Multivariate time series clustering based on common principal component analysis

Clément Weinreich cweinrei@ens-paris-saclay.fr
Théo Danielou theo.danielou@telecom-sudparis.eu

January 23, 2024

1 Introduction and contributions

Time, as a dimension, influences various phenomena in both the physical and digital world. Examples include physical environments, healthcare monitoring, stock exchange, or financial markets, all of which are subject to change over time. Data extracted from these phenomena are called time series, i.e., a sequence of observations ordered by time. In real-world applications, it is not uncommon to encounter Multivariate Time Series (MTS), which are multiple sequential observations along the time dimension that are correlated or dependent. This type of data can be analyzed using a variety of statistical methods such as classification [14], clustering [5], forecasting [1] or anomaly detection [17]. The paper we are studying [9] focuses on the task of clustering, an unsupervised machine learning method that divides observations into clusters such that data associated with a cluster reflects similar properties compared to the data in different clusters.

Previous work from this paper has paid much attention to univariate time series clustering, especially using Dynamic Time Wrapping (DTW) [10] as a similarity measure ([12],[13]). Despite the capability of DTW to consider meaningful differences in shapes and values of time series, the computation of similarity measures is expensive. Some methods such as K-shape or K-MS [11] use a measure of cross-correlation to measure similarity, and offer fast and accurate clustering and classification, but these methods are not suitable for multivariate time series. Clustering multivariate time series is more complex than univariate time series due to intrinsic characteristics like high dimensionality. Most of the work for MTS clustering and classification is based on PCA and its variants in order to reflect the relationships between the variables of the MTS ([18], [7], [6], [15]). However, when applied to clustering, these methods frequently overlook the comparison of the original values in the MTS. The work presented in the paper we present [9], proposes an approach based on common principal component analysis (CPCA) [3] and K-Means to cluster MTS. This method takes into account both the original values of the MTS and the relationships between the variables, while being computationally efficient.

In our project, where no official implementation of the method exists, we focus on a thorough analysis and independent implementation from scratch. We evaluated this method on a new dataset, along with two of the datasets originally used by the authors. Our exploration also extended to examining the method's limitations and potential for improvement through four specific experiments: assessing performance with unbalanced data, evaluating sensitivity to noise, determining the optimal number of principal components to use, and experimenting with different distance metrics for calculating the reconstruction error. Accompanying this report, we provide our code (`https://github.com/theodanielou/Time-Series-Mc2PCA`), along with CSV files containing the

results for each experiment. We employed pair-programming for the large majority of this project, including the development of the method itself, and have worked in parallel on both the design and running of the experiments and the writing of the report. We still separated the work for the experiments, each of us focusing more on two of the four experiments.

2 Method

In this section we detail the method proposed by the authors of [9]. We first present the mathematical concepts involved in their approach, and then explain the clustering algorithm.

2.1 Preliminaries

PCA and its variant CPCA are techniques used for dimensionality reduction and feature representation in complex datasets such as MTS. While PCA focuses on maximizing the variance within each individual group of data (in our case a single MTS), CPCA extends this principle by identifying a common subspace that can represent multiple groups of data simultaneously (in our case a cluster of MTS), maintaining consistency in feature space across different MTS.

Let X a group of data (or a dataset) consisting of N MTS X_i, where each X_i is a matrix in $\mathbb{R}^{n_i \times m}$ with n_i denoting the length of the series in X_i and m the number of variables. Assuming that each time series of X_i has been centered by subtracting their mean so each time series have an expectation of 0, the covariance of each MTS X_i is computed: $\Sigma_i = \frac{1}{n_i} X_i^T X_i$. Then, CPCA averages the individual covariance matrices Σ_i to create a common covariance matrix Σ that represent the entire group of data X: $\Sigma = \frac{1}{N} \sum_{i=1}^{N} \Sigma_i$.

