2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

An Invariance-guided Stability Criterion for Time Series Clustering Validation

Florent Forest*‡, Alex Mourer†‡, Mustapha Lebbah*, Hanane Azzag* and Jérôme Lacaille‡
*LIPN, Université Sorbonne Paris Nord, 93430 Villetaneuse, France
†SAMM, Université Paris 1 Panthéon Sorbonne, 75231 Paris, France
‡Safran Aircraft Engines, 77550 Réau, France
e-mail: forest@lipn.univ-paris13.fr

***Abstract*—Time series clustering is a challenging task due to the specificities of this type of data. Temporal correlation and invariance to transformations such as shifting, warping or noise prevent the use of standard data mining methods. Time series clustering has been mostly studied under the angle of finding efficient algorithms and distance metrics adapted to the specific nature of time series data. Much less attention has been devoted to the general problem of model selection. Clustering stability has emerged as a universal and model-agnostic principle for clustering model selection. This principle can be stated as follows: an algorithm should find a structure in the data that is resilient to perturbation by sampling or noise. We propose to apply stability analysis to time series by leveraging prior knowledge on the nature and invariances of the data. These invariances determine the perturbation process used to assess stability. Based on a recently introduced criterion combining between-cluster and within-cluster stability, we propose an invariance-guided method for model selection, applicable to a wide range of clustering algorithms. Experiments conducted on artificial and benchmark data sets demonstrate the ability of our criterion to discover structure and select the correct number of clusters, whenever data invariances are known beforehand.**

I. Introduction

Time series are a type of data naturally organized as sequences with a temporal dimension, such as values collected by sensors. Large volumes of unlabeled data are ubiquitous across various domains such as healthcare, industry, biology, astronomy, economy, the internet of things (IoT) and many others. Clustering, a widely used technique to gain insights from such data, consists in finding groups of elements called clusters such that elements sharing the same cluster are similar, and elements belonging to different clusters are dissimilar. Time series clustering (TSC) [1], [2] is a challenging task due to the temporal nature of the data, which implies high dimensionality [3], temporal feature correlation, invariance to transformations, and different lengths. Model selection for TSC in particular is not well studied in literature [2]. For instance, methods to select the number of clusters are rarely provided, although selecting the *best* or *natural* number of clusters is known to be one of the crucial problems in cluster analysis [4]–[6]. When external labels are unavailable, model selection is done using internal clustering validity indices [7]. Most indices are based on between-cluster and within-cluster distances, and could be used with any distance other than Euclidean (e.g. Silhouette with Manhattan distance [4]), but their application to time series has not been well studied [1]. These indices are generally used on extracted features, not raw time series (e.g. Davies-Bouldin in [8]). Heuristic methods with cross-correlation dissimilarity have been developed in [9]. For TSC based on autoregressive models, distances between ARMA/ARIMA models have been devised [10], [11]. In case of model-based clustering, such as mixture models, the AIC, BIC and ICL criteria have been widely used [12]–[15]. Still, the validation of time series clustering is unsolved in general.

Clustering stability [5], [6] has emerged as a natural and model-agnostic principle: an algorithm should find stable structures in the data. "To be meaningful, a clustering must be both good and the only good clustering of the data, *up to small perturbations*. Such a clustering is called *stable*. Data that contains a stable clustering is said to be clusterable" [16]. In statistical learning terms, if data sets are sampled from the same underlying distribution, an algorithm should find similar partitions. The data-generating distribution is unavailable in model-free clustering, thus perturbed data sets are obtained either by resampling or injecting noise into the original data. Limitations of this principle, in particular its ability to select the number of clusters, have been studied in [17]. It has been shown that a novel criterion called *Stadion* (stability difference criterion) is able to successfully discover structure and select the number of clusters when using additive noise perturbation. We base ourselves onto this work and extend it to time series which have their own specificities. It is known that temporal data are resilient to particular perturbations, which depend on the application and the physical nature of the observed phenomena. Thus, we leverage prior knowledge on the invariances of the data in order to assess stability of a clustering.

Invariant perturbations are already used for data augmentation, to improve the generalization capability of supervised classifiers. Suitable perturbations for various applications can be found in this literature, for example for time series [18], [19] or images [20], [21]. Transformation-invariant clustering algorithms have also been developed [22], [23]. To our knowledge, the first application of stability analysis to time series clustering comes from the financial field [24]. In their work, authors study the price of financial derivatives, namely credit default swaps. They compare the stability of weighted linkage clustering with different dissimilarities (Euclidean distance,

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

Pearson and Spearman correlations, and a combination of correlation and Hellinger distance between distributions). In order to assess stability, they devise a specialized perturbation framework for financial time series. This idea of leveraging prior knowledge on the nature and properties of the data is also what we would like to develop in this work. However, the approach remains focused on their business field and is application-specific. In addition, no quantitative stability scores are computed, and results are interpreted by visualizing the partitions. Finally, it does not tackle the problem of selecting the number of clusters. A second recent work [25] uses stability to evaluate fuzzy *over-time* clustering to detect correlated subsequences in multivariate time series. This approach is interested in time-point clustering (i.e. clustering individual time points of several series) and in particular the evolution of cluster structure over time. Differently, our work focuses on whole time series clustering. Moreover, they compute stability scores based on a resampling approach [26], whereas we adopt the framework of [17], using perturbation by noise.

The main contributions of this paper are the following:

- To our knowledge, this work is the first general application of stability analysis to time series clustering validation.
- We show that the stability difference criterion (Stadion) from [17] can effectively perform model selection, by letting data invariances guide the perturbation process. In particular, the criterion is able to select the number of clusters.
- Implementations are made available as part of `skstab`[1], a toolkit for clustering stability analysis in Python with a scikit-learn compatible API.

