On the Effectiveness of Distance Measures for Similarity Search in Multi-Variate Sensory Data*

Yash Garg
CIDSE, Arizona State University
Tempe, USA 85281
ygarg@asu.edu

Silvestro Roberto Poccia
University of Turin
Corso Svizzera 185
Turin, Italy 10149
poccia@di.unito.it

ABSTRACT

Integration of rich sensor technologies with everyday applications, such as gesture recognition and health monitoring, has raised the importance of the ability to effectively search and analyze multi-variate time series data. Consequently, various time series distance measures (such as Euclidean distance, edit distance, and dynamic time warping) have been extended from uni-variate to multi-variate time series. In this paper, we note that the naive extensions of these measures may not necessarily be effective when analyzing multi-variate time series data. We present several algorithms, some of which leverage external metadata describing the potential relationships, either learned from the data or captured from the metadata, among the variates. We then experimentally study the effectiveness of multi-variate time series distance measures against human motion data sets.

CCS CONCEPTS

•Information systems →Similarity measures; Multimedia and multimodal retrieval;

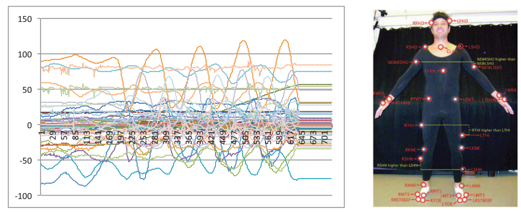
KEYWORDS

Action recognition, similarity measures, DTW, SAX

ACM Reference format:
Yash Garg and Silvestro Roberto Poccia. 2017. On the Effectiveness of Distance Measures for Similarity Search in Multi-Variate Sensory Data. In *Proceedings of ICMR '17, June 6–9, 2017, Bucharest, Romania, ,* 5 pages.
DOI: http://dx.doi.org/10.1145/3078971.3079009

1 INTRODUCTION

In recent years, time series data have become increasingly critical in organizational awareness, prediction, and decision making. Applications that require time series search and analysis include healthcare [8], surveillance [11], and motion and gesture recognition [16]



(a) Sample motion data (b) Sensors

Figure 1: A sample multi-variate time series, tracking 62 sensors, created by body motion capture [1]

(Figure 1). When comparing two sequences or time series, exact alignment is not required in most applications. Instead, whether two sequences are to be treated as matching or not depends on the amount of difference, quantified through *distance* measures[7] (see Section 1.1). Edit-distance measures quantify the minimum number (or cost) of symbol *insertions*, *deletions*, and *substitutions* needed to convert one sequence to the other [12]. *Dynamic time warping (DTW)* distance [3, 6], used commonly when comparing *continuous sequences* or time series (especially in scenarios where the sequences carry similar underlying patterns, but are different from each other due to temporal deformations, such as shifts and stretches), is a related distance measure.

1.1 Related Work

Euclidean distance and, more generally $L_p - norm$ measures, were among the first used to determine the similarity between two time series. Euclidean distance requires that the time series being compared are of *same temporal length* and, since it assumes a strict synchrony among time series, it is not suitable when two time series can have different speeds or are offset in time [6]. Other measures that require equal length and perfect synchrony across time series include cosine and correlation similarity [14].

In the 1970s, Sakoe [13] and in the 1990s, Berndt [3] proposed an edit distance like 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 the last two decades have seen several innovations [4, 6]. For example, while the original DTW is not metric (does not satisfy the triangular inequality), [4] proposed an extended version of DTW that does satisfy the triangular inequality. [10] proposed a SAX representation of input

*Both authors have equal contribution. This work is done under the supervision of Prof. K Selçuk Candan (candan@asu.edu) and Prof. Maria Luisa Sapino (mlsapino@di.unito.it). This work is supported by NSF Grant # 1339835 "E-SDMS", and NSF Grant # 1318788 "Data Mgmt. for Real-Time Data Driven Epidemic Spread Simulation".

ICMR '17, June 6–9, 2017, Bucharest, Romania

DOI: http://dx.doi.org/10.1145/3078971.3079009

sequences, which divides them into fixed-sized "frames", thus transforming the sequences into a reduce space where distance can be computed efficiently.