The singular value decomposition (SVD) of the matrix Σ (real and symmetric) is $\Sigma = U \Lambda U^T$, where U is an $m \times m$ orthogonal matrix with eigenvectors u_j of Σ, and Λ is an $m \times m$ diagonal matrix with singular values σ_j (square roots of eigenvalues λ_j) in non-increasing order. By selecting the first p eigenvectors ($p \leq m$), a common space $S = [u_1, u_2, \ldots, u_p]$ is created for this group of data. For each MTS X_i, the principal components P_i are computed by projecting onto the space S: $P_i = X_i S$, which represents the transformed MTS X_i in the common feature space. This transformation into the space S ensures that the original MTS X_i is represented in a lower dimension while retaining the majority of its information.

Each MTS X_i within the group of data can be projected on the common projection axes S. This projection enables the representation of the original MTS X_i and any other MTS X_j in a reduced-dimensional space, enhancing their comparability. Conversely, the original MTS can be approximated by projecting P_i back to the original feature space using the transpose of S: $Y_i = P_i S^T = X_i S S^T$. Given that the reconstruction uses a subset of principal components (the first p components that form P_i), the reconstructed series Y_i may not perfectly match the original MTS X_i. This difference can be quantified by the reconstruction error E_i which measures the Euclidean distance between Y_i and X_i: $E_i = ||Y_i - X_i||_2 = \sqrt{\sum_{j=1}^{n_i} \sum_{k=1}^{m} (y_{jk} - x_{jk})^2}$. The reconstruction error E_i arises due to the loss of information from the reduced dimension of P_i, and the fidelity of the common space S as characterized by its eigenvectors.

2.2 Clustering multivariate time series

The clustering task partitions the set X into K clusters $C = \{C_1, C_2, \ldots, C_k\}$. For this purpose, the authors propose a clustering framework involving two primary steps: assignment of each MTS

to a cluster, and determination of a prototype representing the cluster's common projection axes. The common projection axes S_k are computed using CPCA and act as a prototype for each cluster. The reconstructed series within the k-th cluster's common projection space can be written as $Y_i^k = X_i S_k S_k^T$. The algorithm assigns the MTS X_i to the cluster C_k by minimizing the reconstruction error $E_{ik} = \min_k ||Y_i^k - X_i||_2$. In this iterative process, each MTS is assigned to the cluster that provides the closest approximation according to the reconstruction error, then the new cluster prototypes are recomputed using CPCA on the new cluster members. The cluster prototypes and memberships are updated until a stopping criterion is met, typically when the sum of the reconstruction errors stabilizes or decreases below a threshold. As the minimized criterion involves the original time series X_i, this method offers a trade-off between good reconstruction and dimensionality reduction.

3 Data

To evaluate our model, we initially focused on datasets used in the referenced article. We employed the Japanese Vowels dataset [8], which comprises two Japanese vowels spoken by various individuals. The objective is to identify individuals through 12 time series corresponding to the coefficients of the speech signal projected by Linear Predictive Coding (LPC). Additionally, we used the CMU MOCAP S16 dataset [2], containing motion capture data of various activities for the 16th subject, aiming to determine if the activity corresponds to walking. Furthermore, we used the Epilepsy dataset [16], not utilized by the authors. This dataset compares signals from people experiencing an epileptic seizure with those engaged in normal activities (providing 3 activity labels, and one for epilepsy). These signals come from a three-axis accelerometer placed on the dominant wrist of the subjects.

Name	Dimension	Length	Classes number	Volume
Japanese Vowels	12	[7-29]	9	370
CMU_MOCAP_S16	62	[127-580]	2	58
Epilepsy	3	206	4	137

Table 1: Multivariate time series datasets used for our experiments

The datasets described above do not require significant preprocessing for use with our method. For instance, the Japanese Vowels dataset already encompasses extensive feature extraction and selection, as it involves selecting coordinates of signals projected by LPC. Additionally, our method includes PCA, which results in dimension reduction. Therefore, conducting further feature extraction on our datasets, such as pre-PCA, could be redundant, particularly considering that information loss occurs due to PCA. It's also worth noting that our method is designed for ease and speed of use, thus avoiding the need for extensive pre-processing.

