Semi-Supervised Time Point Clustering for Multivariate Time Series

Benjamin Ertl[†,*], Jörg Meyer[†], Matthias Schneider[‡], Achim Streit[†]
[†] Steinbuch Centre for Computing (SCC),
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[‡] Institute for Meteorology and Climate Research (IMK-ASF),
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Abstract

The formation and analysis of clusters in multivariate time series can reveal interesting patterns and complex correlations in temporal data. However, traditional clustering methods based on distance metrics fall short to discover interpretable characteristics and structures reflected by these clusters. This paper provides a new method for semi-supervised time point clustering based on the temporal proximity of time points and the correlation of their corresponding values. For this purpose, we utilize CoExDBSCAN, a recently developed density-based clustering algorithm with constrained expansion. CoExDBSCAN allows to identify clusters of temporal neighbourhoods that are only expanded with regards to a priori constraints in defined subspaces. Adopting this algorithm to time series data and grouping segments with similar correlations allows us to find accurate and interpretable structures. We provided a comparison to state-of-the-art methods and verification of our approach on a synthetic dataset and an experimental evaluation on a real-world dataset. The experimental assessment shows that our clustering results can further serve as an effective basis for time series classification.

Keywords: Semi-Supervised Clustering, Time Point Clustering, Multi-variate Time Series

1. Introduction

The increasing amount of data produced over time by a variety of sensors and scientific instruments available through new technologies and increasing storage capacity provides unique opportunities to discover characteristics and structures reflected by meaningful clusters in such time series. Especially recurring subsequences in streams of multiple measurements, that can be organized as multivariate time series, can be interpreted as recurring events or actions. These recurring events can be used to discover repeating patterns, understanding trends, detect anomalies and in general better interpret large and high-dimensional datasets [1]. For this purpose, time series have to be segmented and clustered in a way that the temporal proximity of time points is taken into account and multiple segments can belong to the same cluster. A number of methods can tackle this task for univariate time series [2], but fall short to discover interpretable clusters for multivariate time series [3]. Specifically methods merely based on distance metrics such as euclidean distance or dynamic time warping [4] can not capture structural similarities based on correlations across time. For static data, there has been a growing interest in semi-supervised clustering methods, for example constrained clustering, where additional a priori information or domain knowledge is incorporated into the clustering process, to better capture complex relations between features [5–7]. In general semi-supervised clustering algorithms can be divided into two groups, pointwise and pairwise algorithms, where the former has pre-labeled points available and the latter is usually expressed in *must-link* and *cannot-link* constraints [8, Chapter 20, Agovic et al.]. In this paper, we propose a new method for semi-supervised time point clustering

[*] benjamin.ertl@kit.edu

based on *CoExDBSCAN* [9], a recently developed density-based clustering algorithm with constrained expansion. Our approach follows the pairwise semi-supervision and extends the concept to cluster-wide constraints. By applying *CoExDBSCAN* to time series data and constraining the cluster extension to the correlation of time point values, we are able to identify clusters of segments with similar correlations. Our experimental evaluation shows that these clusters can be associated with recurring events and therefore can be utilized to serve as an effective basis for time series classification.

Our contributions can be summarized as follows:

- We apply the *CoExDBSCAN* algorithm to time series data by defining the time space of the data as subspace for the distance based density computations. In this way, we are able to find temporal neighbourhoods whose expansions are restricted by additional constraints.
- We propose a constraint formulation to restrict the cluster expansion to the correlation of time point values.
- We form groups of segments with similar correlations. These groups can be interpreted as recurring events or actions.
- Finally, we provide an experimental assessment on real-world data and demonstrate a concept that can serve as an effective basis for time series classification utilizing the clustering results.

The remainder of this paper is organized as follows: Section 2 provides an overview of related work and marks out the differences to our approach. Section 3 details the adaption of the *CoExDBSCAN* algorithm and constraint formulation for time series data. In Section 4 we present the verification of our approach and the comparison to state-of-the-art methods, while the experimental evaluation on a real-world dataset is given in Section 5. Section 6 concludes this paper with a discussion of the results and outlook on future research.

2. **Related Work**

Time series clustering is a well-established and active research field across different application domains, for example in industry, biology, energy, medicine, finance or climate. Multiple surveys provide a clear and structured overview of past and current research in time series clustering and its subdomains **whole time series clustering**, **subsequence time series clustering** and **time point clustering** [2, 10, 11]. Our approach falls into the category of time point clustering. Zolhavarieh et al. [11] describe in their review of subsequence time series clustering time point clustering as

> "[...] the clustering of time points on the basis of a combination of their temporal proximity and the similarity of their corresponding values. This approach is similar to time series segmentation. However, time point clustering is different from segmentation in the sense that all points do not need to be assigned to the cluster; that is, some of [the] points are considered noise."

We follow along this distinction and will give a more formal definition in Section 3. Without the differentiation on noise points, i.e. points that do not belong to any cluster, algorithms and methods developed for subsequence clustering are inter-comparable to those developed for time point clustering in terms of extracting similar segments from individual time series. Zolhavarieh et al. [11] provide a comprehensive overview of methods for subsequence time series clustering, especially within the context of the discussion if any method can produce meaningful results at all or if all methods for subsequence time series clustering are actually meaningless [12].

