Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-023-00979-9

Somtimes: self organizing maps for time series clustering and its application to serious illness conversations

Ali Javed[1,3] · Donna M. Rizzo[2,3] · Byung Suk Lee[3] · Robert Gramling[4]

Received: 20 February 2023 / Accepted: 22 August 2023

Abstract

There is demand for scalable algorithms capable of clustering and analyzing large time series data. The Kohonen self-organizing map (SOM) is an unsupervised artificial neural network for clustering, visualizing, and reducing the dimensionality of complex data. Like all clustering methods, it requires a measure of similarity between input data (in this work time series). Dynamic time warping (DTW) is one such measure, and a top performer that accommodates distortions when aligning time series. Despite its popularity in clustering, DTW is limited in practice because the runtime complexity is quadratic with the length of the time series. To address this, we present a new a self-organizing map for clustering TIME Series, called SOMTimeS, which uses DTW as the distance measure. The method has similar accuracy compared with other DTW-based clustering algorithms, yet scales better and runs faster. The computational performance stems from the pruning of unnecessary DTW computations during the SOM's training phase. For comparison, we implement a similar pruning strategy for K-means, and call the latter K-TimeS. SOMTimeS and K-TimeS pruned 43% and 50% of the total DTW computations, respectively. Pruning effectiveness, accuracy, execution time and scalability are evaluated using 112 benchmark time series datasets from the UC Riverside classification archive, and show that for similar accuracy, a 1.8× speed-up on average for SOMTimeS and K-TimeS, respectively with that rates vary between 1× and 18× depending on the dataset. We also apply SOMTimeS to a healthcare study of patient-clinician serious illness conversations to demonstrate the algorithm's utility with complex, temporally sequenced natural language.

Keywords Time series clustering · Self-organizing maps · Dynamic time warping · Clustering · Serious illness conversations

Responsible editor: Eamonn Keogh.

Extended author information available on the last page of the article

Published online: 20 October 2023

1 Introduction

By 2025, it is estimated that more than four hundred and fifty exabytes of data will be collected and stored daily (WorldEconomicForum 2019). Much of that data will be collected continuously and represent phenomena that change over time. We propose that fully understanding the meaning of these data will require methods capable of efficiently visualizing and analyzing large amounts of time series. Examples include data mined by the Internet of Things (CRS 2020; Evans 2011), lexicon and natural language (e.g., Bentley et al. 2018; Ross et al. 2020; Reagan et al. 2016; Chu et al. 2017), biomonitors (Gharehbaghi and Lindén 2018), streamflow, barometric pressure and other environmental sensor data (e.g., Hamami and Dahlan 2020; Javed et al. 2020a; Ewen 2011), social media interactions (e.g., De Bie et al. 2016; Javed and Lee 2018, 2016, 2017), and hourly financial data reported by fluctuating world stock and currency markets (Lasfer et al. 2013). In response to the increasing amounts of time-oriented data available to analysts, the applications of time-series modeling are growing rapidly (e.g., Minaudo et al. 2017; Dupas et al. 2015; Mather and Johnson 2015; Bende-Michl et al. 2013; Iorio et al. 2018; Gupta and Chatterjee 2018; Pirim et al. 2012; Souto et al. 2008; Flanagan et al. 2017).

Time series modeling is computationally “expensive” in terms of processing power and speed of analysis. Indeed, as the numbers of observations or measurement dimensions for each observation increase, the relative efficiency of time series modeling diminishes, creating an exponential deterioration in computational speed. Under conditions where computing power is in excess or when the speed for generating results is not of concern, these challenges would be less pressing. Currently, however, these conditions are rarely met, and the accelerating rate of data collection promises to outpace the computational infrastructure available to most analysts.

In this work, we embed distance-pruning into a new artificial neural network—**S**elf-**O**rganizning **M**ap for **time s**eries (SOMTimeS) and **K**-means for **time s**eries (K-TimeS) to improve the execution time of clustering methods that used Dynamic Time Warping (DTW) for large time series applications. The single layered SOMTimeS is computationally faster than the deep layered classifiers (Wang and Jiang 2021) making it suitable for time series analysis. The popular K-Means clustering of K-TimeS is fast, particularly for well-separated clusters, but lacks the outstanding visualization capabilities of SOMTimeS. The computational efficiency of these algorithms is attributed to the pruning of unnecessary DTW computations during the training phases of each algorithm. When assessed using 112 time series datasets from the University of California, Riverside (UCR) classification archive, SOMTimeS and K-TimeS prunded 43% and 50% of the DTW computations, respectively. While the pruning efficiency and resulting speed-up vary depending on the dataset being clustered, on average, there is a 1.8× speed-up over all 112 of the archived datasets. To the best of our knowledge, K-TimeS and SOMTimeS are the fastest DTW-based clustering algorithms to date.

SOMTimeS is designed for users who wish to leverage DTW (i.e., optimally align two time series) as well as the SOM’s visualization capabilities when clustering time series data of high complexity. To explore the potential utility of SOMTimeS in this regard, we show the algorithm's ability to cluster and visualize time series data (i.e., conversational narratives) from highly emotional doctor-family-patient conversations. Understanding and improving serious illness communication is a national priority for 21st century healthcare; but, our existing methods for measuring and analyzing such data are cumbersome, human intensive, and far too slow to be relevant for large epidemiological studies, communication training, or time-sensitive reporting. Here, we use data from an existing multi-site epidemiological study of serious illness conversations as one example of how efficient computational methods can add to the science of healthcare communication.

The remainder of this paper is organized as follows. Section 2 provides background information on SOMs and DTW. Section 3 presents the SOMTimeS algorithm. Sections 4 and 5 provide the performance measures for SOMTimeS, K-TimeS, and another DTW-based clustering algorithm – TADPole using the UCR benchmark datasets, and then show proof-of-concept of the SOMTimeS algorithm as an exploratory data analysis tool involving health care conversations. Section 6 discusses the results. Section 7 concludes the paper and suggests future work.

2 Background

Similar to the work of Silva and Henriques (2020), Li et al. (2020), Parshutin and Kuleshova (2008), and Somervuo and Kohonen (1999), SOMTimeS is a new artificial neural network that embeds a distance-pruning strategy into a DTW-based Kohonen self-organizing map. While the original Kohonen SOM (see details in Sect. 2.1) is linearly scalable with respect to the number of input data, it often performs hundreds of iterations (i.e., epochs) when self-organizing or clustering the training data. Each epoch requires $n \times M$ distance calculations, where n is the number of observations and M is the number of nodes in the network map. While work has been performed to improve the SOM speed using both hardware (Dias et al. 2021; de Abreu de Sousa et al. 2020) and algorithmic (Conan-Guez et al. 2006) solutions, the large number of required distance calculations is problematic, particularly when the distance measure is computationally expensive, as is the case with DTW (see Sect. 2.2).

Originally introduced in 1970 s for speech recognition (Sakoe and Chiba 1978), DTW continues to be one of the more robust, top performing, and consistently chosen learning algorithms for time series data (Xi et al. 2006; Ding et al. 2008; Paparrizos and Gravano 2016, 2017; Begum et al. 2015; Javed et al. 2020b). Its ability to shift, stretch, and squeeze portions of the time series helps address challenges inherent to time series data (e.g., optimize the alignment of two temporal sequences). Unfortunately, the ability to align the temporal dimension comes with increased computational overhead that has hindered its use in practical applications involving large datasets or long time series clustering (Javed et al. 2020b; Zhu et al. 2012). The first subquadratic-time algorithm ($O(m^2/\log\log m)$) for DTW computation was proposed by Gold and Sharir (2018), which for comparison is still more computationally expensive than the simpler Euclidean distance ($O(m)$).

To address the computational cost, several studies have presented approximate solutions (Zhu et al. 2012; Salvador and Chan 2007; Al-Naymat et al. 2009). To the best of our knowledge, TADPole by Begum et al. (2015) is the only clustering algorithm (see supplementary material Section 8.1) that speeds up the DTW computation without using an approximation. It does so by using a bounding mechanism to prune the expensive DTW calculations. Yet, when coupled with the clustering algorithm (i.e., Density Peaks of Rodriguez and Laio 2014), it still scales quadratically. Thus, even after decades of research (Zhu et al. 2012; Begum et al. 2015; Lou et al. 2015; Salvador and Chan 2007; Wu and Keogh 2020), the almost quadratic time complexity of DTW-based clustering still poses a challenge when clustering time series in practice.