For the CMU MOCAP dataset, 58 samples are available in the original dataset [2]. However, the authors only used 29 samples, so twice as less. By analyzing the whole dataset, we noticed that only 9 out of the 58 samples correspond to the activity 'not walking' (false), resulting in an unbalanced dataset with only 15% of true labels. Dividing the size of the dataset by two and keeping the 9 true labels allows us to get a less unbalanced dataset with 30% of true labels. This strategy has potentially been adopted by the authors and would justify the use of only 29 samples for their experiments.

Moreover, while feature selection steps like change point analysis, frequency analysis, and PCA are vital for enhancing results, we chose not to focus on feature extraction and selection processes, which are partly specific to each dataset and might require particular attention for each one. Our study aims to concentrate on analyzing our model by testing various criteria using different datasets, thereby providing a more comprehensive evaluation of the method's capabilities.

4 Results

To compare our implementation of the method with the results presented in the paper, we first calculated the precision following the same methodology outlined in the article. This calculation was performed on the Japanese Vowels dataset which was also utilized for their experiments. We were able to replicate similar results as those presented in the paper (exact same 4 decimals for $p = 3$), see Table 2 (a). This replication can be obtained under a specific condition: the data (observation and labels) had to be sorted by labels, which is extremely favorable for their initialization scheme that sequentially assigns the samples to clusters. This finding highlights a significant concern in the author's evaluation of their clustering algorithm. Notably, when we randomized the data order before running the algorithm, the precision of the results noticeably decreased (see Table 2 (b)), demonstrating the algorithm's sensitivity to how the clusters are initially assigned.

To evaluate the proposed algorithm, the paper we study presented results solely based on Precision, which they deemed "a stable evaluation criterion". However, assessing an algorithm's performance solely based on precision does not accurately reflect its real-world effectiveness. This is particularly true in cases with unbalanced data, where one might envisage a scenario where all predicted classes predominantly populate the actual class with a high volume of data, thereby yielding high precision while failing to genuinely distinguish between the actual classes. Therefore, to gain a better understanding of the results obtained, we use three metrics: the Precision as presented in [9], the Recall, which is derived from Precision formula, and the Adjusted Rand Score [4], a measure of randomness in our clustering (values range from -0.5 to 1, with a value of 0 indicating random clustering and 1 indicating perfect clustering).

$$\text{Precision} = \sum_{j=1}^{K} \frac{|C_j|}{N} \times \max_{G_i} \left(\frac{|G_i \cap C_j|}{|C_j|} \right), \text{Recall} = \sum_{j=1}^{K} \left(\frac{|G_j|}{N} \times \max_{C_i} \left(\frac{|G_j \cap C_i|}{|G_j|} \right) \right)$$

We incorporated these metrics for the previous experiment on initialization (see Table 2). We also computed all these metrics on the datasets considered, and displayed the results in Table 3. We note that the obtained precision and recall often show correct performances for each dataset. However, for the full MOCAP dataset (58 samples), we observe poor results for the ARI metric compared to the results obtained on the reduced MOCAP dataset (which is less unbalanced). Note that the precision and recall do not reflect the poor clustering performance on these unbalanced data, which confirms that these metrics are not sufficient, and using the ARI adds an informative insight into the clustering performance. The bad performances on the complete MOCAP dataset raise questions about the sensitivity of the clustering algorithm to unbalanced datasets. This is actually not a surprising result as the algorithm clusters in a similar way to K-means. When some clusters have far more points than others, smaller clusters are easily absorbed by larger ones or incorrectly divided as their influence on the overall centroid positioning is comparatively minimal.

As the method is based on a PCA algorithm, we studied the impact of the number of retained components p for the dimensionality reduction step on the Reduced MOCAP and Japanese Vow-

els datasets. First, we analyzed the distribution of the explained variance on the principal components. Figure 2 illustrates that most of the relevant information is stored on the very first components. Then, Figure 1 illustrates the impact of selecting different values of p (from 1 m) over the three metrics. Overall, we observe a similar trend on the two datasets and on the three metrics: the performance is increasing until $p = \frac{1}{3}m$, and then decreases. For these datasets, an optimal choice of p seems to be around $\frac{1}{6}m$. This remains a hypothesis, experimenting on more datasets would help to get a better insight on the optimal choice of p.