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. [5] proposed a feature-extraction algorithm that extracts minimal distinguishing subsequences that can be used as features. [16] proposed to extract and use SIFT-like robust multi-variate temporal features to determine similarity between multi-variate time series. In contrast, edit distance [9] measures aim to determine the minimum sequence of *edit* operations that are required to measure similarity.

1.2 Comparing Multi-Variate Time Series

Naive extensions of the uni-variate measures discussed above might not always work, as we may need to take into account (a) asynchrony among variates, (b) varying scales and importance of the different variates, as well as (c) dependency among the various variates that constitute a multi-variate time series . For example, nearby sensors in a sensor network may observe similar values, or sensors located on a human limb may be constrained to move together, except for some local variations. In this paper, we present, discuss, and evaluate several approaches to computing distances among multi-variate time series.

2 DISTANCE MEASURES FOR MULTI-VARIATE TIME SERIES

In this section, we start by formally defining uni-variate and multi-variate time series.

Definition 2.1 (Uni-Variate Time Series). *A* uni-variate time series, *$T = (d_1, d_2, ..., d_N)$, is a finite sequence of data values. Here, N denotes the length of the time series; and $d_i \in \mathbb{R}$.*

Multi-variate time series extend the uni-variate time series definition above by encapsulating multiple uni-variate series:

Definition 2.2 (Multi-Variate Time Series). *A* multi-variate time series, *$\mathbb{T}$, is an ordered set of equi-length time series; that record different observations (or* variates*):*

$$\mathbb{T} = [T_1, T_2, ..., T_v].$$

Here, $v > 1$ is the number of variates for time series, $\mathbb{T}$.

2.1 Metadata-Enriched Multi-Variate Series

Following [16], we define metadata-enriched multi-variate time seriesas follows:

Definition 2.3. *(Metadata-Enriched Multi-Variate Time Series) A* metadata-enriched multi-variate time series, *$\mathbb{T}_R$, is a multi-variate time series annotated with a data structure, R, that encodes the relationships among the variates; i.e. $\mathbb{T}_R = [\mathbb{T}, R]$.*

While in general the metadata R can be complex, here we assume that R is encoded in the form of a relationship matrix:

Definition 2.4 (Relationship Matrix). *A $v \times v$* relationship matrix*, R, encodes the pairwise relationships among the variates; i.e.,*

$$\forall_{1 \leq i, j \leq v} \quad R[i, j] = \rho(T_i, T_j)$$

where $\rho()$ is an application-specific function to determine the relationship between a given pair of variates.

Example 2.1 (Encoding Spatial Relationships). Let us consider a set, $\mathbb{S}$, of in-situ sensors distributed in a spatial area, and let us assume that we can compute the spatial distance, Δ_{sp}, between any given pair of sensors. Under the assumption that two nearby sensors are likely to observe more similar processes than sensors that are far apart from each other, we can encode a spatial relationship matrix, R_{sp}, as follows:

$$\forall_{s_i, s_j \in S} \quad R_{sp}[i, j] = \max_{s_h, s_k \in \mathbb{S}} \left(\Delta_{sp}(h, k)\right) - \Delta_{sp}(i, j).$$

Intuitively, the difference operation above converts distances to proximities, such that the larger values in the relationship matrix correspond to more-related sensor pairs.

2.2 Synchronized Distance Measures

Dynamic time warping (DTW)[3, 6] is a common technique for comparing sequences or time series by searching for optimal alignments, described in terms of *warp paths.* Recently, various extensions of DTW have been proposed for *multi*-dimensional time series [15]. The most prevalent of these are the *vectorized* and *independent* extensions. In the *vectorized* approach, a multi-variate time series is considered as a sequence of vectors, where the length of a 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 the *independent* approach, however, each variate is treated independently from the others and DTW is applied separately to each; finally, these independent distances are added to compute the overall distance between the given pair of multi-variate time series.

