Contrastive Multivariate Singular Spectrum Analysis

Abdi-Hakin Dirie[†] Abubakar Abid[‡] James Zou[‡]

***Abstract*— We introduce contrastive multivariate singular spectrum analysis, a novel unsupervised method for dimensionality reduction and signal decomposition of time series data. By utilizing an appropriate background dataset, the method transforms a target time series dataset in a way that evinces the sub-signals that are *enhanced* in the target dataset, as opposed to only those that account for the greatest variance. This shifts the goal from finding signals that *explain* the most variance to signals that *matter* the most to the analyst. We demonstrate our method on an illustrative synthetic example, as well as show the utility of our method in the downstream clustering of real electrocardiogram and electromyogram signals. We end with a physical interpretation of what the algorithm is doing.**

***Index Terms*— PCA, SSA, time series, contrastive analysis, electrocardiogram, signal power, convolution**

I. Introduction

Unsupervised dimensionality reduction is a key step in many applications, including visualization [1] [2], clustering [3] [4], and preprocessing for downstream supervised learning [5]. Principal Component Analysis (PCA) is one well-known technique for dimensionality reduction, which notably makes no assumptions about the ordering of the samples in the data matrix $X \in \mathbb{R}^{N \times D}$. Multivariate Singular Spectrum Analysis (MSSA) [6] is an extension of PCA for time series data, which been successfully applied in applications like signal decomposition and forecasting [7] [8] [9]. In MSSA, each row is read at a certain time step, and thus is influenced by the ordering of the samples. MSSA works primarily by identifying key oscillatory modes in a signal, which also makes it useful as a general-purpose signal denoiser. However, MSSA (like PCA, upon which it is based) is limited to finding principal components that capture the maximal variance in the data. In situations where the information of interest explains little overall variance, these methods fail to reveal it.

Recently, *contrastive learning* methods such as contrastive PCA (cPCA) [10], [11], [12] have shown that utilizing a background dataset $Y \in \mathbb{R}^{M \times D}$ can help better discover structure in the foreground (target) X that is of interest to the analyst. The goal of contrastive learning is to take two datasets—a target data and a background data chosen by the user—and identifies low dimensional patterns that are enriched in the target dataset. It has found successful applications in diverse settings ranging from genomics, health to images [10]. However no contrastive method has been developed for time-series data.

We develop Contrastive Multivariate Singular Spectrum Analysis (cMSSA), which generalizes cPCA, and applies it to time series data. Figure 1 visualizes the relationships between the four methods. As a contrastive method, cMSSA emphasizes salient and unique sub-signals in time series data rather than just the sub-signals that comprise the majority of the structure. So while standard MSSA is useful for denoising a signal, cMSSA additionally "denoises" signals of structured but irrelevant information. We make cMSSA available as a software package for the general practitioner[1].

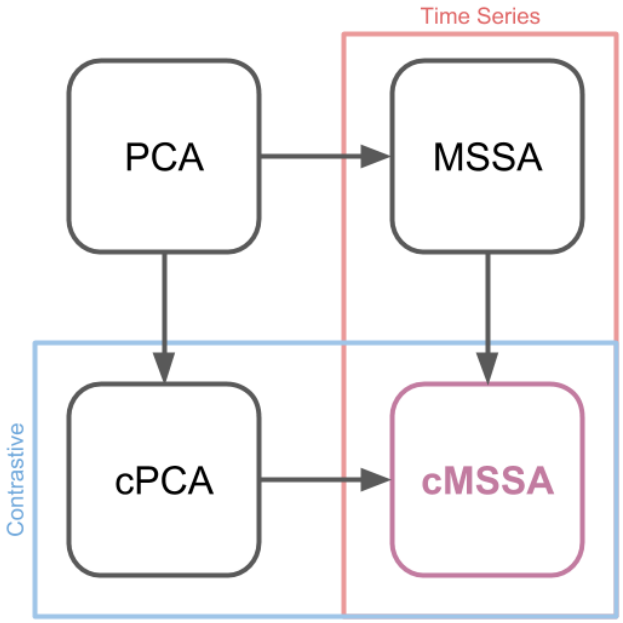


Fig. 1: Schematic illustrating the relations among PCA, cPCA, MSSA, and cMSSA.