In the article, the rationale for using Euclidean distance over other metrics for calculating the reconstruction error E_i was not specified. Therefore, we sought to compare the results obtained by varying the distance measure used in computing the reconstruction error. To this end, we employed three additional distance measures: Cosine distance, which assesses the angle between two time series; Dynamic Time Warping (DTW), accounting for the shape of the time series; and Manhattan Distance (L1), beneficial in contexts where differences in one direction are more significant. We displayed the results in Table 4 and observed that for the Japanese Vowels dataset, it was not the Euclidean distance that yielded the best results but the Cosine distance. This suggests that the choice of distance metric might be significant and warrants testing different distances on various datasets. It is possible that no single distance metric consistently outperforms others across all datasets, but rather the effectiveness depends on the type of data being processed. Additionally, it is crucial to remember that the method is intended to be quick and efficient, and a metric like DTW significantly increases the complexity of our algorithm.

To deepen our analysis, we performed a noise sensitivity experiment in order to assess the robustness of the method. The method uses CPCA so we can suppose that the method is robust to a certain level of noise as it focuses on the directions that maximize variance and ignore the less significant components that might be capturing noise. As we have worked previously with the Japanese Vowels dataset, we also used it for this experiment. We altered data with additive noise on each independent time series: $x_{ij} = x_{ij} + \alpha(\bar{x}_{ij}\epsilon)$ with α a noise scaling factor, $\bar{x}_{ij}$ the empirical mean of the time series x_{ij}, and ϵ a random vector of the same length as x_{ij}, sampled with $\mathcal{N}(0,1)$. We fixed $p = 3$ and tested for different values of noise factor from 0 to 5. To enhance the reproducibility and robustness of the experiment, the results are averaged on 50 different random seeds (0 to 49). The results of this experiment are available in the appendix section. Figure 4 and 3 respectively plot the Adjusted Rand Index mean scores and Precision/Recall for each noise factor α.

The ARI mean score seems to increase with the noise factor up to a certain point, suggesting that the clustering method might be robust to noise. Similarly, both precision and recall show a trend of increasing mean scores as noise is introduced, indicating that the method is not only robust to noise but may also benefit from it in terms of clustering performance. This is in line with our hypothesis. The improved performances upon noise addition might come from the regularization effect caused by the noise. Adding noise prevents the clustering algorithm from overfitting to the noise-free, high-variance components of the data that may not be relevant to the actual structure of the data. It can also indicate that the noise is helping to delineate the clusters more clearly. This could be due to the noise introducing a degree of separation between clusters that were not as distinct in the noise-free data. However, we observe that there are limits to the benefits of added noise, beyond a certain level, too much noise can degrade performance as it can obscure meaningful data patterns. To confirm our interpretation, it would have been necessary to run these experiments on more datasets with different configurations.

References

[1] André Bauer et al. "Libra: A benchmark for time series forecasting methods". In: *Proceedings of the ACM/SPEC International Conference on Performance Engineering*. 2021, pp. 189–200.

[2] Carnegie Mellon University Motion Capture Database. *CMU Motion Capture Database S16 dataset*. link: http://mocap.cs.cmu.edu/.

[3] Bernhard N Flury. "Common principal components in k groups". In: *Journal of the American Statistical Association* 79.388 (1984), pp. 892–898.

[4] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2 (1985), pp. 193–218.

[5] Ali Javed, Byung Suk Lee, and Donna M. Rizzo. "A benchmark study on time series clustering". In: *Machine Learning with Applications* 1 (Sept. 2020), p. 100001. ISSN: 2666-8270. DOI: 10.1016/j.mlwa.2020.100001. URL: http://dx.doi.org/10.1016/j.mlwa.2020.100001.

