A New Symbolic Time Series Representation Method Based on Data Fuzzification

Agnieszka Jastrzębska
Warsaw University of Technology
Warsaw, Poland *A.Jastrzebska@mini.pw.edu.pl*

Zofia Matusiewicz
University of Information Technology and Management
Rzeszów, Poland *zmatusiewicz@wsiz.edu.pl*

Gonzalo Nápoles
Tilburg University
Tilburg, The Netherlands *G.R.Napoles@tilburguniversity.edu*

Abstract

Time series classification is an essential data processing task that relies on assigning class labels to sequences of temporal data. A fundamental component of any time series classification method is data representation. There exist several approaches to that task ranging from straightforward sequence distance-based methods to neural networks. We focus on symbolic time series representation-based methods. The literature of the domain repeatedly underlines their flexibility and good classification quality. We propose a new approach to convert numeric time series into symbolic ones based on fuzzy clustering. The goal is to reduce noise in the data. The proposed method utilizes cluster membership values to determine symbols that characterize the time series. The new approach was tested in an empirical procedure to validate its correctness while achieving satisfying results.

Keywords: time series classification, fuzzy sets, symbolic representation, fuzzy c-means.

1. Introduction

The specificity of the time series classification task advocates for the development of dedicated methods. The machine learning research community answered this call by proposing a range of interesting approaches to this problem. Generally speaking, three families of methods prevail, namely: distance-based, feature-based, and neural methods. Distance-based methods opt for computing differences between time series to extract the most similar ones. A review of these methods was last published by Abanda et al. [1]. Feature-based methods require the extraction of time series features that are then processed in a manner where their order ceases to matter. Two examples of feature extractors for time series data are catch22 [5] and tsfresh [2]. Notwithstanding, many time series classifiers utilize this approach quietly. The extremely popular ROCKET and its modifications [6] utilize this technique. Neural approaches make the third family of methods. They learn time series features and classify data in one model. An example algorithm of this type is InceptionTime [3], which was developed as an extension of the simple AlexNet neural network architecture.

The second aspect of time series classification tackles not the nature of the classifying procedure but data representation. This refers to the base on which we perform computations. In that regard, we may reiterate that the group of feature-based approaches mentioned above refers to a distinct group. Moreover, we shall mention that there are two more groups. One is based on shapelets, which are time series segments (modified or raw) best representing the underlying data. Ji et al. [4] elaborated on selection methods for shapelets to represent time series. Another

type of representation transforms the raw time series into a sequence of symbols defined in a fixed alphabet. The method addressed in this paper belongs to this family.

We present a novel approach to symbolic time series representation based on fuzzy clustering. The use of clustering in this role is new to the time series classification domain. Our goal is to obtain a reliable representation of the time series instances leading to improved results during further processing steps of time series data. The motivation for the use of fuzzy clustering can be summarized as follows:

- We can find a natural grouping of sequences of values with the use of clustering.
- We leverage data fuzzification to reduce dimensionality.
- We operate on a symbolic time series representation that reduces noise in the data.

The proposed method uses cluster membership values to determine symbols for the new data representation format. Symbolic time series is then subjected to classification.

2. The proposed method

The proposed approach relies on two steps: symbolic time series representation and classification. Algorithm 1 shows subsequent steps leading to the creation of a trained model.

Algorithm 1 Model training algorithm

Input: Train set of time series.
Parameters: Number of clusters k, segment length w, stride s.
Output: k centroids, trained random forest classifier.
Segmentation:
1: **for** each time series **do**
2: Split time series into segments by using a moving window of length w and stride s.
3: **end for**
Fuzzification:
4: Create an $N \times w$ matrix with segments arranged row-by-row, where N is the number of all segments created for all time series.
5: Run **Fuzzy C-Means** for the segments matrix to get k clusters.
6: **for** each time series **do**
7: **for** each segment **do**
8: Compute membership value to all clusters using the default fuzzy c-means membership function.
9: Get the largest membership value, generate a symbol corresponding to the appropriate cluster.
10: **end for**
11: This step produces a symbolic data representation with cluster labels being the symbols.
12: **end for**
Classifier training:
13: Train **Random Forest** to symbolic time series.