II. Contrastive Multivariate Singular Spectrum Analysis

Before delving into how cMSSA works, we describe in detail how standard MSSA operates.

A. Standard MSSA

Consider a weakly-stationary single-channel times series $\mathbf{x} \in \mathbb{R}^T$. Furthermore, assume that $\mathbf{x}$ has already been centered. As a non-parametric tool, MSSA seeks to decompose the time series signal into components representing patterns at various time scales without assuming the form of these components. This is done by constructing a matrix from lagged versions of the time series. The so-called "Hankel"

[†]Diffeo Labs
[‡]Stanford University, Department of Electrical Engineering

[1]https://github.com/aadah/cMSSA

matrix $H_{\mathbf{x}} \in \mathbb{R}^{T' \times W}$ with window size W is defined as

$$H_{\mathbf{x}} = \begin{pmatrix} x_1 & x_2 & \dots & x_W \\ x_2 & x_3 & \dots & x_{W+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{T'} & x_{T'+1} & \dots & x_T \end{pmatrix}$$

where $T' = T - W + 1$. To extend to the multivariate case, let $X \in \mathbb{R}^{T \times D}$ be a D-channel time series that runs for T steps. We construct the Hankelized matrix H_X with window W by horizontally concatenating the per-channel Hankel matrices into a T'-by-DW matrix: $H_X = [H_{\mathbf{x}^{(1)}}; H_{\mathbf{x}^{(2)}}; \dots; H_{\mathbf{x}^{(D)}}]$. Next we compute the covariance matrix $C_X \in \mathbb{R}^{DW \times DW}$ as

$$C_X = \frac{1}{T'} H_X^T H_X.$$

The next step is to perform the eigendecomposition on C_X, yielding DW eigenvectors. Of these we take the top K vectors with the largest corresponding eigenvalues, analogous to selecting the top principal components in principal components analysis. We denote $\mathbf{e}^{(k)}$ as the eigenvector with the kth largest eigenvalue. We collect the vectors into a matrix $E \in \mathbb{R}^{DW \times K}$.

To transform our original time series X, we have two options: (a) Project X into the principal component (PC) space defined by E:

$$A = H_X E$$

or (b) use A to compute the kth reconstructed component (RC) $R^{(k)}$ as done in the SSA literature:

$$R_{tj}^{(k)} = \frac{1}{W_t} \sum_{t'=L_t}^{U_t} A_{t-t'+1,k} \cdot \mathbf{e}_{(j-1)W+t'}^{(k)}$$

where $L_t = \max(1, t - T + W)$, $U_t = \min(t, W)$, and $W_t = U_t - L_t + 1$. Equivalently, each $R^{(k)}$ can be computed by taking the outer product between the kth column of A and $\mathbf{e}^{(k)}$ and then averaging the anti-diagonals. The rows of $R^{(k)}$ are indexed by time $t \in \{1, \dots, T\}$ and the columns by channel $j \in \{1, \dots, D\}$.

Summing up the reconstructed components reproduces a denoised version of the original signal. This occurs because noise accounts for a minority of the signal's variance, so is filtered out when dropping the axes associated with the smallest eigenvalues.

For our purposes, we opt instead to take the horizontal concatenation of the reconstructed components as the second transform: $R = [R^{(1)}; R^{(2)}; \dots; R^{(K)}]$. To handle multiple time series, one simply vertically stacks each Hankelized matrix. The algorithm proceeds identically from there.