2.1 Self organizing maps

The Kohonen self-organizing map (Kohonen et al. 2001; Kohonen 2013) may be used to either cluster or classify observations, and has advantages when visualizing complex, nonlinear data (Alvarez-Guerra et al. 2008; Eshghi et al. 2011) Additionally, it has been shown to outperform other parametric methods on data with outliers or high variance (Mangiameli et al. 1996). Similar to methods such as logistic regression and principal component analysis, SOMs may be used for feature selection, as well as mapping input data from a high-dimensional space to a lower-dimensional (typically two-dimensional) space/map. To demonstrate the utility of 2-D visualization, we used the DTW-based SOM to cluster data from one of the UCR archive datasets (InsectEPGRegularTrain) onto a 2-D mesh (see Fig. 1). The input time series data represent voltage changes from an electrical circuit designed to capture the interaction between insects and their food source (e.g., plants). While these data had already been classified into three categories, we show the results from the DTW-based SOM clustering in Fig. 1a–c. The self-organized time series may also be plotted in 2-D space (see map of Fig. 1d); each gray dot represents a input time series. The gray shading represents what is known as a unified distance matrix or U-matrix (Ultsch 1993), and is obtained by calculating the average difference between the weights of adjacent nodes in the trained SOM. These differences are plotted in a gray scale on the trained 2-D mesh. Darker shading represents higher U-matrix values (larger average distance between observations). In this manner, the U-matrix may be used to help assess the quality and the number of clusters. In Fig. 1e, we color-code the three clustered observations; labels (should they exist) could also be superimposed. Finally, any information (i.e, input features or metadata) associated with the observations may also be visualized/superimposed in the same 2-D space. The latter allows users or domain experts to explore associations and the importance of individual input features with the self-clustered results. As an

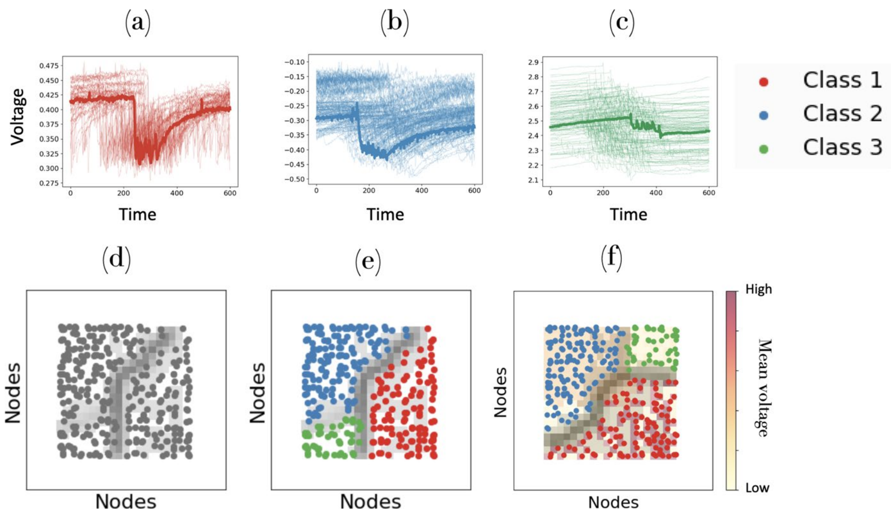

Fig. 1 Results of clustering times series from one of the UCR archive datasets—InsectEPGRegularTrain, using a Self-Organized Map with DTW as the distance measure (a.k.a. SOMTimeS). **a**–**c** Three sets of classified time series associated with voltage changes that capture interactions between insects and their food source (e.g., plants). In each panel a single observation (time series) is highlighted. **d** Unified distance matrix (gray shading) that separates the clustered observations on the 2D grid, **e** color-codes the observations by cluster ID, and **f** superimposes the mean voltage value in the background

example we visualize the mean voltage value in the background in Fig. 1f. The ability to visualize individual input features in the same space as the clustered observations (known as component planes) makes the SOM a powerful tool for exploratory data analysis and feature selection.

2.2 Dynamic time warping

DTW is recognized as one of the most accurate similarity measures for time series data (Paparrizos and Gravano 2017; Rakthanmanon et al. 2012; Johnpaul et al. 2020; Javed et al. 2020b). While the most common measure, Euclidean distance,

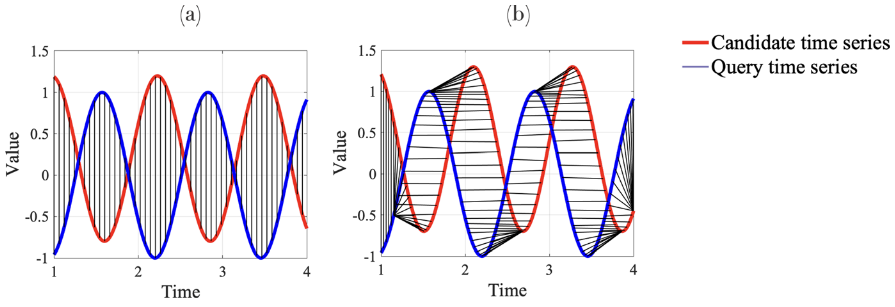

Fig. 2 Alignment between two times series for calculating **a** Euclidean distance and **b** DTW distance

uses a one-to-one alignment between two time series (e.g., labeled candidate and query in Fig. 2a), DTW employs a one-to-many alignment that warps the time dimension (see Fig. 2b) in order to minimize the sum of distances between time series samples. As such, DTW can optimize alignment both globally (by shifting the entire time series left or right) and locally (by stretching or squeezing portions of the time series). The optimal alignment should adhere to three rules:

1. Each point in the query time series must be aligned with one or more points from candidate time series, and vice versa.
2. The first and last points of the query and a candidate time series must align with each other.
3. No cross-alignment is allowed; that is, the aligned time series indices must increase monotonically.

DTW is often restricted to aligning points only within a moving window of a fixed size to improve accuracy and reduce computational cost. The window size may be optimized using supervised learning on the training data. When supervised learning is not possible (i.e., clustering), a window size amounting to 10% of the observation data is usually considered adequate (Ratanamahatana and Keogh 2004). We fixed the DTW window constraint at 5% of the length of the observation data as it was shown to be the optimal window size for the UCR archive Paparrizos and Gravano (2016, 2017).

2.2.1 Upper and lower bounds for DTW-based distance

SOMTimeS uses distance bounding to prune the DTW calculations performed during the SOM unsupervised learning. This distance bounding involves finding a tight upper and lower bound. Because DTW is designed to find a mapping that minimizes the sum of the point-to-point distances between two time series, that mapping can

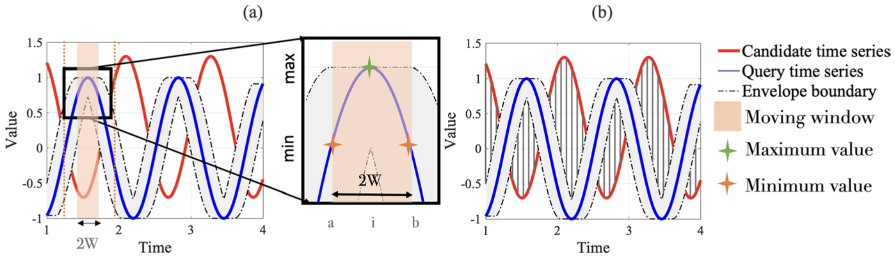

Fig. 3 Two steps of calculating the LB_Keogh tight lower bound for DTW in linear time: **a** determine the envelope around a query time series, and **b** sum the point to point distance shown in grey lines between the envelope and a candidate time series as LB_Keogh (Eq. 2)

never result in a summed distance that is greater than the sum of point-to-point Euclidean distance. Hence, finding the tight upper bound is straight forward—it is the Euclidean distance (Keogh 2002). To find the lower bound, we use a method—the LB_Keogh method (Keogh 2003), common in similarity searches (Keogh 2003; Ratanamahatana et al. 2005; Wei et al. 2005) and clustering (Begum et al. 2015). The LB_Keogh method comprises two steps (see Fig. 3a, b). Given a fixed DTW window size, W, one of the two time series (called the query time series, Q) is bounded by an envelope having an upper (U_i) and lower boundary (L_i) calculated at time step i, respectively, as:

$$\begin{aligned} U_i &= \max(q_a, \dots, q_i, \dots, q_b) \\ L_i &= \min(q_a, \dots, q_i, \dots, q_b), \end{aligned} \tag{1}$$

where $a = i - W$, and $b = i + W$ (see Fig. 3a). In the second step, the LB_Keogh lower bound is calculated as the sum of Euclidean distance between the candidate time series and the envelope boundaries (see vertical lines of Fig. 3b). Equation 2 shows the formula for calculating the LB_Keogh lower bound:

$$\text{LB\_Keogh} = \sqrt{\sum_{i=1}^{m} \begin{cases} (t_i - U_i)^2, & \text{if } t_i > U_i \\ (t_i - L_i)^2, & \text{if } t_i < L_i \\ 0, & \text{otherwise} \end{cases}} \tag{2}$$

where t_i, U_i, and L_i are the values of a candidate time series, the upper and lower envelope boundary, respectively, at time step i.

2.2.2 The UCR archive

The UCR time series classification archive (Dau et al. 2018), with thousands of citations and downloads, is arguably the most popular archive for benchmarking time series clustering algorithms. The archive was born out of frustration, with studies on clustering and classification reporting error rates on a single time series dataset, and then implying that the results would generalize to other datasets. At the time of this writing, the archive has 128 datasets comprising a variety of synthetic, real, raw and pre-processed time series data, and has been used extensively for benchmarking the performance of clustering algorithms (e.g., Paparrizos and Gravano 2016, 2017; Begum et al. 2015; Javed et al. 2020b; Zhu et al. 2012). We excluded sixteen of the archive datasets because they contained only a single cluster, or had time series lengths that vary, and the latter prohibited a fair comparison of SOMTimeS to K-TimeS. Thus, 112 datasets are used to evaluate the accuracy, execution time, and scalability of SOMTimeS.