The rest of the paper is structured as follows: first, an overview of TSC algorithms and the concept of invariance is provided. Second, we introduce formally the definition of clustering stability and the Stadion criterion. The last section exposes the challenges of TSC stability and applies it to several model selection tasks.

II. Invariances and time series clustering

Clustering algorithms are always based on a notion of distance between elements of the data set. Distances between time series are only meaningful if they satisfy certain invariances: in other words, some sequences should be considered similar even if their raw feature values are different. It is not possible to choose an adequate distance measure without knowing what invariances are desirable for the specific task. For the same data set, several clustering solutions are possible, depending on these invariances. Hence, the problem of multiple clusterings is amplified [27]. Example of invariances are:

- **Scale** and/or **offset invariance**. In many cases, we want two series to be considered similar if they differ by an affine transformation (for example, if a value was measured in different physical units, like Celsius and Fahrenheit degrees).
- **Shift invariance**. If a same phenonemon is observed at different time points in two series, they should be considered identical.
- **Warping invariance**. This invariance is necessary if the phenonemon may have different speeds or delays, which is ubiquitous in motion and biological signals. Series can be aligned and matched using Dynamic Time Warping (DTW) [28].
- **Uniform temporal scaling invariance**. Unlike local scaling in warping, global scaling is necessary to match behaviors at different speeds or frequencies, yielding series with different lengths. A solution is to stretch series by a constant factor.
- **Occlusion invariance**. Parts of the input being unobserved should not change cluster membership.
- **Complexity** or **noise invariance**. [29] have shown that time series can have different *complexities*, and that complex time series tend to be closer to simpler time series than to other complex time series under Euclidean distance.

Euclidean distance, used in most traditional clustering algorithms that operate on tabular data (i.e. *flat* vectors of features), does not satisfy any of these invariances. Thus, a variety of dissimilarity measures between time series has been devised [30].

Time series clustering methods can be broadly divided into three categories [2]. Whole time series clustering considers each series as an invidual object. Subsequence clustering consists in clustering subsequences of a single time series, for example a measurement over a long period of time or real-time, streaming data. Time point clustering clusters the individual time observations, and is similar to segmentation. In this work, we only experiment with whole time series clustering.

On another level, clustering algorithms can be either based on raw time series, feature-based, or model-based [1]. Raw time series clustering algorithms define a distance between raw values in the time domain. Agglomerative clustering with single, complete or average linkage, and K-medoids (also called PAM for Partitioning Around Medoids) [4], [31] can be used with any distance between time series. Other widely used methods require the computation of an average in the sense of specific distance, such as K-DBA [32], K-SC [33] and K-shape [34]. Another approach uses *shapelets*, which are short salient subsequences that discriminate between classes. First proposed in supervised learning, unsupervised shapelets are also used for clustering [35], [36].

Feature-based approaches consist in removing the temporal dimension by extracting higher-level features and projecting the data into a space where euclidean distance and generic algorithms (e.g. k-means, agglomerative clustering, SVMs, decision trees) can be used. For instance, statistical features can be extracted, such as mean, variance, minimum and maximum values, number of peaks, etc. Then, a time series

[1] https://github.com/FlorentF9/skstab

can be projected into the frequency domain using a Fourier transform, extracting spectral features. Wavelets are another option. Another approach is to discretize the values taken by the series and aggregate the sequence into a *bag-of-features*, removing the temporal dimension, called piecewise aggregate approximation [37], [38]. Many successful methods in classification and clustering are based on combining bags of multiple time- and frequency-domain features [39], [40]. Finally, this includes deep learning approaches where a neural network learns representations from the raw time series values [41]–[44].

Another kind of approach learns the temporal behavior through autoregressive models, such as ARMA or recurrent neural networks, and cluster the resulting model parameters [10], [11]. Finally, model-based clustering estimates cluster membership probabilities using probabilistic models such as functional mixture models [45].

In this work, we will experiment with two widely used algorithms: K-medoids and K-shape. K-medoids is a center-based algorithm, but differently from K-means, instead of computing the mean (*centroid*) of each cluster, the center is the element minimizing the sum of the distances to every other element (and is called the *medoid*). It can be used with any dissimilarity measure. Here, we will use Euclidean (EUC), Correlation (COR) and Dynamic Time Warping (DTW) [28] distances, defined between two same-length series $\mathbf{x} = (x_1, \ldots, x_T)$ and $\mathbf{y} = (y_1, \ldots, y_T)$ as

$$\mathrm{EUC}(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2 = \sqrt{\sum_{t=1}^{T} (x_t - y_t)^2} \tag{1}$$

$$\begin{aligned} \mathrm{COR}(\mathbf{x}, \mathbf{y}) &= 1 - \mathrm{NCC}_0(\mathbf{x}, \mathbf{y}) \\ &= 1 - \frac{\sum_{t=1}^{T} (x_t - \bar{x}_t)(y_t - \bar{y}_t)}{||\mathbf{x} - \bar{\mathbf{x}}||_2 ||\mathbf{y} - \bar{\mathbf{y}}||_2} \end{aligned} \tag{2}$$

$$\mathrm{DTW}(\mathbf{x}, \mathbf{y}) = \min \sqrt{\sum_{i=1}^{P} w_i} \tag{3}$$

where the warping path $W = \{w_1, \ldots, w_P\}$ with $P \geq T$ is obtained using a dynamic programming approach on the pairwise distance matrix between the two series, based on following recurrence: $d(i,j) = \mathrm{EUC}(i,j) + \min\{d(i-1, j-1), d(i-1, j), d(i, j-1)\}$. It is common to constrain the warping path to a band around the diagonal, e.g. the Sakoe-Chiba band [28]. The invariances of K-medoids depend on the disance used: no invariance with EUC, scale invariance with COR and warping invariance with DTW. K-shape is a center-based algorithm using the shape-based distance (SBD), based on normalized cross-correlation:

$$\mathrm{SBD}(\mathbf{x}, \mathbf{y}) = 1 - \max_{w} \mathrm{NCC}_w(\mathbf{x}, \mathbf{y}) \tag{4}$$

where $w \in [-T, T]$ is the shifting of $\mathbf{x}$. K-shape is thus invariant to scaling and shifting, and is meant to be computationally efficient in its computation of averages.