B. Contrastive MSSA

The modification to MSSA we introduce is via a new variable $\alpha \geq 0$ we call the *contrastive* hyperparameter. We construct H_Y for another D-channel times series Y (the background data) via the same process. It is not required that X and Y run for the same number of time steps, only that their channels are aligned. We compute a contrastive covariance matrix $C = C_X - \alpha C_Y$ and perform the eigendecomposition on C instead of C_X. The intuition for this is that by subtracting out a portion of the variance in Y, the remaining variance in X is likely to be highly specific to X but not Y. This is the key additional mechanism behind cMSSA — if $\alpha = 0$, then no contrast is performed, and cMSSA reduces down to just MSSA. The effectiveness of cMSSA depends on selecting a suitable background time series Y that contains the components that need to be removed from X, but not the components that should be preserved from X.

Algorithm 1 Spectral α-Search

Require: Minimum α to consider $\alpha_{\min}$, maximum α to consider $\alpha_{\max}$, number of αs to consider n, number of αs to return m, foreground signal X, background signal Y, window W, and number of components K.

procedure
  $Q \leftarrow$ LOGSPACE($\alpha_{\min}$, $\alpha_{\max}$, n) $\cup \{0\}$
  for $\alpha^{(i)} \in Q$ **do**
    H_X, $H_Y \leftarrow$ HANKEL(X, W), HANKEL(Y,W)
    C_X, $C_Y \leftarrow$ COV(H_X), COV(H_Y)
    $E^{(i)} \leftarrow$ EIGENDECOMP($C_X - \alpha^{(i)} C_Y$, K)
  $S \leftarrow$ EMPTY($\mathbb{R}^{n+1 \times n+1}$)
  for $i \in \{1, \dots, n+1\}$, $j \in \{i, \dots, n+1\}$ **do**
    $S_{i,j}, S_{j,i} \leftarrow \left\| {E^{(i)}}^T E^{(j)} \right\|_*$
  $Z \leftarrow$ SPECTRALCLUSTER(S, Q, m)
  $Q^* \leftarrow \{0\}$
  for $z \in Z$ **do**
    if $0 \notin z$ **then**
      $\alpha^* \leftarrow$ CLUSTERMEDIOD(z, S)
      $Q^* \leftarrow Q^* \cup \{\alpha^*\}$
return Q^*, set of m best αs, including zero.

The choice of α is non-trivial. Algorithm 1 outlines a routine for auto-selecting a small number of promising values for α. Because cMSSA is designed to assist data exploration, Algorithm 1 uses spectral clustering to identify a diverse set of α values corresponding to diverse eigenspaces and representations of X. The procedure works by first generating a large number of αs spread evenly in log-space. For each candidate α, we use cMSSA to compute its corresponding eigenvector matrix E. The procedure then performs spectral clustering, which requires a pairwise distance matrix as input. The distance metric used takes the nuclear norm of the matrix computed by multiplying the eigenvector matrices E for any pair of α's. After specifying the number of clusters desired, we take the mediod α of each cluster and return them as output. We always include 0 in this set, as the analyst may want to perform their analysis without contrast as control.

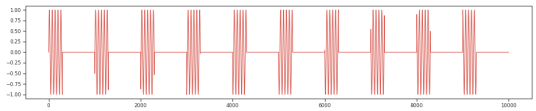
III. EXPERIMENTS

A. Synthetic example

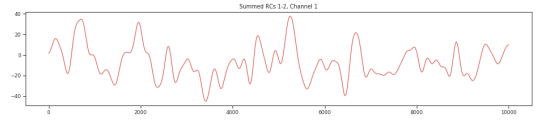
To illustrate cMSSA, we present a simple synthetic example. We generate an artificial one-channel signal Y by sampling 500 sinusoids with different frequencies, amplitudes, phases,

and vertical shifts. White Gaussian noise sample from $\mathcal{N}(0,1)$ is added in as well. We generate X in the same manner, but add in a very specific sub-signal (Figure 2a) that has comparatively low variance compared to the whole time series. The signals X and Y are generated independently as to rule out simple signal differencing as an explanation. We take X as foreground and Y as background.