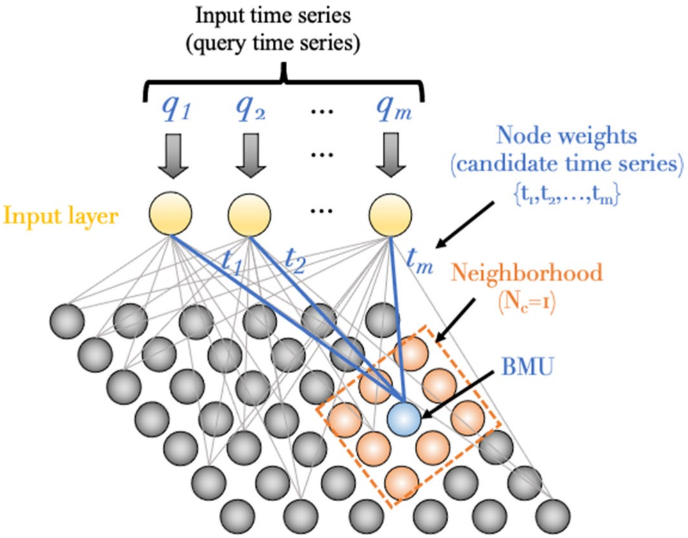

Fig. 4 Schematic of the Kohonen self-organizing map (after Kohnen, 2001) showing weights (candidate time series) of the Best Matching Unit (BMU) in blue surrounded by a user-specified neighborhood (N_c)

3 The SOMTimeS algorithm

SOMTimeS (see Pseudocode 1) is a variant of the SOM, where each input observation (i.e., query time series) is compared with the weights (i.e., candidate time series) associated with each node in the 2-D SOM mesh (see Fig. 4). During training, the comparison (or distance calculation) between these two time series is performed to identify the SOM node whose weights are most similar to a given input time series; this node is identified as the “Best Matching Unit (BMU)”. Once the nodal weights (candidate time series) of the BMU have been identified, these weights (and those of the neighborhood nodes) are updated to more closely match the query time series (Line 17 of Pseudocode 1). This same process is performed for all query time series in the dataset—defined as one iteration. While iterating through some user-defined fixed number of iterations, both the neighborhood size and the magnitude of change to nodal weights are incrementally reduced. This allows the SOM to converge to a solution (stable map of clustered nodes), where the set of weights associated with these self-organized nodes now approximate the input time series (i.e., observed data). In SOMTimeS, the distance calculation is done using DTW with bounding, which helps prune the number of DTW calculations required to identify the BMU.

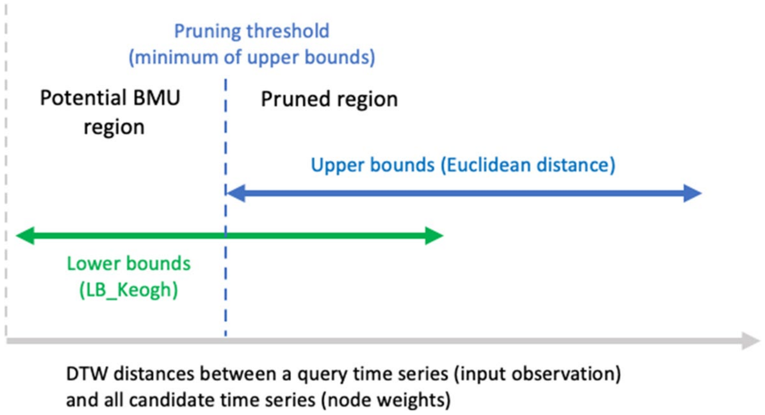
Pruning threshold
(minimum of upper bounds)
Potential BMU region
Pruned region
Upper bounds (Euclidean distance)
Lower bounds (LB_Keogh)
DTW distances between a query time series (input observation) and all candidate time series (node weights)

Fig. 5 Identification of a qualification region in SOMTimeS

3.1 Pruning of DTW computations

Pruning is performed in two steps. First, an upper bound (i.e., Euclidean distance) is calculated between the input observation and each weight vector associated with the SOM nodes (Line 9 of Pseudocode 1). The minimum of these upper bounds is set as the pruning threshold (see dotted line in Fig. 5). Next, for each SOM node, we calculate a lower bound (i.e., LB_Keogh; see Line 10). If the calculated lower bound is greater than the pruning threshold, that respective node is pruned from being the BMU. If the lower bound is less than the pruning threshold, then that SOM node lies in what we refer to as the potential BMU region (see Fig. 5, and Line 12). As a result, the more expensive DTW calculations are performed only for the nodes in this potential BMU region. The node having the minimum summed distance is the BMU.

After identifying the BMUs for each input time series, the BMU weights, as well as the weights of nodes in some neighborhood of the BMUs, are updated to more closely match the respective input time series using a traditional learning algorithm based on gradient descent (Line 17 of Pseudocode 1). Both the learning rate and the neighborhood size are reduced (see lines 19 and 20) over each epoch until the nodes have self-organized (i.e., algorithm has converged). In this work, unless otherwise stated, SOMTimeS is trained for 100 epochs. To further reduce the SOM execution time, the set of input (i.e., query) time series may be partitioned in a manner similar to Wu et al. (1991), Obermayer et al. (1990) and Lawrence et al. (1999) for parallel processing (see Line 5). We should also note that after convergence, SOMTimeS may be used to *classify* observations into a given number of clusters should a known number of classes exist. This is done by setting the mesh size equal to *k* (i.e., desired number of classes), and using the weights of the BMUs for direct class assignment. Python implementation of SOMTimeS is available at Python Package Index (Javed 2021a) and Github (Javed 2021b).

```
Algorithm SOMTimeS pseudocode
Input: a set $\mathcal{S}$ of query time series $\{Q_1, Q_2, ..., Q_n\}$, $epochs$: number of $epochs$, $W$: warping window size
Assumption: Similarity between two observation is the DTW distance between their time series.
Output: Best Matching Units
Procedure
mesh_size $M := 5 \times \sqrt{n}$          // $n$ = the number of time series in $\mathcal{S}$
Create and randomly initialize a mesh of Nodes[$\sqrt{M}$,$\sqrt{M}$], where the weights of each node are a randomly-generated time series (candidate time series) of length equal to the query time series.
neighborhood size $N_c := \sqrt{M}/2$
learning rate $r := 0.9$
Split $\mathcal{S}$ into subsets of equal size, $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_c$, where $c$ is the number of available CPU cores in the machine.
for each epoch $p$ do
    for each split $\mathcal{S}_i$ $(i = 1, 2, ..., c)$ assigned to the core $i$ in parallel do
        for each input time series $Q_{i_j}(j = 1, 2, ..., n/c)$ in $\mathcal{S}_i$ do
            upper bounds:= Euclidean distances between $Q_{i_j}$ and weights of each node.
            lower bounds:= LB_Keogh between $Q_{i_j}$ and weights of each node using the $W$.
            // Prune the set of all nodes to the set of qualified nodes.
            Qualified:= Set of nodes whose weights have a lower bound with $Q_{i_j} \leq$ min(upper bounds).
            Best Matching Unit:= Compute DTW distance between $Q_{i_j}$ and weights of nodes in Qualified. The Best Matching Unit (BMU) is the node whose weights are most similar to $Q_{i_j}$.
        end
    end
    for each time series $Q_i(i = 1, 2, ..., n)$ do
        // Update the node weights of the BMU (and its neighborhood) identified for $Q_i$ using a gradient descent based on learning rate, to more closely match $Q_i$.
        for Weights $(t_{1..m})$ of BMU and its neighbor nodes do
            $t_m = t_m + r \times (q_m - t_m)$
        end
    end
    // Update the neighborhood size and the learning rate in SOM.
    $N_c := \sqrt{M}/2 \times (1 - p/\#epochs)$
    $r := 0.9 \times (1 - p/\#epochs)$
end
return Best Matching Units
```

Pseudocode 1: SOMTimeS algorithm pseudocode

3.2 K-TimeS: generalization of the pruning strategy to K-means

For a more equitable comparison of the speed-up and clustering quality, we embed DTW with a similar pruning strategy into a K-means clustering algorithm, and call this algorithm—K-TimeS. Given some desired number of classes, *k*, training comprises two phases. In phase one, centroids are calculated, and in the phase two, the observations (i.e., query time series) are assigned to their closest centroid using DTW as the distance measure. This continues until some termination condition is met (e.g., number of iterations or convergence). The DTW distance bounding is used in the second phase to prune unnecessary DTW computations required in identifying the closest centriod. Specifically, we assign a centroid to the pruned region of Fig. 5 when it's lower bound (i.e., *LB_Keogh*) distance to a given query time series is greater than the minimum of the upper bound (i.e., Euclidean) distances between the k centroids and the query time series (see pruning threshold in Fig. 5). The pruned centroid cannot be the closest centroid to the given query time series, since there exists at least one other centroid whose DTW distance to the query time series is less than or equal to the pruned centroid's. The computational speed-up achieved in the K-TimeS algorithm results from avoiding the expensive DTW computation between the pruned centroid and the query time series.

4 Performance evaluations

The performance of SOMTimeS may be quantified in three important ways: (1) clustering accuracy (Sect. 4.1), (2) pruning speedup (Sects. 4.2), and (3) scalability (Sect. 4.3). Accuracy is reported using the six assessment metrics of Table 1. These include the Adjusted Rand Index (ARI) (Santos and Embrechts 2009), Adjusted Mutual Information (AMI) (Romano et al. 2016), the Rand Index (RI) (Hubert and Arabie 1985), Homogeneity (Rosenberg and Hirschberg 2007), Completeness (Rosenberg and Hirschberg 2007), and Fowlkes Mallows index (FMS) (Fowlkes and Mallows 1983). We report the speed-up achieved for SOMTimeS and K-TimeS using the 112 UCR datasets. For scalability, we report the number of DTW computations and execution time as a function of *problem size*, defined as $\sum_{i=1}^{n} |Q|_i$, where $|Q|$ is the length of times series Q, and n is the total number of time series in the dataset. The presence of a few large datasets in the archive makes it more informative to visualize problem size as the natural logarithm (see Figure S1 in Supplementary Material). Additionally, we report clustering accuracy in terms of ARI for the original Kohonen self-organizing map (i.e., using Euclidean distance); see Supplemental Material Table S1.