Since the traditional Euclidean distance metric as a similarity measure for clustering algorithms is not taking the order of the data points into account, a similarity measure called *Dynamic Time Warping (DTW)* has been proposed [13] and improved over time, for example by averaging a set of sequence to be used with similarity-based methods like *k-means* [14]. Another prominent clustering algorithm for static data that has also been adopted for spatial and temporal data is *DBSCAN* introduced by Ester et al. [15–17]. Schubert et al. showed that *DBSCAN* continues to be relevant even for high-dimensional data, although the choice of parameters becomes more challenging [18]. The algorithm has also been modified for constrained clustering, for example *C-DBSCAN* [19] and *CoExDBSCAN* [9]. *CoExDBSCAN* has been demonstrated to be especially suited for spatio-temporal data, where one subspace of features defines the spatial or temporal extend of the data and another subspace of features defines the inherent correlations between features.

Besides distance-based algorithms, model-based clustering algorithms for time series have been proposed as well. In a recent publication, Hallac et al. [1] use graph representations for time series subsequences from Markov random fields (MRF) to group similar sequences into clusters, called *Toeplitz inverse covariance-based clustering (TICC)*. *TICC* simultaneously segments and clusters the data based on its correlation and has been demonstrated to be able to find structural similarities in real-world data [1]. Also in the field of unsupervised learning and deep unsupervised learning, model-based clustering algorithms for time series data are subject to recent and continuous research. Qin Zhang et al. proposed a method for unsupervised salient subsequence learning (*USSL*) to extract salient subsequence features from time series, called shapelets [20].

In this paper, we propose a semi-supervised time point clustering method, which sets it apart from complete unsupervised methods, for example *USSL*, model-based methods like *TICC* and solely similarity-based methods like *k-means* with *DTW*. Since *CoExDBSCAN* offers the flexibility to cluster points that are close in one subspace and the expansion of clusters complies with a priori constraints, we choose to adopt this algorithm for time point clustering and the extraction of similar subsequences. Our adaptation of the algorithm and formal definitions of the relevant concepts is explained in detail in the next section.

3. Semi-Supervised Time Point Clustering

In this section, we provide the necessary definitions and details about our adaptation of the *CoExDBSCAN* algorithm, as well as the formulation of our constraint to restrict the cluster expansion. Rodpongpun et al. [21] provide following definitions on subsequence time series clustering, i.e. Definition 1 and 2.

Definition 1. *A time series T of size m is an ordered sequence of real-value data, where $T = (t_1, t_2, ..., t_m)$.*

Definition 2. *A subsequence $T_{i,n}$ of length n of time series T is $T_{i,n} = (t_i, t_{i+1}, ..., t_{i+n-1})$, where $1 \leq i \leq m - n + 1, n < m$.*

Definition 2 can be extended to allow elements to be omitted.

Definition 3. *A subsequence T_S of length n of time series T is an arranged sequence of data that omits some elements without changing the order of the remaining elements. $T_S = (t_{s_1}, t_{s_2}, ...t_{s_n})$, where $|T_S| = n$ and $\forall s_i \in [1, m] : s_i < s_{i+1}$.*

One of the six main definitions that are essential for the *DBSCAN* and *CoExDBSCAN* algorithms is the definition of the ϵ-neighbourhood [15] and its modification the *CoExDBSCAN* ϵ-neighbourhood [9].

Definition 4. *Let DB be a database of points. The ϵ-neighbourhood of a point p, denoted by $N_\epsilon(p)$, is defined by*

$$N_\epsilon(p) = \{q \in DB | dist(p, q) \leq \epsilon\} \tag{3.1}$$

Definition 5. *Let DB be a database of points. The CoExDBSCAN ϵ-neighbourhood of a point p, denoted by $N_\epsilon(p)$, is defined by*

$$N_\epsilon(p) = \{q \in DB | dist(p_S, q_S) \leq \epsilon \wedge constraints(p_R, q_R)\} \tag{3.2}$$

where p_S, q_S are the subspace representations of point p and q of the user-defined spatial subspace S, p_R, q_R are the subspace representations of point p and q of the user-defined constraint subspace R and the `constraints` *function evaluates* `true` *for each constraint C_i in a user-defined set of constraints $C = \{C_1, C_2, ..., C_n\}$.*

Another essential definition is the direct density-reachability, which requires points to belong to the same ϵ-neighbourhood with at least $minPts$ within the neighbourhood to form a cluster. If and only if these conditions are met, the algorithm starts to expand the cluster from every point within the ϵ-neighbourhood.

Definition 6. *A point p is directly density-reachable from a point q wrt. ϵ and $minPts$ if*

(1) $p \in N_\epsilon(q)$ and
(2) $|N_\epsilon(q)| \geq minPts$ (core point condition).

For the remaining definitions of the original *DBSCAN* and *CoExDBSCAN* algorithms as well as the pseudo code of the algorithms, we refer to the original papers by Ester et al. [15] and Ertl et al. [9] respectively.

The transition from Definition 4 to Definition 5 allows us to define the temporal order of the data points as the spatial subspace and to provide a constraint function that is evaluated in another subspace for the clustering algorithm. With this transition, the ϵ-neighbourhood describes a neighbourhood of lagged points, similar to a time window, where the maximum lag in time for the initial data points is defined by the ϵ parameter and the minimal amount of data points that are required to form a cluster is defined by the $minPts$ parameter, see Definition 6, direct density-reachable points of the original *DBSCAN* algorithm.