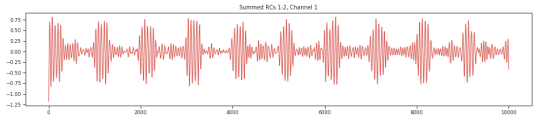
We set $W = 100$, $\alpha = 2$, and use only the top $K = 2$ RCs. Fig. 2 displays the reconstructions computed by MSSA versus cMSSA, alongside the sub-signal that was injected into X. Specifically, we see that the cMSSA reconstruction shown in Fig. 2c yields a noisy approximation of the sub-signal of interest, Fig. 2a. The variance of the noise here is comparable to the variance of the sub-signal—more noise would eventually overpower cMSSA's ability to extract the sub-signal.



(a) Sub-signal specific to the foreground data X, which is of much lower amplitude than the other sinusoidal sub-signals in X.



(b) Without contrast, the reconstructed time series consists of the high-amplitude sinusoidal sub-signals in X.



(c) With contrast, the reconstructed time series is able to identify the unique sub-signal in X.

Fig. 2: Results of a synthetic experiment that demonstrates that cMSSA is able to identify unique sub-signals in a time series, even when they are of much lower amplitude than background components.

B. Clustering of electrocardiograms

The data used in this experiment is taken from the public MHEALTH dataset [13]. In the dataset, 10 individuals were asked to perform 12 physical activities as several sensors recorded varied motion data. The researchers also collected two-lead electrocardiogram (ECG) readings, which we take as dual-channel time series data. In addition to the 12 activities, there is a 13th NULL class that represents ECG signals collected between each activity but which don't have labels themselves. To increase the number of individual time series, we partition each one in half.

For our experiments, the foreground data are all time series labelled as either JOGGING, RUNNING, JUMPING, or CYCLING, 20 time series each for a total of 80. These four, being the more cardio-intensive of the 12, had much more signal activity that would be needed to be sifted through, exactly the type of environment cMSSA is intended to handle. For background data, we take all 272 time series belonging to the NULL class.

To evaluate the effectiveness of cMSSA over its non-contrastive counterpart, we run both cMSSA and MSSA with a variety of hyperparameter settings. For each fitted model, we transform the foreground data to both the PC and RC spaces. Once the transformations are had, we perform spectral clustering into 4 clusters and compare the resulting clusters to the activity labels on the time series data, which were hitherto withheld from the algorithms. There are 3 hyperparameters: the window size $W \in \{8, 16, 32, 64, 128\}$, the number of desired components $K \in \{1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$, and the contrastive parameter α. We set K only if the value is less than or equal to DW (where $D = 2$ in this case). For α, we used our automatic routine to compute five key values to try for each setting of W and K. For each run of the routine, a total of 300 candidate αs we considered, with the minimum and maximum αs being 10^{-3} and 10^{3}, respectively. Of the five ultimately returned, one was zero, representing standard MSSA. Altogether, we run 530 experiments, 106 of which are standard MSSA, and the remaining cMSSA.

The spectral clustering requires an affinity matrix $S \in \mathbb{R}^{N \times N}$ which contains the similarities between any pair of time series, where N is the number of times series we wish to cluster. Let $X^{(i)}$ and $X^{(j)}$ be two time series. Using the FastDTW metric [14] with a euclidean norm[2], we define the similarity S_{ij} to be $\frac{1}{\text{FASTDTW}(X^{(i)}, X^{(j)})+1}$. The cluster evaluation uses the well-rounded BCubed metric [15] to compute the precision, recall, and F1 harmonic mean for a particular cluster prediction. We also perform the evaluation in the model-free sense where we simply cluster the time series with no transformation as a basic baseline.

TABLE I: Clustering results for cMSSA and MSSA in terms of maximum F1 score on the MHEALTH dataset. *average* means average metric over all settings of hyperparameters K and W (and non-zero α for cMSSA), regardless of whether using PC or RC spaces. Model-free clustering baseline also included along with best cMSSA and MSSA models. Best cMMSA run used $\alpha = 25.98$.