4.1 Clustering accuracy

The use of DTW distance bounding during pruning does not effect the clustering accuracy (see values of the assessment indices in Table 1). When the number of iterations (i.e., epochs) through the dataset are fixed at 10, the assessment indices for SOMTimeS and K-TimeS algorithms are comparable for the 112 datasets from the UCR archive. In practice, however, the SOM typically requires more passes through the dataset than K-means to achieve optimal accuracy. Thus, when 100 epochs are used, the SOMTimeS achieves a higher accuracy than K-TimeS for 5 of the 6 measures. Density Peaks, on the other hand, has lower average values for all six assessment indices.

Because the performance (i.e., accuracy measures) for a given clustering algorithm is dependent on the dataset (see Table 1), we use the ARI (Adjusted Rand Index) to assess accuracy between the three DTW-based algorithms in this work. The latter is recommended as one of the more robust measures for assessing accuracy across datasets (Milligan and Cooper 1986; Javed et al. 2020b). The ARI scores

Table 1 Clustering performance shown for K-TimeS, SOMTimeS, TADPole, and their respective non DTW-distance pruning counterparts using six assessment indices

Algorithm	ARI[a]		AMI[b]		RI[c]		H[d]		C[e]		FMS[f]	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
K-TimeS—10-iterations (with and without pruning)	0.22	0.23	0.28	0.26	0.66	0.22	0.29	0.26	0.40	0.32	0.49	0.19
SOMTimeS—10-iterations (with and without pruning)	0.21	0.23	0.27	0.24	0.70	0.15	0.28	0.24	0.31	0.26	0.47	0.20
SOMTimeS—100-iterations (with and without pruning)	0.24	0.23	0.30	0.26	0.71	0.16	0.31	0.25	0.35	0.28	0.50	0.19
TADPole—non-iterative (with and without pruning)	0.16	0.25	0.24	0.27	0.62	0.18	0.25	0.26	0.36	0.31	0.51	0.20

Values represent averages over the 112 datasets in the UCR archive; indices closer to 1 represent better performance

*All algorithms were executed on same computational machine—dual 20-Core Intel Xeon E5-2698 v4 2.2 GHz machine with 512 GB 2,133 MHz DDR4 RDIMM

[a] Adjusted Rand Index (Santos and Embrechts 2009)

[b] Adjusted Mutual Information (Romano et al. 2016)

[c] Rand Index (Hubert and Arabie 1985)

[d] Homogeneity (Rosenberg and Hirschberg 2007)

[e] Completeness (Rosenberg and Hirschberg 2007)

[f] Fowlkes Mallows index (Fowlkes and Mallows 1983)

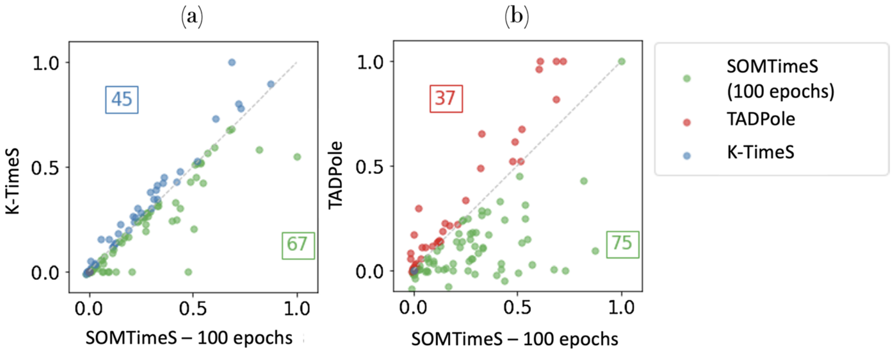

Fig. 6 ARI scores for the SOMTimeS (shown in green) vs. **a** K-TimeS (blue), and **b** TADPole (red) across all 112 of the UCR datasets (Color figure online)

Table 2 Execution time in hours for clustering 112 of the UC Riverside datasets using K-TimeS, SOMTimeS, TADPole and their respective ”no-DTW distance pruning” counterparts

Algorithm	Iterations	Execution time (h)	
		(with pruning)	(without pruning)
K-TimeS	10	6	13
SOMTimeS	10	7	14
SOMTimeS	100	58	148
TADPole	Non-iterative	1011	> 1011

for SOMTimeS at 100 epochs is plotted against (a) K-TimeS and (b) TADPole in Fig. 6 for each of the 112 URC datasets. The ARI scores shown in green (67 of the 112 datasets) lying below the 45-degree line of panel (a) represent higher accuracy for SOMTimeS; those above the diagonal (in blue) indicate that K-TimeS outperforms SOMTimeS for 45 of the 112 datasets. Fig. 6b shows SOMTimeS has higher accuracy for 75 of the 112 datasets compared to TADpole (in green), and lower accuracy for the remaining 37 datasets (red).

4.2 Pruning speed-up

While the speed-up times vary by dataset, the DTW distance pruning improves the execution time by a factor of 2x (on average over all 112 datasets) for both SOMTimeS and K-TimeS (see Fig. 7). K-TimeS is faster because it requires fewer iterations through the data, followed by SOMTimeS and TADPole, respectively. When the epochs are fixed at 10, both K-TimeS and SOMTimeS have comparable execution times. However, because SOMTimeS may achieve higher accuracy with more passes through the dataset (see Table 1), the execution time increases sub-linearly from ∼ 7 to ∼ 58 h when the number of epochs are increased from 10 to 100 (see Table 2). TADPole, with an algorithm complexity of $O(n^2)$, took 1011 h to cluster all the 112 datasets. Execution time for Density Peaks is listed as > 1011 h.

Total number of pruned DTW computations K-TimeS pruned more than 50% of the DTW calculations for 34 of the datasets; whereas, SOMTimeS (with epochs set to 10 and 100, respectively) pruned more than 50% of the DTW calculations for 8 and 21 of the 112 UCR datasets, respectively. TADPole pruned more than 50% of the DTW calculations for 40 of the datasets (see Figure S2 in Supplementary Material). Despite the pruning advantage of TADPole, its quadratic complexity O(n^2) results in more DTW computations (compared to O(n) in SOMTimeS), particularly for larger datasets. As a result, SOMTimeS is 17x faster when clustering the 112 datasets.

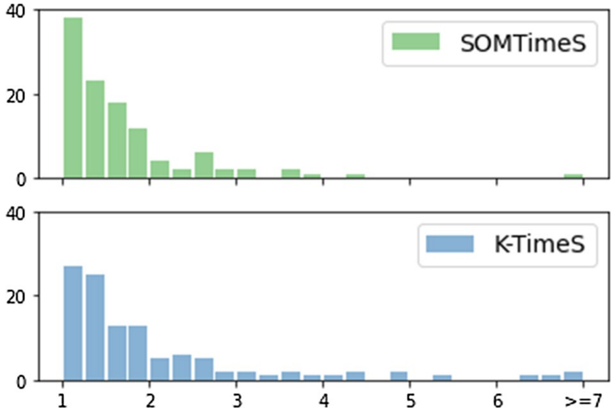

Fig. 7 Speed-up factor achieved for the UCR archive datasets

SOMTimeS is particularly well-suited to parallel execution, and when implemented in parallel, the execution time decreases as a function of available CPU. To cluster the 112 datasets in the UCR archive, SOMTimeS (at 100 epochs) took 3 h using 20 CPUs, and only 20 min when the number of SOM epochs was set to 10.

4.3 Scalability

We studied the scalability of SOMTimeS under two difference scenarios—(1) scaling of DTW computations performed as a function of the number of input time series, and (2) change in the SOMTimeS pruning rate as a function of epochs.

Scaling of DTW computations performed as a function of number of input time series Because TADPole has complexity O(n^2), the number of calls to DTW increases quadratically with the number (n) of input time series. As a result, there is a threshold (in terms of n) at which the number of calls to the DTW function is less for SOMTimeS than that of TADPole (see Fig. 8). This threshold depends on the number of training epochs, and is empirically observed to be approximately n = 100 and n = 2500, for 10 and 100 epochs, respectively. SOMTimeS and K-TimeS have similar complexities, both theoretical and empirical, when the mesh size in SOMTimeS is equal to k.

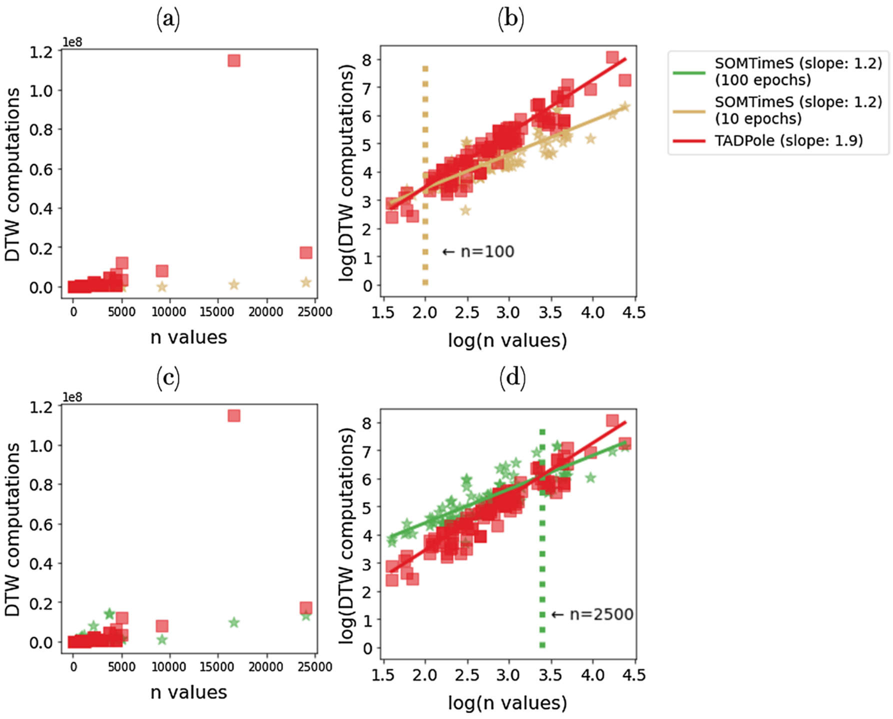

Fig. 8 DTW computations performed as a function of dataset size using linearly scaled axes (**a**, **c**) and log-log (**b**, **d**) for TADPole (200 million DTW computations shown in red) and SOMTimeS (13 million computations at 10 epochs shown in gold; and 100 million computations at 100 epochs shown in green) (Color figure online)

When clustering over all 112 of the UCR archived datasets, SOMTimeS computed the DTW measure 13 million and 100 million times (at 10 and 100 epochs), respectively. K-TimeS computed the DTW measure 8 millions times; while TADPole by comparison computed DTW 200 million times (see Fig. 8). At the dataset level, SOMTimeS had fewer calls for only 12 of the datasets (when using 10 epochs) compared to K-TimeS at 10 iterations. In comparison to TADPole, SOMTimeS had fewer calls for 88 of the datasets (when using 10 epochs), and 26 of the datasets (for 100 epochs). However, the quality of clustering for SOMTimeS at 100 epochs increases for 4 of the six assessment indices compared to K-TimeS, and TADPole.

