Temporal Clustering with Self-Supervised Image Networks

Richard Wong, Daniel Kim, Inwhee Joe
Hanyang Univ.

Abstract

In this paper we introduce a method, Image Temporal Clustering (ITC), to perform temporal clustering. This method utilizes self-supervised image representation learning on image transformations of time series data. The use of image transforms of time series allows for the application of established image-based machine learning methods on time series data. This is the first time that a joint-embedding architecture is applied to image transformations of time series data for the purpose of clustering time series.

I. Introduction

Time series clustering is a self-supervised technique to analyze time series data without the use of labels. This is traditionally performed with feature-based methods to extract relevant patterns from the data. These features usually require domain-knowledge and are non-trivial to identify. Also, techniques that utilize such features are only applicable to time series datasets that are similar.

Recent time series clustering methods that utilize deep learning [1, 2] were able to learn representations of time series data using recurrent neural networks (RNN) in the auto-encoder to extract features from the time series data. The use of the RNN allows for temporal features to be identified in the sequential data. However, Wang et al. [3] proposed 2 methods, Gramian Angular Fields (GAF) and Markov Transition Field (MTF), to transform time series data into images. This allows for the use of convolutional neural nets (CNN) on time series data as if they were regular 2D images, which presents an alternative to the current methods of using RNNs for temporal re-construction to identify features.

A similar work by Anand et al. [4] performed feature extraction using a pre-trained CNN model. However, they used images of line plots of time series data instead of the GAF or MTF transforms. Also, the use of a pre-trained CNN means that the model is fixed and cannot be tailored to the features of new datasets.

We propose the use of a joint-embedding network on the images of transformed time series data with a Variance-Invariance-Covariance (VICReg) [5] objective. It is a self-supervised technique to find invariant features across images of time series and presents a new application of image-representation learning on image transformations of time series. The resulting representations will be used for clustering via the method of K-means.

II. Methodology

We propose a 3-stage process: the transformation of time series data into GAF and MTF images, the use of a joint-embedding network to produce image vector representations, and the use of K-means clustering on the vector representations to assign cluster labels. We term this method as Image Temporal Clustering (ITC). The process flow is summarized in Figure 1.

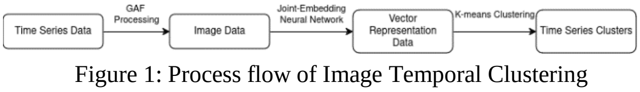

Figure 1: Process flow of Image Temporal Clustering

Let X be a time series where $X = \{x_1, x_2, \ldots, x_n\}$, and each $x_i \ \epsilon \ \boldsymbol{R}$. X is then normalized to obtain $\tilde{X}$ where $\tilde{x_i} \ \epsilon \ [-1,1]$. The time series is then transformed to polar coordinates via $\phi_i = \arccos(\tilde{x_i})$ and $r = \frac{t_i}{N}$. The GAF matrix is obtained from the polar coordinate angular terms via $G_{ij} = \cos(\phi_i + \phi_j)$.

Given the same time series X, the data x_i is sorted into Q quantile bins. We can then count the transitions between the quantile bins associated with the data points and compute the transition probability between 2 quantile bins. The MTF is computed with each element M_{ij} representing the transition probability from q_i to q_j, where q_i and q_j represent the quantiles that x_i and x_j belong to respectively.

The 2 resulting matrices are saved as a single channel image each. We follow the same processing method as Wang et al., where the 2 matrices are combined into a single GAF-MTF image coded by different channels.

A joint-embedding neural network is then used to learn representations based on the set of images obtained from the earlier steps. The original image is augmented with a random crop of varying scale, producing alternate views of the image. The 2 views are then fed into a CNN encoder, producing a representation vector. The representation vector is then fed into an expander network, which projects the representations onto a common space. The resulting 2 embedding vectors are evaluated with the VICReg objective to produce the loss. The objective helps the network to identify invariant features between the views and produce a dimensionally reduced vector representation of the images. This has the potential to find useful features from the unique characteristics of the time series data in an unsupervised manner. A modification from the original VICReg method is the use of the projected vectors for clustering as opposed to using the encoder output. The reason is that the expander network helps to project the encoder outputs onto the same space, allowing for more fair distance-based comparisons in the K-means clustering algorithm.

III. Experiments and Results

A choice of ResNet-18 was made for the CNN-encoder in the joint-embedding network. A linear layer serves as the expander network to project the output of the ResNet network to a final vector of size 64, which is the embedding we will perform clustering on.

Our method was evaluated on the 2018 UCR time series classification archive [6]. We utilized a similar data selection as Zhong et al. [2], where of the 128 datasets in the UCR archive, 10 datasets were chosen: Beef, DistPhalOutlAgeGroup, ECG200, ECGFiveDays, Meat, MoteStrain, OSULeaf, Plane, ProxPhalOutAgeGroup, and ProximalPhalanxTW. We believe that this represents a diverse set of real-world time series datasets to evaluate our clustering technique.

