe known relationships among variates. *(Step 2): Identifying feature candidates:* Next, the process identifies candidate features of interest across multiple scales of the given multi-variate time series by searching over multiple scales and variates of the given series. Each candidate RMT feature has a *temporal-scope* (a beginning and an end in time) and a *variate-scope* (a set of variates involved in the feature). These candidate features of interest are those with the largest variations with respect to their neighbors in time, variates, and scale. We describe this process in Section 6. *(Step 3): Eliminating poor candidates:* At the following step, those candidate features identified in the previous step that are sensitive to noise are eliminated. These include features that are poorly localized (and hence are difficult to match). This is described in Section 7. *(Step 4): RMT feature descriptor creation:* In the final step, for each RMT feature, a local descriptor is extracted using the information obtained in the previous steps. More specifically, the algorithm computes and samples gradients within the scope of the RMT feature. To avoid sudden changes in the descriptor with small changes in the position and to give less emphasis to gradients that are far from the center of the descriptor, a weighing function is used to assign a weight to the magnitude of each sample point based on its distance from the center of the feature. Note that while gradient computation is well understood for uni-variate data [4, 48], this is not the case for multi-variate time series. We describe how this is achieved in Section 8.

The above approach has three key advantages: **Advantage 1:** First of all, the identified salient features are robust against noise and common transformations, such as temporal shifts or dropped/missing variates. **Advantage 2:** Scale invariance enables the extracted salient features to be robust against variations in speed and enables multi-resolution searches. **Advantage 3:** The temporal and relationship scales at which a multi-variate feature is located give an indication about the *scope* (both in terms of duration and the number of variates involved) of the multi-variate feature.

4 TEMPORAL AND VARIATE SMOOTHING OF MULTI-VARIATE TIME SERIES

Let $\mathbf{Y} = (\mathcal{V}, \mathcal{M}, \mathbb{Y}, \mathcal{D})$ be a metadata-enriched multi-variate time series, as defined in Section 3.1. The first step in identifying multi-variate features of $\mathbf{Y}$ is to generate a scale-space representing versions of the given multi-variate series with different amounts of details. As shown in Figure 3, the

[2]This is different from what is known as "multi-variate exponential smoothing", a forecasting technique where the multi-variate models include the so-called "smoothing parameters" and these are learned to obtain models with a better fit to the data [47].

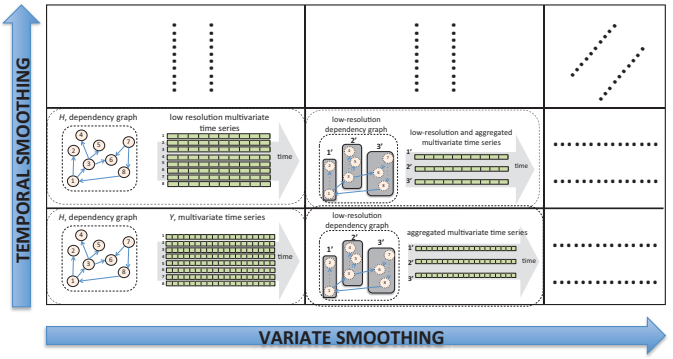


Fig. 3. Scale-space construction through iterative smoothing in time and variates

scale-space, $\mathfrak{Y}$, of $\mathbf{Y}$ is obtained through iterative smoothing across both time and variate relationships, starting with an initial smoothing parameter $\Sigma_0 = \langle \sigma_{time,0}, \sigma_{var,0} \rangle$ and iteratively increasing the smoothing degree up to $\Sigma_{max} = \langle \sigma_{time,max}, \sigma_{var,max} \rangle$, obtaining differently smoothed versions of the time series.

The values of Σ_0 and Σ_{max} control the sizes of the smallest and largest features sought in the data (see Section 9). In the rest of this section, we will first describe temporal and variate smoothing techniques. We will then describe optimizations to reduce the cost of the scale-space construction step of the process. For each of the techniques, we will also discuss the relationship among Σ_0, Σ_{max}, and the sizes of the features identified.

4.1 Temporal Smoothing

Let $\mathbf{Y} = (\mathcal{V}, \mathcal{M}, \mathbb{Y}, \mathcal{D})$ be a metadata-enriched multi-variate time series and let $Y_v = \mathbb{Y}[*, v]$ be a uni-variate time series corresponding to one of its variates. Let $Y_v^{(\sigma)}$ indicate a version of the uni-variate time series, Y_v, smoothed through convolution with the Gaussian function, $G(t, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-t^2}{2\sigma^2}}$, with temporal smoothing parameter σ (Figure 4). Given this, $\mathbf{Y}^{(time,\sigma)} = (\mathcal{V}, \mathcal{M}, \mathbb{Y}^{(time,\sigma)}, \mathcal{D})$, is a version of the multi-variate time series, $\mathbf{Y}$, where each row of $\mathbb{Y}^{(time,\sigma)}$ is a uni-variate time series smoothed with temporal smoothing parameter σ, independently of the other uni-variate series.

4.1.1 Temporal Scope. Let us consider a time instant t on which we are applying Gaussian smoothing with parameter σ. Since, under Gaussian smoothing, 3 standard deviations (i.e. 3σ both directions) would cover $\sim 99.73\%$ of the contributions to the smoothed values, we can define the corresponding *temporal scope* as a time interval, centered at t, of length 6σ; in other words, we have $scope_T(t, \sigma) = [t - 3\sigma, t + 3\sigma)$. Consequently, if the temporal length of a multi-variate time series is L, then we must have $\sigma_{time,max} \leq L/6$. Similarly, since we expect that the smallest feature should involve a time instant and at least its two immediate neighbors, we also have $\sigma_{time,0} \geq 2/6$.

4.1.2 Octaves of Temporal Smoothing. Let σ_1 and σ_2 be two smoothing parameters. The parameter σ_2 is said to be an *octave* larger than σ_1 if $\sigma_2 = 2\sigma_1$. Intuitively, σ_2 defines features twice as large as σ_1 by using a Gaussian smoothing parameter twice as large.

4.2 Variate Smoothing

As described above, the temporal smoothing process relies on a convolution operation that leverages the temporal ordering of the time instants in the series. The challenge is that a similar total

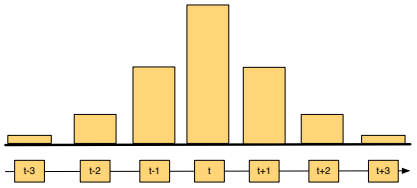


Fig. 4. Gaussian smoothing of a uni-variate series for time instant, t

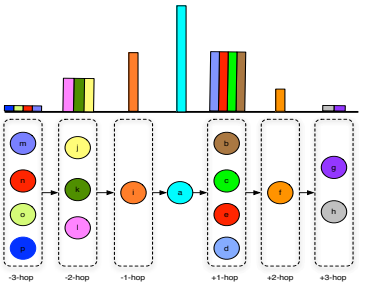


Fig. 5. Graph smoothing for a node

order does not necessarily exist among the variates – therefore, the definition of variate smoothing is not as straightforward.

4.2.1 Gaussian Smoothing of Graph-Organized Variates. Let $\mathbf{Y} = (\mathcal{V}, \mathcal{M}, \mathbb{Y}, \mathcal{D})$ be a metadata-enriched multi-variate time series, where the metadata $\mathcal{M}$ is graph-structured; i.e., there is a graph $G(V, E, W)$ that relates the variates, V, of the data. Let us further define $frwd_G(v_l, \delta)$ and $bkwd_G(v_l, \delta)$, as the forward and backward neighbors of variate $v_l \in V$ at a distance $\geq (\delta - 0.5)$ and $< (\delta + 0.5)$ on G. Intuitively, $frwd_G()$ and $bkwd_G()$ functions order all the variates into a partial order relative to the variate v_l.

Given the partial order defined by the $frwd_G()$ and $bkwd_G()$ functions and a non-negative smoothing parameter σ, we then obtain the Gaussian smoothed version, $\mathbb{Y}^{(var,\sigma)}$ of the matrix $\mathbb{Y}$ as follows: Let $\mathbb{Y}[t]$ be a column vector extracted from $\mathbb{Y}$ corresponding to the observations at time t. Then, for all v_l, we have $\mathbb{Y}^{(var,\sigma)}[t, l]$ equal to

$$\left(\underset{\substack{v_h \in \underset{bkwd_G(v_l,0)}{\underset{frwd_G(v_l,0)}{\cup}}}}{AVG} \mathbb{Y}[t,h] \right) + \sum_{\delta=1}^{\infty} G(\delta,\sigma) \left(\underset{v_h \in frwd_G(v_l,\delta)}{AVG} \mathbb{Y}[t,h] \right) + \sum_{\delta=1}^{\infty} G(\delta,\sigma) \left(\underset{v_h \in bkwd_G(v_l,\delta)}{AVG} \mathbb{Y}[t,h] \right).$$

Figure 5 shows how we apply Gaussian smoothing over a relationship graph. The lower half of the figure shows a variate a and its forward and backward k-hop neighbors in the relationship graph. As shown in the upper half of the figure, when identifying the contributions of the variate on a, Gaussian smoothing is applied along the hop distance. Since at a given hop distance there may be more than one variate, all the variates at the same distance have the same degree of contribution and the degree of contribution gets progressively smaller as we get away from the variate for which the smoothing is performed.

4.2.2 Variate Scope under Gaussian Smoothing. Similarly to the temporal scope, we define the variate scope corresponding to variate v_l at smoothing level σ as

$$scope_V(v_l, \sigma) = \{v_l\} \cup \left(\bigcup_{\delta \leq 3\sigma} frwd_G(v_l, \delta) \right) \cup \left(\bigcup_{\delta \leq 3\sigma} bkwd_G(v_l, \delta) \right).$$

The variate smoothing parameters, $\sigma_{var,0}$ and $\sigma_{var,max}$, must be selected such that for each variate v_l, $\sigma_{var,0}$ includes its immediate (one hop) graph neighbors, $forward\_neighbors(v_l)$ and $backward\_neighbors(v_l)$ on G, and the value of δ corresponding to $\sigma_{var,max}$ should be compatible with the diameter of the graph G.