III. Stability analysis

A data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ consists in N independent and identically distributed (i.i.d.) univariate or multivariate time series, drawn from a data-generating distribution $\mathcal{P}$ on an underlying space $\mathcal{X}$. We assume a clustering algorithm $\mathcal{A}$ takes as input the data set $\mathbf{X}$, the number of clusters $K \geq 1$, and outputs a clustering $\mathcal{C}_K = \{C_1, \ldots, C_K\}$ of $\mathbf{X}$ into K disjoint sets.

Let $\mathbf{X}$ and $\mathbf{X}'$ be two different data sets drawn from $\mathcal{P}$ and note $\mathcal{C}_K$ and $\mathcal{C}'_K$ their respective clusterings. Let s be a similarity measure such that $s(\mathcal{C}_K, \mathcal{C}'_K)$ measures the agreement between the two clusterings. Following the study in [17], we adopt the adjusted Rand index (ARI) measure [46]. For a given sample size N, the stability of a clustering algorithm $\mathcal{A}$ is defined as the expected similarity between two clusterings $\mathcal{C}_K$, $\mathcal{C}'_K$ on different data sets $\mathbf{X}$ and $\mathbf{X}'$, sampled from the distribution $\mathcal{P}$,

$$\mathrm{Stab}(\mathcal{A}, K) = \mathbb{E}_{\mathbf{X}, \mathbf{X}' \sim \mathcal{P}^N} \left[s(\mathcal{C}_K, \mathcal{C}'_K) \right]. \tag{5}$$

The expectation is taken with respect to the i.i.d. sampling of the sets from $\mathcal{P}$. This quantity is unavailable in practice, as we have a finite number of samples, so it needs to be estimated empirically. Various methods are listed in [6], [17], based either on resampling or noise. Stability is determined by the number of data points changing clusters under perturbation. In the case of algorithms that minimize an objective function, sources of instability have been discussed in [6]. In a context of large sample size and effective algorithm initialization, [17] identified *jittering* as a sufficient source of instability. Jittering is caused by data points changing side at cluster boundaries after perturbation. Therefore, strong jitter is produced when a cluster boundary cuts through high-density regions. Other sources of instability do not occur in our setting.

Let $\{\mathbf{X}_1, \ldots, \mathbf{X}_D\}$ be D perturbed versions of the original data set $\mathbf{X}$. Between-cluster stability of algorithm $\mathcal{A}$ with parameter K estimates the expectation (5) by the empirical mean of the similarities s between the reference clustering $\mathcal{C}_K = \mathcal{A}(\mathbf{X}, K)$ and the clusterings of the perturbed data sets,

$$\mathrm{Stab_B}(\mathcal{A}, \mathbf{X}, \mathcal{C}_K, K) = \frac{1}{D} \sum_{d=1}^{D} s\left(\mathcal{C}_K, \mathcal{A}(\mathbf{X}_d, K)\right). \tag{6}$$

Since s is a similarity measure, this quantity needs to be maximized. Then, within-cluster stability has been introduced to assess the presence of stable structures inside each cluster. To this aim, [17] propose to *cluster again* the data within each cluster of $\mathcal{C}_K$. Formally, let $\mathbf{\Omega} \subset \mathbb{N}^*$ be a set of numbers of clusters. The k-th cluster in the reference clustering is noted C_k, its number of elements N_k, and $\mathcal{Q}^{(k)}_{K'} = \mathcal{A}(C_k, K')$ denotes a partition of C_k into K' clusters. Within-cluster stability is defined as

$$\begin{aligned} &\mathrm{Stab_W}(\mathcal{A}, \mathbf{X}, \mathcal{C}_K, K, \mathbf{\Omega}) = \\ &\qquad \sum_{k=1}^{K} \left(\frac{1}{|\mathbf{\Omega}|} \sum_{K' \in \mathbf{\Omega}} \mathrm{Stab_B}(\mathcal{A}, C_k, \mathcal{Q}^{(k)}_{K'}, K') \right) \times \frac{N_k}{N}. \end{aligned} \tag{7}$$

As a good clustering is unstable within each cluster, this quantity needs to be minimized. Hence, the *Stadion* validity index (standing for *stability difference criterion*) combines between-cluster and within-cluster stability by computing the difference between both quantities. Omitting $\mathcal{A}$, K and $\mathbf{X}$ in the notations, its expression is:

$$\text{Stadion}(\mathcal{C}_K, \mathbf{\Omega}) = \text{Stab}_\text{B}(\mathcal{C}_K) - \text{Stab}_\text{W}(\mathcal{C}_K, \mathbf{\Omega}). \quad (8)$$

Since we use an effective initialization scheme, the same reference partition $\mathcal{C}_K$ is used in both terms of (8). Thus, Stadion evaluates the stability of an algorithm w.r.t. a reference partition. An important assumption behind our implementation of within-cluster stability is that, for non-clusterable structures (w.r.t. an algorithm), the algorithm must place cluster boundaries in high-density regions to produce instability through jittering.