Change in the SOMTimeS pruning rate as a function of epochs When we examine the pruning effect as a function of epochs, both the number of DTW calls and the execution time decrease as the number of epochs increases. As the nodes of the SOM mesh organize, more nodes get pruned; and hence, fewer nodes exist in the unpruned region (i.e., potential BMU region of Fig. 5), which decreases the need for DTW calls. Figure 9a shows the total number of calls to the DTW function made for each dataset, normalized over all epochs. The dashed

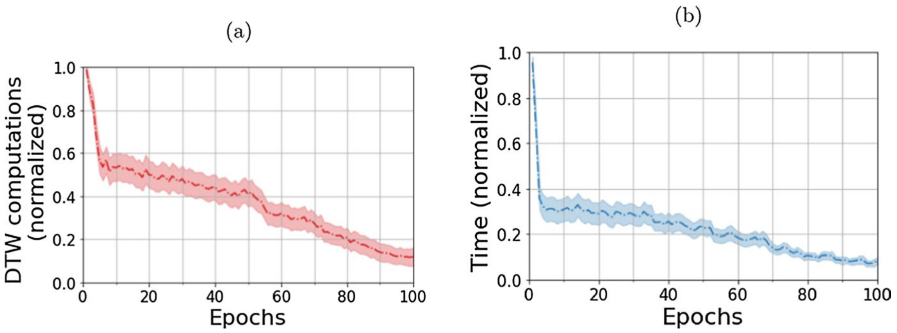

Fig. 9 Change in the SOMTimeS pruning effect as the number of epochs increases measured as the normalized **a** number of calls to the DTW function and **b** execution time. The dashed line represents the mean value for all datasets after individually normalizing each dataset run over all epochs. The shaded region corresponds to 95% confidence interval around the mean

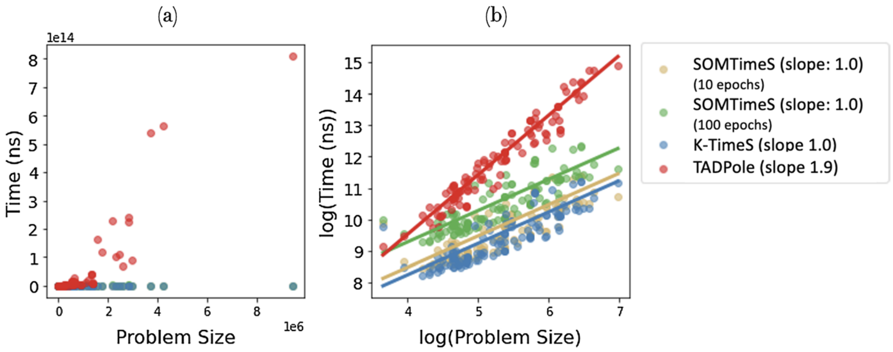

Fig. 10 Execution time of SOMTimeS, K-TimeS, and TADPole for the 112 archived UCR datasets on a **a** linear, and **b** log scaled axes

line represents the average number of calls over all datasets and the shaded region shows the 95% confidence interval. Figure 9b shows the corresponding normalized execution time. Both normalized DTW calls and execution time per epoch steadily decrease with increasing number of epochs and iterative updating of SOM weights. The elbow point, where further epochs result in a diminishing reduction of DTW calculations, is at the 6th epoch. This is called the swapover point and occurs when the self-organizing map moves from gross reorganization of the SOM weights to fine-tuning of the weights.

Finally, Fig. 10 shows how SOMTimeS execution time scales with the problem size ($\sum_{i=1}^{n} |Q|_i$, where $|Q|$ is the length of times series Q, and n is the total number of time series in the dataset). It increases at a lower rate than TADPole and is similar to K-TimeS. TADPole increases at the highest rate, consistent with its O(n^2) complexity of DTW calculations, followed by K-TimeS with a complexity of O($n \times k \times$ number of iterations), where k is the number of clusters. SOMTimeS has complexity of O($n \times k \times e$), where e is the number of epochs. K-TimeS (at 10 iterations) is slightly faster per unit problem size than SOMTimeS (at 10 epochs) because it (1) has pruned slightly more DTW calculations, and (2) does not require weight updates.

5 Application to serious illness conversations

We apply SOMTimeS to healthcare communication research in order to demonstrate the utility for identifying clinically relevant cluster analysis during complex patient-clinician interactions. We use direct observation data collected as part of the Palliative Care Communication Research Initiative (PCCRI) cohort study (Gramling et al. 2015) to demonstrate the algorithm's utility with temporally sequenced natural language. The PCCRI is a multisite, epidemiological study that includes verbatim transcriptions of audio-recorded palliative care consultations involving 231 hospitalized people with advanced cancer, their families, and 54 palliative care clinicians. The conversations have been transcribed by human experts for natural language processing (NLP) tasks with the goal of incentivizing high-quality communication. In order to avoid sparse decile-level data in shorter conversations, we selected 171 of the conversations that were longer than 15 min in duration. The average duration of conversations was 36 ± 16 minutes, with an average of 327 ± 176 speaker turns and 4734 ± 2280 words. All patients completed pre- and post-conversation surveys; information gathered included age, gender, personal identity, and a short number of questions that ranged from self-rated optimism to satisfaction with the consultation.

5.1 Discovering a clinically meaningful taxonomy of healtcare communication

Understanding and improving healthcare communication requires scalable approaches for characterizing what actually happens when patients, families, and clinicians interact in samples that are large enough to represent the diverse cultural, dialectical, decisional and clinical contexts in which these phenomena occur (Gramling et al. 2021; Tarbi et al. 2022; Clarfeld et al. 2021). Unfortunately, historical approaches to conversation analyses are too cumbersome, costly, and time-intensive to achieve this scalability, thus limiting our existing empirical knowledge about serious illness communication (Tulsky et al. 2017). Discovering and exploring clinically important patterns (i.e., clusters) of inter-personal communication amid the complex, dynamic and relational nature of clinical conversations presents a timely opportunity for scalable unsupervised machine learning methods to define a provisional taxonomy of healthcare communication. SOMTimeS is equipped to meet the need in two important ways. First, as described above, SOMTimeS offers substantial improvement in analytic efficiency necessary for identifying patterns in communication dynamics and content that unfold naturally over the temporal course of conversations. Second, as we will describe further below, the capacity for SOMTimeS to intuitively present results without pre-defining the number of clusters offers scientists of complementary disciplinary training (e.g., linguistics, anthropology, medicine, nursing, epidemiology, computer science) to empirically evaluate the number

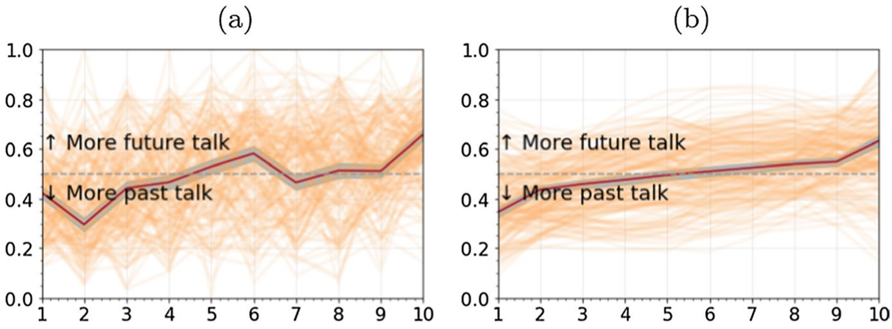

Fig. 11 Temporal plot showing the **a** raw time series, and **b** smoothed time series for all conversations superimposed in brown; the red line represents the mean values, and the shaded region around the red lines represents 95% confidence interval (Color figure online)

and clinical meaningfulness of communication types. Here, we use one feature of health communication (i.e., temporal ordering of conversation content) (Jaworski 2014; Labov 2013, 1980) foundational to narrative medicine (Charon 2001; Charon et al. 2016; Charon and Montello 2006, 2002).

Narrative medicine is a robust field of healthcare practice (Charon et al. 2016; Charon and Montello 2006, 2002), training and science that is grounded in the near culturally ubiquitous ways in which people explore, find and share meaning about their life experiences through stories, particularly in contexts of serious and life-threatening illness (Gramling et al. 2021; Labov 2013; Reblin et al. 2022; Barnato et al. 2016; Edlmann et al. 2019). Our previous work demonstrates that computational narrative analysis is a useful framework for characterizing the complex temporal "story arc" of naturally occurring healthcare conversations between seriously ill persons, their clinicians, and their family members (Ross et al. 2020). A central feature of narrative analysis is called temporal reference and indicates how conversation participants dynamically order the story "events" (e.g., topics, experiences, worries) happening in the past, present or future (Jaworski 2014; Labov 2013, 1980; Ross et al. 2020) Here, we demonstrate how SOMtimeS can be useful to explore clinically meaningful clusters of conversation "story arcs" using the tense of verbs and verb phrases as lexical markers of temporal reference.