4.2.3 Octaves of Gaussian Variate Smoothing. Let σ_1 and σ_2 be two Gaussian graph smoothing parameters. Under Gaussian smoothing, the graph smoothing parameter σ_2 is said to be an *octave* larger than σ_1 if $\sigma_2 = 2\sigma_1$.

4.3 Combined Time and Variate Smoothing

Given the above definitions of temporal and variate smoothing functions, we now define combined time and variate smoothing of metadata-enriched multi-variate time series:

Definition 4.1 (TV-Smoothing of a Multi-Variate Time Series). Let $\mathbf{Y} = (\mathcal{V}, \mathcal{M}, \mathbb{Y}, \mathcal{D})$ be a metadata-enriched multi-variate (MM) time series. Recall that $\mathbb{Y}$ is a $(d_1 + d_2 + \ldots + d_m) \times l$ data matrix, where l is the temporal length of the multi-variate time series, $d_i = |V_i|$, and $\mathbb{Y}$ takes values from the data domain $\mathcal{D}$. For a given smoothing parameter, $\Sigma = \langle \sigma_{time}, \sigma_{var} \rangle$, the TV-smoothed version, $\mathbf{Y}\{\Sigma\}$, of the multi-variate time series, $\mathbf{Y}$, is defined as $\mathbf{Y}\{\Sigma\} = \left(\mathbb{Y}^{(time, \sigma_{time})}\right)^{(var, \sigma_{var})}$, where,

- $\mathbb{Y}^{(time, \sigma_{time})}$ is a version of $\mathbb{Y}$ where each row (i.e., each uni-variate time series) is temporally smoothed with smoothing parameter σ_{time}, independently from the rest; and
- $\mathbb{X}^{(var, \sigma_{var})}$ is a version of $\mathbb{X}$ where each column (i.e., time instant) is smoothed with smoothing parameter σ_{var}, using the variable relationships and modalities described by the metadata $\mathcal{M}$. ◇

5 STEP 1: SCALE-SPACE CONSTRUCTION FOR MULTI-VARIATE TIME SERIES

As we have seen in Section 4, given a metadata-enriched multi-variate time series, $\mathbf{Y} = (\mathcal{V}, \mathcal{M}, \mathbb{Y}, \mathcal{D})$, first step in identifying multi-variate features of $\mathbf{Y}$ is to generate a scale-space representing versions of the multi-variate series with different amounts of details. In this paper, we consider two types of scale-spaces: *diagonal* and *full* scale-spaces, described below.

5.1 Diagonal Scale-Spaces

Let $\Sigma_0 = \langle \sigma_{time,0}, \sigma_{var,0} \rangle$ be the user provided smallest temporal and variate smoothing parameters and let l indicate the total number of layers in the scale space. An l-layer *diagonal* state space, $\mathfrak{Y}_{diag}$, is defined as a set of data matrices $\{\mathbb{Y}_0, \ldots, \mathbb{Y}_l\}$, where $\mathbb{Y}_i = \mathbb{Y}\{\langle \sigma_{time,0} \times k^i, \sigma_{var,0} \times k^i \rangle\}$, for some scaling parameter $k > 1$. Note that, in this case, we have $\sigma_{time,max} = \sigma_{time,0} \times k^l$ and $\sigma_{var,max} = \sigma_{var,0} \times k^l$. This will generate only the diagonal entries in the scale-space shown in Figure 3.

5.2 Full Scale-Spaces

In contrast, the complete scale-space shown in Figure 3 is generated as follows: Let

- $\Sigma_0 = \langle \sigma_{time,0}, \sigma_{var,0} \rangle$ be the smallest temporal and variate smoothing parameters,
- $\mathcal{L} = \langle l_{time}, l_{var} \rangle$ indicate the number of temporal and variate smoothing layers, and
- $\mathcal{K} = \langle k_{time,}, k_{var} \rangle$ be scaling parameters for temporal and variate smoothings.

An $\mathcal{L}$-layer *full* state space, $\mathfrak{Y}_{full}$, is defined as a set of data matrices $\langle \mathbb{Y}_{0,0}, \ldots, \mathbb{Y}_{i,j}, \ldots, \mathbb{Y}_{l_{time}, l_{var}} \rangle$, where $\mathbb{Y}_{i,j} = \mathbb{Y}\{\langle \sigma_{time,0} \times k_{time}^i, \sigma_{var,0} \times k_{var}^j \rangle\}$. In this case, we have $\sigma_{time,max} = \sigma_{time,0} \times k_{time}^{l_{time}}$ and $\sigma_{var,max} = \sigma_{var,0} \times k_{var}^{l_{var}}$.

5.3 Optimization: Time and Variate Subsampling

In the process described above, the multi-variate time series is incrementally smoothed both in time and relationships, halving details at each octave. We note that, once the details have been

halved at an octave boundary, performing the feature extraction operation at the same level detail is going to be wasteful. To avoid such waste, we subsample the multi-variate time series at octave boundaries (Figure 3). More specifically, at temporal octave boundaries (where temporal details have been halved) we drop one out of every two consecutive temporal observations, reducing the size of the data by half. Similarly, at variate octave boundaries (where variate relationship details have been halved) we reduce the numbers of variates by half by applying a variate clustering algorithm[3].

6 STEP 2: IDENTIFYING MULTI-VARIATE TEMPORAL FEATURE CANDIDATES

Building on the observation [4, 31] that robust localized features are often located where the differences between neighboring regions (possibly in different scales) are large, we seek RMT features of the given multi-variate time series at the *local extrema* of the scale space defined by the difference-of-smoothing (DoS) series. Naturally, the DoS generation and feature identification process will be slightly different depending on whether a diagonal or full scale-space is used.

6.1 Local Extrema in Diagonal Scale-Spaces

An l-layer *diagonal* state space of a metadata-enriched multi-variate time series, $\mathbf{Y} = (\mathcal{V}, \mathcal{M}, \mathbb{Y}, \mathcal{D})$, is defined as a set of data matrices $\{\mathbb{Y}_0, \ldots, \mathbb{Y}_l\}$, where $\mathbb{Y}_i = \mathbb{Y}\{\langle \sigma_{time,0} \times k^i, \sigma_{var,0} \times k^i \rangle\}$, for some scaling parameter $k > 1$. Given this, we create the corresponding DoS by considering a sequence of difference matrices $\{\mathbb{D}_0, \ldots, \mathbb{D}_{l-1}\}$, where $\mathbb{D}_i = |\mathbb{Y}_{i+1} - \mathbb{Y}_i|$. We detect RMT feature candidates by seeking the local maxima and minima of the resulting DoS: each variate-time-scale (VTS) triple, $\langle v, t, s \rangle$, is compared to its neighbors (both in time and variate relationships) in the same scale as well as the scales above and below, and the triple is selected as a candidate only if it is close to being an extremum; i.e., each $\langle v, t, s \rangle$ is compared against its 26 ($= 3^3 - 1$) *neighbors* in time, scale, and variate relationships[4]. More specifically, for each $\langle v, t, s \rangle$, we compare $\mathbb{D}_s[v, t]$ against

$$max \left\{ \begin{array}{ccc} \mathbb{D}_{s-1}[v, t-1] & \mathbb{D}_s[v, t-1] & \mathbb{D}_{s+1}[v, t-1] \\ \mathbb{D}_{s-1}[v, t] & & \mathbb{D}_{s+1}[v, t] \\ \mathbb{D}_{s-1}[v, t+1] & \mathbb{D}_s[v, t+1] & \mathbb{D}_{s+1}[v, t+1] \\ \mathbb{FD}_{s-1}[v, t-1] & \mathbb{FD}_s[v, t-1] & \mathbb{FD}_{s+1}[v, t-1] \\ \mathbb{FD}_{s-1}[v, t] & \mathbb{FD}_s[v, t] & \mathbb{FD}_{s+1}[v, t] \\ \mathbb{FD}_{s-1}[v, t+1] & \mathbb{FD}_s[v, t+1] & \mathbb{FD}_{s+1}[v, t+1] \\ \mathbb{BD}_{s-1}[v, t-1] & \mathbb{BD}_s[v, t-1] & \mathbb{BD}_{s+1}[v, t-1] \\ \mathbb{BD}_{s-1}[v, t] & \mathbb{BD}_s[v, t] & \mathbb{BD}_{s+1}[v, t] \\ \mathbb{BD}_{s-1}[v, t+1] & \mathbb{BD}_s[v, t+1] & \mathbb{BD}_{s+1}[v, t+1] \end{array} \right\},$$

where $\mathbb{FD}_s[v, t] = \Big(F\mathbb{D}_s\Big)[v, t]$, $\mathbb{BD}_s[v, t] = \Big(B\mathbb{D}_s\Big)[v, t]$, and F and B are two matrices describing forward and backward relationships among variates. Intuitively, $\mathbb{FD}$ accounts for the combined DoS values of the forward neighbors and $\mathbb{BD}$ accounts for the combined DoS values of the backward neighbors of v. We declare the triple, $\langle v, t, s \rangle$, a candidate if the corresponding DoS value, $\mathbb{D}_s[v, t]$, is greater than $\Theta\%$ of the maximum of its 26 scale-neighbors in DoS, for some user provided $\Theta \sim 100$.

6.2 Local Extrema in Full Scale-Spaces

As seen earlier, an $\mathcal{L}$-layer *full* state space, $\mathfrak{Y}_{full}$, is defined as a set of data matrices $\langle \mathbb{Y}_{0,0}, \ldots, \mathbb{Y}_{i,j}, \ldots, \mathbb{Y}_{l_{time}, l_{var}} \rangle$, where $\mathbb{Y}_{i,j} = \mathbb{Y}\{\langle \sigma_{time,0} \times k^i_{time}, \sigma_{var,0} \times k^j_{var} \rangle\}$. Given this, for

[3]In the experiments reported in Section 11, we use a k-means algorithm, where k is equal to the half of the number of variates, based on the distances among sensors on the underlying sensor-distance graph

[4]The number of neighboring triples may be less than 26 if the triple is at the boundary in terms of time, scale, or variate relationship graph.

each $s = \langle i, j \rangle$ pair, we can define three differences:

$$\mathbb{D}^t_{i,j} = \left|\mathbb{Y}_{i+1,j} - \mathbb{Y}_{i,j}\right|, \quad \mathbb{D}^v_{i,j} = \left|\mathbb{Y}_{i,j+1} - \mathbb{Y}_{i,j}\right|, \quad \text{and} \quad \mathbb{D}^{t,v}_{i,j} = \left|\mathbb{Y}_{i+1,j+1} - \mathbb{Y}_{i,j}\right|.$$

Local extrema are then identified by considering each variate-time-scale (VTS) triple, $\langle v, t, s \rangle$, and comparing $max(\mathbb{D}^t_{i,j}, \mathbb{D}^v_{i,j}, \mathbb{D}^{t,v}_{i,j})$ to 78 ($= 3 \times 26$) neighboring triples[5] of $\langle v, t, s \rangle$ in time, scale, and relationships for each of the $\mathbb{D}^t$, $\mathbb{D}^v$, and $\mathbb{D}^{t,v}$. We finally declare the triple, $\langle v, t, s \rangle$, a candidate if the corresponding DoS value, $\mathbb{D}_s[v, t]$, is greater than Θ% of the maximum of its 78 neighbors in DoS, for some user provided $\Theta \sim 100$.