2.2.1 Independent DTW. Given two sequences, $\mathbb{X}$ and $\mathbb{Y}$, each with v variates, the independent DTW measure is computed by treating each variate separately from the others:

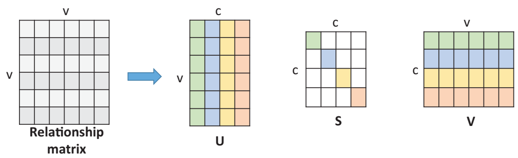
$$I\_DTW = \sum_{i=1}^{v} DTW(\mathbb{X}_i, \mathbb{Y}_i).$$

2.2.2 Vectorized DTW. In contrast, vectorized DTW considers $\mathbb{X}$ and $\mathbb{Y}$ as sequences of vectors, where the length of a vector is equal to the number of variates in the time series; i.e, $\mathbb{X} = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N)$ and $\mathbb{Y} = (\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_M)$, such that each $\vec{x}_*$ and $\vec{y}_*$ is a v-dimensional vector. The DTW algorithm is then applied considering the differences among these vectors; i.e.,

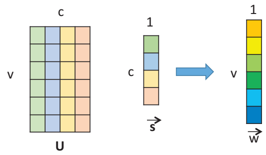
$$V\_DTW = DTW_v(\mathbb{X}, \mathbb{Y}),$$

2.2.3 SAX based DTW. SAX [10] transforms a sequence of length n into a string of a length determined by the size of window w. At the same time, SAX can transform two sequences to strings of equal length, thus allowing linear comparison between them. A *uni-variate time series*, $T = (d_1, d_2, ..., d_N)$, can be written as $T_S = \langle C_1, ..., C_m \rangle$ where m is the number of words extracted using aggregation. Similar to the independent DTW, DTW measure between two sequences in SAX representation, $\mathbb{X}_\mathbb{S}$ and $\mathbb{Y}_\mathbb{S}$, each with v variates, can be computed as follows:

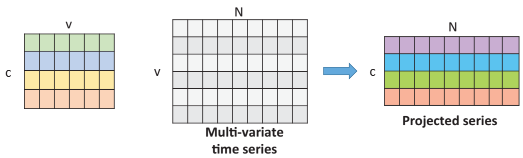
$$SAX\_DTW = \sum_{i=1}^{v} DTW(\mathbb{X}_{\mathbb{S}i}, \mathbb{Y}_{\mathbb{S}i}).$$



(a) Eigen-decomposition of the relationship matrix



(b) Weight extraction



(c) Projecting time series onto orthogonal variates

Figure 2: Weight extraction and projection of a multi-variate time series onto an orthogonal variate basis using relationship matrix R.

2.3 Metadata-Driven Distance Measures

In this section, we discuss two approaches to account for the contextual relationships among the variates:

- metadata-driven weighting, and
- orthogonal latent variates.

2.3.1 Metadata-Driven Weights. Given a relationship matrix $R \rightarrow v \times v$, representing the contextual relationship between the variates, we can leverage eigen-decomposition to extract the patterns encoded in R,

$$[U^R, S^R, V^R] = eigen(R),$$

where the $c \times c$ diagonal matrix S^R and the $V \times c$ matrix U^R encode the strengths of pattern and the participation of the variates to these patterns.

Given this, we can calculate variate weights, $\vec{w}_R$, by combining the pattern membership degrees of the variates encoded in U^R with the pattern strengths scores encoded in S^R. In particular, we consider both positive and negative weights:

$$\vec{w}_R^+ = U^R * \vec{s}_R,$$
$$\vec{w}_R^- = U^R * 1/\vec{s}_R,$$

where the vector $\vec{s}_R = diag(S^R)$ encodes the strengths of the latent patterns in the relationship matrix R. Once these weights are extracted, we can adapt uni-variate similarity and distance (such as DTW) measures to account for metadata-driven weights; e.g.,

$$weighted\_DTW = \sum_{i=1}^{v} \vec{w}[i] \times DTW(\mathbb{X}_i, \mathbb{Y}_i).$$

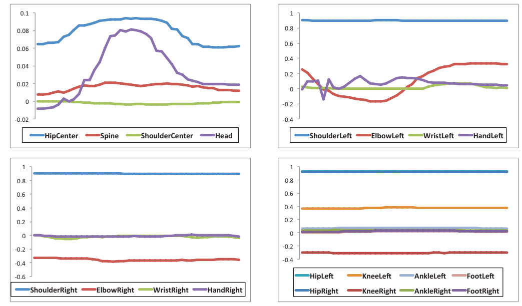


Figure 3: A sample gesture from the Kaggle data set and the corresponding structural groupings of the body sensors [2]

2.3.2 Orthogonal Latent Variates. As discussed in Section 1.1, a major difficulty with existing multi-variate distance/similarity measures is that they assume either that (a) the different variates in the data are completely independent from each other, or that (b) the variates move in perfectly synchronized lock-step fashion.