By varying ε from 0 to a maximum value ε_max, a so-called *stability path* is obtained, i.e. the evolution of a stability score as a function of ε (see examples in Figures 2, 7 and 9). A straightforward method to fix ε_max beyond which comparisons are not meaningful anymore is as follows. The perturbation corresponding to ε_max is meant to destroy the cluster structure of the original data. This corresponds to the value where the data are no longer clusterable, i.e. $K = 1$ becomes the best solution w.r.t. Stadion. As shown in [17], a first guess at $\varepsilon_\text{max} = \sqrt{p}$ (where p is the data dimension) works well in practice. Visualizing the stability paths helps interpreting the structures found by an algorithm, hence improving the usefulness of results.

IV. Invariance-guided stability for time series

Stability methods based on resampling are data-independent and therefore directly applicable to time series. However, it has been shown in a realistic setting that these methods cannot work in the general case [17]. On the contrary, noise-based perturbations such as uniform or Gaussian ε-Additive Perturbation produce instability through jittering of cluster boundaries. The underlying assumption is that a clustering should be resilient to low levels of noise (no points change clusters), unless a boundary lies in a high-density region, where a large number of points change clusters. While adding uniform or Gaussian noise to every dimension is meaningful for tabular vectors of normalized features where Euclidean distance is used, it is irrelevant for raw time series. Algorithms for clustering raw time series use different distance metrics, thus there is no reason that random noise would or would not make points change clusters, depending on the cluster boundaries. The notion of cluster boundary itself becomes unclear when using different distances, as it is no longer a simple hyperplane. Clusters of time series are not clusters in the sense of euclidean distance and are resilient to different types of perturbation. For example, if time series are invariant to shifting, clusters should be resilient to perturbation by random shifting. As another example, if a set of time series is clustered under DTW distance with Sakoe-Chiba band w, clusters should be resilient to perturbation by warping, with a warping level not exceeding w. Most importantly, these invariances are determined by the physical nature of the observed phenomenon and not by the data set itself. The practitioner needs to know in advance which transformations are invariant and which are not. Only then, a suited distance and algorithm can be used and evaluated for model selection.

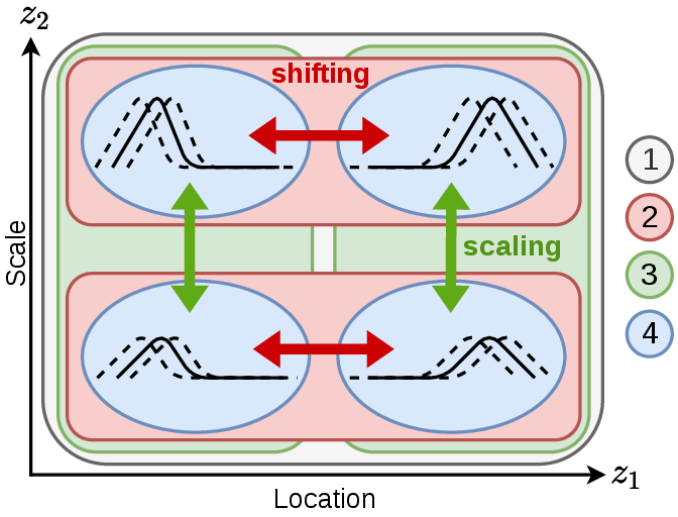

Fig. 1. Artificial time series data set consisting in one-dimensional bumps at two different locations and scales. The data distribution is represented in the (*location*, *scale*) latent factor space. Invariance to perturbation by random shifting (red) or scaling (green) determines the cluster structure, leading to 4 different solutions with 1, 2 or 4 clusters.

A. Perturbing invariant latent factors

The computation of stability needs to be adapted to the case of time series, and the perturbation process depends on the invariances of the data (shifting, scaling, offsetting, uniform or local warping, noise, etc). Let us illustrate this discussion with a simple artificial example, displayed on Figure 1. A data set consists in one-dimensional time series with "bumps" at two different time locations and with two different scales on the y-axis.

The data are generated by only two underlying latent factors: location (z_1) and scale (z_2). Had we access to the variables underlying the time series data-generating process, the task would be traditional clustering in a two-dimensional Euclidean vector space: the latent data distribution is simply 4 Gaussians. At a first glance, the model selection task can now seem straightforward: take any clustering algorithm based on Euclidean distance (e.g. k-means or Ward linkage), with ε-AP, and the Stadion criterion surely outputs the solution with $K = 4$. However, it is clearly false, because the true solution depends on the invariances of the original time series. The perturbation used in latent space must also take into account these invariances. Shift (or scale) invariance implies the variable z_1 (respectively z_2) should be ignored in the perturbation. There is a duality between perturbations in original time series space and in latent factor space, represented on Figure 1. The true cluster structure consists in:

- Shift and scale invariance: solution (1) with $K = 1$

- Shift invariance only: solution (2) with $K = 2$
- Scale invariance only: solution (3) with $K = 2$
- No invariance: solution (4) with $K = 4$ clusters

In the next paragraphs, we will focus on two model selection tasks, using the stability principle introduced in the previous section. First, we show that stability indicates whether a distance is adapted to the data invariances, and second, we select the number of clusters K using the Stadion internal validity index.