7 STEP 3: ELIMINATING POOR RMT FEATURE CANDIDATES

Local extrema of DoS can include candidate triples that are poorly localized. In order to identify whether a triple $\langle v, t, s \rangle$ is well or poorly localized in the scale-space, we can consider the principal curvatures at the point $\langle v, t, s \rangle$ of the scale-space generated earlier: a poorly defined peek in the difference-in-smoothing will have a large principal curvature in the scale space in one direction, but a small one in the perpendicular direction. Consequently, as was observed in [15, 31], we can search for well-localized candidates by considering the ratio of the eigenvalues of the 2×2 Hessian matrix, which describes the local curvature of the scale-space in terms of the second-order partial derivatives.

Given the above observation, the major challenge, in this case, is to define and compute the partial derivatives for metadata enhanced multi-variate time series to obtain the Hessian matrix we seek. More specifically, for each VTS triple, $\langle v, t, s \rangle$, we need to construct a 2×2 time/variates Hessian matrix, $\mathfrak{D}^{TV}_{v,t,s} = \begin{bmatrix} D_{T,T} & D_{T,V} \\ D_{V,T} & D_{V,V} \end{bmatrix}$, where

- $D_{T,T} = D_T D_T$ is the second derivate along time for the triple $\langle v, t, s \rangle$,
- $D_{V,V} = D_V D_V$ is the second derivative along "variate relationships" for $\langle v, t, s \rangle$,
- $D_{T,V} = D_T D_V$ is the partial derivative along time of the partial derivate along variate relationships of the triple $\langle v, t, s \rangle$, and
- $D_{V,T} = D_V D_T$ is the partial derivative along variate relationships of the partial derivate along time of the triple $\langle v, t, s \rangle$.

In this paper, we propose to estimate the derivatives along time and variate relationships by taking differences of neighboring sample points:

$$\begin{aligned} D_T(v, t, s) &= \mathbb{Y}_s[v, t+1] - \mathbb{Y}_s[v, t-1], \\ D_V(v, t, s) &= \begin{cases} (\mathbb{F}\mathbb{Y}_s[v, t] - \mathbb{B}\mathbb{Y}_s[v, t]) & \text{for directed relationships} \\ (\mathbb{F}\mathbb{Y}_s[v, t] - \mathbb{Y}_s[v, t]) & \text{for undirected relationships} \end{cases} \end{aligned}$$

Here, $\mathbb{F}\mathbb{Y}_s$ and $\mathbb{B}\mathbb{Y}_s$, account for the (weighted) averages of the forward and backward variate neighbors at the corresponding scale: i.e., $\mathbb{F}\mathbb{Y}_s[v, t] = \left(F\mathbb{Y}_s\right)[v, t]$ and $\mathbb{B}\mathbb{Y}_s[v, t] = \left(F\mathbb{Y}_s\right)[v, t]$, where (as was discussed in the previous section) F and B are two matrices describing forward and backward relationships among variates.

Once the Hessian matrix, $\mathfrak{D}^{TV}_{v,t,s}$, is constructed for the triple $\langle v, t, s \rangle$, whether the triple is poorly localized can be checked using eigenvalue-based techniques [15, 31]. Note that derivatives (with respect to time) will be high at the boundaries of time (i.e., the beginning and end of the time series). Similarly, in directed variate relationship graphs, source and sink nodes are likely to have large derivatives with respect to the relationship space. Since many of these triples at the boundary of

[5]The number of neighboring triples may be less than 78 if the triple is at the boundary in terms of time, scale, or variate relationship graph.

time and relationship do not correspond to real features of the data, but are essentially boundary *noise*s, such candidate triples are removed even if they are well-localized.

8 STEP 4: RMT FEATURE DESCRIPTOR CREATION

For data objects that can be represented as 2D matrices (such as images), [31] proposed that a gradient histogram based descriptor around the given point $\langle x, y\rangle$ on the matrix could be constructed by computing a gradient for each element in the neighborhood of the point [31]. The resulting gradients are then quantized into c orientations. Finally a $2a \times 2b$ grid is superimposed on the neighborhood region centered around the point and the gradients for the elements that fall into each cell are aggregated into a c-bin gradient histogram. This process leads to a feature descriptor vector of length $2a \times 2b \times c$. In [4], we have shown that gradient histograms (created from data vectors instead of data matrices) are also effective in describing temporal features of uni-variate time series. In the case of multi-variate time series, however, we cannot directly apply the above techniques. Instead, we first need to construct an *extractor matrix* to enable the gradient extraction process.

8.1 Extractor Matrix

Let $\mathfrak{Y}$ be a scale space defined over the given metadata-enriched multi-variate time series and the VTS triple, $\langle v, t, s\rangle$, be an RMT feature identified from $\mathfrak{Y}$. The multi-variate feature defined by a variate-time-scale triple, $\langle v, t, s\rangle$, has an associated scope, defined by the scale, s, in which it is identified. The pair, $\langle v, t\rangle$, forms the center of the feature in time and variates. Given this feature center, under scale, s, which corresponds to temporal and variate smoothing parameter pair, $\Sigma = \langle \sigma_{time}, \sigma_{var}\rangle$, the temporal and variate scopes of the feature are computed as described in Sections 4.1.1 and 4.2.2, respectively.

As we have also seen in Section 4, observations closer in time and relationships to the triple will have significantly larger contributions to the feature than the points closer to the boundaries of the scope. Therefore, to identify gradients across time and variate relationships, we first construct an N-step aggregation series:

Definition 8.1 (N-Step Aggregation Series). For directed variate relationships, we define the N-step aggregation series corresponding to scale s as follows: For $-N < a \leq N$,

$$W_s[a] \quad = \quad \begin{cases} \text{if } a > 0 & \left(F^a \mathbb{Y}_s\right) \\ \text{if } a = 0 & \mathbb{Y}_s \\ \text{if } a < 0 & \left(B^a \mathbb{Y}_s\right), \end{cases}$$

where, as before, F and B are two matrices describing forward and backward relationships among variates. Similarly, in the case of undirected variate relationships, we define the N-step aggregation series, such that for $0 \leq a \leq N$ we have

$$W_s[a] \quad = \quad \begin{cases} \text{if } a > 0 & \left(F^a \mathbb{Y}_s\right) \\ \text{if } a = 0 & \mathbb{Y}_s. \end{cases} \quad \diamond$$

Once the N-step aggregation series are obtained, we can then construct the extractor matrices from which the feature descriptors will be obtained:

Definition 8.2 (Extractor Matrix). Let $\langle v, t, s\rangle$ be a VTS triple on the scale space. In the case of directed variate relationships, we define the corresponding extractor matrix as a $2N \times 2M$ matrix, $X_{v,t,s}$, such that for $-N < a \leq N$ and $-M < b \leq M$, we have $X_{v,t,s}[a, b] = \Big(W_s[a]\Big)[v, t + b]$. In the case of undirected variate relationships, we define the extractor matrix as a $(N + 1) \times 2M$

matrix, $X_{v,t,s}$, such that for $-N < a \leq N$ and $0 \leq b \leq M$, we have $X_{v,t,s}[a,b] = \Big(W_s[a]\Big)[v, t+b]$. ◇

The values of N and M should be selected to roughly cover the scope of the feature.

8.2 Descriptor Extraction

Given this extractor matrix, $X_{v,t,s}$, the feature descriptor is created as a c-directional gradient histogram of this matrix, sampling the gradient magnitudes around the salient point using a $2a \times 2b$ grid (or $2a \times b$ grid for undirected relationship graphs) superimposed on the matrix, $X_{v,t,s}$. To give less emphasis to gradients that are far from the point $\langle v, t \rangle$, a Gaussian weighting function is used to reduce the magnitude of elements further from $\langle v, t \rangle$.

This process leads to a feature descriptor vector of length $2a \times 2b \times c$ (or $2a \times b \times c$ for undirected graphs). The descriptor size must be selected in a way that reflects the temporal characteristics of the time series; if a multi-variate time series contains many similar features, it might be more advantageous to use large descriptors that can better discriminate: these large descriptors would not only include information that describe the corresponding features, but would also describe the temporal contexts in which these features are located.

9 RMT FEATURE SET OF A MULTI-VARIATE TIME SERIES

Given the above, the RMT features of a metadata-enriched multi-variate (MM) time series, $\mathbf{Y} = (\mathcal{V}, \mathcal{M}, \mathbb{Y}, \mathcal{D})$, with respect to the parameters

- $\Sigma_0 = \langle \sigma_{time,0}, \sigma_{var,0} \rangle$; i.e., the smallest temporal and variate smoothing parameters,
- $\mathcal{L} = \langle l_{time}, l_{var} \rangle$; i.e., the number of temporal and variate smoothing layers, and
- $\mathcal{K} = \langle k_{time,}, k_{var} \rangle$; i.e., the scaling parameters for temporal and variate smoothings,

is defined as a set, $\mathcal{F}$, where each feature, $f \in \mathcal{F}$, extracted from $\mathbf{Y}$, is a pair of the form, $f = \langle pos, \vec{d} \rangle$:

- $pos = \langle v, t, s \rangle$ is a VTS triple denoting the position of the feature in the scale-space of the multi-variate time series, where v is the index of the variate at which the feature is *centered*, t is the time instant around which the duration of the feature is *centered*, and s is the temporal/variate smoothing scale in which the feature is identified. Note that this triple also defines the *temporal and variates scopes* of the RMT feature.
- $\vec{d}$ is a vector of length $2a \times 2b \times c$ for directed relationship graphs and $2a \times b \times c$ for undirected graphs, as described in the previous section.