For any data point t_i at time i the algorithm considers all points t_j at times $j \in [i - \epsilon, i + \epsilon]$ as candidates for an initial subsequence. If all constraints are satisfied for any t_j, t_i and t_j belong to the same subsequence, which is further extended at point t_j. All resulting subsequences follow Definition 3 and all points within each subsequence satisfy all constraints. The omitted elements from one subsequence, if any, are either belonging to another overlapping subsequence or are disregarded as noise.

We have determined a constraint formulation through empirical evaluation that has been proven to be especially suited for correlated data. For each evolving subsequence we compute the residuals of an ordinary least squares linear regression and include neighbouring points in this subsequence if and only if the square of the residual of a neighbouring point deviates from the mean of the square of the residuals of the current points in the subsequence only by a certain factor δ. This δ has to be determined either via parameter selection, for example grid-search, or via a priori knowledge about the nature of the time series.

Definition 7. *A point t_j belongs to a subsequence T_S of length n of a time series T of length m, with $T_S = (t_{s_1}, t_{s_2}, ...t_{s_n})$ where $|T_S| = n$ and $\forall s_i \in [1, m] : s_i < s_{i+1}$, iff*

$$(Y_{t_j} - \hat{Y}_{t_j})^2 < \delta \cdot \frac{1}{n} \sum_{k=s_1}^{s_n} (Y_{t_k} - \hat{Y}_{t_k})^2 \tag{3.3}$$

where Y and $\hat{Y}$ are the dependent variable and fitted value of the linear regression respectively.

After splitting the time series into subsequences, first, we label all sequences with less than required data points ($reqPts$) as noise; such sequences can appear if the $minPts$ parameter has been set to a small number and the sequence could not be expanded due to the given constraint. Second, we compute the regression coefficients for each remaining subsequence and group sequences with equal or slightly different regression coefficients for the dependent variable into the same cluster. The threshold for different coefficients has to be determined the same way as the δ parameter, either via parameter selection, for example grid-search, or via a priori knowledge about the nature of the time series. This process can be repeated for multiple time series and will result in clusters of subsequences as following.

Definition 8. *A cluster C is a set of subsequences of one or multiple time series, $C = \{T_S^{(l)}\}$ for $1 \leq l \leq N$, where N is the number of total subsequences, where each time point in every subsequence satisfies the conditions formulated in Definition 6 and Definition 7, and each subsequence of points $T_S^{(l)}$ satisfies following conditions:*

(1) $\forall T_S^{(l)} : |T_S^{(l)}| \geq reqPts$ *(subsequences with more than reqPts)*

(2) $\forall T_S^{(l)}, T_S^{(o)} \in C : \|\beta_l - \beta_o\| < \theta$ *(regression coefficients close)*

where β_l, β_o are the regression coefficients of a linear regression of all time points in subsequences $T_S^{(l)}, T_S^{(o)}$ respectively for a threshold θ.

It should be noted, that our constraint (Definition 7) has been specifically formulated to cluster linear segments with similar regression coefficients for a given time series. This makes our approach especially suited for time series that exhibit such inherent characteristic, for example correlated events in the feature space. However, our approach could be used to find non-linear segments as well by designing an appropriate constraint or multiple constraints, but finding and expressing suitable constraints for the *CoExDBSCAN* algorithm remains a challenging task [9].

Following Definition 1 to 8, our approach for semi-supervised time point clustering for multivariate time series can be summarized in four steps.

(1) Compute the *CoExDBSCAN* clustering result for each time series with the time dimension as the spatial subspace and the correlated features as the correlation subspace with the constraint formulated in Definition 7. Each cluster is equivalent to a subsequence. (**adaptation of the algorithm**)
(2) Label all subsequences with less than the minimum required number of time points as noise, if any. (**modification of the algorithm**)
(3) Compute the linear regression coefficients between the correlated features for each subsequence. (**adaptation of the algorithm**)
(4) Group all subsequences with equal or close regression coefficients up to a certain threshold into one cluster. The resulting clusters, see Definition 8, contain segments of one or multiple time series that are similar to each other in terms of temporal proximity and correlation of the comprising time points. (**improvement through innovation**)

4. Verification

To verify our approach detailed in the previous section, we compared the results of our semi-supervised time point clustering method to a baseline *k-means* clustering for time-series data with dynamic time warping (DTW), a Gaussian Mixture Model (GMM) and the Toeplitz inverse covariance-based clustering ($TICC$) method. The *k-means* with DTW algorithm is a well established similarity-based method for time series and time point clustering and is available in a variety of programming languages. The Gaussian Mixture Model

Table 1. Value range and generation methods.

Sequence	Points	Feature x1	Feature x2	Noise
1	10	$x1 \in [-50, 100] + \xi$	$x2 = 0.1 \cdot x1 + \xi$	$\xi \sim \mathcal{N}(0, 4)$
2	20	$x1 \in [100, 250] + \xi$	$x2 = -0.2 \cdot x1 + 39.65 + \xi$	$\xi \sim \mathcal{N}(0, 4)$
3	20	$x1 \in [100, 250] + \xi$	$x2 = 0.5 \cdot x1 - 106.94 + \xi$	$\xi \sim \mathcal{N}(0, 4)$
4	10	$x1 \in [-50, 100] + \xi$	$x2 = -0.6 \cdot x1 + 4.52 + \xi$	$\xi \sim \mathcal{N}(0, 4)$