[6] M Johannesmeyer. "Abnormal Situation Analysis Using Pattern Recognition Techniques and Histrical Data". In: *AIChE Annual meeting, Dallas, TX, Oct. 31-Nov. 5 1999*. 1999.

[7] Leonidas Karamitopoulos, Georgios Evangelidis, and Dimitris Dervos. "PCA-based time series similarity search". In: *Data mining: Special issue in annals of information systems*. Springer, 2009, pp. 255–276.

[8] Mineichi Kudo, Jun Toyama, and Masaru Shimbo. *Japanese Vowels*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NS47.

[9] Hailin Li. "Multivariate time series clustering based on common principal component analysis". In: *Neurocomputing* 349 (2019), pp. 239–247. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.03.060. URL: https://www.sciencedirect.com/science/article/pii/S092523121930400X.

[10] Meinard Müller. "Dynamic time warping". In: *Information retrieval for music and motion* (2007), pp. 69–84.

[11] John Paparrizos and Luis Gravano. "Fast and accurate time-series clustering". In: *ACM Transactions on Database Systems (TODS)* 42.2 (2017), pp. 1–49.

[12] François Petitjean, Alain Ketterlin, and Pierre Gançarski. "A global averaging method for dynamic time warping, with applications to clustering". In: *Pattern recognition* 44.3 (2011), pp. 678–693.

[13] François Petitjean et al. "Dynamic time warping averaging of time series allows faster and more accurate classification". In: *2014 IEEE international conference on data mining*. IEEE. 2014, pp. 470–479.

[14] Alejandro Pasos Ruiz et al. "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances". In: *Data Mining and Knowledge Discovery* 35.2 (Dec. 2020), pp. 401–449. ISSN: 1573-756X. DOI: 10.1007/s10618-020-00727-3. URL: http://dx.doi.org/10.1007/s10618-020-00727-3.

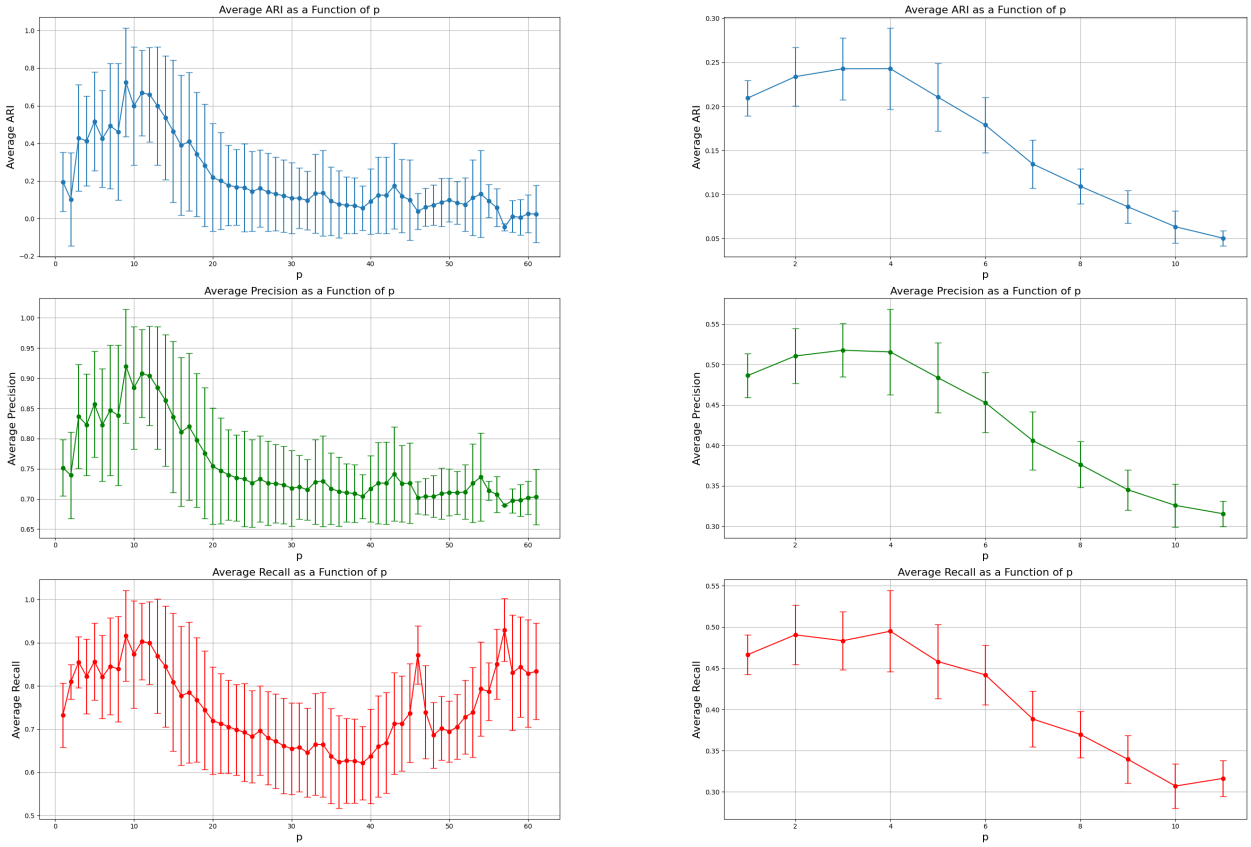
[15] Ashish Singhal and Dale E Seborg. "Pattern matching in multivariate time series databases using a moving-window approach". In: *Industrial & engineering chemistry research* 41.16 (2002), pp. 3822–3838.