Note that this set contains RMT features of potentially different sizes. In particular, we have $\sigma_{time,min} = \sigma_{time,0}$, $\sigma_{var,min} = \sigma_{var,0}$, $\sigma_{time,max} = \sigma_{time,0} \times k_{time}^{l_{time}}$, $\sigma_{var,max} = \sigma_{var,0} \times k_{var}^{l_{var}}$, and these define the minimum and maximum temporal and variate scopes of the features identified from the given multi-variate time series.

10 TIME SERIES MATCHING USING RMT FEATURES

This feature set can be used for various applications, including alignment, indexing, and classification of multi-variate series. Let us be given two metadata-enriched multi-variate (MM) time series, $\mathbf{Y}_1$ and $\mathbf{Y}_2$, and their feature sets $\mathcal{F}_1$ and $\mathcal{F}_2$. We rely on the alignments of the feature pairs in $\mathcal{F}_1$ and $\mathcal{F}_2$ to measure how well these two series match each other.

10.1 Alignment of Feature Pairs

Let $f_1 = \langle\langle v_1, t_1, s_1\rangle, \vec{d}_1\rangle$ and $f_2 = \langle\langle v_2, t_2, s_2\rangle, \vec{d}_2\rangle$ be two RMT features in $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. When matching f_1 and f_2, we consider how well aligned as well as how important these two features are.

10.1.1 Temporal Alignment of a Pair of Features. Two features are said to be temporally aligned if their temporal scopes overlap significantly and temporal centers are close.

Definition 10.1 (Temporal Overlap). Let $[ts_1, te_1)$ denote the temporal scope of the first feature defined by t_1 and the temporal smoothing parameter corresponding to the feature scale s_1. Similarly, let $[ts_2, te_2)$ denote the temporal scope of the first feature defined by t_2 and the feature scale s_2. We define the temporal overlap score of the two features as $Overlap_T(f_1, f_2) = \frac{min(te_1, te_2) - max(ts_1, ts_2)}{max(te_1, te_2) - min(ts_1, ts_2)}$. ◇

Definition 10.2 (Temporal Center Proximity). We define the temporal proximity score as $Prox_T(f_1, f_2) = 1 - \frac{|t_1 - t_2|}{maxLength}$, where $maxLength$ is the length of the time series. ◇

Given these, we define temporal alignment score as follows:

Definition 10.3 (Temporal Alignment). We define the temporal alignment score of the two features as $Align_T(f_1, f_2) = \frac{Overlap_T(f_1, f_2) + Prox_T(f_1, f_2)}{2}$. ◇

10.1.2 Variate Alignment of a Pair of Features. Two features are said to be variate aligned if their variate scopes overlap significantly:

Definition 10.4 (Variate Alignment). Let $scope(v_1, \sigma_{var,1})$ denote the variate scope of the first feature defined by parameter $\sigma_{var,1}$ corresponding to feature scale s_1. Similarly, let $scope(v_2, \sigma_{var,2})$ be the variate scope of the second feature. We define the variate alignment score of the two features as $Align_V(f_1, f_2) = \frac{scope(v_1, \sigma_{var,1}) \cap scope(v_2, \sigma_{var,2})}{scope(v_1, \sigma_{var,1}) \cup scope(v_2, \sigma_{var,2})}$. ◇

10.1.3 Descriptor Alignment of a Pair of Features. Two features are said to be descriptor aligned if their descriptor vectors are similar to each other:

Definition 10.5 (Descriptor Alignment). We define the descriptor alignment score of the two features as $Align_D(f_1, f_2) = sim(\vec{d}_1, \vec{d}_2)$ or as $Align_D(f_1, f_2) = \left(1 + \Delta(\vec{d}_1, \vec{d}_2)\right)^{-1}$ for a given similarity, $sim()$, or distance, $\Delta()$, function. ◇

10.1.4 Amplitude Alignment of a Pair of Features. Two features are said to be amplitude aligned if the average amplitudes of the time series within the corresponding feature scopes are similar to each other:

Definition 10.6 (Amplitude Alignment). We define the amplitude alignment score as $Align_A(f_1, f_2) = \left(1 + |ampl_1 - ampl_2|\right)^{-1}$, where $ampl_1$ and $ampl_2$ are the average amplitudes of the time series, Y_1 and Y_2, within the scopes of f_1 and f_2. ◇

10.2 Feature Significance

10.2.1 Scope Significance of a Given Pair of Features. The size of the temporal and/or variate scopes may impact the significance of a feature.

Definition 10.7 (Temporal Scope Significance). The combined temporal scope significance of f_1 and f_2 is defined as $Sig_T(f_1, f_2) = \frac{\sigma_{time,1} + \sigma_{time,2}}{2}$, where $\sigma_{time,1}$ and $\sigma_{time,2}$ are the two temporal smoothing parameters corresponding to temporal scales, s_1 and s_2. ◇

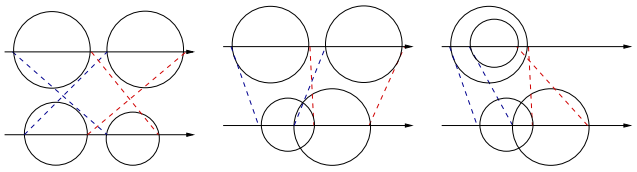


Fig. 6. Example scope boundary conflicts: blue lines mark corresponding starting points of the matching scopes, whereas red lines mark the corresponding end points

Definition 10.8 (Variate Scope Significance). The combined variate scope significance of f_1 and f_2 is defined as $Sig_V(f_1, f_2) = \frac{\sigma_{var,1}+\sigma_{var,2}}{2}$, where $\sigma_{var,1}$ and $\sigma_{var,2}$ are the two variate smoothing parameters corresponding to feature scales, s_1 and s_2. ◇

10.2.2 Contextual Significance of Features. In many applications, we also need to consider how discriminating or contextually important a feature is. For example, in a classification scenario, when comparing features f_1 and f_2 from these time series, we may also consider how representative (frequent and discriminating) f_1 and f_2 are as the contextual importance measure, $Sig_C(f_1, f_2)$.

10.3 Overall Feature Matching Score

Given the above, the overall matching score of two features is a combination of the individual measures of alignment and importance:

Definition 10.9 (Overall Feature Matching Score). We define the overall matching score, $match(f_1, f_2)$, of the two features as

$$\mu\left(\begin{array}{l} Align_D(f_1, f_2), Align_T(f_1, f_2), Align_V(f_1, f_2), Align_A(f_1, f_2), \\ Sig_T(f_1, f_2), Sig_V(f_1, f_2), Sig_C(f_1, f_2) \end{array}\right),$$

where μ is a merge function that combines the individual scores. ◇

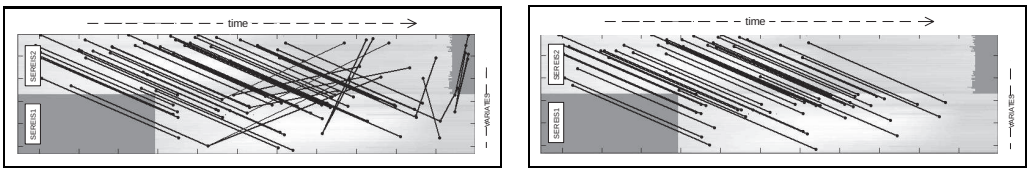
While there exist different merge functions (such as *min*, *max*, *sum*, *avg*, *product*), in the experiments reported in Section 11 we use *product*, which approximates the boolean operator *and* when individual scores are zeros and ones [5].

10.4 Identifying Candidate Matching Pairs

Given a query time series, $\mathbf{Y}_q$, and a data series, $\mathbf{Y}_d$, and their feature sets $\mathcal{F}_q$ and $\mathcal{F}_d$, the next step is to identify a set $\mathcal{P} \subseteq \mathcal{F}_q \times \mathcal{F}_d$ of candidate feature pairs, such that

- $\forall f_{q,i} \in \mathcal{F}_q \; \exists f_{d,j} \in \mathcal{F}_d \;\; s.t. \;\; \langle f_{q,i}, f_{d,j}\rangle \in \mathcal{P}$ (i.e., for each query RMT feature on the query object, at least one matching feature on the data object is located),
- $\forall \langle f_{q,i}, f_{d,j}\rangle, \langle f_{q,h}, f_{d,k}\rangle \in \mathcal{P} \;\; (f_{q,i} = f_{q,h}) \rightarrow (f_{d,j} = f_{d,k})$ (i.e., for each query RMT feature on the query object, at most one matching feature on the data object is located), and
- $\sum_{\langle f_{q,i}, f_{d,j}\rangle \in \mathcal{P}} match(f_{q,i}, f_{d,j})$ is maximized.

It is easy to see that, since, for each query feature, $\mathcal{P}$ contains one and only one matching data feature and since we aim to maximize the overall matching score, the set $\mathcal{P}$ can be obtained by considering each feature $f_{q,i} \in \mathcal{F}_q$ and selecting the feature $f_{d,j} \in \mathcal{F}_d$ with the maximum $match(f_{q,i}, f_{d,j})$ value. However, the feature pairs in $\mathcal{P}$ obtained this way may not be mutually consistent and such inconsistencies need to be eliminated.