B. Selecting the right distance with stability

Between-cluster stability can be used to select a distance or algorithm with appropriate invariances. An algorithm should obtain a high between-cluster stability when perturbing the data under invariant transformations. Concretely, we consider the toy data set shown in Figure 1, and the widely used K-medoids algorithm, where the number of clusters is fixed to $K = 2$. For effective initialization, required by [17], we use K-medoids++ initialization and take the best result over 10 runs. In the first experiment, we assume the data is scale-invariant. Thus, we use perturbation by randomly scaling the whole time series by a factor drawn uniformely in the $[1/(1+\varepsilon), 1+\varepsilon]$ interval. The value ε controls the perturbation level, similarly to the noise level in [17]. Then, we evaluate between-cluster stability for three distances: Euclidean, correlation and DTW. Figure 2 displays the resulting stability paths, and unsurprisingly, correlation distance (K-medoids+COR) is the most stable. The second experiment assumes shift-invariance of the data. The whole time series are shifted temporally by a fraction of the time series length, drawn uniformely in $[0, \varepsilon]$. The perturbation level ε now represents the maximum shift length. The between-cluster stability paths now indicate that K-medoids+DTW is the most stable algorithm.

This toy task is rather a sanity check, because one generally knows in advance which invariances an algorithm satisfies, but we can imagine more complex algorithms where invariances are not clearly determined.

C. Selecting the number of clusters

The second model selection task is the selection of the number of clusters K. First, we consider an artificial data set consisting in one, two or three bumps located around three different time locations in the series, displayed on Figure 3. The desired invariance is warping invariance. Thus, the true number of clusters is $K = 3$, corresponding to the number of bumps.

We evaluate the K-medoids algorithm with DTW distance using the Stadion criterion and warping perturbation, for $K = 1 \dots 9$. Warping level is controlled by two parameters: first, α controls the maximum fraction of the series that will be warped, and ε controls the warping level, drawn uniformely in $[1/(1+\varepsilon), 1+\varepsilon]$. We fix $\alpha = \varepsilon = 0.2$. The hyperparameters of Stadion are set to $D = 10$ and $\mathbf{\Omega} = \{2 \dots 5\}$ without need for any tuning (see [17] for discussions on hyperparameters). Stadion scores and standard deviations over $D = 10$ perturbations are shown on Figure 4. Clearly, our method has

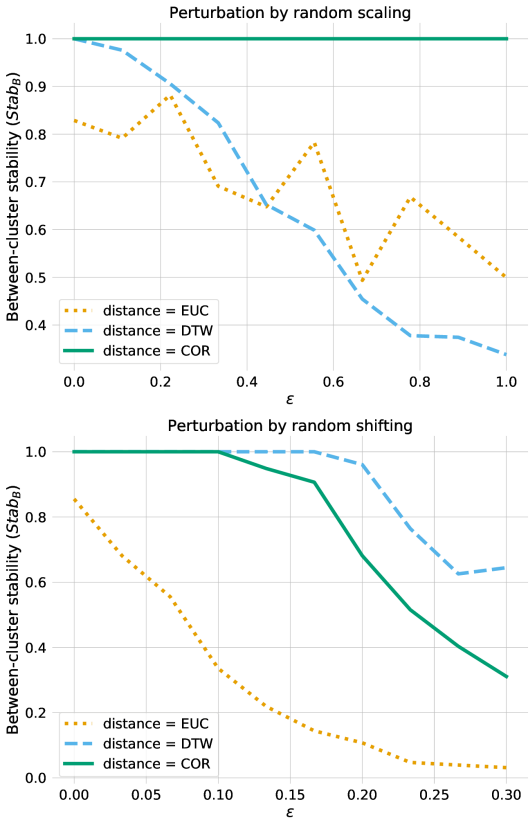

Fig. 2. Between-cluster stability paths under perturbation by random scaling (top) and shifting (bottom) for the K-medoids algorithm, with euclidean (EUC), correlation (COR) and dynamic time warping (DTW) distances. COR is resilient to scaling and DTW is more resilient to shifting. ε controls the level of perturbation.

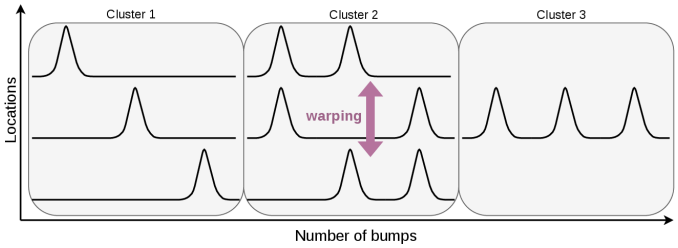

Fig. 3. Artificial time series data set consisting in one, two or three bumps at different locations, represented in the (*number of bumps*, *locations*) latent factor space. Under the assumption of warping invariance, the true number of clusters is 3, corresponding to the number of bumps.

selected the desired solution $K = 3$. This means that the most natural structure is three clusters, with respect to the considered algorithm and invariance. Whenever the algorithm is not able to find any structure resilient to the perturbation, our method outputs $K = 1$, i.e. the data is not clusterable. This happens if we use Euclidean distance instead of DTW, as shown in Figure 5. Interestingly, the second-best solution is $K = 7$, as there are 7 different configurations for the locations of the bumps.

Experiments were then conducted on univariate data sets from the UCR/UEA archive [47] (although our stability frame-

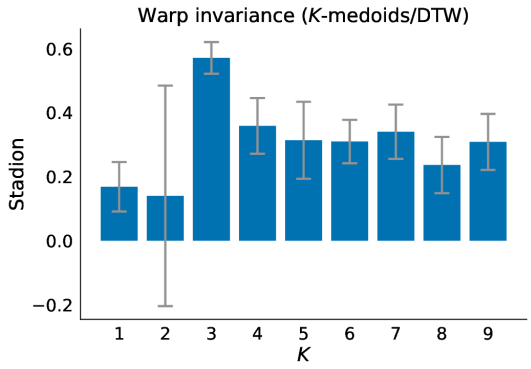

Fig. 4. Stadion criterion with perturbation by random warping (here with $\alpha = \varepsilon = 0.2$ and $D = 10$) for the K-medoids algorithm with DTW distance, for $K = 1 \dots 9$. The correct solution $K = 3$ is selected.