5.2 Data pre-processing: verb tense as a time series

We used a temporal reference tagger (Ross et al. 2020) to assign temporal reference (past, present, or future) to verbs and verb modifiers in the verbatim transcripts. Specifically, the Natural Language Toolkit (www.nltk.org) was used to classify each word in the transcripts into a part of speech and for any word classified as a verb, the preceding context is used to assign that verb (and any modifiers) to a given temporal reference. Then, each conversation was stratified into deciles of "narrative time" based on the total word count for each conversation, and a temporal reference (i.e., verb tense) time series was generated for each conversation as the proportion of all future tense verbs relative to the total number of past and future tense verbs

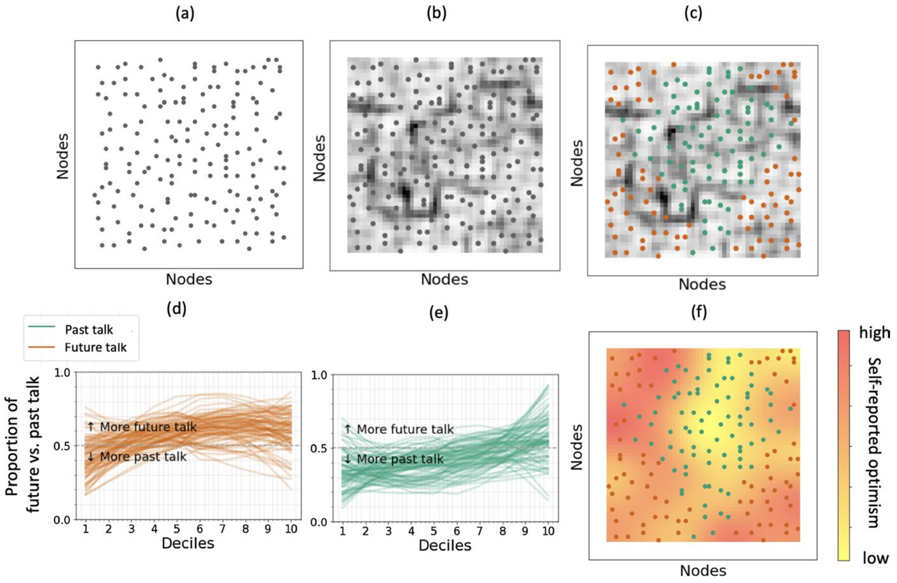

Fig. 12 Temporal reference time series data from 171 serious illness conversations **a** self-organized on a trained 2-D map, **b** with U-matrix added, **c** color-coded into two clusters. **d**, **e** Input time series clustered as future and past talk, respectively, and **f** a heatmap of trait optimism superimposed on the self-organized map

(see Supplemental Material Figure S3a). The vertical axis in Fig. 11 represents the proportion of future vs. past talk on a per decile basis, where any value above the threshold (dashed line = 0.5) represents more future talk. Each of the 171 temporal reference time series (see Fig. 11a) were then smoothed using a 2nd-order, 9-step Savitzky-Golay filter (Savitzky and Golay 1964) (see Fig. 11b and Supplemental Material S3b). The latter smooths the time series by fitting a polynomial to data within a moving window and then uses the polynomial to replace data values. Visual inspection of the time series plots (e.g. Figure 11) showed that a DTW window size of 10% (or 1 decile) to be adequate for aligning the future vs. past talk time series.

5.3 SOM clustering and graphical representation of temporal reference "arcs"

The SOMTimeS graphical user interface (GUI) offers important methodological advantages for conversation scientists to directly evaluate the potential number and meaningfulness of clusters. SOMTimeS maps the temporal reference time series for each conversation onto a standard SOM two-dimensional graph (Fig. 12a), thus allowing human analysts to "see" multi-dimensional relationships in two-dimensional space. Because the absolute graphical distance between observations is not a direct measure of temporal reference arc similarity, adding a U-matrix (Fig. 12b) offers "topographical" boundaries marking prominent differences in time-series patterns. When considering a two-cluster solution (Fig. 12c), we observe that the peripheral region of the toroidal SOM mesh is characterized by one cluster; while a second cluster is observed in the central region. This GUI offers analysts visually intuitive access to evaluate the observations in two important and complementary ways. The first uses qualitative methods, such as those used in linguistic and anthropological sciences, to systematically sample conversations from seemingly discrete "geographic regions" of the map to explore whether and how conversation dynamics (e.g., turn-taking etiquette, power dynamics of voice and topics, empathic expressions) indicate clinically meaningful sub-types of interactions.

The second advantage of the SOMTimeS GUI is the ability to overlay onto the same map information about each observation that was not used for clustering, but that may be conceptually relevant to evaluating the clinical relevance of the SOM regions. For example, other work identifies that an important personality trait - the tendency for how optimistically or pessimistically seriously ill people react to uncertainty—has important clinical implications for the process and outcomes of how clinicians, patients and families discuss the future (Robinson et al. 2008; Ingersoll et al. 2019). At the time of study enrollment, patient participants in the PCCRI self-reported their degree of trait optimism on an ordinal scale from high to low. Figure 12f shows a heatmap of trait optimism scores that visually indicate a potentially strong association with regions of the SOM. When considering the two-cluster solution, we observe that the peripheral region is characterized by relatively higher optimism and more future-oriented temporal reference "story arcs" (Fig. 12d) versus a central region with lower optimism and less future-orientated shapes of conversations (Fig. 12e). When considered statistically, trait optimism is significantly associated with peripheral versus central cluster assignment (chi square $p < 0.05$).

We propose that the scalability of SOMTimeS for efficiently evaluating time-series phenomena in dynamic clinical conversations, such as story arcs, and the intuitive GUI offers an exceptional opportunity for multi-method research focused on discovering and evaluating clinically meaningful types of healthcare communication.

6 Discussion

We present SOMTimeS as a clustering algorithm for time series that exploits the competitive learning of the Kohonen self-organizing map, a pruning strategy and the distance bounds of DTW to improve execution time. SOMTimeS contrasts with other DTW-based clustering algorithms in both its ability to both reduce the dimensionality of, and visualize input features associated with clustering temporal data. We also implemented a similar DTW-distance pruning strategy in K-means for the first time to demonstrate performance gains achieved for what is likely the most popular clustering algorithm to date. The resulting algorithm, K-TimeS, is faster than SOMTimeS because it requires fewer iterations through the data. In terms of accuracy, SOMTimeS has higher assessment indices compared to TADPole, and while the assessment indices are statistically similar with K-TimeS, the additional functionality of the SOM comes with a higher computational cost.

The benchmark experiments in this work are intended to put SOMTimeS in context with state-of-the-art time series clustering algorithms. Keeping the study objectives in mind, execution times are used to demonstrate scalability, and highlight the feasibility of analyzing large time series datasets using SOMTimeS. K-means is perhaps the most popular clustering algorithm and has been proven time and again to outperform state-of-the-art algorithms; however, because of its simplicity, it lacks the interpretability and visualization capabilities of the SOM. TADPole on the other hand, is a state-of-the-art clustering algorithm that organizes data differently from SOMTimeS (and by extension K-TimeS), as evident from the difference in ARI scores (see Fig. 6a), and choice of centroids (i.e., density peaks; see Supplementary Material Section 8.1). For these reasons, the algorithms tested are not direct competitors of one another and each has advantages in their own right.

SOMTimeS learns (i.e., self-organizes) in an iterative manner such that as the number of SOM epochs increase, the execution time per epoch decreases (see Fig. 9b), making higher number of epochs (and thus, corresponding assessment indices) feasible. This reduction in time is also directly proportional to the number of calls to the DTW function at each epoch. The elbow point (at 6 for SOMTimeS with 100 epochs) indicates quick gains in pruning DTW calculations. This same gain is observed when the total number of epochs is set to 10 or 50 (see Supplementary Material Figure S4). SOMTimeS took 40 min to cluster the entire UCR archive using 10 epochs, and less than 300 min when the number of epochs was increased 10-fold. Similarly, the largest dataset in terms of problem size took 5 min to cluster using 10 epochs, and 35 min to cluster at 100 epochs. SOMTimeS demonstrates sub-linear scalability when it comes to increasing the number of epochs. The scalability, fast execution times, and the ease of saving the state (weights) of a SOM make SOMTimeS a potential candidate for an *anytime* algorithm. It possesses the five most desirable properties of anytime algorithms (Zilberstein and Russell 1995; Zhu et al. 2012).

Concluding remarks This paper presents a computationally efficient variant of the SOM that uses DTW as a distance measure and a DTW-distance pruning strategy. For comparison purposes, we also implement a similar pruning strategy for K-means, called K-TimeS, and to put its performance in context, we present another state-of-the-art algorithm, TADPole. All three use DTW-distance pruning, and each has their own strengths and weaknesses. They each organize data differently, and the SOM is known for data visualization, dimensionality reduction, and feature selection. For these reasons a direct comparison of the advantages and disadvantages of each algorithm is not possible. SOMTimeS has unique data visualization abilities that require the mesh size to be increased to a value higher then k. The pruning strategy presented in this work makes the latter feasible. However, if only classification (or hard clusters) are required, then $K - TimeS$ is the faster and equally accurate clustering algorithm.