[16] Jose R Villar et al. "Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition". In: *International journal of neural systems* 26.06 (2016), p. 1650037.

[17] Phillip Wenig, Sebastian Schmidl, and Thorsten Papenbrock. "TimeEval: A benchmarking toolkit for time series anomaly detection algorithms". In: *Proceedings of the VLDB Endowment* 15.12 (2022), pp. 3678–3681.

[18] Kiyoung Yang and Cyrus Shahabi. "A PCA-based similarity measure for multivariate time series". In: *Proceedings of the 2nd ACM international workshop on Multimedia databases*. 2004, pp. 65–74.

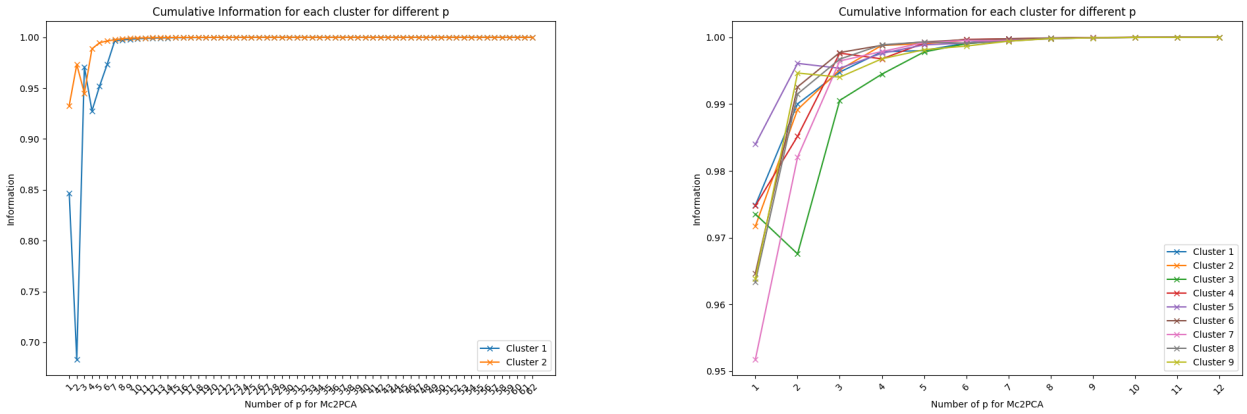
A Figures



(a) Reduced_CMU_MOCAP_S16 dataset

(b) Japanese Vowels dataset

Figure 1: Number of retained principal components p versus the metrics ARI (blue), Precision (Green), and Recall (Red). The metrics values are averaged over 50 runs on shuffled data and fixed random seeds (0 to 49). The vertical lines illustrate the standard deviation.