Here, we note that one way to leverage groupings of variates is to map (or project) the given time series onto an alternative set of latent, orthogonal variates. Intuitively, since each orthogonal variate would correspond to a different structural grouping of the original variates, assuming that these groupings reflect the way temporal observations of the variates vary together, we expect that the observations mapped on to these latent variates would be independent from each other. We can achieve this goal by relying on orthogonal variate bases obtained through eigen-decompositions of the relationship matrix[1]. More specifically, given a relationship matrix, R, we first obtain its eigen-decomposition as before:

$$[U^R,\ S^R,\ V^R] = eigen(R).$$

However, instead of relying on U^R and S^R to assign weights to the existing variates, we leverage the $c \times v$ matrix V^R to help project the two time series, $\mathbb{X}$ and $\mathbb{Y}$, onto a set of orthogonal latent variates:

$$\mathbb{X}^\dagger = V^R\ \mathbb{X} \ \text{ and } \ \mathbb{Y}^\dagger = V^R\ \mathbb{Y}.$$

This process is visualized in Figure 2. The resulting time series, $\mathbb{X}^\dagger$ and $\mathbb{Y}^\dagger$, can now be used as inputs to any independent or synchronized multi-variate distance/similarity measure, as discussed in Section 2.2. While evaluating the proposed approach, we will consider orthogonal projection-based independent and vectorized approaches.

3 EXPERIMENTAL EVALUATION

In this section, we evaluate the presented approaches on two human action datasets, Mocap[1] and Kaggle[2] and assess their effectiveness in similarity search. While both record human motion, the Mocap and Kaggle data sets are very different from each other: as we see in Figure 1, the variates in the Mocap data all record varying degrees of repeated motion; in contrast, as Figure 3 illustrates, the motions in Kaggle show significant degrees of variate localization (with only a few sensors recording active motion during a gesture)

[1]In this case, eigen-decompositions of the input time series are not useful, as individually decomposing the two time series would project them onto different orthogonal bases.

Algorithms	Top-5 % Accuracy	Class Tightness
Original Data Space		
V-DTW	87.83	1.62
V-SAX	67.83	1.38
I-DTW	95.43	1.91
I-SAX	71.74	1.49
MW-DTW(w_R^+)	95.33	1.91
MW-DTW(w_R^-)	**95.87**	**1.93**
MW-SAX(w_R^+)	72.28	1.49
MW-SAX(w_R^-)	71.52	1.48
Orthogonal Latent Space		
V-DTW	72.07	1.43
V-SAX	71.41	1.41
I-DTW	77.72	1.57
I-SAX	65.98	1.26
MW-DTW(S^R)	69.46	1.56
MW-DTW($1/S^R$)	**81.74**	**1.92**
MW-SAX(S^R)	56.20	1.23
MW-SAX($1/S^R$)	63.80	1.27

Table 1: Percent retrieval accuracy & class tightness for Mocap data set for DTW & SAX (window size =20) measures.

Algorithms	Top-5 % Accuracy	Class Tightness
Original Data Space		
V-DTW	**44.63**	1.30
V-SAX	35.78	1.31
I-DTW	41.78	1.32
I-SAX	32.81	1.32
MW-DTW(w_R^+)	41.49	**1.35**
MW-DTW(w_R^-)	41.81	1.32
MW-SAX(w_R^+)	22.44	1.34
MW-SAX(w_R^-)	33.16	**1.35**
Orthogonal Latent Space		
V-DTW	23.65	1.12
V-SAX	**27.88**	1.18
I-DTW	21.85	1.15
I-SAX	24.91	1.22
MW-DTW(S^R)	21.85	1.15
MW-DTW($1/S^R$)	21.84	1.15
MW-SAX(S^R)	18.15	**1.25**
MW-SAX($1/S^R$)	21.90	1.20

Table 2: Percent retrieval accuracy & class tightness for Kaggle data set for DTW & SAX (window size =2) measures.

and do not include repeated patterns. These differences among the motion data sets enable us to study the various algorithms within different contexts. As metadata, we use the sensors' spatial relationship; for latent-space approaches, we preserve 100% energy (variance). For this experiment, to generate SAX representations, we use window size 20 and 2 for Mocap and Kaggle respectively.

3.1 Evaluation Criteria

To assess the effectiveness of the algorithms, we considered two main criteria: *retrieval accuracy* and *class tightness*.