The first step relies on a moving window method that splits time series into segments. There are two parameters associated with this process: segment length w and stride s. Segments are extracted for a given train set and form an $N \times w$ segment matrix, where N is the number of segments. Subsequently, we perform a fuzzification procedure involving the fuzzy c-means algorithm. The goal is to generate k centroids that would represent the underlying properties of extracted segments.

The maximal membership principle determines which cluster must be selected in each case. In turn, each segment is replaced with a single value symbolizing the most important cluster that represents it best.

In the subsequent step, we proceed with the classification of the symbolic time series. In this study, we resorted to the random forest classifier (RF) as it pertained to our preliminary attempts with this approach. After the model is trained, we can classify unseen instances.

3. Empirical evaluation

3.1. Experimental procedure

The experiments in this section concern 17 time series classification datasets, which are publicly available under `https://www.timeseriesclassification.com`. The datasets are already partitioned into training and test sets, such that the former is used for model construction, while the latter is used to test the model's generalization capabilities.

The study concerned 17 datasets: *Beef, BeetleFly, Computers, DiatomSizeReduction, DistalPhalanxOutlineAgeGroup, Herring, Earthquakes, MiddlePhalanxOutlineAgeGroup, MiddlePhalanxOutlineCorrect, Phoneme, RefrigerationDevices, ScreenType, ShapeletSim, SonyAIBORobotSurface1, SonyAIBORobotSurface2, ToeSegmentation2, WormsTwoClass*.

Each experiment was repeated 10 times, and the results were averaged in order to obtain a single estimate per dataset. We establish plain Random Forest as a baseline method, directly run on the data. Comparing our results with it enables us to ascertain the value added by the fuzzification and word generation to the recognition scheme.

3.2. Parameter impact analysis

The experiments concerned a range of values of the number of clusters. We experimented with $k = 3, 5, 7, 11, 17$. The experiments were run separately for each dataset. We tested word lengths w of 3, 5, 7, 11, and 23. We tested stride values s of 1, 2, 4, 5, 12, and 24. As mentioned above, each experiment (concerning a specific value of k, s, and w) was repeated 10 times, and the results averaged.

As a result of the parameter sensitivity analysis, we assumed $s = 2$ and $w = 11$ for the remaining empirical studies conducted in this section. Subsequently, we run experiments searching for the right values of parameter k, which, as we suspected turned out to be dataset-dependent. We do not place detailed results, due to space constraints.

3.3. Classification quality

Let us compare the results provided by our approach with those achieved with state-of-the-art algorithms. Table 1 concerns other dictionary-based methods.

Our method outperformed **SAX-VSM** in 6 cases (for the following datasets *Computers, DistalPhalanxOutlineCorrect, Earthquakes, Herring, MiddlePhalanxOutlineAgeGroup, Phoneme, SonyAIBORobotSurface1*), the same is for one (*Herring*). We should note that *MiddlePhalanxOutlineAgeGroup* scored 20.20 percentage points better. In addition, the new method obtained better results compared to **WEASEL** in 5 cases (*DistalPhalanxOutlineAgeGroup, Earthquakes, Herring, MiddlePhalanxOutlineAgeGroup, SonyAIBORobotSurface1*). The average quality improvement is 4.19 percentage points. Similar results were obtained for the **BoP** method, for which the new method was also found to be superior in 10 cases. The average improvement was 10.71 percentage points. Finally, over 20 percentage points of advantage were achieved for the datasets *MiddlePhalanxOutlineAgeGroup* and *SonyAIBORobotSurface1*.

For 7 datasets, this is for *DistalPhalanxOutlineAgeGroup DistalPhalanxOutlineCorrect, Earthquakes, Herring, MiddlePhalanxOutlineAgeGroup, RefrigerationDevices, SonyAIBORobotSurface1* our method resulted in an average improvement of 11.06 percentage points compared to **BOSS**. The best improvement was achieved in relation to the set *SonyAIBORobotSurface1*.