We compared ITC with the existing methods of K-means, u-shapelet [7], DTC [1], and DTCC [2]. K-means was used as a baseline measurement. u-shapelet represents a non-deep learning method of temporal clustering, whereas DTC and DTCC represents some recent deep learning-based methods. We used the normalized mutual information (NMI) and Rand Index (RI) as the evaluating metrics. The results for RI and NMI are found in Table 2 and Table 3 respectively. The results for ITC were obtained from an average of 10 runs.

Table 2: RI scores on the 10 time series datasets

Dataset	**K-means**	**u-shapelet**	**DTC**	**DTCC**	**ITC**
Beef	0.671	0.697	0.635	**0.763**	0.608
DistPhalOutAgeGrp	0.617	0.627	**0.781**	0.752	0.643
ECG200	0.632	0.576	0.602	**0.673**	0.645
ECGFiveDays	0.478	0.597	0.502	**0.658**	0.580
Meat	0.660	0.674	0.322	0.720	**0.779**
MoteStrain	0.495	0.479	0.506	**0.774**	0.600
OSULeaf	0.562	0.552	0.733	**0.755**	0.744
Plane	0.908	**1.000**	0.904	0.937	0.941
ProxPhalOutAgeGrp	0.529	0.521	0.743	**0.809**	0.738
ProxPhalTW	0.479	0.479	0.838	**0.861**	0.814
Average	0.603	0.620	0.656	**0.770**	0.709

Table 3: NMI scores on the 10 time series datasets

Dataset	**K-means**	**u-shapelet**	**DTC**	**DTCC**	**ITC**
Beef	0.293	0.341	0.275	**0.475**	0.235
DistPhalOutAgeGrp	0.188	0.258	0.341	**0.460**	0.302
ECG200	0.140	0.132	0.092	**0.301**	0.210
ECGFiveDays	0.001	0.150	0.002	**0.362**	0.124
Meat	0.251	0.272	0.225	0.610	**0.621**
MoteStrain	0.055	0.008	0.009	**0.465**	0.164
OSULeaf	0.021	0.020	0.220	0.235	**0.272**
Plane	0.860	**1.000**	0.868	0.925	0.845
ProxPhalOutAgeGrp	0.064	0.033	0.415	**0.532**	0.468
ProxPhalTW	0.008	0.011	0.620	**0.621**	0.531
Average	0.188	0.223	0.307	**0.499**	0.377

From Tables 2 and 3, we can observe that ITC performs competitively with the other methods. The performance is close to recent deep learning temporal clustering methods. Among the methods presented, ITC achieves the 2nd best average score for both RI and NMI scores.

For the RI scores, ITC achieved the best score in the Meat dataset and achieved close to best performance in the ECG200, OSULeaf, and Plane datasets. For the NMI scores, ITC achieved the best score in the Meat and OSULeaf datasets and achieved good performance in the ECG200, ECGFiveDays, MoteStrain, and the ProxPhalOutlAgeGroup datasets when compared with the other methods.

Although it was not able to exceed the state-of-the-art results of DTCC in most of the results, the performance of ITC is well-balanced across the datasets and it does not underperform very badly in datasets where other methods might have very low performance with respect to the NMI score (MoteStrain, OSULeaf, ProxPhalOutlAgeGroup, and ProxPhalTW). The other methods other than DTCC exhibit NMI scores that were close to 0 in some of these time series datasets. This suggests that some of the clusterings were almost random since the shared information was so low. ITC was able to have NMI scores across datasets that were not close to 0, demonstrating that the method was able to find some clustering features across all the datasets.

IV. Conclusion

We introduced a new self-supervised method of performing temporal clustering by using a joint-embedding network on transformed time series images. The method is simple in principle as it applies an established image representation learning method in a new way by learning on images obtained from processing time series. The results obtained is already competitive with recent deep temporal clustering methods despite having a relatively simple objective of only VICReg to find time series features. This technique demonstrates an alternative to the current crop of RNN auto-encoder-based deep learning methods on time series sequential data for time series clustering.

The other deep temporal clustering methods additionally utilize a joint-optimization of learning representations and clustering. A possible future work could be to enhance this image-based temporal clustering method with a joint-optimization objective that involves clustering.

REFERENCES

[1] Madiraju, N.S., Sadat, S.M., Fisher, D., & Karimabadi, H. (2018). Deep Temporal Clustering : Fully Unsupervised Learning of Time-Domain Features. ArXiv, abs/1802.01059.

[2] Zhong, Y., Huang, D., & Wang, C. (2022). Deep Temporal Contrastive Clustering. Neural Processing Letters, 55, 7869-7885.

[3] Wang, Z., & Oates, T. (2014). Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks.

[4] G. Anand, R. Nayak, DeLTa: Deep local pattern representation for time-series clustering and classification using visual perception, Knowledge-Based Systems 212 (2021) 106551.

[5] Bardes, A., Ponce, J., & LeCun, Y. (2021). VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. ArXiv, abs/2105.04906.

[6] Dau, H. A., Keogh, E., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, Gustavo, & Hexagon-ML. (2018). The UCR Time Series Classification Archive. https://www.cs.ucr.edu/%7Eeamonn/time_series_data_2018/.

[7] Zakaria, A. Mueen, E. Keogh, Clustering time series using unsupervised-shapelets, in: Proc. of IEEE International Conference on Data Mining (ICDM), 2012, pp. 785–794.