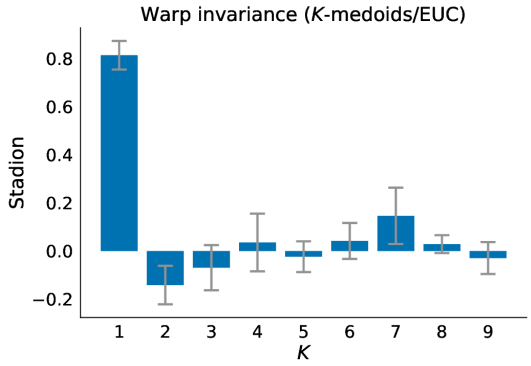

Fig. 5. Stadion criterion with perturbation by random warping for K-medoids with Euclidean distance, for $K = 1 \dots 9$. Our method outputs $K = 1$, meaning that the data is not clusterable w.r.t. the considered algorithm and invariance.

work also applies in the multivariate case). We present results for the CBF and Trace data sets, with two algorithms: K-medoids+DTW and K-shape [34]. For each algorithm we keep the best out of 10 runs. In order to speed up computations, we use only a subsample of 50 time series with balanced ground-truth class labels.

First, we evaluate K-medoids+DTW on CBF (see Figure 6). We choose to perturbate the data by random shifting and adding uniform noise, as CBF consists in three different noisy shapes at different locations. As previously, the shifting level is controlled by ε, varied from 0 to 0.3 to obtain the Stadion paths on Figure 7. The uniform noise is fixed and drawn in $[-0.3, 0.3]$. Choosing the right perturbation seems to be a difficult task and to require profound knowledge of the data set; however, it is not strictly necessary. On CBF, warping invariance could also be correctly used, but shifting is sufficient to discover the right structure. Results are presented on Figure 7: our method successfully selects the solution $K = 3$ (by taking the highest maximum or average Stadion value over the path until $\varepsilon_{\max}$, as explained in [17]). It also corresponds to the partition with the best ARI (ARI $= 0.93$).

A second experiment on the Trace data set with K-shape

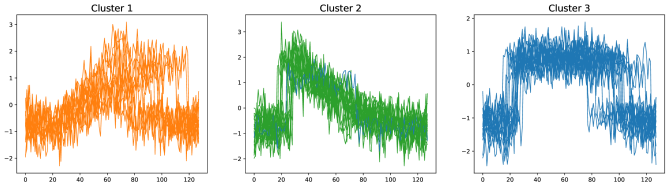

Fig. 6. Partitions obtained on the CBF data set by K-medoids+DTW for $K = 3$. The best solution w.r.t. the ARI is $K = 3$ (ARI $= 0.93$).

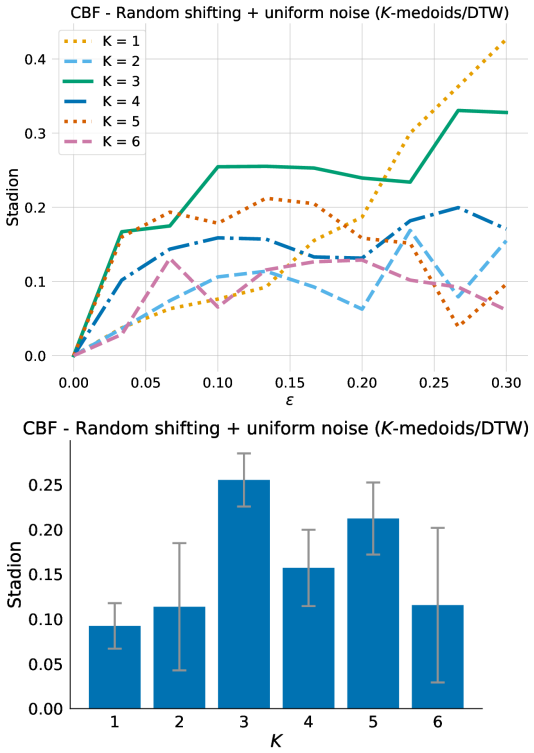

Fig. 7. Stadion criterion for K-medoids+DTW on CBF under shifting and uniform noise perturbation, evaluated for $K = 1 \dots 6$. (Top) Stadion paths as a function of shifting level ε. The solution is selected by the highest maximum or average Stadion value. (Bottom) Stadion scores taken at $\varepsilon = 0.15$ with standard deviations over $D = 10$ perturbations.

presents a case where the algorithm cannot recover the ground-truth partition (see Figure 8). We choose a warping-based perturbation, with $\alpha = \varepsilon$ and $\varepsilon_{\max} = 0.5$, and evaluate parameters $K = 1 \dots 5$ ($K > 5$ produces clusters with too few points). As can been seen on the results Figure 9, Stadion selects the solution with $K = 3$, although the ground-truth partitions has 4 clusters. However, two of the clusters cannot be distinguished by K-shape, thus $K = 3$ is objectively the best solution (as measured by ARI with ground-truth labels). As a conclusion, our method evaluates the *quality of a given partition with respect to a given algorithm and a given set of invariances*, and yields sensible and interpretable results.