(a) Reduced_CMU_MOCAP_S16 dataset

(b) Japanese Vowels dataset

Figure 2: Percentage of cumulative information for each cluster with varying number of retained principal components p.

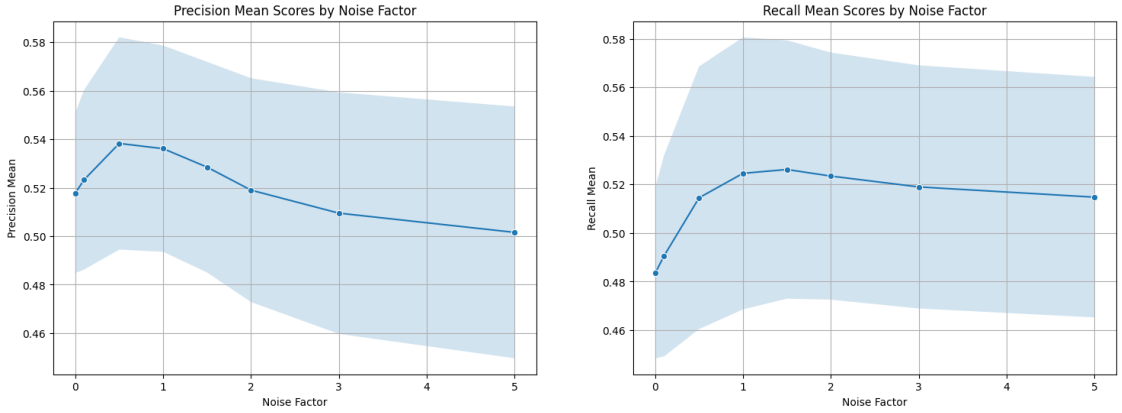


Figure 3: Precision and Recall mean scores as the noise factor increases from 0 to 5, on the Japanese Vowels dataset with $p = 3$ fixed for the clustering algorithm. Precision measures the fraction of true positives among the positive results, and Recall measures the fraction of true positives that were identified correctly by the algorithm.

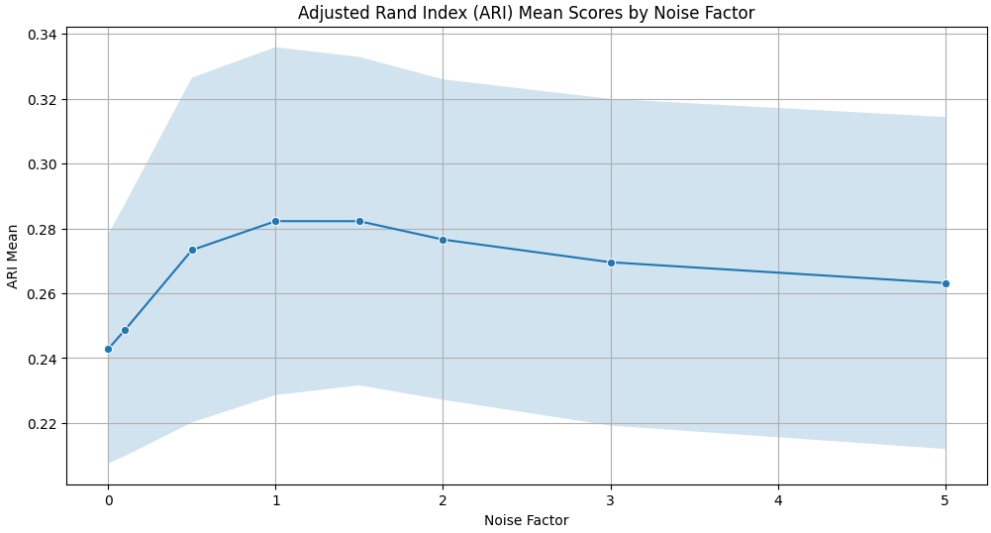


Figure 4: Adjusted Rand Index (ARI) mean scores as the noise factor increases from 0 to 5, on the Japanese Vowels dataset with $p = 3$ fixed for the clustering algorithm. The shaded area represents the standard deviation around the mean ARI score, providing an indication of the variability in ARI scores at each noise level. The ARI is a measure of the similarity between two data clusterings, and a higher ARI means a better match between the clustering result and the ground truth.