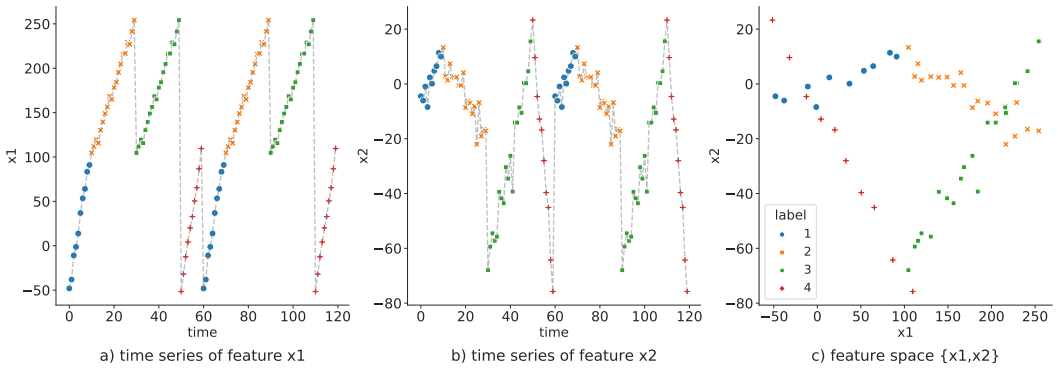

Figure 1. Synthetic dataset; a) time series for feature x1, b) time series for feature x2 and c) joint feature space {x1,x2}. Labels are true labels.

is a general, model-based approach that provides a sound mathematical-based approach for statistical modelling of a wide variety of random phenomena [22, 23]. As a state-of-the-art comparison we choose *TICC* [1], since the authors have shown that their method outperforms a range of model-based and distance-based clustering methods for clustering multivariate time series subsequences. Since our own implementation is based on Python, we choose the *k-means* with *DTW* implementation from the tslearn machine learning toolkit for time series data [24] and the *GMM* implementation from the scikit-learn machine learning package [25] in Python as well; the code of the *TICC* method is also provided in Python by the authors [1].

We use a synthetic dataset for verification and comparison that has known correlations between two features and known temporal subsequences. Because the correlations and the order of subsequences are known, we can evaluate all methods against the ground truth. We use two metrics, the adjusted Rand index and the clustering accuracy. The Rand index measures the similarity between two data clusterings by counting equal elements in subsets created by the two partitions of the data, the true partition (true labels) and the computed partition (predicted labels) [26]. Since the expected value of the Rand index of two random partitions does not take a constant value [27], Hubert et al. introduced an adjustment for chance to the Rand index [28]. The adjusted Rand index is thus ensured to have a value close to zero for random labeling independently of the number of clusters and samples and exactly one when the clusterings are identical, up to a permutation [25]. Our second metric, the cluster accuracy, finds the best match between the true labels and the predicted labels. The greater the clustering accuracy, the better the clustering performance [29].

Our synthetic dataset has four temporal sequences that are repeated once: "1, 2, 3, 4, 1, 2, 3, 4", illustrated in Figure 1a) and b). Each sequence has two correlated features (x1 and x2), generated according to Table 1.

For each sequence, we choose one feature evenly distributed on the given interval plus some randomly distributed noise and the other feature according to a specific linear equation plus some randomly distributed noise that leads to overlapping areas in the joint feature

Table 2. Summary of clustering results for the synthetic data using the adjusted Rand index (ARI) and cluster accuracy (ACC) metrics.

Clustering Method	ARI	ACC	Parameters
Modified CoExDBSCAN	**0.88**	**0.93**	$\epsilon = 2, minPts = 1, \delta = 4, \theta = 0.01$
GMM	0.67	0.75	$n = 4, n\_init = 10, init\_params = kmeans$
TICC	0.36	0.58	$n = 4, w = 1, \lambda = 0.11, \beta = 0$
k-means with DTW	0.26	0.52	$n = 4$

space as depicted in Figure 1c). This overlap in feature space is particularly challenging for cluster algorithms, since distance-based and density-based algorithms can not distinguish between the overlapping clusters without a priori information.

Our modified and adapted version of *CoExDBSCAN* yields the best clustering result for the synthetic data with the highest adjusted Rand index (0.88) and highest cluster accuracy (0.93), as summarized in Table 2. This approach significantly outperforms the baseline *k-means* with *DTW*, the general clustering approach with *GMM* and also the state-of-the-art *TICC* algorithm. Figure 4 shows the clustering results of our semi-supervised time point clustering method. The subsequences have been accurately identified and clustered together, with six data points labeled as noise (5% of all data points). We choose the input parameters based on a grid search with ϵ, $minPts$ and δ in the range of $[1, 5]$ with a step size of one and kept the parameters with the highest adjusted Rand index. Subsequences with two or less points have been labeled as noise.

The second best clustering results are obtain by the Gaussian Mixture Model with all dimensions included in the clustering process (see Figure 3). Similar results have been shown by Hallac et al. [1] in their comparison for the same order of subsequences. More notably, the results between our baseline *k-means* and state-of-the-art *TICC* method are surprisingly indifferent in terms of the cluster accuracy and with a slightly higher ARI score for the *TICC* algorithm. Figure 2 illustrates the clustering result for the *TICC* algorithm.

Varying the *TICC* input parameters show no tendencies for improving the results, furthermore higher values for the smoothness penalty parameter β have resulted in numerical errors. However, decreasing the noise on the synthetic data can lead to a better performance. This increase in accuracy can be observed in all compared algorithms except the baseline *k-means* algorithm, while increasing the noise leads to the opposite, decreasing accuracy. Therefore, the comparison with the presented synthetic data is sound and holds true for a variation in noise also.