V. Conclusion and Future Work

In this work, we introduced an invariance-guided criterion for model selection in time series clustering. The method

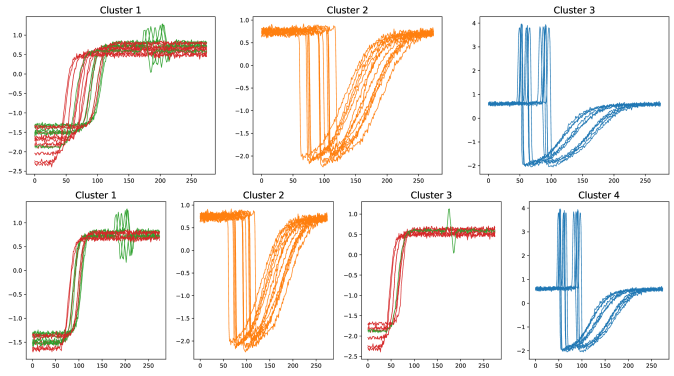

Fig. 8. Partitions obtained on the Trace data set by K-shape for $K = 3$ (top) and $K = 4$ (bottom). The algorithm is unable to recover the ground-truth partition into 4 clusters. The best solution w.r.t. the ARI is $K = 3$ (ARI $= 0.80$), followed by $K = 4$ (ARI $= 0.75$).

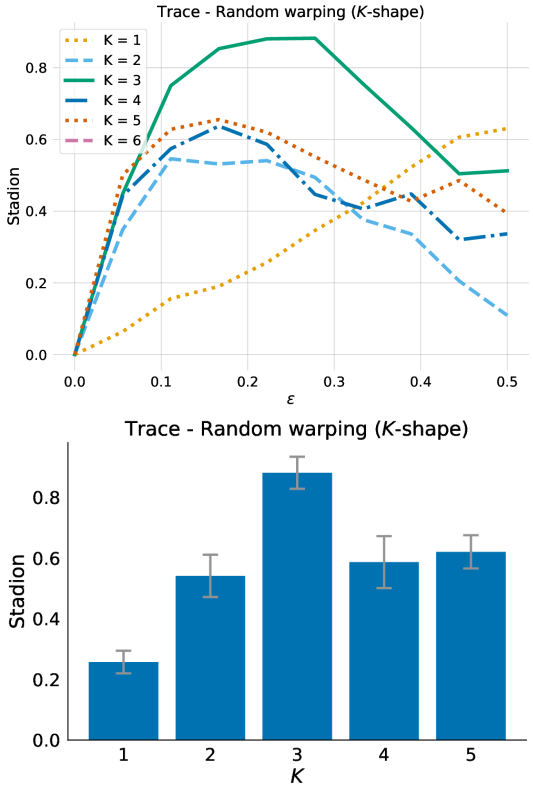

Fig. 9. Stadion criterion for K-shape on Trace under warping perturbation, evaluated for $K = 1 \ldots 5$. (Top) Stadion paths as a function of warping level ε. (Bottom) Stadion scores taken at $\varepsilon = 0.25$ with standard deviations over $D = 10$ perturbations.

is based on the principle that a good clustering is stable under particular perturbations. We use prior knowledge on the invariances of time series data to compute stability scores, based on the recent stability difference criterion (Stadion). Encouraging results were obtained on several toy and benchmark data sets, using well-known center-based time series clustering algorithms. The criterion was able to correctly determine the number of clusters given a set of invariances, and provides an interpretable visualization tool called stability paths. An important drawback is its high computational cost, as it requires to run the algorithm multiple times for each evaluated parameter. There is a need for efficient algorithms or algorithms with an extension operator, able to assign new points to clusters without re-training. The extended version was not tackled in this paper and is left for future work. Moreover, this implementation of within-cluster stability is not valid for all classes of algorithms [17]. Future work will focus on reducing the computational burden, and exploring more complex data sets. Insights on data perturbations can be gained from the vast literature on invariant transformations and data augmentation. Finally, we are convinced that interesting links could be made between clustering stability and adversarial attacks [48].

Acknowledgment

This research was funded by the French agency for research and technology (ANRT) through the CIFRE grant 2017/1279 and by Safran Aircraft Engines (Safran group). We used the algorithm implementations of the `tslearn` library [49] for K-shape, and `sklearn-extra` for K-medoids. The CBF and Trace data sets were taken from the UCR/UEA archive [47].

References

[1] T. Warren Liao, "Clustering of time series data - A survey," *Pattern Recognition*, 2005.
[2] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, "Time-series clustering - A decade review," *Information Systems*, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.is.2015.04.007
[3] M. Verleysen and D. François, "The Curse of Dimensionality in Data Mining," in *IWANN*, 2005. [Online]. Available: http://www.springerlink.com/index/n65tna6vwt3b1pw6.pdf
[4] H. J. Ng, Raymond T, "Efficient and Effective Clustering Data Mining Methods for Spatial Data Mining," in *International Conference on Very Large Data Bases (VLDB)*, 1994.
[5] S. Ben-David, U. Von Luxburg, and D. Pál, "A sober look at clustering stability," *Lecture Notes in Computer Science*, 2006.
[6] U. Von Luxburg, "Clustering stability: An overview," *Foundations and Trends in Machine Learning*, 2009.
[7] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, 2013.
[8] J. Neel, "Cluster analysis methods for speech recognition," Ph.D. dissertation, KTH, 2005.
[9] R. Baragona, "A simulation study on clustering time series with meta-heuristic methods," *Quaderni di Statistica*, 2001.
[10] E. A. Maharaj, "Clusters of time series," *Journal of Classification*, 2000.
[11] D. Piccolo, "A Distance Measure for Classifying ARIMA Models," *Journal of Time Series Analysis*, 1990.
[12] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a Mixture Model for Clustering with Integrated Completed likelihood," *IEEE Transactions on Pattern Analysis and Machine Learning*, 2000.
[13] C. Bouveyron, E. Côme, and J. Jacques, "The discriminative functional mixture model for a comparative analysis of bike sharing systems," *Annals of Applied Statistics*, 2015.
[14] É. Goffinet, M. Lebbah, H. Azzag, and L. Giraldi, "Clustering de séries temporelles par construction de dictionnaire," in *EGC*, 2020.
[15] ——, "Autonomous Driving Validation With Model-Based Dictionary Clustering," in *ECML-PKDD*, 2020.
[16] M. Meila, "How to tell when a clustering is (approximately) correct using convex relaxations," in *NeurIPS*, 2018.