Model	W	K	Space	P / R / F1
Model-free	-	-	-	52.65 / 55.25 / 53.62
MSSA (best)	16	12	PC	54.24/ 60.38 / 57.14
cMSSA (best)	128	20	PC	**67.12 / 74.63 / 70.67**
MSSA (average)	-	-	-	44.05 / 50.96 / 47.19
cMSSA (average)	-	-	-	58.87 / 65.48 / 61.97

Table I reports the results of the clustering task on ECG data. We observe a number of things. First, the best runs of both MSSA and cMSSA outperform the model-free baseline. Second, we see that the best run cMSSA outperforms the best run MSSA. Note that the α value used was one of the

[2]FastDTW is not a symmetric metric, so we take the minimum between the two orderings of the operands.

values recommended by Algorithm 1. We also note that the best runs of MSSA and cMSSA both find that clustering in the PC space yielded better results. Third, if we simply look at the average-case performance, cMSSA has a 14-point F1 gain over MSSA. Finally, we note the efficiency. If we look at the best MSSA run, we see that it used 12 components. The number of components available is DW, so MSSA used 37.5% (12 out of 32) of the available basis. The same arithmetic applied to the best cMSSA run shows that it achieved its performance using only about 7.8% (20 out of 256) of the available basis.

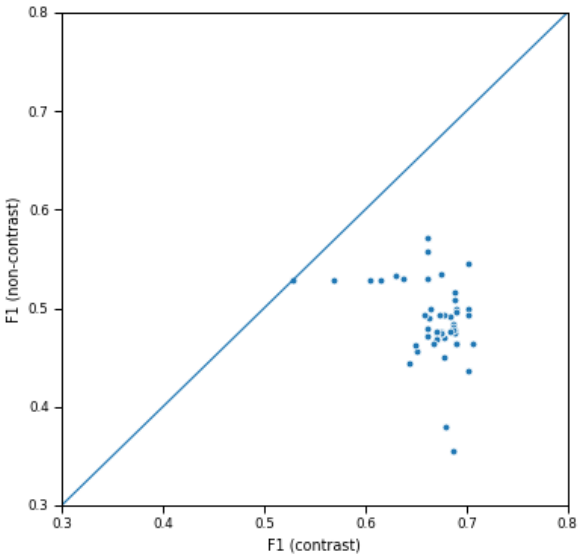


Fig. 3: Plot of paired F1 scores. Each point is for a particular setting of W and K. The contrastive F1 score used is the maximum of the four runs (one per automatically selected α, including $\alpha = 0$) for that setting of the hyperparams. $x = y$ line drawn as guidance. The points look at only those where the transform used is A.

Figure 3 shows a more granular view of the general gains to be had from using cMSSA. For a particular setting of W and K, we plot the F1 score for the non-contrastive case vs the contrastive case. Of the four auto-discovered values of αs used in the contrastive case, we take the model that had the greatest F1. Points below the diagonal line mean that the contrast was useful for a particular setting of the hyperparameters.

Finally, Figure 4 shows a visual comparison of MSSA versus cMSSA, using their respective hyperparameters settings as shown in Table I. Each row depicts how a random signal is processed with contrast on or off. We immediately see that cMSSA finds different signals than those found by MSSA. In the case of MSSA, the processed signals do not look substantially different from the originals. This is due to the fact that the high variance signals are shared across activities, so MSSA favors them during reconstruction. This is not the

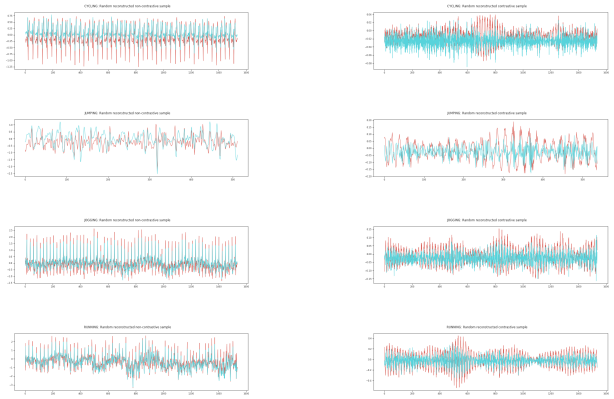


Fig. 4: Reconstructed time series after performing MSSA ($W = 16$, $K = 12$) and cMSSA ($W = 128$, $K = 20$, $\alpha = 25.98$). Each row is the same random signal reconstructed using MSSA (left) and cMSSA (right), one row per activity. The two colors correspond to the dual recording channels.

case with cMSSA, which identifies the differentiating signals that can disambiguate the activities.

