Enhancing Interpretability in Multivariate Time Series Classification through Dimension and Feature Selection

Zed Lee*

Department of Computer and Systems Sciences, Stockholm University, Stockholm, Sweden

Abstract

Interpretability in multivariate time series classification is crucial for understanding model decisions. However, the complexity of these classifiers often results in overwhelming feature spaces, hindering interpretability. To address this issue, we propose two novel methods: 1) Dimension selection based on segmentation of time series (DST) and 2) Feature selection based on discretization similarity (FDS). DST segments time series data and applies dimension selection to each segment, capturing distinct properties across different time ranges. FDS reduces feature redundancy by comparing discretization techniques and eliminating those with similar bin boundaries. Experiments on 24 UEA multivariate datasets demonstrate that our methods can significantly reduce the number of features while maintaining accuracy, offering a practical solution for enhancing interpretability in multivariate time series classification.

Keywords

Multivariate Time Series, Interpretability, Dimension Selection, Feature Selection,

1. Introduction

Time series datasets involve large quantities of data across multiple dimensions. The complexity of multivariate time series classification can quickly become overwhelming due to the interactions between different dimensions, which may negatively impact the classification outcome. Consequently, multivariate time series classifiers have grown increasingly complex in their model structures and feature spaces to enhance predictive performance. However, these classifiers often lack interpretability, posing both a challenge and a requirement.

Interpretable time series classifiers, such as MR-SEQL [1] and MR-PETSC [2], have been developed using symbolic discretization. Although these symbolic features have specific meanings and are linked to an interpretable linear classifier, several issues hinder full interpretability. First, as ensemble-based methods, both classifiers define multiple event sequence patterns for the same time points under various parameter settings for discretization to create bag-of-word patterns, resulting in inconsistencies in value ranges for the same discretized patterns, undermining interpretability. Z-Time [3] addresses this issue by eliminating the ensemble structure and applying various discretization techniques across the time series with unique event labels, ensuring each event label corresponds to a specific value range. However, the second problem

HI-AI@KDD, Human-Interpretable AI Workshop at the KDD 2024, 26th of August 2024, Barcelona, Spain

*Corresponding author.

zed.lee@dsv.su.se (Z. Lee)

is the sheer number of features used by all three classifiers, making human interpretation impractical.

In this paper, we suggest that interpretability should be evaluated not only by the architecture of models and features but also by the number of features used for classification. While dimensionality reduction is a common approach in various machine learning tasks [4], it is not suitable for interpretability as it can distort values. Initial efforts in dimensionality selection for multivariate time series often assume the selection of specific dimensions throughout the entire time series [5], which might not be optimal. This paper proposes two techniques leveraging previous work [5], Z-Time's segmentation properties, and multiple discretization techniques. First, we segment the time series and select different dimensions based on the properties within each segment. Second, we measure the similarity of different discretized bins and remove those with the highest similarity using the elbow method.

Our main **contributions** and **novelty** of this paper include:

- **Novelty.** We introduce the use of segmentation and discretization similarity to reduce the number of interpretable features in multivariate time series.
- **Effectiveness and efficiency.** Our proposed techniques can reduce the number of features by up to 86% while maintaining accuracy, with an average accuracy drop of only up to 9% on the UEA multivariate time series datasets [6].
- **Reproducibility.** Our code is publicly available on our GitHub repository[1].

2. Related Work

While many algorithms for multivariate time series classification have leveraged ensembles [7, 8] and deep learning techniques [9], recent attention has been directed towards interpretable time series classification. Most state-of-the-art interpretable time series classifiers utilize symbolic discretization [10] to create feature spaces, combined with linear classifiers. MR-SEQL [1] integrates a symbolic sequential learner with two discretization techniques: symbolic aggregate approximation (SAX) [10] and symbolic Fourier approximation (SFA) [11], to form the feature space representation. Similarly, MR-PETSC [2] employs standard frequent pattern mining with a relative duration constraint, instead of a sequential learner, to capture non-contiguous patterns as well as subsequences. Although both MR-SEQL and MR-PETSC can be applied to multivariate time series classification, their interpretability has been studied primarily for univariate problems, without addressing relationships between variables. The most recent work, Z-Time [3], offers the best efficiency (i.e., runtime) and effectiveness (i.e., accuracy) for multivariate time series classification. Unlike MR-PETSC and MR-SEQL, Z-Time is designed to consider the relationships between dimensions by incorporating temporal relations through *temporal abstraction*. Z-Time enhances interpretability by avoiding ensemble structures with multiple sliding windows and instead applying different discretization techniques, ensuring each event label has a single definition and value range. For feature reduction, earlier methods focused on dimension selection based on correlation [12] or similarity scores [13]. The most recent approach [5] selects dimensions based on the prototype distance between classes, which

[1] https://github.com/zedshape/dim-reduce

has also been tested in [14] for HIVE-COTE 2.0 [8], the most accurate classifier on the UCR dataset [6].