As an addition to Table 1, we also add the values of k, for which the best results were

Table 1. Comparison between the proposed approach and dictionary-based methods. The performance metric concerns accuracy expressed as a percentage.

Dataset	SAX-VSM	WEASEL	BOP	BOSS	S-BOSS	ours best
BeetleFly	90.00	88.67	70.00	90.00	**93.67**	80.00
BirdChicken	**100.00**	86.50	75.00	95.00	96.83	85.00
Computers	62.00	77.85	66.80	75.60	**82.00**	65.60
DistalPhalanxOutlineAgeGroup	**84.17**	79.28	69.06	74.82	82.13	80.75
DistalPhalanxOutlineCorrect	72.83	**81.92**	71.38	72.83	81.10	77.00
Earthquakes	74.82	74.75	74.10	74.82	74.75	**81.99**
Herring	**62.50**	60.21	56.25	54.69	60.83	**62.50**
MiddlePhalanxOutlineAgeGroup	54.55	66.04	51.95	54.55	65.91	**74.75**
MiddlePhalanxOutlineCorrect	67.70	**82.83**	70.79	78.01	80.66	67.50
Phoneme	10.5	25.95	12.97	26.48	**27.96**	18.25
RefrigerationDevices	65.33	73.97	46.13	49.87	**77.84**	53.07
ScreenType	51.20	**59.59**	41.87	46.40	58.90	38.67
ShapeletSim	71.67	99.74	70.00	**100.00**	**100.00**	50.00
SonyAIBORobotSurface1	81.36	90.93	71.55	63.23	89.53	**92.18**
SonyAIBORobotSurface2	81.64	**93.53**	81.11	85.94	88.43	77.54
ToeSegmentation2	86.15	92.85	**94.62**	96.15	96.31	74.62
WormsTwoClass	71.43	80.04	63.64	**83.12**	80.78	60.77

achieved BeetleFly $\{5, 7, 9, 11\}$, BirdChicken $\{5, 11\}$, Computers $\{9\}$, Dist.Phal.OutAG $\{11\}$, Dist.Phal.OutCor $\{9\}$, Earthquakes $\{5, 9, 11\}$, Herring $\{5\}$, Mid.Phal.OutAG $\{11\}$, Mid.Phal.OutCor $\{7\}$, Phoneme $\{11\}$, Refrig.Dev $\{11\}$, ScreenType $\{11\}$, ShapeletSim $\{5, 7, 9, 11\}$, SonyAIBORobot1 $\{11\}$, SonyAIBORobot2 $\{5\}$, ToeSegm2 $\{7\}$, Worms2Class $\{9\}$.

References

[1] Abanda, A., Mori, U., and Lozano, J. A.: A review on distance based time series classification. In: *Data Mining and Knowledge Discovery* 33.2 (2019), pp. 378–412.

[2] Christ, M., Braun, N., Neuffer, J., and Kempa-Liehr, A. W.: Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). In: *Neurocomputing* 307 (2018), pp. 72–77.

[3] Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F.: InceptionTime: Finding AlexNet for time series classification. In: *Data Mining and Knowledge Discovery* 34.6 (2020), pp. 1936–1962.

[4] Ji, C., Liu, S., Yang, C., Pan, L., Wu, L., and Meng, X.: A Shapelet Selection Algorithm for Time Series Classification: New Directions. In: *Procedia Computer Science* 129 (2018). Proc. of IIKI 2017, pp. 461–467.

[5] Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., and Jones, N. S.: catch22: CAnonical Time-series CHaracteristics. In: *Data Mining and Knowledge Discovery* 33.6 (2019), pp. 1821–1852.

[6] Tan, C. W., Dempster, A., Bergmeir, C., and Webb, G. I.: MultiRocket: multiple pooling operators and transformations for fast and effective time series classification. In: *Data Mining and Knowledge Discovery* 36.5 (2022), pp. 1623–1646.