(a) Pairs of matching RMT features (b) Remaining pairs of matches

Fig. 7. (a) Candidate RMT feature pairs for two multi-variate time series, and (b) the remaining subset of matching RMT feature pairs after inconsistency pruning

10.5 Inconsistency Pruning of Candidate Pairs

Intuitively, we call a set of feature matchings *temporally consistent* if the corresponding features are similarly ordered in both time series. Figure 6 shows several temporal inconsistencies, where temporal scope boundaries of matching features are not similarly ordered in two time series. Formally, we define temporal consistency as follows:

Definition 10.10 (Temporal Consistency). Let us be given two metadata-enriched multi-variate (MM) time series, $\mathbf{Y}_1$ and $\mathbf{Y}_2$, and their feature sets $\mathcal{F}_1$ and $\mathcal{F}_2$. Let $p_1 = \langle f_1^1, f_2^1 \rangle, p_2 \in \langle f_1^2, f_2^2 \rangle \in \mathcal{P}$ be two candidate feature pairs and $bounds_b^a = \{ts_b^a, te_b^a\}$ be the start and end points of the temporal scope of feature f_b^a for $a, b \in \{1, 2\}$. We call p_1 and p_2 *temporally consistent* if and only if

- for all pairs of end points $t_i^1, t_j^1 \in bounds_1^1 \cup bounds_2^1$ in the first pair, we have

$$((t_i^1 > t_j^1) \rightarrow (t_i^2 \not< t_j^2)) \wedge ((t_i^1 < t_j^1) \rightarrow (t_i^2 \not> t_j^2)),$$

  where $t_i^2, t_j^2 \in bounds_1^2 \cup bounds_2^2$ are the two end points in the second pair corresponding to t_i^1 and t_j^1; and
- for all pairs of end points $t_i^2, t_j^2 \in bounds_1^2 \cup bounds_2^2$ in the second pair, we have

$$((t_i^2 > t_j^2) \rightarrow (t_i^1 \not< t_j^1)) \wedge ((t_i^2 < t_j^2) \rightarrow (t_i^1 \not> t_j^1)),$$

  where $t_i^1, t_j^1 \in bounds_1^1 \cup bounds_2^1$ are the two end points in the first pair corresponding to t_i^2 and t_j^2. ◇

Figure 7 provides an example with inconsistent matches: here we see that the matching process identified some very distant pairs of RMT features as matches. Note also that there are many matching pairs that cross each other in time, implying temporal features that are differently ordered in time in two time series. To improve the accuracy of the matching process, we need to eliminate such inconsistencies. The outline of the process to eliminate inconsistencies is as follows

(1) For each pair, $\langle f_1, f_2 \rangle \in \mathcal{P}$ of matching features, we compute a dominance score, $dom(f_1, f_2)$, as

$$\rho \begin{pmatrix} Align_D(f_1, f_2), Align_T(f_1, f_2), Align_V(f_1, f_2), Align_A(f_1, f_2), \\ Sig_T(f_1, f_2), Sig_V(f_1, f_2), Sig_C(f_1, f_2) \end{pmatrix}.$$

Note that this dominance score may, but is not required to, be the same as the overall matching score discussed in Section 10.3.

(2) We next initialize an empty set ($\mathcal{R}$) to collect the committed consistent feature pairs and two empty lists ($list_1$ and $list_2$) to keep track of their temporal scopes: i.e., we set $\mathcal{R} = \emptyset$, $list_1 = \perp$, and $list_2 = \perp$.

(3) Next, we consider all pairs of matching features in $\mathcal{P}$ in descending order of their dominance scores. Let $\langle f_1, f_2 \rangle \in \mathcal{P}$ be the pair we are currently considering.

(a) *Temporal consistency verification:* Let $\langle ts_1, te_1 \rangle$ and $\langle ts_2, te_2 \rangle$ be the temporal scopes of f_1 and f_2, respectively
  (i) We *attempt* to insert the ts_1 and te_1 into $list_1$ ordered in increasing order of time; similarly we *attempt* to insert ts_2 and te_2 into the list, $list_2$, also ordered in increasing order of time.
  (ii) Let $rank(ts_1)$, $rank(ts_2)$, $rank(te_1)$, and $rank(te_2)$ be the corresponding ranks of the time points in their respective time ordered lists.
  (iii) If $rank(ts_1) = rank(ts_2)$ and $rank(te_1) = rank(te_2)$, then we confirm the insertion and we keep the pair[6].
  (iv) Else, we drop the pair $\langle f_1, f_2 \rangle$ and eliminate the corresponding scope boundaries from the lists $list_1$ and $list_2$.

(b) If the candidate pair $\langle f_1, f_2 \rangle$ has not been dropped due to temporal inconsistency, then insert the pair in $\mathcal{R}$: i.e., $\mathcal{R} \rightarrow \mathcal{R} \cup \{\langle f_1, f_2 \rangle\}$.

Note that the reason why the feature pairs are considered in descending order of dominance scores is that, when an inconsistency is identified, the most recently considered pair –which is less dominant (relatively less aligned, smaller, and less similar) –can be eliminated without affecting the already committed boundaries.

10.6 RMT-Based Multi-variate Time Series Matching Score

Given two metadata-enriched multi-variate (MM) time series, $\mathbf{Y}_1$ and $\mathbf{Y}_2$, and their feature sets $\mathcal{F}_1$ and $\mathcal{F}_2$, the above process results in a set $\mathcal{R}(\mathcal{F}_1, \mathcal{F}_2) = \{\langle f_{1,i}, f_{2,i} \rangle\}$, where $f_{1,i} \in \mathcal{F}_1$ and $f_{2,i} \in \mathcal{F}_2$, respectively. We define the overall matching score, $score(\mathbf{Y}_1, \mathbf{Y}_2)$, of the two multi-variate series, using this set of matching feature pairs:

$$\sum_{\langle f_1, f_2 \rangle \in \mathcal{R}(\mathcal{F}_1, \mathcal{F}_2)} \phi \left(\begin{array}{l} Align_D(f_1, f_2), Align_T(f_1, f_2), Align_V(f_1, f_2), Align_A(f_1, f_2), \\ Sig_T(f_1, f_2), Sig_V(f_1, f_2), Sig_C(f_1, f_2) \end{array} \right),$$

where ϕ is a combined scoring function.

While there exist different merge functions (such as *min*, *max*, *sum*, *avg*, *product*), in the experiments reported in Section 11 we use *product*, which approximates the boolean operator *and* when individual scores are zeros and ones [5].

11 EVALUATION

In this section, we present experiment results that assess the efficiency and effectiveness of the *robust multi-variate temporal* (RMT[7]) feature extraction algorithms. In our preliminary work [51], we had shown that the *diagonal* scale-space based RMT features (Section 5.1) are more effective in partial time series search and classification tasks than alternative techniques, including **SVD,** where we created a single fingerprint for each multi-variate time series using the SVD transformation; and **DTW,** where distances were computed directly using dynamic time warping [9]. In the appendix, we also consider **SAX**[28] **DTW**, which provides time savings over DTW, possibly at the expense of accuracy. Therefore, instead of replicating the experiments reported in [51], we focus on the impact of *full* scale-space based RMT (Section 5.2) features with respect to the use of *diagonal* scale space based RMT (Section 5.1) and also investigate the impacts of the alternative matching and inconsistency removal strategies described in Section 10, within the context of a motion recognition task.

[6]The process is slightly more complex in that there can be exceptions where the ranks are different, but time values are the same. We also confirm the insertion in these special cases.

[7]RMT source code is available at [13].

Table 1. (a) Default configuration and (b) alternative matching/pruning strategies

(a) Default configuration

RMT	
# iterations, L	6
# of octaves, o	3
initial smoothing for time, $\sigma_{time,0}$	2.8
initial smoothing for relationships,$\sigma_{var,0}$	0.5
candidate pruning threshold, ω_τ	10
descriptor size, $2a \times 2b \times c$	$(4 \times 4 \times 8 =)$ 128
relationship reduction algorithm	k-means
SVD	
degree of energy preservation	95%

(b) Matching/pruning strategies

TO	Temporal overlap (Definition 10.1)
TP	Temporal proximity (Definition 10.2)
TA	Temporal alignment (Definition 10.3)
VA	Variate alignment (Definition 10.4)
DA	Descriptor alignment (Definition 10.5)
AA	Amplitude alignment (Definition 10.6)
TS	Temporal scope significance (Definition 10.7)
VS	Variate scope significance (Definition 10.7)

11.1 Settings

11.1.1 Hardware/Software. In order to ensure results are comparable to those reported in [51], all experiments were run on the identical set up, including 4-core Intel Core i5-2400 3.10GHz machines with 8GB memory, running 64-bit Windows 7 Enterprise, using Matlab.

11.1.2 Data Set. For the experiments in this section, we use the Mocap time series data set [34]: The data set consists of movement records from markers placed on subjects' bodies as they perform 8 types of tasks. We use ASF/AMC format where the original coordinate readings are converted into 62 joint angles data. We treat readings for each joint angle as a different uni-variate time series. The hierarchical spatial distribution (e.g. left foot, right foot, left leg, etc.) of the joint angles on the body is used to create the underlying correlation matrix used as metadata[8].

We consider additional data sets in the online appendix.

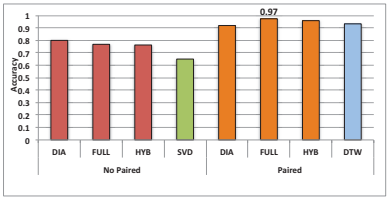
11.1.3 Evaluation Metrics. For evaluating accuracy, we use take-one-out methodology with the following criteria: *(a) top-5 precision:* the number of series, among the nearest 5 results, that are of the same class of movement as the query series, and *(b) top-$\|c\|$ precision:* the number of series, among the nearest $\|c\|$ results (where $\|c\|$ is the size of the movement class containing the query series) that are of the same class as the query series. The first measure reflects how effective a particular approach is for nearest-neighbor classification, whereas the second measure reflects how well defined the classes.

In addition, we also report pairwise matching times for the alternative approaches. Note that since we are using top-5 and top-$\|c\|$ classification, the classification time is a function of the value of $\|c\|$, the number of labeled data in the training data set, and the pairwise matching time. To ensure that the efficiency different algorithms can be compared independently of the value of $\|c\|$ and the training data set, in the paper, we report the pairwise matching time as an indicator of the classification cost.