3. Proposed Methods

3.1. Background

3.1.1. Multivariate time series

Let $\mathbf{t} = \{t_1, \ldots, t_m\}$ represent a time series spanning m time points. A collection of such time series forms a time series instance $\mathbf{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_d\}$, consisting of d variables or dimensions. If $d = 1$, $\mathbf{T}$ is univariate; if $d > 1, \mathbf{T}$ is multivariate. Each time series instance $\mathbf{T}_k$ is assigned a class label $y_k \in \mathbf{y}$, where $\mathbf{y}$ is a list of class labels corresponding to each instance. The goal of time series classification models is to predict these class labels correctly.

3.1.2. Dimension selection techniques

Recent work by [5] has proposed two supervised dimension selection methods, which our suggestions build upon:

- **Elbow class sum (ECS)**: This method calculates the distance matrix between class centroid values and sums all the pairwise distances calculated for each dimension. The elbow method is then applied to find a cut-off point.
- **Elbow class pairwise (ECP)**: This method introduces an additional step to ECS. Instead of summation, it applies the elbow method to the pairwise distances for each dimension and then unions the eligible dimensions obtained from the distances for each dimension.

These methods assume that the selected dimensions span the entire time series. While ECP is regarded as the best among them, it sometimes fails to choose smaller dimensions, returning the whole set of dimensions. We address this issue by suggesting a segmentation-based application.

3.1.3. Discretization techniques

Discretization techniques have been actively used in interpretable time series classifiers to convert time series into sets of symbols. Each time step $t_i \in \mathbf{t}$ is converted into an event e_i, creating an event sequence $\mathbf{e}$. Each event value can take a unique event type ϵ. Z-Time uses the following three techniques:

- **Equal width discretization (EWD)**: Assuming $\mathbf{t}$ follows a uniform distribution, discretization boundaries are defined so that all event labels have value ranges of equal length, i.e., $t^{\epsilon_a}_{max} - t^{\epsilon_a}_{min} = t^{\epsilon_b}_{max} - t^{\epsilon_b}_{min}$ [15].
- **Equal frequency discretization (EFD)**: Discretization boundaries are defined so that each event label occurs with the same frequency in $\mathbf{e}$, i.e., $|e_i \in \mathbf{e} : e_i = \epsilon_a| = |e_i \in \mathbf{e} : e_i = \epsilon_b|$ [15].

- **Symbolic aggregate approximation (SAX)**: SAX uses a window size w and an event label size to perform both discretization and summarization. The discretization boundaries are defined assuming $\mathbf{t}$ follows a normal distribution [10], using the points that produce equi-sized areas under the normal distribution curve.

3.2. Dimension selection based on segmentation of time series (DST)

Our assumption is that important dimensions may vary across different time ranges, whereas current methods select dimensions by treating the time series as a whole. Selecting dimensions from segments of the time series could potentially enhance performance. This approach might not be feasible for interpretable classifiers that use sliding windows and ensemble structures, but Z-Time applies segmentation to capture the distinct properties of different time periods. This method is effective for time series with many unrelated parts or where the distribution changes over time. Unlike sliding windows, which overlap over time points and discretize values within these windows, segmentation offers a more straightforward approach to interpretability since each time point is discretized only once.

First, a time series instance $\mathbf{T}$ is divided into k equal-length segments $\{\mathbf{T}^1, \ldots, \mathbf{T}^k\}$. Then, a dimension selection algorithm such as ECP or ECS is applied to each segment $\mathbf{T}^i$, resulting in different dimensions being selected for each segment. When multiple time series instances are considered, the dimension selection algorithm is applied to a set of instances to ensure consistent dimension selection. Second, after segmentation, Z-Time is applied to each segment individually. This results in k different feature sets, which are concatenated to create a single feature set for input to an interpretable linear classifier.