7 Conclusion and future work

The explosion in volume of time series data has resulted in the availability of large unlabeled time datasets. In this work, we introduce **S**elf-**O**rganizing **M**aps for **t**ime **s**eries (**SOMTimeS**). SOMTimeS is a self-organizing map for clustering and classifying time series data that uses DTW as a distance measure of similarity between time series. To reduce run time and improve scalability, SOMTimeS prunes DTW calculations by using distance bounding during the SOM training phase. This pruning results in a computationally efficient and fast time series clustering algorithm that is linearly scalable with respect to increasing number of observations. SOMTimeS clustered 112 datasets from the UCR time series classification archive in 7 hours with state-of-art accuracy. We also implemented a similar pruning strategy in K-means for the first time to demonstrate performance gains achieved for what is likely the most popular clustering algorithm to date. K-TimeS clustered all the 112 datasets in 6 hours with accuracy scores comparable to SOMTimeS; however, the former lacks the visualization capabilities of the SOM. We applied SOMTimeS to 171 conversations from the PCCRI dataset. The resulting clusters showed two fundamental shapes of conversational stories.

To further improve computational efficiency and clustering accuracy, newer and state-of-the-art variations of SOMs may be used that leverage the same pruning strategy in this work. Improving computational time of DTW-based algorithms is an active area of research, and any improvement in computational speed of DTW can be incorporated in SOMTimeS for the unpruned DTW computations. Finally, SOMTimeS is a uni-variate time series clustering algorithm. To create a multivariate time series clustering algorithm, the pruning strategy will have to be revisited to accommodate the variations of DTW for multi-variate time series. SOMTimeS is a fast and linearly scalable algorithm that recasts DTW as a computationally efficient distance measure for time series data clustering.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10618-023-00979-9.

Acknowledgements This project was supported by the Richard Barrett Foundation and Gund Institute for Environment through a Gund Barrett Fellowship. Additional support was provided by the Vermont EPSCoR BREE Project (NSF Award OIA-1556770), and the National Science Foundation under Grant No. EAR 2012123. We thank Dr. Patrick J. Clemins of Vermont EPSCoR, for providing support in using the EPSCoR Pascal high-performance computing server for the project. Computations were performed, in part, on the Vermont Advanced Computing Core. We also thank all the curators and administrators of the UCR archive. Data used to illustrate concepts in this paper arise from the Palliative Care Conversation Research Initiative (PCCRI). The PCCRI was funded by a Research Scholar Grant from the American Cancer Society (RSG PCSM124655; PI: Robert Gramling).

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Al-Naymat G, Chawla S, Taheri J (2009) Sparsedtw: a novel approach to speed up dynamic time warping. In: Proceedings of the eighth Australasian data mining conference—volume 101, AusDM '09. AUS. Australian Computer Society, Inc, pp 117–127

Alvarez-Guerra M, González-Piñuela C, Andrés A, Galán B, Viguri JR (2008) Assessment of self-organizing map artificial neural networks for the classification of sediment quality. Environ Int 34(6):782–790

Barnato A, Schenker Y, Tiver G, Dew M, Arnold R, Nunez E, Reynolds C (2016) Storytelling in the early bereavement period to reduce emotional distress among surrogates involved in a decision to limit life support in the icu: a pilot feasibility trial. Crit Care Med 45:1

Begum N, Ulanova L, Wang J, Keogh E (2015) Accelerating dynamic time warping clustering with a novel admissible pruning strategy. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 49–58

Bende-Michl U, Verburg K, Cresswell HP (2013) High-frequency nutrient monitoring to infer seasonal patterns in catchment source availability, mobilisation and delivery. Environ Monit Assess 185(11):9191–9219

Bentley F, Luvogt C, Silverman M, Wirasinghe R, White B, Lottridge D (2018) Understanding the long-term use of smart speaker assistants. Proc ACM Interact Mob Wearable Ubiquitous Technol 2(3):1–24

Charon R (2001) Narrative medicine: a model for empathy, reflection, profession, and trust. JAMA 286(15):1897–1902

Charon R, Montello M (2002) Stories matter: the role of narrative in medical ethics. Routledge, New York

Charon R, Montello M (2006) Narrative medicine: honoring the stories of illness. Oxford University Press, Oxford

Charon R, DasGupta S, Hermann N, Irvine C, Marcus ER, Rivera Colsn E, Spencer D, Spiegel M (2016) The principles and practice of narrative medicine. Oxford University Press, Oxford

Chu E, Dunn J, Roy D, Sands G (2017) Ai in storytelling: machines as cocreators. https://www.mckinsey.com/industries/technology-media-andtelecommunications/our-insights/ai-in-storytelling

Clarfeld LA, Gramling R, Rizzo DM, Eppstein MJ (2021) A general model of conversational dynamics and an example application in serious illness communication. PLoS ONE 16(7):1–19

Conan-Guez B, Rossi F, El Golli A (2006) Fast algorithm and implementation of dissimilarity self-organizing maps. Neural Netw 19(6):855–863 (**Advances in Self Organising Maps - WSOM &apos;05**)

CRS (2020) The internet of things (iot): an overview. https://fas.org/sgp/crs/misc/IF11239.pdf

Dau HA, Keogh E, Kamgar K, Yeh C-CM, Zhu Y, Gharghabi S, Ratanamahatana CA, Yanping, Hu B, Begum N, Bagnall A, Mueen A, Batista G, Hexagon-ML (2018) The UCR time series classification archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

de Abreu de Sousa MA, Pires R, Del-Moral-Hernandez E (2020) Somprocessor: a high throughput fpga-based architecture for implementing self-organizing maps and its application to video processing. Neural Netw 125:349–362

De Bie T, Lijffijt J, Mesnage C, Santos-Rodríguez R (2016) Detecting trends in twitter time series. In: 2016 IEEE 26th international workshop on machine learning for signal processing (MLSP), pp 1–6

Dias LA, Damasceno AM, Gaura E, Fernandes MA (2021) A full-parallel implementation of self-organizing maps on hardware. Neural Netw 143:818–827

Ding H, Trajcevski G, Scheuermann P, Wang X, Keogh E (2008) Querying and mining of time series data: experimental comparison of representations and distance measures. Proc VLDB Endow 1(2):1542–1552

Dupas R, Tavenard R, Fovet O, Gilliet N, Grimaldi C, Gascuel-Odoux C (2015) Identifying seasonal patterns of phosphorus storm dynamics with dynamic time warping. Water Resour Res 51(11):8868–8882

Edlmann T, Hayes T, Brown L (2019) Storytelling: global reflections on narrative, pp 1–4

Eshghi A, Haughton D, Legrand P, Skaletsky M, Woolford S (2011) Identifying groups: a comparison of methodologies. J Data Sci 9:271–291

Evans D (2011) The internet of things. How the next evolution of the internet is changing everything. Technical Report MSU-CSE-06-2, Cisco Systems

Ewen J (2011) Hydrograph matching method for measuring model performance. J Hydrol 408(1):178–187

Flanagan K, Fallon E, Connolly P, Awad A (2017) Network anomaly detection in time series using distance based outlier detection with cluster density analysis. In: Proceedings of the 2017 Internet technologies and applications, pp 116–121

Fowlkes EB, Mallows CL (1983) A method for comparing two hierarchical clusterings. J Am Stat Assoc 78(383):553–569

Gharehbaghi A, Lindén M (2018) A deep machine learning method for classifying cyclic time series of biological signals using time-growing neural network. IEEE Trans Neural Netw Learn Syst 29(9):4102–4115

Gold O, Sharir M (2018) Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. ACM Trans Algorithms 14(4):1–17

Gramling R, Gajary-Coots E, Stanek S, Dougoud N, Pyke H, Thomas M, Cimino J, Sanders M, Chang Alexander S, Epstein R, Fiscella K, Gramling D, Ladwig S, Anderson W, Pantilat S, Norton S (2015) Design of, and enrollment in, the palliative care communication research initiative: a direct-observation cohort study. BMC Palliat Care 14:40

Gramling R, Javed A, Durieux BN, Clarfeld LA, Matt JE, Rizzo DM, Wong A, Braddish T, Gramling CJ, Wills J, Arnoldy F, Straton J, Cheney N, Eppstein MJ, Gramling D (2021) Conversational stories & self organizing maps: innovations for the scalable study of uncertainty in healthcare communication. Patient Educ Couns 104:2616–2621

Gupta K, Chatterjee N (2018) Financial time series clustering. In: Information and communication technology for intelligent systems (ICTIS 2017), vol 2, pp 146–156

Hamami F, Dahlan IA (2020) Univariate time series data forecasting of air pollution using lstm neural network. In: 2020 international conference on advancement in data science, E-learning and information systems (ICADEIS), pp 1–5

Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218

Ingersoll L, Chang Alexander S, Ladwig S, Anderson W, Norton S, Gramling R (2019) The contagion of optimism: the relationship between patient optimism and palliative care clinician overestimation of survival among hospitalized patients with advanced cancer. Psycho-Oncology 28:1286–1292

Iorio C, Frasso G, D'Ambrosio A, Siciliano R (2018) A P-spline based clustering approach for portfolio selection. Expert Syst Appl 95:88–103

Javed A (2021a) Somtimes: self-organizing maps for time series. https://pypi.org/project/somtimes/

Javed A (2021b) Somtimes: self-organizing maps for time series. https://github.com/ali-javed/somtimes

Javed A, Lee BS (2016) Sense-level semantic clustering of hashtags in social media. In: Proceedings of the 3rd annual international symposium on information management and big data

Javed A, Lee BS (2017) Sense-level semantic clustering of hashtags. In: Lossio-Ventura JA, Alatrista-Salas H (eds) Information management and big data. Springer, Cham, pp 1–16