Table 2 summarizes the metric scores and lists the parameters used for each algorithm in detail. Since *k-means*, *GMM* and *TICC* require the number of clusters as a parameter, we fixed this parameter to the true number of subsequences. Other parameters have been set to the cluster algorithms' default values, besides the number of initializations for *GMM* that has been set to ten iterations with the best results to keep, and the window size for *TICC* that has been set to one due to empirical evaluation; also the smoothness penalty parameter β for the *TICC* algorithm has been set to zero, i.e. without a temporal consistency constraint [1], according to our Definition 3.

Our approach does not require the number of clusters as an input parameter, which is usually unknown a priori and should rather be discovered in the clustering process. The other parameters are intuitively comprehensible, with ϵ corresponding to the time window, δ corresponding to the factor of maximum deviation from the residuals mean of the linear regression and θ corresponding to the similarity threshold of the linear regression coefficients for each subsequence. Given these parameters either through empirical evaluation or expert knowledge, our approach captures the inherent structure of the data best compared to the selected algorithms.

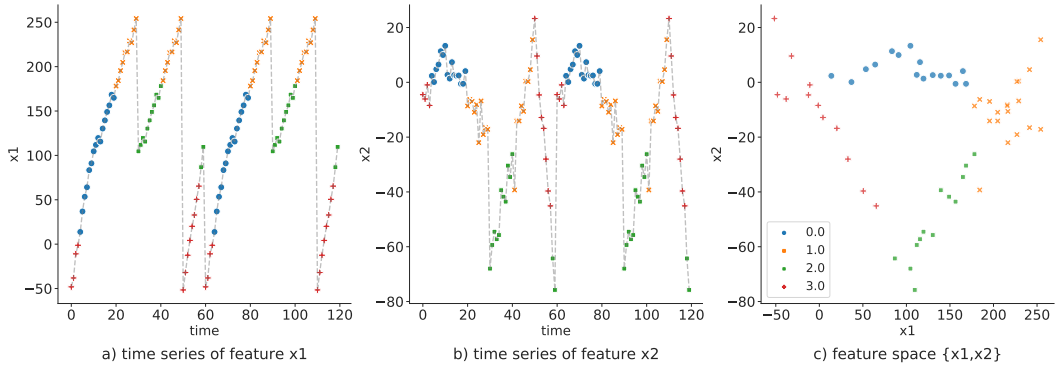

Figure 2. Time point clustering with the **TICC** algorithm; a) time series for feature x1 with predicted labels, b) time series for feature x2 with predicted labels and c) joint feature space {x1,x2} with predicted labels.

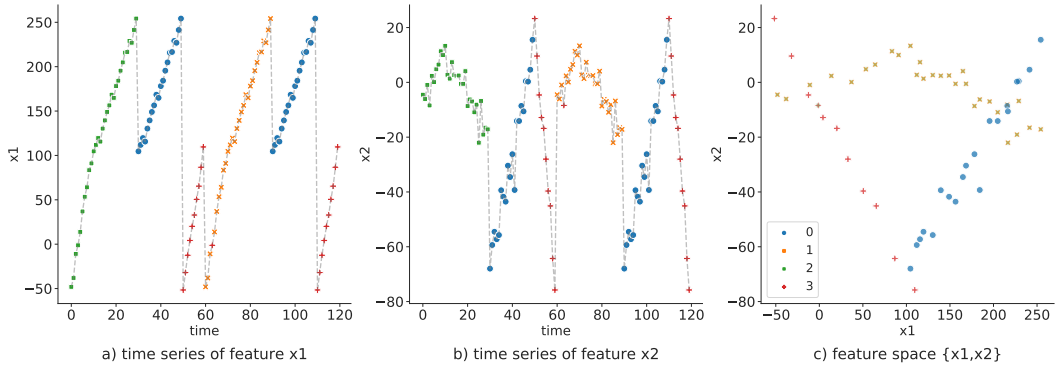

Figure 3. Time point clustering with the Gaussian mixture model (**GMM**); a) time series for feature x1 with predicted labels, b) time series for feature x2 with predicted labels and c) joint feature space {x1,x2} with predicted labels (orange crosses and green squares are overlaid).

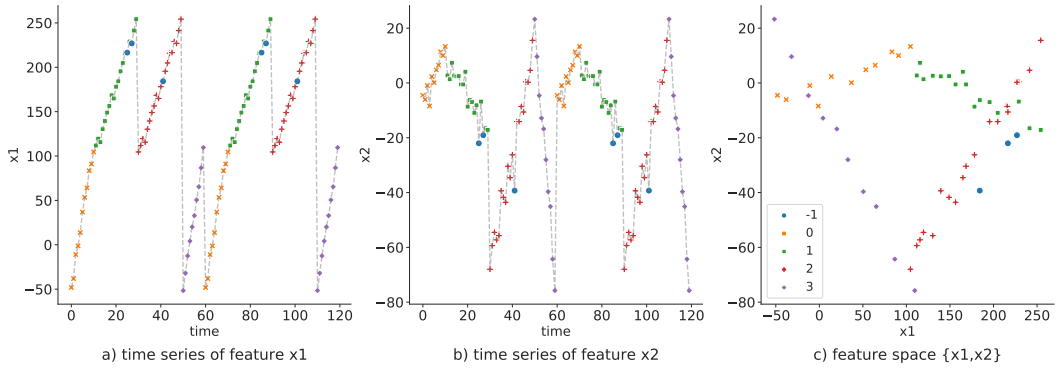

Figure 4. Semi-supervised time point clustering with the modified **CoExDBSCAN** algorithm; a) time series for feature x1 with predicted labels, b) time series for feature x2 with predicted labels and c) joint feature space {x1,x2} with predicted labels.