[17] A. Mourer, F. Forest, M. Lebbah, H. Azzag, and J. Lacaille, "Selecting the Number of Clusters K with a Stability Trade-off: an Internal Validation Criterion," 2020. [Online]. Available: https://arxiv.org/abs/2006.08530

[18] Q. Pan, X. Li, and L. Fang, "Data Augmentation for Deep Learning-Based ECG Analysis," *Feature Engineering and Computational Intelligence in ECG Monitoring*, 2020.

[19] B. Fu, F. Kirchbuchner, and A. Kuijper, "Data Augmentation for Time Series : Traditional vs Generative Models on Capacitive Proximity Time Series," in *ACM International Conference on PErvasive Technologies Related to Assistive Environment (PETRA)*, 2020.

[20] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, "Adaptive data augmentation for image classification," *International Conference on Image Processing (ICIP)*, 2016.

[21] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0

[22] B. J. Frey and N. Jojic, "Transformation-Invariant Clustering and Dimensionality Reduction Using EM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[23] T. Monnier, T. Groueix, and M. Aubry, "Deep Transformation-Invariant Clustering," 2020. [Online]. Available: http://arxiv.org/abs/2006.11132

[24] G. Marti, P. Very, P. Donnat, and F. Nielsen, "A proposal of a methodological framework with experimental guidelines to investigate clustering stability on financial time series," *International Conference on Machine Learning and Applications (ICMLA)*, 2016.

[25] G. Klassen, M. Tatusch, L. Himmelspach, and S. Conrad, "Fuzzy Clustering Stability Evaluation of Time Series," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*. Springer International Publishing, 2020. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-50146-4_50

[26] V. Roth, T. Lange, M. Braun, and J. Buhmann, "A Resampling Approach to Cluster Validation," *Compstat*, 2002.

[27] I. Färber, S. Günnemann, H.-P. Kriegel, P. Kröger, E. Müller, E. Schubert, T. Seidl, and A. Zimek, "On Using Class-Labels in Evaluation of Clusterings," *International Workshop on Discovering, Summarizing and Using Multiple Clusterings (MultiClust), KDD*, 2010.

[28] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1978.

[29] G. E. Batista, X. Wang, and E. Keogh, "A Complexity-Invariant Distance Measure for Time Series," in *SIAM International Conference on Data Mining*, 2011. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611972818.60

[30] R. Giusti and G. E. Batista, "An empirical comparison of dissimilarity measures for time series classification," *Brazilian Conference on Intelligent Systems (BRACIS)*, 2013.

[31] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[32] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, 2011.

[33] J. Yang and J. Leskovec, "Patterns of Temporal Variation in Online Media," in *WSDM*, 2011.

[34] J. Paparrizos and L. Gravano, "k-Shape: Efficient and Accurate Clustering of Time Series," *ACM SIGMOD*, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=2723372.2737793

[35] J. Zakaria, A. Mueen, and E. Keogh, "Clustering time series using unsupervised-shapelets," in *International Conference on Data Mining (ICDM)*, 2012.

[36] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang, "Salient Subsequence Learning for Time Series Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[37] P. Patel, E. Keogh, J. Lin, and S. Lonardi, "Mining motifs in massive time series databases," 2003.

[38] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a Novel Symbolic Representation of Time Series," 2007. [Online]. Available: http://cs.gmu.edu/~jessica/SAX_DAMI_preprint.pdf

[39] P. Schäfer, "The BOSS is concerned with time series classification in the presence of noise," *Data Mining and Knowledge Discovery*, 2015.

[40] P. Schäfer and U. Leser, "Multivariate Time Series Classification with WEASEL + MUSE," in *ACM*, 2016.

[41] N. S. Madiraju, S. M. Sadat, D. Fisher, and H. Karimabadi, "Deep Temporal Clustering: Fully Unsupervised Learning of Time-Domain Features," 2018. [Online]. Available: http://arxiv.org/abs/1802.01059

[42] Q. Ma, J. Zheng, S. Li, and G. W. Cottrell, "Learning Representations for Time Series Clustering," in *NeurIPS*, 2019.

[43] V. Fortuin, M. Hüser, F. Locatello, H. Strathmann, and G. Rätsch, "SOM-VAE: Interpretable Discrete Representation Learning on Time Series," in *International Conference on Learning Representations (ICLR)*, 2019.

[44] L. Manduchi, M. Hüser, G. Rätsch, and V. Fortuin, "Variational PSOM: Deep Probabilistic Clustering with Self-Organizing Maps," 2019. [Online]. Available: http://arxiv.org/abs/1910.01590

[45] F. Chamroukhi and H. D. Nguyen, "Model-Based Clustering and Classification of Functional Data," Tech. Rep., 2018. [Online]. Available: http://arxiv.org/abs/1803.00276

[46] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, 1985.

[47] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The UEA multivariate time series classification archive, 2018," 2018. [Online]. Available: http://arxiv.org/abs/1811.00075

[48] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Adversarial Attacks on Deep Neural Networks for Time Series Classification," in *International Joint Conference on Neural Networks*, 2019.

[49] R. Tavenard, "tslearn: A machine learning toolkit dedicated to time-series data," 2017. [Online]. Available: https://github.com/rtavenar/tslearn