Retrieval Accuracy (ACC). To quantify retrieval accuracy provided by a given distance function, Δ, we execute k-nearest neighbor ($k-NN$) queries (for varying k, excluding itself) and measure accuracy as

$$ACC(k, \Delta) = \underset{C \in Classes}{AVG} \ \underset{s_i \in C}{AVG} \left(\frac{\#match(s_i, k, \Delta, C)}{k} \right),$$

Class Tightness (CT). Given a distance function, Δ, we consider a set of classes *tight* if the following measure is proportionately large:

$$CT(\Delta) = \underset{C \in Classes}{AVG} \frac{avg\_inter\_dist(C, \Delta)}{avg\_intra\_dist(C, \Delta)}.$$

where,

$$avg\_inter\_dist(C, \Delta) = \underset{s_i \in C}{AVG} \left(\underset{s_j \notin C}{AVG} \left(\Delta(s_i, s_j) \right) \right)$$

$$avg\_intra\_dist(C, \Delta) = \underset{s_i \in C}{AVG} \left(\underset{s_j (\neq s_i) \in C}{AVG} \left(\Delta(s_i, s_j) \right) \right).$$

3.2 Results

Mocap dataset: Table 1 presents the results. It can be seen that metadata adds more discriminatory power to the distance measure, which in turn leads to a better retrieval accuracy and class tightness as compared to the naive extensions of both DTW and SAX. While latent space mappings overall result in reduced accuracy, we see that metadata support again provides a significant boost.

Kaggle dataset: Table 2 presents the results. Note that for this data set, retrieval is more difficult, and vectorized approaches, which assume full synchrony, provide an overall better retrieval accuracy. Yet, once again, metadata weighting helps improve the separation between classes, both in original and latent spaces.

4 CONCLUSIONS AND FUTURE WORK

As seen in Section 3, overall we observe a positive impact from the use of metadata, which describes contextual relationships, in computing distances among multi-variate time series.

Naturally, the impact of the use of metadata depends on the type of series we consider, the degree of inherent synchrony/asynchrony amongst the variates in the data, and the quality of metadata representation used to capture the relationships among the variates. Therefore, significant research needs to be done on automated mechanisms to characterize multi-variate data and extract informative metadata. We also note that the degree of synchrony/asynchrony may be different for different groupings of sensors; therefore, we may also seek to identify and leverage sub-groupings of relatively synchronous sensors to improve accuracy.

Another challenge is that, while here we assume that the metadata is common to all series, it is possible that in practice, each individual series will have its own metadata (e.g., each user has their own body shape), and the alignment of these metadata needs to be considered while measuring the similarities and differences among the corresponding multi-variate time series.

In this paper we disregarded (a) feature-based approaches to multi-variate time series comparison, (b) use of distance measures for searching for multi-variate motifs, (c) metadata-supported dynamic topic modeling and deep learning over multi-variate time series, and (d) execution time issues and other performance optimizations. These constitute our future work.

REFERENCES

[1] CMU graphics lab motion capture database, 2015.

[2] Kaggle gesture dataset, 2015.

[3] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[4] Lei Chen and Raymond Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803. VLDB Endowment, 2004.

[5] Xiaonan Ji, James Bailey, and Guozhu Dong. Mining minimal distinguishing subsequence patterns with gap constraints. *Knowledge and Information Systems*, 11(3):259–286, 2007.

[6] Eamonn Keogh. Exact indexing of dynamic time warping. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 406–417. VLDB Endowment, 2002.

[7] Eamonn Keogh. Data mining and information retrieval in time series/multimedia databases. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 10–10. ACM, 2006.

[8] Duk-Jin Kim and Balakrishnan Prabhakaran. Multimedia aspects in health care. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 921–922. ACM, 2009.

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

[10] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill 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*, pages 2–11. ACM, 2003.

[11] Yang Liu, Feng Zhou, Wei Liu, Fernando De la Torre, and Yan Liu. Unsupervised summarization of rushes videos. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 751–754. ACM, 2010.

[12] Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.

[13] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[14] Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

[15] Parinya Sanguansat. Multiple multidimensional sequence alignment using generalized dynamic time warping. *WSEAS Transactions on Mathematics*, 11(8):668–678, 2012.

[16] Xiaolan Wang, K Selçuk Candan, and Maria Luisa 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.