C. Clustering of electromyograms

We also applied cMSSA to electromyogram (EMG) time series data publicly available[3] via the UCI Machine Learning Repository [16]. The researchers who compiled the data had 4 individuals each perform 20 physical tasks like hammering and walking. Ten of these activities were categorized as NORMAL while the other ten were categorized as AGGRESSIVE. Each individual had 8 sensors attached to them: two for the biceps, two for the triceps, two for the quadriceps, and two for the hamstrings. Thus we have 80 total time series, 4 for each activity. To increase the total number of time series, we again partition each series in half, yielding 8 series per activity. Even after partitioning, each time series runs for around 5000 time steps, far more than the ECG data. We sample every third time step to bring the length to a computationally tractable size, while only sacrificing a small amount of fidelity.

The experimental setup is very similar to the ECG experiments. We designated AGGRESSIVE and NORMAL as foreground and background, respectively. We considered the same set of values for W and K used in the ECG experiments. Unlike the ECG experiments, which clustered two-channel time series into 4 classes, the EMG experiments looked at cMSSA's effectiveness at clustering 8-channel time series into 10 classes. Again, spectral clustering was used with the FastDTW distance metric. The ground-truth labels for each activity were used only for the evaluation metrics.

Table II shows the results of the experiments. We see that the clustering task here is noticeably more difficult than clusering ECG, resulting in relatively lower precision, recall and F1. Nonetheless, cMSSA still significantly outperforms MSSA and the baseline, when the best hyperparameter choices

[3] https://archive.ics.uci.edu/ml/datasets/EMG+Physical+Action+Data+Set

TABLE II: Clustering results for cMSSA and MSSA in terms of maximum F1 score on the EMG dataset. *average* means average metric over all settings of hyperparameters K and W (and non-zero α for cMSSA), regardless of whether using PC or RC spaces. Model-free clustering baseline also included along with best cMSSA and MSSA models. Best cMMSA run used $\alpha = 3.10$.

Model	W	K	Space	P / R / F1
Model-free	-	-	-	25.13 / 27.50 / 26.26
MSSA (best)	8	18	RC	25.51 / 32.50 / 28.58
cMSSA (best)	16	6	PC	**34.04 / 41.56 / 37.43**
MSSA (average)	-	-	-	21.97 / 23.65 / 22.75
cMSSA (average)	-	-	-	26.65 / 29.62 / 28.02

for each method is used. In order to further validate the robustness of MSSA and cMSSA to the hyperparameters, we also report in Table II the average performance of each method when it's systematically ran over a diverse set of hyperparameters. cMSSA still demonstrates better average performance compared to MSSA.

IV. Physical Interpretation of cMSSA

While cMSSA is simple to implement, the mechanism by which it works has been thus far been explained only in terms of covariance. Just as [10] illustrates a geometric interpretation of how cPCA works, we present a physical interpretation of cMSSA using conventional signal processing language.

We begin with the objective that cMSSA tries to solve, which is finding the unit vector that accounts for the most variance in the contrastive covariance matrix:

$$\arg\max_{\mathbf{v} \in \mathbb{R}^{DW}_{\text{unit}}} \mathbf{v}^T (C_X - \alpha C_Y)\mathbf{v}.$$

Let $\lambda_X(\mathbf{v}) = \mathbf{v}^T C_X \mathbf{v}$ and define $\lambda_Y(\mathbf{v})$ similarly. [10] shows that for a universe $\mathcal{U} = \{(\lambda_X(\mathbf{v}), \lambda_Y(\mathbf{v})) \mid \mathbf{v} \in \mathbb{R}^{DW}_{\text{unit}}\}$ of target-background variance pairs, cMSSA would discover the subset that bounds the portion of the convex hull induced by $\mathcal{U}$ associated with the contrastive directions. However, this geometric interpretation becomes less useful in the context of cMSSA because of the intermediate step of computing the requisite Hankel matrices, from which the covariance matrices are actually computed. Instead, the Hankel matrices enable us to discover a physical interpretation.