11.1.4 Alternative RMT Features. We consider different types of RMT features:

- *diagonal scale-space based RMT (DIA):* This is the version of the RMT features studied in our prior work. As described in Section 5.1, these features are extracted only by considering the diagonal scales of the scale space; in other words, the features' temporal and variate scopes grow in synch to each other.
- *full scale-space based RMT (FULL):* This is the version of the RMT features proposed in this paper. As described in Section 5.2, these are extracted by considering all scales of the scale

[8]Note that this provides an *intentionally rough* metadata, enabling us to observe accuracy of RMT features under imperfect domain knowledge



(a) Average top-5 precision (%)

Average Top-5 Precision (%)					
		Non-paired		Paired	
Class	num	RMT	SVD [51]	RMT	DTW [51]
climb	18	58.9	52.2	85.6	68.9
dribble	14	32.9	28.6	87.1	84.3
jumping	30	100	82.0	100	100
running	19	100	100	100	100
salsa	30	50.0	59.3	100	87.1
soccer	6	43.3	30.0	93.3	96.7
walk	36	100	89.4	100	100
walk (uneven)	76.1	100	58.7	100	98.7
Average	184	**76.9**	69.0	**97.4**	93.3
Confidence		**72.8-**	65.2-	**96.5-**	91.7-
Interval		**80.8**%	72.8%	**98.3**%	94.9%

(b) Per-class top-5 precision

Average Pairwise Matching Time			
Non-paired		Paired	
RMT	SVD [51]	RMT	DTW [51]
0.18s	0.003s	0.19s	0.38s

(c) Matching time (in seconds)

Fig. 8. Top-5 matching accuracy and matching time – default configuration: descriptor alignment (DA) based feature matching and DA based inconsistency pruning and overall score computation

space; consequently, the features' temporal and variate scopes grow independently from each other, enabling heterogeneously shaped features.

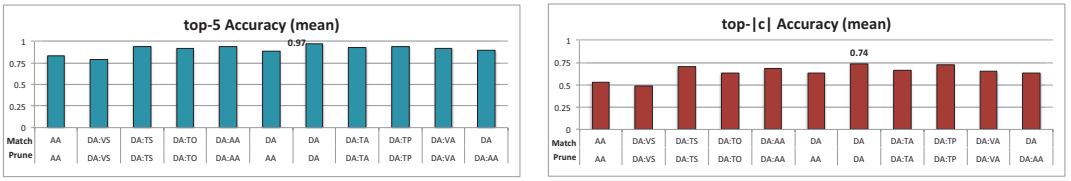
- *hybrid RMT (HYB):* We also consider hybrid feature sets, where diagonal scale-space features and full scale-space features are combined. Note that due to the feature candidate elimination strategy described in Section 7, feature set obtained using the full scale-space is not necessarily the superset of the features obtained using the diagonal scale-space. This hybrid strategy re-introduces the diagonal scale-space features which may have been eliminated due to some features in the non-diagonal scales of the space.
- *diagonal scale-space based RMT - alt. 2 (DIA2):* Note that diagonal scale-space based RMT features can be obtained either by using only the diagonal scales of the scale-space as described in Section 5.1, or can be obtained by selecting the subset of the full scale-space based RMT features such that the temporal and variate scales are the same. We refer to this second alternative as DIA2.

Table 1(a) provides the outline of the default parameter configuration and describes how these parameters are varied in the experiments.

11.1.5 Alternative Alignment Strategies. In this section, we experiment with the various temporal and variate alignment metrics presented in Section 10.1 and listed in Table 1(b). When needed, for combining these measures in Table 1(b), we use multiplication as the merge function. In addition, we consider two alignment strategies: *(a) all octaves alignment (AoA):* Under this strategy, any two pair of features can be considered for alignment irrespective of their scales. *(b) same octave alignment (SoA):* Under this strategy, only those pairs of features that have the same time and variate octaves are considered for alignment.

11.1.6 Alternative Inconsistency Elimination (Pruning) Strategies. In this section, we also consider the impact of the measures presented in Table 1 on inconsistency elimination process (Section 10.5). In addition, we consider two pruning strategies: *(a) all octaves pruning (AoP):* Under this strategy, any two pairs of features can be considered inconsistent irrespective of their scales. *(b) same octave pruning (SoP):* Under this strategy, only those pairs of features that have the same time and variate octaves can be considered inconsistent.

11.2 Discussion of the Results



(a) top-5 precision (b) top-||c|| precision

Fig. 9. The impact of the alternative feature matching and inconsistency pruning strategies – full (FULL) feature set, same octave alignment (SoA), and same octave pruning (SoP)

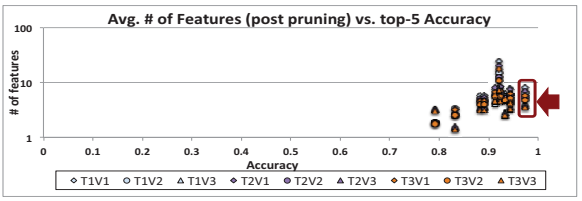


Fig. 10. Average number of feature pairs for the alternatives considered in Figure 9. Here T*i*V*j* refers to query features (remaining after inconsistency pruning) that are of time octave *i* and variate octave *j*

11.2.1 Overview. Figure 8(a) compares the classification accuracy of RMT using the 8 classes with 184 motions in the Mocap data set against the alternative approaches reported in [51]:

- *variate-paired* alignment: This is the default configuration where we assume that the pairing of the variates in the query and in the database are known in advance. DTW requires that this pairing is known. In the case of RMT, we leverage the pairing information by ignoring feature matches during the feature alignment phase unless at least 50% of the variates are common. As we see in Figure 8, paired RMT provides the best overall accuracy.
- non-*variate-paired* alignment: Both SVD and RMT can operate without requiring pairing of the variates. Given two multi-variate time series, SVD uses the decomposed series rather than the series themselves, thus it does not require the series to be variate paired. Similarly, RMT can be implemented in such a way that variate alignments are completely ignored during the matching phase. As we see in Figure 8, non-paired RMT works better than SVD – and thus is applicable when pairing information is not available. While SVD supports fast matching, the accuracy is significantly lower to render it a feasible approach.

Note that Figure 8(b) also includes confidence intervals for the accuracies of various techniques. As we see here, RMT's confidence intervals do not overlap with the other techniques' accuracy confidence intervals, providing additional evidence for the advantage of using RMT features. Moreover, the confidence intervals of RMT are significantly tighter than the confidence intervals of other techniques, again providing evidence that RMT is more robust than the other approaches.

Figure 8(a) shows that, as expected, we obtain highest accuracy when we consider the full scale space. It is also important to note that in these experiments we have not leverage RMT feature significance (FS) to boost matching accuracy. Unlike DTW, RMT can further boost accuracy through relevance feeback and other (semi-)supervised learniing techniques.

11.2.2 Impact of the Alternative Feature Matching and Inconsistency Pruning Strategies. As we have seen in Section 10 and Table 1, one can use several strategies to match features across multivariate time series and prune inconsistencies. For the results above, as the default configuration,

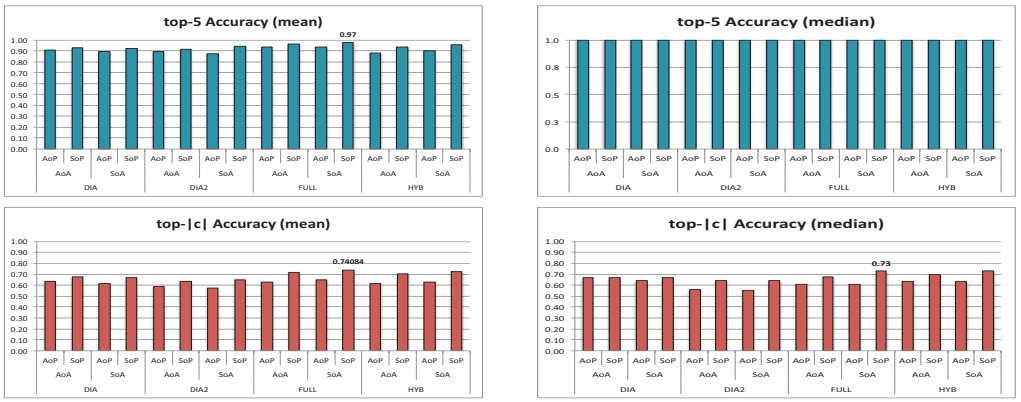


Fig. 11. Impact of feature discovery and octave management – using descriptor alignment (DA) for feature matching, inconsistency pruning, and overall scoring

Table 2. # feature pairs before and after inconsistency pruning for the optimal configuration in Figure 11

	Before inconsistency pruning	After inconsistency pruning
DIA	952.2	21.5
FULL	1252.1	48.2
HYB	2204.3	56.3

we considered descriptor alignment (DA) based feature matching, inconsistency, pruning and overall score computation. While the best strategy is application dependent, as we see in Figure 9, in this application, RMT is able to achieve high accuracy by using descriptor alignment (DA) for feature matching, inconsistency pruning, and overall score computation. The result also shows that considering additional criteria, such as temporal or variate alignment, is not necessary (and can, in fact, be harmful) in this particular application.

This shows that the RMT feature descriptors are highly informative. This is further confirmed by Figure 10, where we see the average number of (post-pruning) matching feature pairs for the alternative strategies considered in Figure 9. As we see in this Figure, a higher number of matching feature pairs does not translate into more accurate matches. This indicates that the resulting RMT features are highly informative and a small number of feature pairs at different scales and shapes are sufficient to characterize different types of motion.

Impact of the Feature Discovery and Octave Management Strategies. For the default results presented above, we leveraged full (FULL) feature set with same octave alignment (AoA) and same octave pruning (SoP) strategies. In Figure 11, we study the impact of these strategies in further detail. As we see in this figure, the default configuation indeed leads to highest accuracy: Firstly, as expected, feature matches and inconsistencies need to be considered at each octave scale separately. Secondly, the figure shows that the full scale space provide more information than the diagonal scale space – in fact, extending the FULL feature set with diagonal features (i.e., using the HYB strategy) does not lead to any better results than just using the FULL or DIA feature sets.

This is further studied in Table 2, which shows the average matching # feature pairs, before and after inconsistency pruning, for the optimal configuration in Figure 11. As we see here, inconsistency pruning eliminates a large number of feature pairs. We also see that the hybrid option (HYB), has more feature pairs than both DIA and FULL, but, as we have seen Figure 11, these additional feature pairs do not contribute to the accuracy.