Javed A, Lee BS (2018) Hybrid semantic clustering of hashtags. Online Soc Netw Media 5:23–36

Javed A, Hamshaw SD, Lee BS, Rizzo DM (2020a) Multivariate event time series analysis using hydrological and suspended sediment data. J Hydrol 593:125802

Javed A, Lee BS, Rizzo DM (2020b) A benchmark study on time series clustering. Mach Learn Appl 1:100001

Jaworski ACN (2014) The discourse reader. Routledge, New York

Johnpaul C, Prasad MV, Nickolas S, Gangadharan G (2020) Trendlets: a novel probabilistic representational structures for clustering the time series data. Expert Syst Appl 145:113119

Keogh E (2002) Exact indexing of dynamic time warping. In: Proceedings of the 28th international conference on very large data bases, VLDB '02. VLDB Endowment, pp 406–417

Keogh E (2003) Efficiently finding arbitrarily scaled patterns in massive time series databases. In: Lavrač N, Gamberger D, Todorovski L, Blockeel H (eds) Knowledge discovery in databases: PKDD 2003. Springer, Berlin, pp 253–265

Kohonen T (2013) Essentials of the self-organizing map. Neural Netw 37:52–65 (**Twenty-fifth Anniversay Commemorative Issue**)

Kohonen T, Schroeder MR, Huang TS (2001) Self-organizing maps, 3rd edn. Springer, Berlin

Labov W (1980) Locating language in time and space. Quantitative analyses of linguistic structure. Academic Press, New York

Labov W (2013) The language of life and death: the transformation of experience in oral narrative. Cambridge University Press, Cambridge

Lasfer A, El-Baz H, Zualkernan I (2013) Neural network design parameters for forecasting financial time series. In: 2013 5th international conference on modeling, simulation and applied optimization (ICMSAO), pp 1–4

Lawrence RD, Almasi GS, Rushmeier HE (1999) A scalable parallel algorithm for self-organizing maps with applications to sparse data mining problems. Data Min Knowl Discov 3(2):171–195

Li K, Sward K, Deng H, Morrison J, Habre R, Franklin M, Chiang Y-Y, Ambite J, Wilson JP, Eckel SP (2020) Using dynamic time warping self-organizing maps to characterize diurnal patterns in environmental exposures. Res Square 11:24052

Lou Y, Ao H, Dong Y (2015) Improvement of dynamic time warping (dtw) algorithm. In: 2015 14th international symposium on distributed computing and applications for business engineering and science (DCABES), pp 384–387

Mangiameli P, Chen SK, West D (1996) A comparison of som neural network and hierarchical clustering methods. Eur J Oper Res 93(2):402–417 (**Neural Networks and Operations Research/Management Science**)

Mather AL, Johnson RL (2015) Event-based prediction of stream turbidity using a combined cluster analysis and classification tree approach. J Hydrol 530:751–761

Milligan G, Cooper M (1986) A study of the comparability of external criteria for hierarchical cluster analysis. Multivar Behav Res 21(4):441–58

Minaudo C, Dupas R, Gascuel-Odoux C, Fovet O, Mellander P-E, Jordan P, Shore M, Moatar F (2017) Nonlinear empirical modeling to estimate phosphorus exports using continuous records of turbidity and discharge. Water Resour Res 53:7590–7606

Obermayer K, Ritter H, Schulten K (1990) Large-scale simulations of self-organizing neural networks on parallel computers: application to biological modelling. Parallel Comput 14(3):381–404

Paparrizos J, Gravano L (2016) K-shape: efficient and accurate clustering of time series. SIGMOD Rec 45(1):69–76

Paparrizos J, Gravano L (2017) Fast and accurate time-series clustering. ACM Trans Database Syst 42(2):8:1-8:49

Parshutin S, Kuleshova G (2008) Time warping techniques in clustering time series. https://api.semanticscholar.org/CorpusID:1293241

Pirim H, Ekşioğlu B, Perkins AD, Yüceer C (2012) Clustering of high throughput gene expression data. Comput Oper Res 39:3046–3061

Rakthanmanon T, Campana B, Mueen A, Batista G, Westover B, Zhu Q, Zakaria J, Keogh E (2012) Searching and mining trillions of time series subsequences under dynamic time warping. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, pp 262–270

Ratanamahatana CA, Keogh E (2004) Everything you know about Dynamic Time Warping is wrong. In: Proceedings of the 3rd workshop on mining temporal and sequential data. Citeseer

Ratanamahatana C, Keogh E, Bagnall AJ, Lonardi S (2005) A novel bit level time series representation with implication of similarity search and clustering. In: Ho TB, Cheung D, Liu H (eds) Advances in knowledge discovery and data mining. Springer, Berlin, pp 771–777

Reagan AJ, Mitchell L, Kiley D, Danforth CM, Dodds PS (2016) The emotional arcs of stories are dominated by six basic shapes. EPJ Data Sci 5(1):1–12

Reblin M, Wong A, Arnoldy F, Pratt S, Dewoolkar A, Gramling R, Rizzo DM (2022) The storylistening project: feasibility and acceptability of a remotely delivered intervention to alleviate grief during the covid-19 pandemic. J Palliat Med 26(3):327–333

Robinson T, Chang Alexander S, Hays M, Jeffreys A, Olsen M, Rodriguez K, Pollak K, Abernethy A, Arnold R, Tulsky J (2008) Patient-oncologist communication in advanced cancer: predictors of patient perception of prognosis. Support Care Cancer 16:1049–57

Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344(6191):1492–1496

Romano S, Vinh NX, Bailey J, Verspoor K (2016) Adjusting for chance clustering comparison measures. J Mach Learn Res 17(1):4635–4666

Rosenberg A, Hirschberg J (2007) V-measure: a conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning, pp 410–420

Ross L, Danforth CM, Eppstein MJ, Clarfeld LA, Durieux BN, Gramling CJ, Hirsch L, Rizzo DM, Gramling R (2020) Story arcs in serious illness: natural language processing features of palliative care conversations. Patient Educ Couns 103(4):826–832

Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process 26(1):43–49

Salvador S, Chan P (2007) Toward accurate dynamic time warping in linear time and space. Intell Data Anal 11(5):561–580

Santos JM, Embrechts M (2009) On the use of the Adjusted Rand Index as a metric for evaluating supervised classification. In: Proceedings of the 19th international conference on artificial neural networks: Part II, pp 175–184

Savitzky A, Golay MJE (1964) Smoothing and differentiation of data by simplified least squares procedures. Anal Chem 36:1627–1639

Silva MI, Henriques R (2020) Exploring time-series motifs through dtw-som. In: 2020 International joint conference on neural networks, IJCNN 2020, Proceedings of the international joint conference on neural networks, pp. 1–8. United States, Institute of Electrical and Electronics Engineers Inc. Conference date: 19-07-2020 to 24-07-2020. https://doi.org/10.1109/IJCNN48605.2020.9207614

Somervuo P, Kohonen T (1999) Self-organizing maps and learning vector quantization for feature sequences. Neural Process Lett 10(2):151–159

Souto Md, Costa I, Araujo D, Ludermir T, Schliep A (2008) Clustering cancer gene expression data: a comparative study. BMC Bioinform 9:497

Tarbi EC, Blanch-Hartigan D, van Vliet LM, Gramling R, Tulsky JA, Sanders JJ (2022) Toward a basic science of communication in serious illness. Patient Educ Couns 105(7):1963–1969

Tulsky JA, Beach MC, Butow PN, Hickman SE, Mack JW, Morrison RS, Street RL, Sudore RLJ, White DB, Pollak KI (2017) A research agenda for communication between health care professionals and patients living with serious illness. JAMA Intern Med 177(9):1361–1366

Ultsch A (1993) Self-organizing neural networks for visualisation and classification. In: Opitz O, Lausen B, Klar R (eds) Information and classification. Springer, Berlin, pp 307–313

Wang J, Jiang J (2021) Unsupervised deep clustering via adaptive gmm modeling and optimization. Neurocomputing 433:199–211

Wei L, Keogh E, Van Herle H, Mafra-Neto A (2005) Atomic wedgie: efficient query filtering for streaming time series. In: Fifth IEEE international conference on data mining (ICDM'05), 8 pp

WorldEconomicForum (2019) How much data is generated each day? Technical report

Wu R, Keogh E (2020) Fastdtw is approximate and generally slower than the algorithm it approximates. IEEE Trans Knowl Data Eng PP:1–1

Wu C-H, Hodges RE, Wang C-J (1991) Parallelizing the self-organizing feature map on multiprocessor systems. Parallel Comput 17(6):821–832

Xi X, Keogh E, Shelton C, Wei L, Ratanamahatana CA (2006) Fast time series classification using numerosity reduction. In: Proceedings of the 23rd international conference on machine learning, ICML '06. Association for Computing Machinery, New York, NY, USA, pp 1033–1040

Zhu Q, Batista G, Rakthanmanon T, Keogh E (2012) A novel approximation to dynamic time warping allows anytime clustering of massive time series datasets. In: Proceedings of the 2012 SIAM international conference on data mining, pp 999–1010

Zilberstein S, Russell S (1995) Approximate reasoning using anytime algorithms. Springer US, Boston, pp 43–62

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Ali Javed[1,3] · Donna M. Rizzo[2,3] · Byung Suk Lee[3] · Robert Gramling[4]

✉ Ali Javed
ajaved@stanford.edu

[1] Department of Medicine, Stanford University, 300 Pasteur Dr, Stanford, CA 94305, USA

[2] Department of Civil and Environmental Engineering, University of Vermont, Burlington, VT, USA

[3] Department of Computer Science, University of Vermont, Burlington, VT, USA

[4] Department of Family Medicine, University of Vermont, Burlington, VT, USA