While segmentation has improved Z-Time's performance, it has the side effect of linearly increasing the number of features, as each feature created from each segment must be distinguishable. This necessitates an additional step to significantly reduce the number of features.

3.3. Feature selection based on discretization similarity (FDS)

The second proposed strategy to reduce features involves finding similarities among discretization techniques. Z-Time uses three different discretizations, each with and without PAA, creating six different representations for each dimension of a time series instance. Sometimes, different techniques can produce similar bin boundaries, making it redundant to retain all of them. This method compares the boundaries of the bins created by each discretization technique by calculating the differences in boundary values. After computing the sum of boundary differences, the elbow method is used to select dimensions with significant differences.

Z-Time uses equal width discretization (EWD), equal frequency discretization (EFD), and SAX. Suppose a set of boundaries for a technique is $\mathbf{g} = \{g_1, g_2, \ldots, g_n\}$. The average difference is calculated as follows:

$$\frac{1}{n}\sum_{i}^{n}(g_{i,k} - g_{j,k})^2$$

The elbow method is then applied to identify the number of techniques with sufficiently high average differences, resulting in a set $\mathbf{g}'$ where $|\mathbf{g}'| \leq |\mathbf{g}|$. While this strategy does not reduce the number of dimensions, it significantly reduces the number of features, since in the worst case, the number is quadratic to the number of discretization techniques. Each technique creates a different set of event labels, thereby enhancing the overall interpretability of the classification model.

4. Experiments

Our experiment aims to evaluate the effectiveness of our proposed methods in reducing the number of features created by Z-Time. We used 26 UEA multivariate datasets with no missing values for our experiments [6]. The properties of these datasets can be found in the original repository. We excluded two datasets: *FaceDetection*, due to a memory limit of 128 GB for the chosen parameters, and *PenDigit*, as it was too small to apply segmentation. We compared different combinations of the following options:

- **Setting 1**: Dimension selection methods (ECP, ECS)
- **Setting 2**: Segmentation (with/without DST)
- **Setting 3**: Feature reduction (with/without FDS)
- **Setting 4**: Segment size ($k = 2, k = 4$)

Table 1
The relative numbers for the number of features, the number of dimensions, accuracy, and the total runtime compared to the default setting without any feature reduction technique. For accuracy, higher is better, while for all the others, lower is better.

Segments	FDS	Techniques	Features %	Dimension %	Accuracy %	Time %
2	FALSE	ECP	0.56	0.64	1.05	0.62
		ECP+DST	0.72	0.78	1.00	0.78
		ECS	0.26	**0.42**	1.02	0.37
		ECS+DST	0.31	0.46	0.95	0.44
	TRUE	ECP	0.25	0.64	**1.06**	0.46
		ECP+DST	0.32	0.78	0.97	0.54
		ECS	**0.12**	**0.42**	1.00	**0.27**
		ECS+DST	0.14	0.46	0.91	0.32
4	FALSE	ECP	0.56	0.64	**1.02**	0.61
		ECP+DST	0.71	0.77	1.00	0.78
		ECS	0.27	**0.42**	0.98	0.38
		ECS+DST	0.30	0.43	0.97	0.42
	TRUE	ECP	0.26	0.64	0.96	0.43
		ECP+DST	0.34	0.77	0.94	0.55
		ECS	**0.13**	**0.42**	0.92	**0.26**
		ECS+DST	0.14	0.43	0.93	0.31

In total, there are 16 different combinations per dataset. While detailed results are available in our repository, we present the average values in Table 1. Table 1 shows relative numbers for

Table 2
The win/lose comparisons on the accuracy per dataset between ECP/ECS with/without DST. The numbers indicate the number of dataset each setting shows the highest accuracy.

Options		ECP			ECS		
Segments	FDS	without DST	tie	with DST	without DST	tie	with DST
2	TRUE	**11**	8	5	9	2	**13**
	FALSE	6	**11**	7	9	2	**13**
4	TRUE	7	8	**9**	6	2	**16**
	FALSE	7	**10**	7	7	3	**14**

the number of features, the number of dimensions, accuracy, and the total runtime compared to the default setting without any feature reduction technique. The best technique is marked in bold, while the second best is underlined.

First, we observe that FDS significantly reduces the number of features, achieving an additional average reduction of 24.9% of the original features for the same setting. Without FDS, the minimum feature number is 26% of the original, but it can be further reduced to 12% with FDS. Additionally, while ECP generally shows better accuracy than ECS, ECP reduces features to 26% of the original, whereas ECS can reduce them to 12% while maintaining the same accuracy with $k = 2$.