5. Experimental Evaluation

For the experimental evaluation we decided to demonstrate the significance of our approach on a popular real-world dataset in time series clustering. The LIBRAS movement dataset [30] is available from the UCI Machine Learning Repository [31] and contains 15 classes of 24 instances each, where each class references to a hand movement type in the LIBRAS Brazilian sign language. All movements were tracked from video analysis, where in each frame, the centroid pixels of the segmented objects (the hand) are found, which compose the discrete version a curve with 45 points. All curves are normalized in the unitary space and mapped in a representation with 90 features, representing the coordinates of the movement [31]. Each hand movement can be further segmented in distinct sub-movements, e.g. upper left to lower right, therefore makes it suitable for subsequence and time point clustering. However, there are no true labels available for individual segments and thus we can not perform external clustering validation with the adjusted Rand index (ARI) or cluster accuracy (ACC). However, Figure 6 provides a visual comparison of the *CoExDBSCAN*, *GMM* and *TICC* algorithms on the LIBRAS dataset on an individual sample from the vertical zigzag class (see sample 123 from Figure 5).

Figure 5 exemplifies the result of our experiment, depicting nine sample time series from the same class (vertical zigzag), without loss of generality, after applying the modified *CoExDBSCAN* algorithm on each series. The last step of our method, grouping all subsequences with equal or close regression coefficients, has been implemented in this case through discretization of the coefficients for each cluster into equal-sized buckets based on their quantiles. This results in three clusters, which correspond to similar, partial motions. The blue dot time points indicate a motion staring left in the coordinate space and ending in a lower right position of the coordinate space. The green square time points indicate a mirrored motion that starts right in the coordinate space and ends in a lower left position. The third class of subsequences, orange crosses, indicate a motion similar to the blue dot sequences, but with a less steep slope, e.g. oriented to a horizontal motion.

The visual comparison in Figure 6 of the *CoExDBSCAN*, *GMM* and *TICC* algorithms on the LIBRAS dataset on an individual sample from the vertical zigzag class shows that *CoExDBSCAN* provides the best qualitative result. For each algorithm we performed the discretization of the linear regression coefficients for each cluster into equal-sized buckets based on their quantiles. With *CoExDBSCAN* similar partial motions are grouped into the same category, see Figure 6 a), while with the *GMM* method, see Figure 6 b), and with the *TICC* algorithm, see Figure 6 c), partial motions with a visual clear change of course could not be separated. Furthermore, *GMM* and *TICC* both require the number of clusters to form as input parameter, which depends on the class of motion and number of partial motions to identify. Therefore, both algorithms can mere express the inherent structure of the dataset, but our modified *CoExDBSCAN* method is able to independently discover the intrinsic properties of the data.

In addition to the clustering of similar subsegements, we can utilize the clustering results by comparing the distributions of coefficients, which can serve as an effective basis for time series classification. If we compute the Kolmogorov-Smirnov statistic [32] for each learned distribution against an unseen distribution, we can classify time series that exhibit similar subsequences up to an accuracy of $\sim 67\%$. Similar accuracy can be achieved by *k-nearest neighbors vote* ($\sim 79\%$) or a time-series specific *Support Vector Classifier* ($\sim 68\%$)from the tslearn toolkit [24] without parameter optimization.

This experimental assessment shows that by adopting the *CoExDBSCAN* algorithm to time series data and grouping segments with similar correlations allows us to find accurate and interpretable structures. Moreover, our clustering results can serve as an effective basis for time series classification, which we plan to elaborate on in future work.

6. Conclusion

In this article we propose a new approach for semi-supervised time point clustering for multivariate time series. We adopt and modify the *CoExDBSCAN* algorithm and apply the algorithm to time series data to find temporal neighbourhoods, whose expansions are restricted to a priori constraints. We propose a constraint formulation to restrict the cluster expansion to the correlation of time point values. This constraint is defined by the deviation of time point residuals from the mean of residuals of time points within a subsequence, based on linear regression. Beyond the presented low-dimensional verification and evaluation datasets, our approach remains relevant even for high-dimensional data. This fact derives from the evaluation by Schubert et al. [18] for the original DBSCAN algorithm for high-dimensional data and becomes apparent in our research with climate data, where we apply the presented method to segment trajectories with high dimensionality.