Before proceeding, we define some notation. Let $\|\cdot\|_F$ and $\langle \cdot, \cdot \rangle_F$ be the Frobenius norm and inner product, respectively. Let $P(\cdot)$ compute the power of a discrete univariate signal. Finally, let $X[t_1, t_2]$ be the contiguous slice of a time series X from time t_1 to t_2. Let $*$ be the 2D convolution operator between two time series.

Theorem 1 (Contrastive Power)**.** *Let $X \in \mathbb{R}^{T_X \times D}$ and $Y \in \mathbb{R}^{T_Y \times D}$ be two time series. For a given $\alpha > 0$ and window size W, cMSSA is solving the following objective:*

$$\arg\max_{V \in \mathbb{R}^{W \times D}} P(X * V) - \alpha P(Y * V)$$

under the constraint that $\|V\|_F = 1$.

Proof. Let $H_X \in \mathbb{R}^{T'_X \times DW}$ and $H_Y \in \mathbb{R}^{T'_Y \times DW}$ be the respective Hankel matrices of X and Y. The contrastive power objective follows algebraically from the original covariance objective. For any $\mathbf{v} \in \mathbb{R}^{DW}_{\text{unit}}$, we have that

$$\begin{aligned}
\mathbf{v}^T(C_X - \alpha C_Y)\mathbf{v} &= \mathbf{v}^T C_X \mathbf{v} - \alpha \mathbf{v}^T C_Y \mathbf{v} \\
&= \frac{1}{T'_X}\mathbf{v}^T H_X^T H_X \mathbf{v} - \frac{\alpha}{T'_Y}\mathbf{v}^T H_Y^T H_Y \mathbf{v} \\
&= \frac{\|H_X \mathbf{v}\|^2}{T'_X} - \alpha \frac{\|H_Y \mathbf{v}\|^2}{T'_Y} \\
&= P(H_X \mathbf{v}) - \alpha P(H_Y \mathbf{v})
\end{aligned}$$

By noting that the quantity $H_X \mathbf{v}$ can be viewed as a univariate signal of length T'_X, we can reimagine it as convolving the original signal X with a reshaped $\mathbf{v}$. We call the reshaped object $V \in \mathbb{R}^{W \times D}$, and it's necessarily the case that $\|V\|_F = 1$. All that is left is to define the discrete 2D convolution for $t' \in \{1, \ldots, T'_X\}$:

$$(X * V)(t') = \langle X[t', t' + W - 1], V \rangle_F.$$

The same treatment is applied to $H_Y \mathbf{v}$, yielding the reformulated objective. □

The contrastive power objective gives a new view of what cMSSA is doing. Instead of thinking in terms of more abstract notions of variance, we can understand what cMSSA is doing using common signal processing concepts of power and convolution. We readily see that the algorithm specifically searches for a convolutional filter V that results in a high-power transformed target signal relative to the power of the corresponding transformed background signal. The geometric interpretation also applies, except it speaks in terms of signal power instead of variance.

V. Conclusion

We have developed cMSSA, a general tool for dimensionality reduction and signal decomposition of temporal data. By introducing a background dataset, we can efficiently identify sub-signals that are enhanced in one time series data relative to another. In an empirical experiment, we find that for virtually any setting of the hyperparameters, cMSSA is more effective at unsupervised clustering than MSSA, contingent on appropriate choices for the foreground and background data. It is worth emphasizing that cMSSA is an unsupervised learning technique. It does *not* aim to discriminate between time series signals, but rather discover structure and sub-signals within a given time series more effectively by using a second time series as background. This distinguishes it from various discriminant analysis techniques for time series that are based on spectral analysis [17], [18].