Since Table 1 only shows average values, it might obscure the effect of DST, as results always appear inferior without DST. While standard ECP and ECS maintain better accuracy on average, ECP and ECS after segmentation show better accuracy in many instances. ECP after segmentation performs better in terms of accuracy on 16 datasets, considering all different settings (FDS and the number of segments). Table 2 shows the number of datasets where better accuracy is achieved by using DST. ECS with DST shows better accuracy than ECS without DST on 16 datasets with $k = 4$ and on 13 datasets with $k = 2$, which is more than half in both cases. However, with ECP, there is no meaningful improvement from applying DST. ECS after segmentation shows a significant drop in specific datasets, affecting the overall average, mainly due to the incorrect choice of the number of segments.

5. Conclusion

In this paper, we introduced two methods to enhance the interpretability of multivariate time series classifiers: 1) Dimension selection based on segmentation of time series (DST) and 2) Feature selection based on discretization similarity (FDS). Our experiments on 24 UEA multivariate datasets demonstrated that these methods could significantly reduce the number of features, by up to 86%, while maintaining accuracy, with only an average accuracy drop of up to 9%. These methods simplify the feature space and enhance interpretability, offering a practical solution for multivariate time series classification without compromising predictive performance. Future work can explore optimizing segmentation processes with dynamic lengths and refining similarity measures in FDS to enhance the quality of features.

References

[1] T. Le Nguyen, S. Gsponer, I. Ilie, M. O'Reilly, G. Ifrim, Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations, Data Mining and Knowledge Discovery 33 (2019) 1183–1222.

[2] L. Feremans, B. Cule, B. Goethals, Petsc: pattern-based embedding for time series classification, Data Mining and Knowledge Discovery 36 (2022) 1015–1061.

[3] Z. Lee, T. Lindgren, P. Papapetrou, Z-time: efficient and effective interpretable multivariate time series classification, Data Mining and Knowledge Discovery 38 (2024) 206–236.

[4] C. O. S. Sorzano, J. Vargas, A. P. Montano, A survey of dimensionality reduction techniques, arXiv preprint arXiv:1403.2877 (2014).

[5] B. Dhariyal, T. L. Nguyen, G. Ifrim, Fast channel selection for scalable multivariate time series classification, in: Advanced Analytics and Learning on Temporal Data: 6th ECML PKDD Workshop, AALTD 2021, Bilbao, Spain, September 13, 2021, Revised Selected Papers 6, Springer, 2021, pp. 36–54.

[6] A. Bagnall, J. Lines, W. Vickers, E. Keogh, The uea & ucr time series classification repository, 2018. URL: http://www.timeseriesclassification.com.

[7] J. Lines, S. Taylor, A. Bagnall, Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification, in: ICDM, IEEE, 2016, pp. 1041–1046.

[8] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, A. Bagnall, Hive-cote 2.0: a new meta ensemble for time series classification, Machine Learning 110 (2021) 3211–3243.

[9] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, F. Petitjean, Inceptiontime: Finding alexnet for time series classification, Data Mining and Knowledge Discovery 34 (2020) 1936–1962.

[10] J. Lin, E. Keogh, L. Wei, S. Lonardi, Experiencing sax: a novel symbolic representation of time series, Data Mining and Knowledge Discovery 15 (2007) 107–144.

[11] P. Schäfer, The boss is concerned with time series classification in the presence of noise, Data Mining and Knowledge Discovery 29 (2015) 1505–1530.

[12] B. Kathirgamanathan, P. Cunningham, A feature selection method for multi-dimension time-series data, in: Advanced Analytics and Learning on Temporal Data: 5th ECML PKDD Workshop, AALTD 2020, Ghent, Belgium, September 18, 2020, Revised Selected Papers 6, Springer, 2020, pp. 220–231.

[13] S. Han, A. Niculescu-Mizil, Supervised feature subset selection and feature ranking for multivariate time series without feature extraction, arXiv preprint arXiv:2005.00259 (2020).

[14] B. Dhariyal, T. Le Nguyen, G. Ifrim, Scalable classifier-agnostic channel selection for multivariate time series classification, Data Mining and Knowledge Discovery 37 (2023) 1010–1054.

[15] Y. Yang, G. I. Webb, A comparative study of discretization methods for naive-bayes classifiers, in: PKAW, volume 2002, 2002.