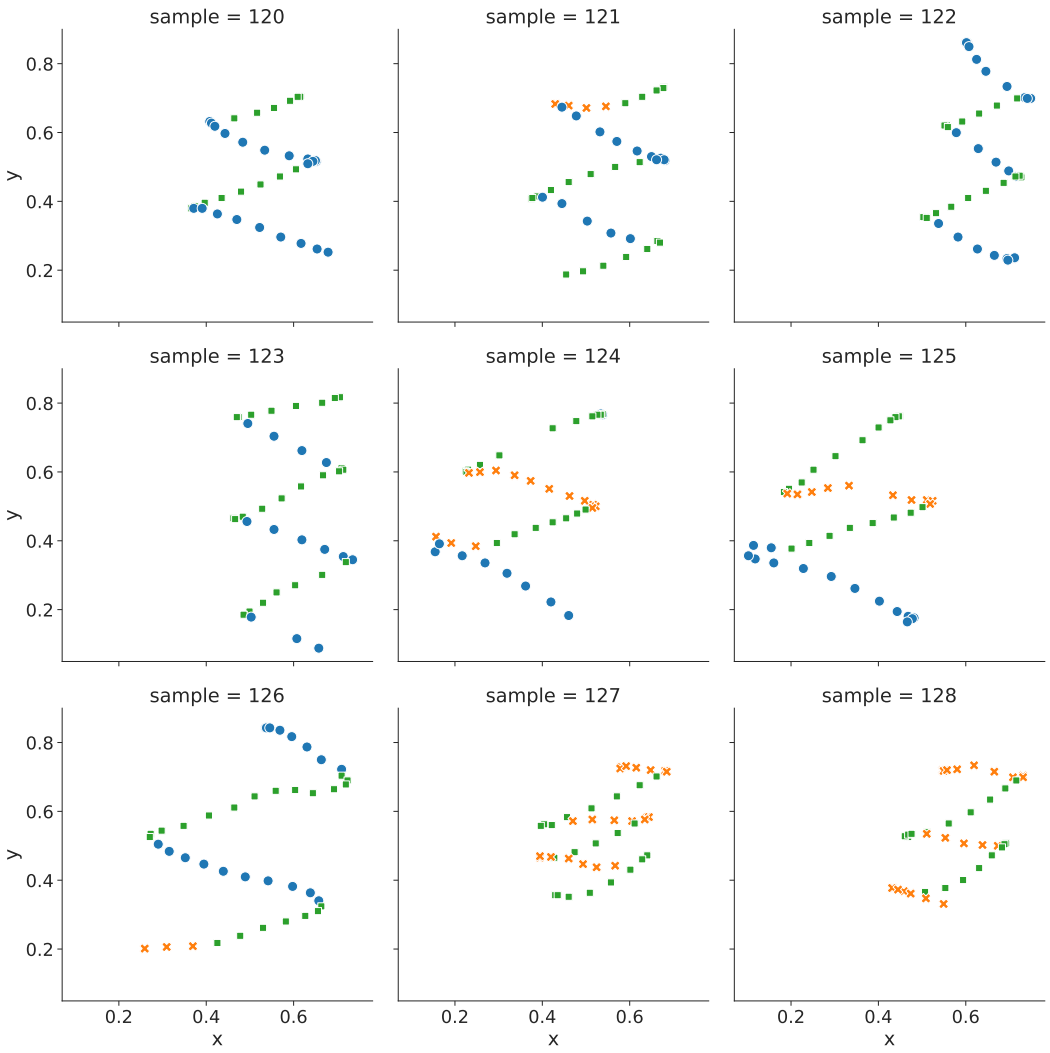

Figure 5. Semi-supervised time point clustering with the modified *CoExDBSCAN* algorithm on the LIBRAS dataset. Nine samples from one class (vertical zigzag) illustrate the successful segmentation of each time series into similar motions.

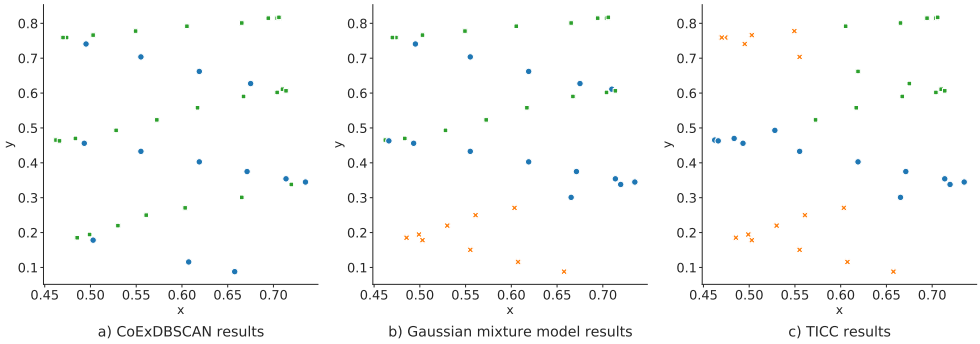

Figure 6. Comparison of clustering algorithms on the LIBRAS dataset on an individual sample from the vertical zigzag class (see sample 123 from Figure 5); a) **CoExDBSCAN** results (points labeled as noise are omitted), b) **GMM** results, and c) **TICC** results.

Forming groups of segments with similar correlations results in clusters of subsequences, which can be interpreted as recurring events or actions; for example similar movements as shown in our experimental evaluation. Our verification on a synthetic dataset indicates that our method significantly outperforms our baseline and state-of-the-art comparisons. Our validation and application to the real-world LIBRAS movement dataset demonstrates that this approach can accurately identify subsequences of similar motions and we are able to extract distributions of coefficients for each subsequence towards an effective classification approach.

For future work, we plan to improve especially this classification aspect and to provide more detailed comparison studies with other algorithms in the field of subsequence and time point clustering for multivariate time series.

References

[1] D. Hallac, S. Vare, S. Boyd, and J. Leskovec. "Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. ACM, 2017, pp. 215–223.