12 CONCLUSIONS

Many time series data sets are (a) multi-variate, (b) interrelated, and (c) multi-resolution. These include motion and gesture data, as described in this paper, as well as data from other domains: In this paper, we presented a metadata-enriched multi-variate time series model, in which a dependency/correlation model relates the individual variates to each other. Recognizing that multi-variate temporal features can be extracted more effectively by simultaneously considering, at multiple scales, differences among individual variates along with the dependency/correlation model that relates them, we further developed algorithms to detect robust multi-variate temporal (RMT) features that are multi-resolution, local, and invariant against various types of noise. Experiments using human motion data, where labeled ground truth is available, confirmed that the RMT features are highly effective in multi-variate series search and classification.

REFERENCES

[1] I. Batal, D. Fradkin, J. Harrison, F. Moerchen, and M. Hauskrecht. Mining recent temporal patterns for event detection in multivariate time series data. In *KDD'12*, pages 280–288, 2012.

[2] D. J. Bemdt and J. Clifford. Using dynamic time warping to find patterns in time series, 1994.

[3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML'06*, pages 113–120, 2006.

[4] K. S. Candan, R. Rossini, M. L. Sapino, and X. Wang. sdtw: Computing dtw distances using locally relevant constraints based on salient feature alignments. *PVLDB*, 5(11):1519–1530, July 2012.

[5] K. S. Candan and M. L. Sapino. *Data Management for Multimedia Retrieval.* Cambridge University Press, New York, NY, USA, may 2010. ISBN-10:0521887399, ISBN-13: 978-0521887397, May 31, 2010.

[6] N. Castro and P. Azevedo. Multiresolution Motif Discovery in Time Series. In *Proceedings of the SIAM International Conference on Data Mining.*, SDM '10, pages 665–676, Columbus, Ohio, USA, 2010. SIAM.

[7] L. Chen. *Similarity Search over Time Series and Trajectory Data.* PhD thesis, University of Waterloo, 2005.

[8] L. Chen and R. Ng. On the marriage of lp-norms and edit distance. In *VLDB*, 2004.

[9] 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/.

[10] F. Chung, T. Fu, R. Luk, and V. Ng. Flexible time series pattern matching based on perceptually important points. *IJCAI Workshop on Learning from Temporal and Spatial Data*, pages 1–7, 2001.

[11] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: Experimental comparison of representations and distance measures. *VLDB*, pages 1542–1552, 2008.

[12] M. Eichler. Granger causality and path diagrams for multivariate time series. *Journal of Econometrics*, 2006.

[13] EmitLab-ASU. Rmt code, 2017. Available upon request.

[14] P. Esling and C. Agon. Time-series data mining. *ACM Comput. Surv.*, 45(1):12:1–12:34, Dec. 2012.

[15] C. Harris and M. Stephens. A combined corner and edge detector. *Fourth Alvey Vision Conference*, pages 147–151, 1988.

[16] A. Harvey and S. Koopman. Multivariate structural time series model. In *System Dynamics in Economic and Financial Models*, pages 269–296. John Wiley and Sons, 1997.

[17] A. S. Hopkins, A. Lekov, J. Lutz, G. Rosenquist, and L. Gu. Simulating a nationally representative housing sample using energyplus. page 55, 2011.

[18] X. Ji, J. Bailey, and G. Dong. Mining minimal distinguishing subsequence patterns with gap constraints. In *KAIS*, 2007.

[19] Y. Jin and B. Prabhakaran. Knowledge discovery from 3d human motion streams through semantic dimensional reduction. *ACM Transactions on Multimedia Computing, Communications and Applications*, 7(2), 2 2011.

[20] E. Keogh. Exact indexing of dynamic time warping. In *VLDB*, pages 406–417, 2002.

[21] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *KAIS*, 2005.

[22] D. Kim and B. Prabhakaran. Motion fault detection and isolation in body sensor networks. *Pervasive and Mobile Computing*, 7(6):727–745, 2011.

[23] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.

[24] J. B. Kruskal. An overview of sequence comparison: Time warps, string edits, and macromolecules. *SIAM Review*, 25(2):201–237, 1983.

[25] W. Krzanowski. Between-groups comparison of principal components. *Journal of the American Statistical Assoc.*, 1979.

[26] C. Li, S. Q. Zheng, and B. Prabhakaran. Segmentation and recognition of motion streams by similarity search. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(3), Aug. 2007.

[27] L. Li, B. A. Prakash, and C. Faloutsos. Parsimonious linear fingerprinting for time series. *PVLDB*, 2010.

[28] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, pages 2–11. ACM, 2003.

[29] S. Liu, Y. Garg, K. S. Candan, M. L. Sapino, and G. Chowell. Notes2: Networks-of-traces for epidemic spread simulations. In *AAAI International Workshop on Computational Sustainability (co-located with AAAI15)*, 2015.

[30] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, ICCV '99, 1999.

[31] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[32] S. Mehta, R. Nallusamy, R. V. Marawar, and B. Prabhakaran. A study of DWT and SVD based watermarking algorithms for patient privacy in medical images. In *ICHI'13*, pages 287–296, 2013.

[33] T. C. Mills. *Time Series Techniques for Economists.* Cambridge University Press, 1990.

[34] Mocap. Cmu mocap data set, 2001. http://mocap.cs.cmu.edu/.

[35] Y. Mohammad and T. Nishida. Constrained motif discovery in time series. *New Generation Computing*, 27(4):319–346, 2009.

[36] F. Mörchen. Time series feature extraction for data mining using dwt and dft. 2003.

[37] J. Peng, H. Wang, J. Li, and H. Gao. Set-based similarity search for time series. In F. Özcan, G. Koutrika, and S. Madden, editors, *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 2039–2052. ACM, 2016.

[38] C. S. Perng, H. Wang, S. R. Zhang, and D. S. P. Jr. Landmarks: a new model for similarity-based pattern querying in time series databases. ICDE'00, pages 33–42, 2000.

[39] S. R. Poccia and Y. Garg. On the effectiveness of distance measures for similarity search in multi-variate sensory data: Effectiveness of distance measures for similarity search. In *ICMR'17*, pages 489–493, 2017.

[40] S. R. Poccia, M. L. Sapino, X. C. Sicong Liu, Y. Garg, S. Huang, J. H. Kim, X. Li, P. Nagarkar, and K. S. Candan. SIMDMS: Data management and analysis to support decision making through large simulation ensembles. In *EDBT'17*, pages 582–585, 2017.

[41] 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*, 2012.

[42] T. Rakthanmanon and E. Keogh. Fast-shapelets: A scalable algorithm for discovering time series shapelets. In *SDM*, 2013.

[43] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. In *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 1978.

[44] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* 1983.

[45] P. Sanguansat. Multiple multidimensional sequence alignment using generalized dynamic time warping. 8(10):668–678, 2012.

[46] L. Shuai, C. Li, X. Guo, B. Prabhakaran, and J. Chai. Motion capture with ellipsoidal skeleton using multiple depth cameras. *IEEE Trans. Vis. Comput. Graph.*, 23(2):1085–1098, 2017.

[47] A. Silva, R.J.Hyndman, and R.Snyder. The vector innovations structural time series framework: A simple approach to multivariate forecasting. *Statistical Modelling*, 10(4):353–374, 2010.

[48] U. Vespier, A. Knobbe, S. Nijssen, and J. Vanschoren. *MDL-Based Analysis of Time Series at Multiple Time-Scales*, pages 371–386. ECML-PKDD'12. 2012.

[49] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing multidimensional time-series. *The VLDB Journal*, 15(1):1–20, Jan. 2006.

[50] X. Wang and K. S. Candan. Relevant shape contour snippet extraction with metadata supported hidden markov models. CIVR '10, pages 430–437, 2010.

[51] X. Wang, K. S. Candan, and M. L. Sapino. Leveraging metadata for identifying local, robust multi-variate temporal (rmt) features. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 388–399. IEEE, 2014.

[52] X. Wang, J. Lin, P. Senin, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, and S. Frankenstein. RPM: representative pattern mining for efficient time series classification. In *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016.*, pages 185–196, 2016.

[53] K. Yang and C. Shahabi. A pca-based similarity measure for multivariate time series. In *MMDB*, pages 65–74. ACM, 2004.

[54] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan. Detecting time series motifs under uniform scaling. In *KDD'07*, pages 844–853. ACM, 2007.

[55] L. Ye and E. Keogh. Time series shapelets: A new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 947–956, 2009.

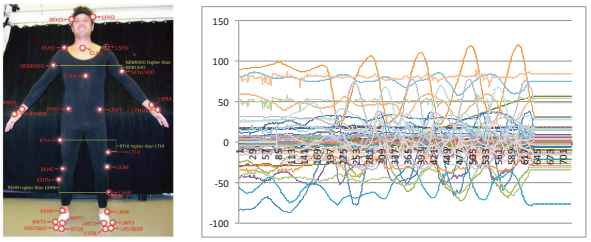


Fig. 12. A multi-variate time series capturing body movement: the structure of the human body relates the positions of the body sensors during the motion capture [34]