Some basic principles should be kept in mind when choosing to use cMSSA. First, the data ideally should exhibit periodic behavior, as MSSA (and by extension, cMSSA) is particularly well suited to finding oscillatory signals. Second, the data of interest X and background Y should not be

identical, but should share common structured signal such that the contrast retains some information in the foreground. As an example, the ECG foreground data consisted of subjects performing very specific activities, whereas the background consisted of unlabelled ECG signals in which the participants performed no specific activity. We would expect a good amount of overlap in signal variance, but signals specific to the four activities would be under-represented in the background. Thus contrast is a plausible way to extract this signal.

Finally, we note that the only hyper parameter of cMSSA is the contrast strength, α. In our default algorithm, we developed an automatic subroutine that selects the most informative values of α. The experiments performed used the automatically generated values. We believe that this default will be sufficient in many use cases of cMSSA, but the user may also set specific values for α if more granular exploration is desired.

REFERENCES

[1] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[2] L. McInnes and J. Healy, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[3] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu, "Dimensionality reduction for k-means clustering and low rank approximation," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. ACM, 2015, pp. 163–172.

[4] D. Niu, J. Dy, and M. Jordan, "Dimensionality reduction for spectral clustering," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 552–560.

[5] M. Pechenizkiy, A. Tsymbal, and S. Puuronen, "Pca-based feature transformation for classification: Issues in medical diagnostics," in *Computer-Based Medical Systems, 2004. CBMS 2004. Proceedings. 17th IEEE Symposium on*. IEEE, 2004, pp. 535–540.

[6] H. Hassani and R. Mahmoudvand, "Multivariate singular spectrum analysis: A general view and new vector forecasting approach," *International Journal of Energy and Statistics*, vol. 1, no. 01, pp. 55–83, 2013.

[7] H. Hassani, S. Heravi, and A. Zhigljavsky, "Forecasting european industrial production with singular spectrum analysis," vol. 25, pp. 103–118, 03 2009.

[8] R. Mahmoudvand, F. Alehosseini, and P. Rodrigues, "Forecasting mortality rate by singular spectrum analysis," vol. 13, pp. 193–206, 11 2015.

[9] K. Patterson, H. Hassani, S. Heravi, and A. Zhigljavsky, "Multivariate singular spectrum analysis for forecasting revisions to real-time data," *Journal of Applied Statistics*, vol. 38, no. 10, pp. 2183–2211, 2011.

[10] A. Abid, M. J. Zhang, V. K. Bagaria, and J. Zou, "Exploring patterns enriched in a dataset with contrastive principal component analysis," *Nature communications*, vol. 9, no. 1, p. 2134, 2018.

[11] J. Y. Zou, D. J. Hsu, D. C. Parkes, and R. P. Adams, "Contrastive learning using spectral methods," in *Advances in Neural Information Processing Systems*, 2013, pp. 2238–2246.

[12] R. Ge and J. Zou, "Rich component analysis." in *ICML*, 2016, pp. 1502–1510.

[13] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: a novel framework for agile development of mobile health applications," in *International Workshop on Ambient Assisted Living*. Springer, 2014, pp. 91–98.

[14] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[15] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[16] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[17] E. A. Maharaj and A. M. Alonso, "Discriminant analysis of multivariate time series: Application to diagnosis based on ecg signals," *Computational Statistics & Data Analysis*, vol. 70, pp. 67–87, 2014.

[18] R. T. Krafty, "Discriminant analysis of time series in the presence of within-group spectral variability," *Journal of time series analysis*, vol. 37, no. 4, pp. 435–450, 2016.

[19] O. Banos, C. Villalonga, R. Garcia, A. Saez, M. Damas, J. A. Holgado-Terriza, S. Lee, H. Pomares, and I. Rojas, "Design, implementation and validation of a novel open framework for agile development of mobile health applications," *Biomedical engineering online*, vol. 14, no. 2, p. S6, 2015.

[20] R. Vautard, P. Yiou, and M. Ghil, "Singular-spectrum analysis: A toolkit for short, noisy chaotic signals," *Physica D: Nonlinear Phenomena*, vol. 58, no. 1-4, pp. 95–126, 1992.