B Tables

Metrics	ARI	Precision	Recall
(a) Japanese Vowels without shuffling			
p = 1	0.2301	0.5324	0.5
p = 2	0.2766	0.5838	0.5324
p = 3	0.3143	0.6108	0.5568
p = 4	0.3611	0.6297	0.6027
(b) Japanese Vowels with shuffling			
p = 1	0.21 ± 0.006	0.486 ± 0.007	0.467 ± 0.007
p = 2	0.234 ± 0.009	0.511 ± 0.009	0.491 ± 0.01
p = 3	0.243 ± 0.01	0.518 ± 0.009	0.483 ± 0.01
p = 4	0.243 ± 0.013	0.516 ± 0.015	0.495 ± 0.014

Table 2: Precision, ARI and Recall computed on the Japanese Vowels dataset, with and without shuffling the dataset. For the runs on the shuffled dataset we averaged the metrics on 50 fixed random seeds (0 to 49), and provided the confidence interval.

Metrics	ARI	Precision	Recall
Japanese Vowels			
p = 1	0.21 ± 0.006	0.486 ± 0.007	0.467 ± 0.007
p = 2	0.234 ± 0.009	0.511 ± 0.009	0.491 ± 0.01
p = 3	0.243 ± 0.01	0.518 ± 0.009	0.483 ± 0.01
p = 4	0.243 ± 0.013	0.516 ± 0.015	0.495 ± 0.014
CMU_MOCAP_S16			
p = 1	0.052 ± 0.032	0.845 ± 0.0	0.697 ± 0.007
p = 2	-0.016 ± 0.057	0.846 ± 0.002	0.841 ± 0.007
p = 3	0.172 ± 0.095	0.864 ± 0.01	0.829 ± 0.021
p = 4	0.264 ± 0.079	0.859 ± 0.009	0.8 ± 0.022
Reduced_CMU_MOCAP_S16			
p = 1	0.195 ± 0.044	0.752 ± 0.013	0.732 ± 0.021
p = 2	0.103 ± 0.068	0.739 ± 0.02	0.81 ± 0.011
p = 3	0.429 ± 0.078	0.837 ± 0.024	0.855 ± 0.016
p = 4	0.412 ± 0.066	0.823 ± 0.023	0.822 ± 0.024
Epilepsy			
p = 1	0.226 ± 0.004	0.511 ± 0.008	0.642 ± 0.004
p = 2	0.11 ± 0.007	0.462 ± 0.01	0.502 ± 0.014
p = 3	0.059 ± 0.019	0.357 ± 0.022	0.847 ± 0.037

Table 3: ARI, Precision, and Recall computed on the 3 datasets we consider. The datasets are shuffled before running the clustering algorithm, we averaged the results and computed the confidence interval on 50 fixed random seeds (0 to 49). We precise the results for $p = 1, \ldots, 4$ as done by the authors in the original article, except for the Epilepsy dataset that only have 3 dimensions.

Metrics	ARI	Precision	Recall
Japanese Vowels (p=3)			
euclidean	0.247 ± 0.013	0.524 ± 0.008	0.493 ± 0.01
dtw	0.226 ± 0.008	0.515 ± 0.01	0.468 ± 0.011
l1	0.247 ± 0.011	0.521 ± 0.013	0.496 ± 0.01
cosine	**0.266 ± 0.009**	**0.531 ± 0.007**	**0.501 ± 0.01**

Table 4: ARI, Precision, and Recall computed on the Japanese Vowels dataset with p fixed to 3 and by varying the distance metric used to compute the reconstruction error E_i.