ONLINE APPENDIX

A- Experiments with Additional Data Sets

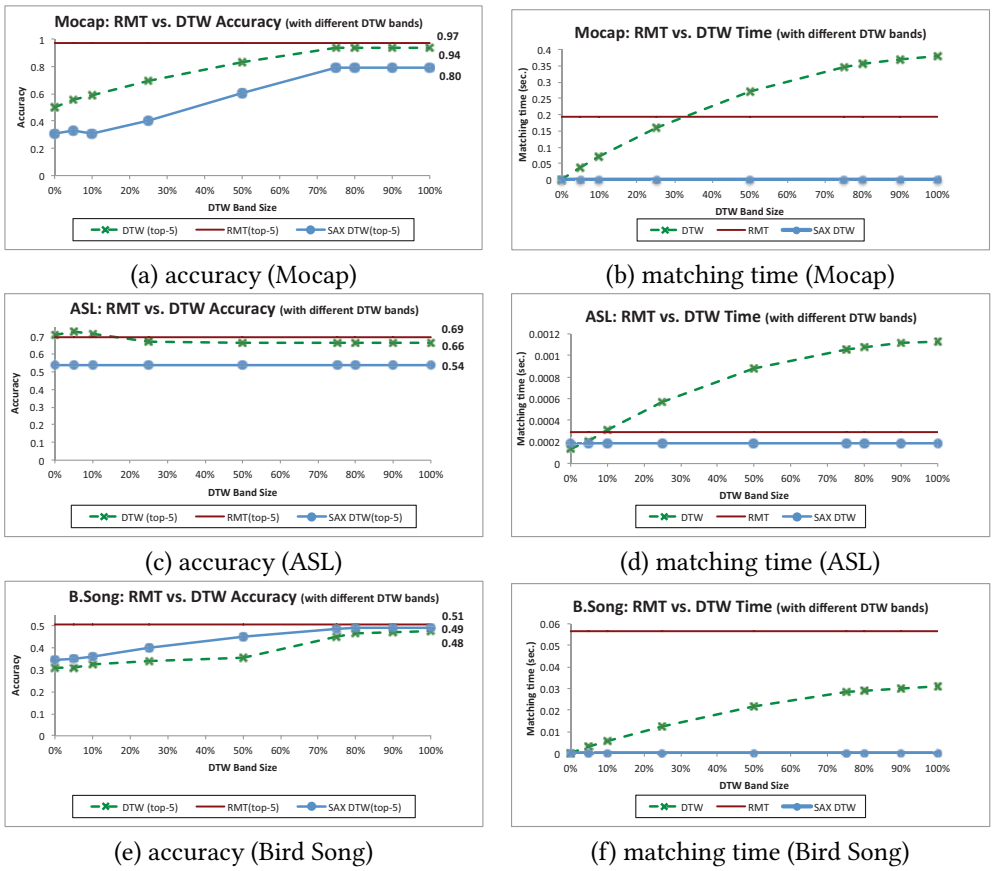
Many time series data sets are (a) multi-variate, (b) interrelated, and (c) multi-resolution: For the experiments reported in this paper, we used several multi-variate data sets that (a) offer the ability to leverage supporting metadata and (b) offer ground truth that can be used for evaluation purposes. The Mocap data sets used in the experiments in Section 11 represented human movement in the form of multi-variate time series (Figure 12, [34]). In this section, we also consider two additional multimedia data sets:

The *Australian sign language* data[9] includes sign gestures captured using a glove-based capture system. The capture data includes 100 per second tracking for all five fingers for both hands: each position tracker provides six degrees of freedom (roll, pitch, yaw, x, y, and z). The data set contains 95 signs, with 27 examples per sign. This data set has 22 variates (11 per hand) and contains a total of 2565 (= 95 × 27) multi-variate time series of average time length, 57. We associated with this data set a metadata file that considers the positions of the fingers within each hand. For this data set we set $\sigma_{time,0}$ to 0.5 (proportional to the average length of the series relative to Mocap - but sufficiently large that the temporal scope of the smallest feature covers more than one time instant). Note that ASL data set is selected because it is a relatively synchronized data set where Euclidean based measures perform well.

The *Bird Song* data set[10] contains Mel-frequency cepstral coefficient (MFCC) features for different bird calls. Intuitively, each MFCC coefficient captures short-term power spectrum of a sound for a given frequency band. The MFCC bands are equally spaced on the Mel scale (indicating that they are judged to be of equal distance from each other by listeners). The data set contains 13 MFCC coefficients (i.e., variates) for 154 bird calls of 8 classes, with the average time length of 397 time stamps. We associated with this data set a metadata file that records which MFCC co-efficient is neighbor to which other MFCC coefficients. For this data set we set $\sigma_{time,0}$ to 1.6 (proportional to the average length of the series, relative to Mocap).

Figure 13 shows top-5 accuracies and matching times for paired RMT, DTW[2], and SAX[28] DTW. SAX (Symbolic Aggregate approXimation [28]) is a symbolic representation for time series, which provides a lower-bound for distance measurements such as dynamic time warping and in general can be computed faster than traditional DTW. Here we also provide SAX[11] as a baseline competitor. We set the parameters for SAX representation: use 10 symbols[11] for representations and 20 segments for each multi-variate time series. Since DTW can be made faster by considering narrower bands [20, 43] (rather than the whole sequences), in this figure, we also consider

[9]https://archive.ics.uci.edu/ml/datasets/Australian+Sign+Language+signs+(High+Quality)
[10]http://www.xeno-canto.org/explore/taxonomy
[11]http://www.cs.ucr.edu/ eamonn/SAX.htm



(a) accuracy (Mocap) (b) matching time (Mocap)

(c) accuracy (ASL) (d) matching time (ASL)

(e) accuracy (Bird Song) (f) matching time (Bird Song)

Fig. 13. (a,c,e) top-5 accuracies for RMT of DTW for different bands (note: 0% band length corresponds to traditional Euclidean distance) and (b,d,f) matching times: for all experiments, the smallest RMT feature is set to be 5% of the average time series in the data set.

accuracies and execution times for different DTW band sizes. As an approximation method, the performance of SAX shares a similar behavior as the DTW method.

We can see in this figure that, while it may help make DTW process faster, placing a significant band length constraint ($\leq$ 80%) on DTW may reduce accuracy (for Mocap and bird song data sets, which are less temporally synchronized than the ASL data set). Most importantly, the figure shows that while SAX's accuracy widely fluctuates from one data set to the other, RMT provides consistently better (and overall *the best*) top-5 accuracies, at a matching time cost comparable to DTW. These results indicate that, whenever (even rough) metadata relating the variates is available, RMT can leverage this information to improve matching and classification accuracy.

B- Experiments with Additional Algorithms

In the previous sections, we compared the proposed RMT algorithm to approaches that are based on SVD, DTW, and SAX-based feature extraction. In this section, we consider two recent systems, namely RPM [52] and STS3 [37], that provide parameter selection and hyper-parameter estimation functionalities for uni-variate time series matching and time-series classification tasks; in

Data set	**RPM Acc.** (SVM)	**STS3 Acc.** (1-NN)	**RMT Acc.** (1-NN)
MoCap	0.847	0.078	0.989
BirdSong	0.436	0.145	0.481
ASL	N/A	0.234	0.715

Table 3. Accuracies for the multi-dimensional extensions of RPM [52] and STS3 [37] algorithms

particular, both use training data to learn feature patterns as well as contextually relevant hyper-parameters.

For both of these techniques, we obtained original code from the authors. However, since both of these approaches were originally designed for uni-variate time series data, we revised their code to account for multi-variate series as follows:

- RPM [52] creates SAX sequences and grammar rules for uni-variate time series from each class. More specifically, RPM concatenates all uni-variate time series from the same class in the training data and then extracts and selects SAX symbol sequences that are most representative for this given class. RPM then uses Sequitur to learn the context free grammars from the SAX representations as the grammar induction rules to represent this class. Given the output of this process, it finally uses SVM classifier for classification tasks.

  Since in this paper, we consider multi-variate time series data, we modified the original implementation to account for the existence of multiple variates. The training phase stays the same: RPM generates a grammar pattern for each variate. We concatenate all variate pattern vectors from the same class into one vector and use these concatenated pattern vectors from testing data for the SVM classifier. In order to ensure that RPM results and other results presented in our paper are comparable, we set the same SAX parameters as it was described in our manuscript: 20 SAX segments for each multi-variate time series data elements and up to 10 symbols for grammar rule-based representations.
- Instead of concatenating all uni-variate time series from the same class together, STS3 [37] learns patterns (sets of cell IDs) for every time series of each class. During testing phase, it computes sets of cells for testing data and it uses Jaccard similarity between training and testing data to assign class labels for the testing class.

  Once again, the original STS3 algorithm is designed for uni-variate time series. Therefore, we modified the implementation such that it extracts sets of cells for each variate of class per training data element and aggregates the final Jaccard similarities for each pair of corresponding variates between two multi-variate time series to measure time series similarity.

RPM [52] provides classification through SVM, whereas the code of STS3 [37] provided by the authors is designed for 1-NN matching. Therefore, to be fair to STS3, in Table 3, we provide 1-NN accuracy for RMT (rather than 5-NN and $\|c\|$-NN accuracies as reported elsewhere in this paper).

As the results in the table shows, the accuracy of RPM is lower than that of RMT, especially for the BirdSong data set, which results in significaanly lower accurcy, Note that, we are not able to report RPM accuracy results for the Australian Sign Language (ASL) dataset because there are multiple variates from various classes with values all zero and these cannot be used to generate grammar rules for classification. The table also shows that STS3 performs significantly worse than both RPM and RMT. While, unlike RPM, STS3 is able to handle the ASL data, it still provides very low accuracy due to the existence of these highly non-discriminating variates.

Data set	**Jaccard SAX Acc.** (1-NN)	**Cosine SAX Acc.** (1-NN)	**RMT Acc.** (1-NN)
MoCap	0.826	0.782	0.989
BirdSong	0.357	0.305	0.481
ASL	0.525	0.504	0.715

Table 4. Accuracies for the multi-dimensional extensions of Jaccard and cosine similarity based extensions of SAX

We note that the reason why STS3 performs rather poorly on multi-variate time series may be due to the way these algorithms learn patterns or the way they compare the series (or both). In order to better understand the underlying reason, we also considered a simple strategy that creates SAX symbols as in RPM, but uses Jaccard similarity of the resulting SAX term vectors for similarity computation as in STS3. More specifically, we counted the frequencies of each SAX symbol within an uni-variate vector and summed up the resulting weighted Jaccard similarities among variate pairs to obtain the similarity between two multi-variate time series: let $\vec{s}$ and $\vec{t}$ represent the symbol frequency vectors for two time series, S and T; the corresponding weighted Jaccard similarity is computed as

$$sim_{Jacc}(S,T) = \frac{||\vec{s}||_1 + ||\vec{t}||_1 - ||\vec{s} - \vec{t}||_1}{||\vec{s}||_1 + ||\vec{t}||_1 + ||\vec{s} - \vec{t}||_1},$$

where $|| * ||_1$ represents norm-1 for the corresponding vector. Furthermore, as a control scenario, we also considered the cosine similarity between the two vectors.

The results under the same evaluation conditions are presented in Table 4. This rather simple technique, based on SAX features matched using Jaccard similarity, approaches to that of RPM (on MoCap and BirdSong data sets where RPM results are available) and significantly improves over that of STS3; however results are still not as good as the RMT accuracies. Moreover, when using cosine similarity, matching accuracy slightly drops under that of the Jaccard similarity, indicating that STS3 is using a good measure for matching, but the core problem is that the underlying pattern extraction scheme cannot be directly expanded for multi-variate series.

Received January 2017