[2] T. Warren Liao. "Clustering of time series dataa survey". In: *Pattern Recognition* 38.11 (2005), pp. 1857 –1874. ISSN: 0031-3203.

[3] D. Ienco and R. Interdonato. "Deep Multivariate Time Series Embedding Clustering via Attentive-Gated Autoencoder". In: *Advances in Knowledge Discovery and Data Mining*. Cham: Springer International Publishing, 2020, pp. 318–329.

[4] D. J. Berndt and J. Clifford. "Using Dynamic Time Warping to Find Patterns in Time Series". In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. AAAIWS'94. Seattle, WA: AAAI Press, 1994, pp. 359370.

[5] M. Pourrajabi, D. Moulavi, R. J. G. B. Campello, A. Zimek, J. Sander, and R. Goebel. "Model Selection for Semi-Supervised Clustering". In: *Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, March 24-28, 2014*. Konstanz: OpenProceedings.org, 2014, pp. 331–342.

[6] S. Basu, I. Davidson, and K. Wagstaff. *Constrained clustering: Advances in algorithms, theory, and applications*. Boca Raton, Florida: CRC Press, Jan. 2008, pp. 1–442.

[7] D. Dinler and M. K. Tural. "A Survey of Constrained Clustering". In: *Unsupervised Learning Algorithms*. Cham: Springer, 2016, pp. 207–235. ISBN: 978-3-319-24211-8.

[8] C. C. Aggarwal and C. K. Reddy. *Data Clustering: Algorithms and Applications*. 1st. Chapman & Hall/CRC, 2013. ISBN: 1466558210.

[9] B. Ertl., J. Meyer., M. Schneider., and A. Streit. "CoExDBSCAN: Density-based Clustering with Constrained Expansion". In: *Proceedings of the 12th International Joint Conference*

on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR, SciTePress, 2020, pp. 104–115.

[10] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah. "Time-Series Clustering - A Decade Review". In: *Inf. Syst.* 53.C (2015), pp. 1638.

[11] S. Zolhavarieh, S. Aghabozorgi, and Y. W. Teh. "A Review of Subsequence Time Series Clustering". In: *The Scientific World Journal* 2014 (July 2014), p. 312521. ISSN: 2356-6140.

[12] E. Keogh and J. Lin. "Clustering of time-series subsequences is meaningless: implications for previous and future research". In: *Knowledge and Information Syst.* 8.2 (2005), pp. 154–177.

[13] H. Sakoe and S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE Trans. on Acoustics, Speech, and Signal Proc.* 26.1 (1978), pp. 43–49.

[14] F. Petitjean, A. Ketterlin, and P. Gançarski. "A Global Averaging Method for Dynamic Time Warping, with Applications to Clustering". In: *Pattern Recogn.* 44.3 (2011), pp. 678–693.

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters a Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.* KDD '96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[16] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications". In: *Data Mining and Knowledge Discovery* 2.2 (1998), pp. 169–194.

[17] L. Kirichenko, T. Radivilova, and A. Tkachenko. "Comparative analysis of noisy time series clustering". In: *CEUR workshop proceedings.* 2019.

[18] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN". In: *ACM Trans. Database Syst.* 42.3 (2017).

[19] C. Ruiz, M. Spiliopoulou, and E. Menasalvas. "C-DBSCAN: Density-Based Clustering with Constraints". In: *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing.* Berlin, Heidelberg: Springer, 2007, pp. 216–223.

[20] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang. "Salient Subsequence Learning for Time Series Clustering". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 41.9 (2019), pp. 2193–2207.

[21] S. Rodpongpun, V. Niennattrakul, and C. A. Ratanamahatana. "Selective Subsequence Time Series clustering". In: *Knowledge-Based Systems* 35 (2012), pp. 361–368. ISSN: 0950-7051.

[22] G. J. McLachlan and K. E. Basford. *Mixture models: Inference and applications to clustering.* Vol. 38. M. Dekker New York, 1988.

[23] G. J. McLachlan and D. Peel. *Finite mixture models.* John Wiley & Sons, 2004.

[24] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods. "Tslearn, A Machine Learning Toolkit for Time Series Data". In: *Journal of Machine Learning Research* 21.118 (2020), pp. 1–6.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[26] W. M. Rand. "Objective Criteria for the Evaluation of Clustering Methods". In: *Journal of the American Statistical Association* 66.336 (1971), pp. 846–850.

[27] K. Y. Yeung and W. L. Ruzzo. "An empirical study on principal component analysis for clustering gene expression data". In: *Bioinformatics* 17.9 (2001), pp. 763–774.

[28] L. Hubert and P. Arabie. "Comparing partitions". In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. ISSN: 1432-1343.

[29] F. Role, S. Morbieu, and M. Nadif. "CoClust: A Python Package for Co-Clustering". In: *Journal of Statistical Software, Articles* 88.7 (2019), pp. 1–29. ISSN: 1548-7660.

[30] D. B. Dias, R. C. B. Madeo, T. Rocha, H. H. Biscaro, and S. M. Peres. "Hand movement recognition for Brazilian Sign Language: A study using distance-based neural networks". In: *2009 International Joint Conference on Neural Networks.* 2009, pp. 697–704.

[31] D. Dua and C. Graff. *UCI Machine Learning Repository.* 2017.

[32] J. L. Hodges. "The significance probability of the smirnov two-sample test". In: *Arkiv för Matematik* 3 (1958), pp. 469–486.